

IN silico Bloinformatics toolboX (inbix)

0.99p

Generated by Doxygen 1.8.13

Contents

1 Namespace Index	1
1.1 Namespace List	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	7
3.1 Class List	7
4 File Index	11
4.1 File List	11
5 Namespace Documentation	17
5.1 insilico Namespace Reference	17
5.1.1 Function Documentation	18
5.1.1.1 best_n()	18
5.1.1.2 get_bits()	19
5.1.1.3 join() [1/3]	19
5.1.1.4 join() [2/3]	19
5.1.1.5 join() [3/3]	20
5.1.1.6 split() [1/10]	20
5.1.1.7 split() [2/10]	20
5.1.1.8 split() [3/10]	20

5.1.1.9 <code>split()</code> [4/10]	21
5.1.1.10 <code>split()</code> [5/10]	21
5.1.1.11 <code>split()</code> [6/10]	21
5.1.1.12 <code>split()</code> [7/10]	21
5.1.1.13 <code>split()</code> [8/10]	22
5.1.1.14 <code>split()</code> [9/10]	22
5.1.1.15 <code>split()</code> [10/10]	22
5.1.1.16 <code>split_if()</code> [1/3]	22
5.1.1.17 <code>split_if()</code> [2/3]	23
5.1.1.18 <code>split_if()</code> [3/3]	23
5.1.1.19 <code>to_lower()</code> [1/3]	23
5.1.1.20 <code>to_lower()</code> [2/3]	23
5.1.1.21 <code>to_lower()</code> [3/3]	24
5.1.1.22 <code>to_upper()</code> [1/3]	24
5.1.1.23 <code>to_upper()</code> [2/3]	24
5.1.1.24 <code>to_upper()</code> [3/3]	24
5.1.1.25 <code>trim()</code> [1/3]	24
5.1.1.26 <code>trim()</code> [2/3]	25
5.1.1.27 <code>trim()</code> [3/3]	25
5.1.1.28 <code>trim_left()</code> [1/3]	25
5.1.1.29 <code>trim_left()</code> [2/3]	25
5.1.1.30 <code>trim_left()</code> [3/3]	25
5.1.1.31 <code>trim_right()</code> [1/3]	26
5.1.1.32 <code>trim_right()</code> [2/3]	26
5.1.1.33 <code>trim_right()</code> [3/3]	26
5.1.1.34 <code>zeroPadNumber()</code>	26

6 Class Documentation	27
6.1 _RipmResult Struct Reference	27
6.1.1 Detailed Description	27
6.1.2 Member Data Documentation	28
6.1.2.1 degrees	28
6.1.2.2 hubDegrees	28
6.1.2.3 hubs	28
6.1.2.4 modules	28
6.1.2.5 sizes	28
6.2 Alleles Class Reference	29
6.2.1 Detailed Description	30
6.2.2 Constructor & Destructor Documentation	30
6.2.2.1 Alleles() [1/3]	30
6.2.2.2 Alleles() [2/3]	30
6.2.2.3 Alleles() [3/3]	30
6.2.3 Member Function Documentation	31
6.2.3.1 matches()	31
6.2.3.2 operator<()	31
6.2.3.3 operator==()	31
6.2.3.4 swap()	31
6.2.4 Member Data Documentation	31
6.2.4.1 a1	32
6.2.4.2 a2	32
6.2.4.3 bp	32
6.2.4.4 chr	32
6.2.4.5 snp	32
6.3 ArgumentHandler Class Reference	33
6.3.1 Detailed Description	35

6.3.2	Constructor & Destructor Documentation	35
6.3.2.1	ArgumentHandler()	35
6.3.2.2	~ArgumentHandler()	35
6.3.3	Member Function Documentation	35
6.3.3.1	checkArguments()	35
6.3.3.2	DISALLOW_COPY_AND_ASSIGN()	35
6.3.3.3	displayHelp()	36
6.3.3.4	displayVersion()	36
6.3.3.5	processArguments()	36
6.3.4	Member Data Documentation	36
6.3.4.1	alpha	36
6.3.4.2	alwayssplitvars	36
6.3.4.3	argc	37
6.3.4.4	argv	37
6.3.4.5	caseweights	37
6.3.4.6	catvars	37
6.3.4.7	depvarname	37
6.3.4.8	file	37
6.3.4.9	fraction	38
6.3.4.10	holdout	38
6.3.4.11	impmeasure	38
6.3.4.12	memmode	38
6.3.4.13	minprop	38
6.3.4.14	mtry	38
6.3.4.15	nthreads	39
6.3.4.16	ntree	39
6.3.4.17	outprefix	39
6.3.4.18	predall	39

6.3.4.19	<code>predict</code>	39
6.3.4.20	<code>predictiontype</code>	39
6.3.4.21	<code>probability</code>	40
6.3.4.22	<code>replace</code>	40
6.3.4.23	<code>savemem</code>	40
6.3.4.24	<code>seed</code>	40
6.3.4.25	<code>splitrule</code>	40
6.3.4.26	<code>splitweights</code>	40
6.3.4.27	<code>statusvarname</code>	41
6.3.4.28	<code>targetpartitionsize</code>	41
6.3.4.29	<code>treetype</code>	41
6.3.4.30	<code>verbose</code>	41
6.3.4.31	<code>write</code>	41
6.4	<code>AttributeRanker</code> Class Reference	42
6.4.1	Detailed Description	43
6.4.2	Constructor & Destructor Documentation	44
6.4.2.1	<code>AttributeRanker()</code>	44
6.4.2.2	<code>~AttributeRanker()</code>	44
6.4.3	Member Function Documentation	44
6.4.3.1	<code>ComputeScores()</code>	44
6.4.3.2	<code>GetClassificationError()</code>	44
6.4.3.3	<code>GetNormalizeFlag()</code>	45
6.4.3.4	<code>GetScores()</code>	45
6.4.3.5	<code>InitializeData()</code>	45
6.4.3.6	<code>NormalizeScores()</code>	45
6.4.3.7	<code>PrintScores()</code>	45
6.4.3.8	<code>ResetForNextIteration()</code>	46
6.4.3.9	<code>SetK()</code>	46

6.4.3.10	SetNormalize()	46
6.4.3.11	WriteScores()	46
6.4.4	Member Data Documentation	47
6.4.4.1	classificationAccuracy	47
6.4.4.2	dataset	47
6.4.4.3	k	47
6.4.4.4	normalizeScores	48
6.4.4.5	scoreNames	48
6.4.4.6	scores	48
6.5	BirdseedData Class Reference	48
6.5.1	Detailed Description	50
6.5.2	Constructor & Destructor Documentation	51
6.5.2.1	BirdseedData()	51
6.5.2.2	~BirdseedData()	51
6.5.3	Member Function Documentation	51
6.5.3.1	GetAlleleCounts()	51
6.5.3.2	GetGenotypeCounts()	51
6.5.3.3	GetMajorAlleleFrequency()	52
6.5.3.4	GetMajorMinorAlleles()	52
6.5.3.5	GetMissingValues()	52
6.5.3.6	GetNumSNPs()	52
6.5.3.7	GetNumSubjects()	53
6.5.3.8	GetSamplePhenotype()	53
6.5.3.9	GetSNPNames()	53
6.5.3.10	GetSubjectCallConfidences()	53
6.5.3.11	GetSubjectGenotypes()	54
6.5.3.12	GetSubjectLabels()	54
6.5.3.13	GetSubjectNames()	54

6.5.3.14	HasPhenotypes()	54
6.5.3.15	HasSubjectLabels()	54
6.5.3.16	LoadData()	55
6.5.3.17	PrintAlleleCounts()	56
6.5.3.18	PrintInfo()	56
6.5.4	Member Data Documentation	56
6.5.4.1	confidences	56
6.5.4.2	excludeSnps	56
6.5.4.3	excludeSnpsFilename	56
6.5.4.4	genotypeCounts	57
6.5.4.5	hasExcludedSnps	57
6.5.4.6	hasIncludedSnps	57
6.5.4.7	hasPhenotypes	57
6.5.4.8	hasSubjectLabels	57
6.5.4.9	includeSnps	58
6.5.4.10	includeSnpsFilename	58
6.5.4.11	missingValues	58
6.5.4.12	phenosFilename	58
6.5.4.13	phenotypes	58
6.5.4.14	snpAlleleCounts	59
6.5.4.15	snpGenotypes	59
6.5.4.16	snpMajorAlleleFreq	59
6.5.4.17	snpMajorMinorAlleles	59
6.5.4.18	snpNames	59
6.5.4.19	snpsFilename	60
6.5.4.20	subjectLabels	60
6.5.4.21	subjectLabelsFilename	60
6.5.4.22	subjectNames	60

6.6 CArgs Class Reference	61
6.6.1 Detailed Description	62
6.6.2 Constructor & Destructor Documentation	62
6.6.2.1 CArgs()	62
6.6.3 Member Function Documentation	62
6.6.3.1 any()	63
6.6.3.2 check_unused_options()	63
6.6.3.3 count()	63
6.6.3.4 find()	63
6.6.3.5 fromPriorLog()	63
6.6.3.6 fromScript()	64
6.6.3.7 parseOptions()	64
6.6.3.8 value() [1/2]	64
6.6.3.9 value() [2/2]	64
6.6.3.10 value_double()	64
6.6.3.11 value_int()	65
6.6.3.12 value_lui()	65
6.6.3.13 varValue()	65
6.6.4 Member Data Documentation	65
6.6.4.1 a	65
6.6.4.2 n	65
6.6.4.3 option	66
6.6.4.4 optionLabel	66
6.6.4.5 original	66
6.6.4.6 parsed	66
6.6.4.7 root_command	66
6.7 CentralityRanker Class Reference	67
6.7.1 Detailed Description	68

6.7.2	Constructor & Destructor Documentation	68
6.7.2.1	CentralityRanker() [1/3]	68
6.7.2.2	CentralityRanker() [2/3]	68
6.7.2.3	CentralityRanker() [3/3]	69
6.7.2.4	\sim CentralityRanker()	69
6.7.3	Member Function Documentation	69
6.7.3.1	CalculateCentrality()	69
6.7.3.2	GaussEliminationSolver()	69
6.7.3.3	GetResultsByVariable()	69
6.7.3.4	Permute()	70
6.7.3.5	PowerMethodSolver()	70
6.7.3.6	ReadGainFile()	70
6.7.3.7	SetGammaVector()	70
6.7.3.8	SetGlobalGamma()	70
6.7.3.9	WriteToConsole()	71
6.7.3.10	WriteToFile()	71
6.7.4	Member Data Documentation	71
6.7.4.1	colsum	71
6.7.4.2	D	71
6.7.4.3	G	71
6.7.4.4	gainFile	72
6.7.4.5	gamma	72
6.7.4.6	gammaVector	72
6.7.4.7	Gdiag	72
6.7.4.8	Gtrace	72
6.7.4.9	n	72
6.7.4.10	r	73
6.7.4.11	ranks	73

6.7.4.12	topN	73
6.7.4.13	variableNames	73
6.8	Chap Class Reference	73
6.8.1	Detailed Description	74
6.8.2	Constructor & Destructor Documentation	74
6.8.2.1	Chap()	74
6.8.3	Member Function Documentation	74
6.8.3.1	build()	74
6.8.3.2	determineTestType()	75
6.8.3.3	isNested()	75
6.8.3.4	setModels()	75
6.8.3.5	setSNPList()	75
6.8.4	Member Data Documentation	75
6.8.4.1	alternate	75
6.8.4.2	current	76
6.8.4.3	H	76
6.8.4.4	null	76
6.8.4.5	P	76
6.9	ChapModel Class Reference	76
6.9.1	Detailed Description	77
6.9.2	Member Function Documentation	77
6.9.2.1	buildFromGroups()	77
6.9.2.2	buildFromSNPs()	77
6.9.2.3	haplotypesInSameGroup()	78
6.9.3	Member Data Documentation	78
6.9.3.1	coef	78
6.9.3.2	df	78
6.9.3.3	group	78

6.9.3.4	haploGroup	78
6.9.3.5	label	79
6.9.3.6	InLk	79
6.9.3.7	masked_conditioning_snps	79
6.9.3.8	model	79
6.9.3.9	p	79
6.9.3.10	rsq	79
6.9.3.11	se	80
6.10	CHIP_SEQ_INFO Struct Reference	80
6.10.1	Detailed Description	81
6.10.2	Member Data Documentation	81
6.10.2.1	chrom	81
6.10.2.2	position	81
6.10.2.3	rsnum	81
6.10.2.4	totalRegionReads	82
6.11	ChiSquared Class Reference	82
6.11.1	Detailed Description	85
6.11.2	Constructor & Destructor Documentation	85
6.11.2.1	ChiSquared()	85
6.11.2.2	~ChiSquared()	86
6.11.3	Member Function Documentation	86
6.11.3.1	ClearTables()	86
6.11.3.2	ComputeScore()	86
6.11.3.3	ComputeScores()	87
6.11.3.4	ComputeScoresWithPValues()	87
6.11.3.5	GetFrequencyCounts()	87
6.11.3.6	PrepareForAttribute()	87
6.11.3.7	PrintScoresWithPValues()	88

6.11.3.8 PrintTables()	88
6.11.3.9 ResetForNextIteration()	88
6.11.3.10 WriteScoresWithPValues()	88
6.11.4 Member Data Documentation	89
6.11.4.1 chiSquaredValues	89
6.11.4.2 expectedContingencyTable	89
6.11.4.3 numClasses	89
6.11.4.4 numLevels	89
6.11.4.5 observedFreqTable	90
6.11.4.6 scoresPvalues	90
6.12 CInfo Class Reference	90
6.12.1 Detailed Description	91
6.12.2 Member Data Documentation	91
6.12.2.1 bpstart	91
6.12.2.2 bpstop	91
6.12.2.3 lstart	91
6.12.2.4 lstop	92
6.13 clump_LD Class Reference	92
6.13.1 Detailed Description	93
6.13.2 Constructor & Destructor Documentation	93
6.13.2.1 clump_LD()	93
6.13.3 Member Function Documentation	93
6.13.3.1 allelePairs()	93
6.13.3.2 clump()	94
6.13.3.3 read_assoc_file()	94
6.13.3.4 set_Id()	94
6.13.3.5 set_pval()	94
6.13.3.6 set_r2()	94

6.13.3.7 <code>set_second_pval()</code>	95
6.13.4 Member Data Documentation	95
6.13.4.1 <code>assoc_results</code>	95
6.13.4.2 <code>clumped</code>	95
6.13.4.3 <code>filename</code>	95
6.13.4.4 <code>hp</code>	95
6.13.4.5 <code>ld_distance</code>	96
6.13.4.6 <code>P</code>	96
6.13.4.7 <code>pval_cutoff</code>	96
6.13.4.8 <code>pvals</code>	96
6.13.4.9 <code>r2_cutoff</code>	96
6.13.4.10 <code>second_pval_cutoff</code>	96
6.13.4.11 <code>snps</code>	97
6.14 ClumpPair Class Reference	97
6.14.1 Detailed Description	98
6.14.2 Member Function Documentation	98
6.14.2.1 <code>operator<()</code>	98
6.14.3 Member Data Documentation	98
6.14.3.1 <code>f</code>	98
6.14.3.2 <code>snp</code>	98
6.15 ClumpResults Class Reference	99
6.15.1 Detailed Description	99
6.15.2 Member Function Documentation	100
6.15.2.1 <code>operator<()</code>	100
6.15.3 Member Data Documentation	100
6.15.3.1 <code>annot</code>	100
6.15.3.2 <code>p</code>	100
6.16 Cluster Class Reference	101

6.16.1 Detailed Description	101
6.16.2 Member Data Documentation	101
6.16.2.1 person	101
6.17 CNVIndivReport Class Reference	102
6.17.1 Detailed Description	103
6.17.2 Constructor & Destructor Documentation	103
6.17.2.1 CNVIndivReport()	103
6.17.3 Member Function Documentation	103
6.17.3.1 calculateResults()	103
6.17.4 Member Data Documentation	103
6.17.4.1 count	103
6.17.4.2 count_baseline	103
6.17.4.3 n	104
6.17.4.4 segCount	104
6.17.4.5 t1	104
6.17.4.6 t10	104
6.17.4.7 t11	104
6.17.4.8 t12	104
6.17.4.9 t2	105
6.17.4.10 t3	105
6.17.4.11 t4	105
6.17.4.12 t5	105
6.17.4.13 t6	105
6.17.4.14 t7	105
6.17.4.15 t8	106
6.17.4.16 t9	106
6.18 CRandom Class Reference	106
6.18.1 Detailed Description	108

6.18.2 Member Function Documentation	108
6.18.2.1 rand() [1/2]	108
6.18.2.2 rand() [2/2]	108
6.18.2.3 srand()	108
6.18.3 Member Data Documentation	108
6.18.3.1 AM	109
6.18.3.2 EPS	109
6.18.3.3 IA	109
6.18.3.4 idum	109
6.18.3.5 IM	109
6.18.3.6 IQ	109
6.18.3.7 IR	110
6.18.3.8 iv	110
6.18.3.9 iy	110
6.18.3.10 NDIV	110
6.18.3.11 NTAB	110
6.18.3.12 RNMX	111
6.19 CSNP Class Reference	111
6.19.1 Detailed Description	111
6.19.2 Member Data Documentation	112
6.19.2.1 one	112
6.19.2.2 two	112
6.20 Data Class Reference	112
6.20.1 Detailed Description	114
6.20.2 Constructor & Destructor Documentation	114
6.20.2.1 Data()	114
6.20.2.2 ~Data()	114
6.20.3 Member Function Documentation	115

6.20.3.1 addSparseData()	115
6.20.3.2 DISALLOW_COPY_AND_ASSIGN()	115
6.20.3.3 get()	115
6.20.3.4 getAllValues()	115
6.20.3.5 getIndex()	116
6.20.3.6 getMaxNumUniqueValues()	116
6.20.3.7 getNumCols()	116
6.20.3.8 getNumRows()	116
6.20.3.9 getNumUniqueDataValues()	116
6.20.3.10 getUniqueDataValue()	117
6.20.3.11 getVariableID()	117
6.20.3.12 getVariableNames()	117
6.20.3.13 loadFromFile()	117
6.20.3.14 loadFromFileOther()	117
6.20.3.15 loadFromFileWhitespace()	118
6.20.3.16 reserveMemory()	118
6.20.3.17 set()	118
6.20.3.18 sort()	118
6.20.4 Member Data Documentation	118
6.20.4.1 externalData	118
6.20.4.2 index_data	119
6.20.4.3 max_num_unique_values	119
6.20.4.4 num_cols	119
6.20.4.5 num_cols_no_sparse	119
6.20.4.6 num_rows	119
6.20.4.7 num_rows_rounded	119
6.20.4.8 sparse_data	120
6.20.4.9 unique_data_values	120

6.20.4.10 variable_names	120
6.21 DataChar Class Reference	120
6.21.1 Detailed Description	123
6.21.2 Constructor & Destructor Documentation	123
6.21.2.1 DataChar() [1/2]	123
6.21.2.2 DataChar() [2/2]	123
6.21.2.3 ~DataChar()	123
6.21.3 Member Function Documentation	124
6.21.3.1 DISALLOW_COPY_AND_ASSIGN()	124
6.21.3.2 get()	124
6.21.3.3 reserveMemory()	124
6.21.3.4 set()	124
6.21.4 Member Data Documentation	125
6.21.4.1 data	125
6.22 DataDouble Class Reference	125
6.22.1 Detailed Description	128
6.22.2 Constructor & Destructor Documentation	128
6.22.2.1 DataDouble() [1/2]	128
6.22.2.2 DataDouble() [2/2]	128
6.22.2.3 ~DataDouble()	129
6.22.3 Member Function Documentation	129
6.22.3.1 DISALLOW_COPY_AND_ASSIGN()	129
6.22.3.2 get()	129
6.22.3.3 loadFromDatasetMask()	129
6.22.3.4 loadFromPlink()	129
6.22.3.5 reserveMemory()	130
6.22.3.6 set()	130
6.22.4 Member Data Documentation	130

6.22.4.1	data	130
6.23	DataFloat Class Reference	131
6.23.1	Detailed Description	133
6.23.2	Constructor & Destructor Documentation	133
6.23.2.1	DataFloat() [1/2]	133
6.23.2.2	DataFloat() [2/2]	133
6.23.2.3	~DataFloat()	133
6.23.3	Member Function Documentation	134
6.23.3.1	DISALLOW_COPY_AND_ASSIGN()	134
6.23.3.2	get()	134
6.23.3.3	reserveMemory()	134
6.23.3.4	set()	134
6.23.4	Member Data Documentation	135
6.23.4.1	data	135
6.24	Dataset Class Reference	135
6.24.1	Detailed Description	144
6.24.2	Constructor & Destructor Documentation	144
6.24.2.1	Dataset()	144
6.24.2.2	~Dataset()	145
6.24.3	Member Function Documentation	145
6.24.3.1	AttributeInteractionInformation()	145
6.24.3.2	CalculateDistanceMatrix() [1/2]	145
6.24.3.3	CalculateDistanceMatrix() [2/2]	146
6.24.3.4	CalculateGainMatrix()	146
6.24.3.5	CalculateInteractionInformation()	147
6.24.3.6	CalculateRegainMatrix()	148
6.24.3.7	CheckHardyWeinbergEquilibrium()	148
6.24.3.8	ComputeInstanceToInstanceDistance()	149

6.24.3.9 CreateDummyAlleles()	149
6.24.3.10 ExcludeMonomorphs()	149
6.24.3.11 ExtractAttributes()	150
6.24.3.12 GetAlternatePhenotypesFilename()	150
6.24.3.13 GetAttribute()	150
6.24.3.14 GetAttributeAlleles()	151
6.24.3.15 GetAttributeIndexFromName()	151
6.24.3.16 GetAttributeMAF()	152
6.24.3.17 GetAttributeMutationType()	152
6.24.3.18 GetAttributeNames()	152
6.24.3.19 GetAttributeRowCol()	153
6.24.3.20 GetAttributeTiTvCounts()	153
6.24.3.21 GetAttributeValues() [1/2]	154
6.24.3.22 GetAttributeValues() [2/2]	154
6.24.3.23 GetClassColumn()	155
6.24.3.24 GetClassIndexes()	155
6.24.3.25 GetClassProbability()	155
6.24.3.26 GetClassValues()	156
6.24.3.27 GetDistanceMetrics()	157
6.24.3.28 GetFileAttributeNames()	157
6.24.3.29 GetFileNumericsNames()	157
6.24.3.30 GetInstance()	157
6.24.3.31 GetInstanceIds()	158
6.24.3.32 GetInstanceIndexForID()	158
6.24.3.33 GetIntForGenotype()	159
6.24.3.34 GetJukesCantorDistance()	159
6.24.3.35 GetKimuraDistance()	159
6.24.3.36 GetMAFs()	159

6.24.3.37 GetMeanForNumeric()	160
6.24.3.38 GetMinMaxForContinuousPhenotype()	161
6.24.3.39 GetMinMaxForNumeric()	161
6.24.3.40 GetNumeric()	162
6.24.3.41 GetNumericIndexFromName()	162
6.24.3.42 GetNumericMatrix()	162
6.24.3.43 GetNumericRowCol()	163
6.24.3.44 GetNumericsFilename()	163
6.24.3.45 GetNumericsNames()	163
6.24.3.46 GetNumericValues() [1/2]	163
6.24.3.47 GetNumericValues() [2/2]	164
6.24.3.48 GetProbabilityValueGivenClass()	164
6.24.3.49 GetRandomInstance()	165
6.24.3.50 GetSnpsFilename()	165
6.24.3.51 GetVariableNames()	165
6.24.3.52 HasAllelicInfo()	166
6.24.3.53 HasAlternatePhenotypes() [1/2]	166
6.24.3.54 HasAlternatePhenotypes() [2/2]	166
6.24.3.55 HasContinuousPhenotypes()	166
6.24.3.56 HasGenotypes()	166
6.24.3.57 HasNumerics() [1/2]	167
6.24.3.58 HasNumerics() [2/2]	167
6.24.3.59 HasPhenotypes()	167
6.24.3.60 IsLoadableInstanceId()	167
6.24.3.61 LoadAlternatePhenotypes()	168
6.24.3.62 LoadDataset() [1/4]	168
6.24.3.63 LoadDataset() [2/4]	169
6.24.3.64 LoadDataset() [3/4]	169

6.24.3.65 LoadDataset() [4/4]	170
6.24.3.66 LoadNumerics()	170
6.24.3.67 LoadOtherDatasetInstances()	170
6.24.3.68 LoadPrivacySim()	171
6.24.3.69 LoadSnps()	171
6.24.3.70 MaskGetAllVariableNames()	172
6.24.3.71 MaskGetAttributeIndices()	172
6.24.3.72 MaskGetAttributeMask()	173
6.24.3.73 MaskGetInstanceIds()	173
6.24.3.74 MaskGetInstanceIndices()	173
6.24.3.75 MaskGetInstanceMask()	174
6.24.3.76 MaskIncludeAllAttributes()	174
6.24.3.77 MaskIncludeAllInstances()	174
6.24.3.78 MaskPopAll()	175
6.24.3.79 MaskPushAll()	175
6.24.3.80 MaskRemoveInstance()	175
6.24.3.81 MaskRemoveVariable()	176
6.24.3.82 MaskRemoveVariableType()	176
6.24.3.83 MaskSearchInstance()	176
6.24.3.84 MaskSearchVariableType()	177
6.24.3.85 MaskWriteNewDataset()	177
6.24.3.86 NumAttributes()	178
6.24.3.87 NumClasses()	178
6.24.3.88 NumInstances()	178
6.24.3.89 NumLevels()	178
6.24.3.90 NumNumerics()	179
6.24.3.91 NumVariables()	179
6.24.3.92 Print()	179

6.24.3.93 PrintAttributeLevelsSeen()	180
6.24.3.94 PrintClassIndexInfo()	180
6.24.3.95 PrintLevelCounts()	180
6.24.3.96 PrintMaskStats()	180
6.24.3.97 PrintMissingValuesStats()	180
6.24.3.98 PrintNumericsStats()	181
6.24.3.99 PrintStats()	181
6.24.3.100PrintStatsSimple()	181
6.24.3.101ProcessExclusionFile()	181
6.24.3.102ResetNearestNeighbors()	182
6.24.3.103RunSnpDiagnosticTests()	182
6.24.3.104SetDistanceMetrics()	182
6.24.3.105SNPHWE()	183
6.24.3.106SwapAttributes()	184
6.24.3.107UpdateAllLevelCounts()	184
6.24.3.108UpdateLevelCounts()	184
6.24.3.109WriteLevelCounts()	185
6.24.3.110WriteNewDataset() [1/2]	185
6.24.3.111WriteNewDataset() [2/2]	186
6.24.3.112WriteNewPlinkCovarDataset()	186
6.24.3.113WriteNewPlinkPedDataset()	187
6.24.3.114WriteSnpTiTvInfo()	187
6.24.4 Member Data Documentation	187
6.24.4.1 alternatePhenotypesFilename	187
6.24.4.2 attributeAlleleCounts	188
6.24.4.3 attributeAlleles	188
6.24.4.4 attributeLevelsSeen	188
6.24.4.5 attributeMinorAllele	188

6.24.4.6 attributeMutationMap	188
6.24.4.7 attributeMutationTypes	189
6.24.4.8 attributeNames	189
6.24.4.9 attributesMask	189
6.24.4.10 attributesMaskPushed	189
6.24.4.11 classColumn	189
6.24.4.12 classIndexes	190
6.24.4.13 continuousPhenotypeMinMax	190
6.24.4.14 engine	190
6.24.4.15 genotypeCounts	190
6.24.4.16 hasAllelicInfo	190
6.24.4.17 hasAlternatePhenotypes	191
6.24.4.18 hasContinuousPhenotypes	191
6.24.4.19 hasGenotypes	191
6.24.4.20 hasNumerics	191
6.24.4.21 hasPhenotypes	191
6.24.4.22 instanceIds	192
6.24.4.23 instanceIdsToLoad	192
6.24.4.24 instances	192
6.24.4.25 instancesMask	192
6.24.4.26 instancesMaskPushed	192
6.24.4.27 levelCounts	193
6.24.4.28 levelCountsByClass	193
6.24.4.29 maskIsPushed	193
6.24.4.30 missingNumericValues	193
6.24.4.31 missingValues	193
6.24.4.32 numDiffFuncPtr	193
6.24.4.33 numDiffMetricName	194

6.24.4.34 numericsFilename	194
6.24.4.35 numericsIds	194
6.24.4.36 numericsMask	195
6.24.4.37 numericsMaskPushed	195
6.24.4.38 numericsMinMax	195
6.24.4.39 numericsNames	195
6.24.4.40 numericsSums	195
6.24.4.41 phenotypesIds	196
6.24.4.42 snpDiffFuncPtr	196
6.24.4.43 snpDiffMetricName	197
6.24.4.44 snpNearestNeighborFuncPtr	197
6.24.4.45 snpNearestNeighborMetricName	198
6.24.4.46 snpsFilename	198
6.25 DatasetInstance Class Reference	198
6.25.1 Detailed Description	202
6.25.2 Constructor & Destructor Documentation	202
6.25.2.1 DatasetInstance()	202
6.25.2.2 ~DatasetInstance()	203
6.25.3 Member Function Documentation	203
6.25.3.1 AddInfluenceFactorD()	203
6.25.3.2 AddNumeric()	203
6.25.3.3 ClearInfluenceFactors()	204
6.25.3.4 GetAttribute()	204
6.25.3.5 GetAttributes()	204
6.25.3.6 GetClass()	205
6.25.3.7 GetCvSetType()	205
6.25.3.8 GetDatasetPtr()	205
6.25.3.9 GetId()	205

6.25.3.10 GetInfluenceFactorD()	205
6.25.3.11 GetNNearestInstances() [1/3]	205
6.25.3.12 GetNNearestInstances() [2/3]	206
6.25.3.13 GetNNearestInstances() [3/3]	206
6.25.3.14 GetNumeric()	207
6.25.3.15 GetNumerics()	207
6.25.3.16 GetPredictedValueTau()	208
6.25.3.17 LoadInstanceFromInstancePtr()	208
6.25.3.18 LoadInstanceFromVector()	208
6.25.3.19 NumAttributes()	208
6.25.3.20 NumNumerics()	209
6.25.3.21 Print()	209
6.25.3.22 PrintDistancePairs()	209
6.25.3.23 ResetNearestNeighbors()	209
6.25.3.24 SetClass()	210
6.25.3.25 SetCvSetType()	210
6.25.3.26 SetDistanceSums() [1/2]	210
6.25.3.27 SetDistanceSums() [2/2]	211
6.25.3.28 SetId()	211
6.25.3.29 SetPredictedValueTau()	211
6.25.3.30 SwapAttributes()	211
6.25.4 Member Data Documentation	212
6.25.4.1 attributes	212
6.25.4.2 bestNeighborhoods	212
6.25.4.3 bestNeighborhoodsDiffClass	212
6.25.4.4 bestNeighborhoodsSameClass	213
6.25.4.5 classLabel	213
6.25.4.6 cvSetType	213

6.25.4.7 dataset	213
6.25.4.8 id	213
6.25.4.9 neighborInfluenceFactorDs	214
6.25.4.10 numerics	214
6.25.4.11 predictedValueTau	214
6.26 DcVar Class Reference	215
6.26.1 Detailed Description	217
6.26.2 Constructor & Destructor Documentation	217
6.26.2.1 DcVar()	217
6.26.2.2 ~DcVar()	217
6.26.3 Member Function Documentation	217
6.26.3.1 CalculateFdrBHThreshold()	217
6.26.3.2 CheckInputs()	218
6.26.3.3 ComputeDifferentialCorrelationZ()	218
6.26.3.4 ComputeDifferentialCorrelationZsparse()	218
6.26.3.5 ComputeDifferentialCorrelationZvals()	218
6.26.3.6 FilterPvalues()	218
6.26.3.7 FindSnps()	219
6.26.3.8 FlattenPvals()	219
6.26.3.9 GetLocalCis()	219
6.26.3.10 GetRadius()	219
6.26.3.11 GetSnpsForTFs()	219
6.26.3.12 GetSnpsForTranscript()	220
6.26.3.13 GetTF()	220
6.26.3.14 GetTFIInfo()	220
6.26.3.15 GetTFRadius()	220
6.26.3.16 IsSnpInTFs()	220
6.26.3.17 LoadDefaultTranscriptionFactorLUT()	221

6.26.3.18 MapPhenosToModel()	221
6.26.3.19 MapSnpIndexToPlinkPhenos()	221
6.26.3.20 PrintState()	221
6.26.3.21 ReadCheckpoint()	221
6.26.3.22 ReadChipSeqFile()	222
6.26.3.23 ReadGeneExpressionFile()	222
6.26.3.24 ReadGenotypesFile()	222
6.26.3.25 ReadSnpLocationsFile()	222
6.26.3.26 ReadTranscriptCoordinates()	222
6.26.3.27 ReadTranscriptFactorCoordinates()	222
6.26.3.28 Run()	223
6.26.3.29 RunOMRF()	223
6.26.3.30 RunOMRFChipSeq()	223
6.26.3.31 RunPlink()	223
6.26.3.32 SetLocalCis()	223
6.26.3.33 SetRadius()	224
6.26.3.34 SetTF()	224
6.26.3.35 SetTFRadius()	224
6.26.3.36 SplitExpressionCaseControl()	224
6.26.3.37 WriteCheckpoint()	224
6.26.3.38 WriteResults()	225
6.26.4 Member Data Documentation	225
6.26.4.1 caselIdxCol	225
6.26.4.2 chipSeqExpression	225
6.26.4.3 chipSeqMode	225
6.26.4.4 coordinates	225
6.26.4.5 ctrlIdxCol	226
6.26.4.6 expressionMatrix	226

6.26.4.7 geneExprNames	226
6.26.4.8 geneExprSubjects	226
6.26.4.9 genotypeMatrix	226
6.26.4.10 genotypeSubjects	226
6.26.4.11 localCis	227
6.26.4.12 nInnerLoop	227
6.26.4.13 nOuterLoop	227
6.26.4.14 numCombs	227
6.26.4.15 pVals	227
6.26.4.16 radius	227
6.26.4.17 snpInputType	228
6.26.4.18 snpLocations	228
6.26.4.19 snpNames	228
6.26.4.20 tfMode	228
6.26.4.21 tfRadius	228
6.26.4.22 tfTableLoaded	228
6.26.4.23 thisTFSnpIndices	229
6.26.4.24 thisTranscriptSnpIndices	229
6.26.4.25 totalTests	229
6.26.4.26 transcriptFactorLUT	229
6.26.4.27 zVals	229
6.27 deref_less Class Reference	230
6.27.1 Detailed Description	231
6.27.2 Member Function Documentation	231
6.27.2.1 operator()()	231
6.28 deref_less_bcw Class Reference	231
6.28.1 Detailed Description	232
6.28.2 Member Function Documentation	232

6.28.2.1 <code>operator()()</code>	232
6.29 Deseq Class Reference	233
6.29.1 Detailed Description	235
6.29.2 Constructor & Destructor Documentation	235
6.29.2.1 <code>Deseq()</code>	235
6.29.2.2 <code>~Deseq()</code>	235
6.29.3 Member Function Documentation	236
6.29.3.1 <code>ComputeScores()</code>	236
6.29.3.2 <code>PrintScores()</code>	236
6.29.3.3 <code>ReadDeseqScores()</code>	237
6.29.3.4 <code>WriteDatasetInMayoFormat()</code>	237
6.29.3.5 <code>WriteScores()</code>	237
6.29.4 Member Data Documentation	237
6.29.4.1 <code>dataset</code>	237
6.29.4.2 <code>scores</code>	238
6.30 DgeData Class Reference	238
6.30.1 Detailed Description	240
6.30.2 Constructor & Destructor Documentation	240
6.30.2.1 <code>DgeData()</code>	240
6.30.2.2 <code>~DgeData()</code>	240
6.30.3 Member Function Documentation	240
6.30.3.1 <code>GetCountsFilename()</code>	240
6.30.3.2 <code>GetGeneCountsSum()</code>	241
6.30.3.3 <code>GetGeneMinMax()</code>	241
6.30.3.4 <code>GetGeneNames()</code>	241
6.30.3.5 <code>GetNormalizationFactors()</code>	241
6.30.3.6 <code>GetNumGenes()</code>	241
6.30.3.7 <code>GetNumSamples()</code>	242

6.30.3.8	GetSampleCounts()	242
6.30.3.9	GetSampleNames()	242
6.30.3.10	GetSamplePhenotype()	242
6.30.3.11	GetSequencingDepthForSample()	242
6.30.3.12	LoadData()	243
6.30.3.13	PrintSampleStats()	243
6.30.4	Member Data Documentation	243
6.30.4.1	counts	243
6.30.4.2	countsFilename	243
6.30.4.3	geneNames	244
6.30.4.4	hasNormFactors	244
6.30.4.5	minMaxGeneCounts	244
6.30.4.6	minMaxSampleCounts	244
6.30.4.7	normFactors	244
6.30.4.8	normsFilename	245
6.30.4.9	phenosFilename	245
6.30.4.10	phenotypes	245
6.30.4.11	sampleNames	245
6.30.4.12	sampleZeroes	245
6.30.4.13	sumGeneCounts	246
6.31	insilico::do_to_lower< charT > Class Template Reference	246
6.31.1	Detailed Description	247
6.31.2	Constructor & Destructor Documentation	247
6.31.2.1	do_to_lower() [1/2]	248
6.31.2.2	do_to_lower() [2/2]	248
6.31.3	Member Function Documentation	248
6.31.3.1	operator()()	248
6.31.4	Member Data Documentation	248

6.31.4.1	m_ctype	248
6.32	insilico::do_to_upper< charT > Class Template Reference	249
6.32.1	Detailed Description	250
6.32.2	Constructor & Destructor Documentation	250
6.32.2.1	do_to_upper() [1/2]	251
6.32.2.2	do_to_upper() [2/2]	251
6.32.3	Member Function Documentation	251
6.32.3.1	operator()()	251
6.32.4	Member Data Documentation	251
6.32.4.1	m_ctype	251
6.33	double2 Class Reference	252
6.33.1	Detailed Description	252
6.33.2	Constructor & Destructor Documentation	252
6.33.2.1	double2() [1/2]	253
6.33.2.2	double2() [2/2]	253
6.33.3	Member Function Documentation	253
6.33.3.1	operator<()	253
6.33.3.2	operator==()	253
6.33.4	Member Data Documentation	253
6.33.4.1	p1	254
6.33.4.2	p2	254
6.34	DPrime Class Reference	254
6.34.1	Detailed Description	255
6.34.2	Member Data Documentation	255
6.34.2.1	dp	255
6.34.2.2	dpl	255
6.34.2.3	dpu	255
6.34.2.4	lod	255

6.35 Edger Class Reference	256
6.35.1 Detailed Description	258
6.35.2 Constructor & Destructor Documentation	258
6.35.2.1 Edger()	258
6.35.2.2 ~Edger()	258
6.35.3 Member Function Documentation	259
6.35.3.1 ComputeScores()	259
6.35.3.2 PrintScores()	259
6.35.3.3 ReadEdgerScores()	260
6.35.3.4 WriteDatasetInMayoFormat()	260
6.35.3.5 WriteScores()	260
6.35.4 Member Data Documentation	260
6.35.4.1 dataset	260
6.35.4.2 scores	261
6.36 Eigen Class Reference	261
6.36.1 Detailed Description	262
6.36.2 Member Function Documentation	262
6.36.2.1 set()	262
6.36.3 Member Data Documentation	262
6.36.3.1 d	262
6.36.3.2 z	262
6.37 EpistasisEQtl Class Reference	263
6.37.1 Detailed Description	265
6.37.2 Constructor & Destructor Documentation	265
6.37.2.1 EpistasisEQtl()	265
6.37.2.2 ~EpistasisEQtl()	265
6.37.3 Member Function Documentation	265
6.37.3.1 CheckInputs()	265

6.37.3.2 GetLocalCis()	265
6.37.3.3 GetRadius()	266
6.37.3.4 GetSnpForTFs()	266
6.37.3.5 GetSnpForTranscript()	266
6.37.3.6 GetTF()	266
6.37.3.7 GetTFInfo()	266
6.37.3.8 GetTFRadius()	267
6.37.3.9 IsSnpInTFs()	267
6.37.3.10 LoadDefaultTranscriptionFactorLUT()	267
6.37.3.11 PrintState()	267
6.37.3.12 ReadTranscriptCoordinates()	267
6.37.3.13 ReadTranscriptFactorCoordinates()	268
6.37.3.14 Run()	268
6.37.3.15 RunEqtl()	268
6.37.3.16 RunIqltICisTrans()	268
6.37.3.17 RunIqltFull()	268
6.37.3.18 SetDebugMode()	269
6.37.3.19 SetLocalCis()	269
6.37.3.20 SetRadius()	269
6.37.3.21 SetTF()	269
6.37.3.22 SetTFRadius()	269
6.37.4 Member Data Documentation	269
6.37.4.1 badModels	270
6.37.4.2 coordinates	270
6.37.4.3 debugMode	270
6.37.4.4 goodModels	270
6.37.4.5 localCis	270
6.37.4.6 nInnerLoop	270

6.37.4.7 nOuterLoop	271
6.37.4.8 outerLoopSnps	271
6.37.4.9 radius	271
6.37.4.10 resultsMatrixBetas	271
6.37.4.11 resultsMatrixPvals	271
6.37.4.12 tfMode	271
6.37.4.13 tfRadius	272
6.37.4.14 tfTableLoaded	272
6.37.4.15 thisTFSnpIndices	272
6.37.4.16 thisTranscriptSnpIndices	272
6.37.4.17 transcriptFactorLUT	272
6.38 EvaporativeCooling Class Reference	273
6.38.1 Detailed Description	275
6.38.2 Constructor & Destructor Documentation	275
6.38.2.1 EvaporativeCooling()	275
6.38.2.2 ~EvaporativeCooling()	276
6.38.3 Member Function Documentation	276
6.38.3.1 ComputeClassificationErrorRandomForest()	276
6.38.3.2 ComputeECScores()	276
6.38.3.3 ComputeFreeEnergy()	276
6.38.3.4 GetAlgorithmType()	277
6.38.3.5 GetECScores()	277
6.38.3.6 GetInteractionScores()	277
6.38.3.7 GetMaineffectScores()	278
6.38.3.8 OptimizeTemperature()	278
6.38.3.9 PrintAllAttributeScores()	278
6.38.3.10 PrintAllScoresTabular()	278
6.38.3.11 PrintAttributeScores()	279

6.38.3.12 PrintInteractionAttributeScores()	279
6.38.3.13 PrintKendallTaus()	279
6.38.3.14 PrintMainEffectAttributeScores()	279
6.38.3.15 RemoveWorstAttributes()	280
6.38.3.16 RunRandomForest()	280
6.38.3.17 RunReliefF()	280
6.38.3.18 WriteAttributeScores()	281
6.38.3.19 WriteClassificationErrors()	282
6.38.3.20 WriteTemperatures()	282
6.38.4 Member Data Documentation	282
6.38.4.1 algorithmType	282
6.38.4.2 analysisType	283
6.38.4.3 bestClassificationError	283
6.38.4.4 classificationErrors	283
6.38.4.5 dataset	283
6.38.4.6 ecScores	284
6.38.4.7 evaporatedAttributes	284
6.38.4.8 freeEnergyScores	284
6.38.4.9 interactionAlgorithm	284
6.38.4.10 interactionScores	284
6.38.4.11 itAlgorithmType	285
6.38.4.12 maineffectAlgorithm	285
6.38.4.13 maineffectScores	285
6.38.4.14 meAlgorithmType	285
6.38.4.15 numRFTThreads	285
6.38.4.16 numTargetAttributes	286
6.38.4.17 numToRemoveNextIteration	286
6.38.4.18 numToRemovePerIteration	286

6.38.4.19 optimalTemperature	286
6.38.4.20 optimizeTemperature	286
6.38.4.21 PP	287
6.38.4.22 temperatures	287
6.39 EvaporativeCoolingPrivacy Class Reference	287
6.39.1 Detailed Description	289
6.39.2 Constructor & Destructor Documentation	289
6.39.2.1 EvaporativeCoolingPrivacy()	289
6.39.2.2 ~EvaporativeCoolingPrivacy()	289
6.39.3 Member Function Documentation	289
6.39.3.1 CheckDetectedAttributes()	290
6.39.3.2 ClassifyAttributeSet()	290
6.39.3.3 ComputeAttributeProbabilities()	290
6.39.3.4 ComputeBestAttributesErrors()	290
6.39.3.5 ComputeImportance()	290
6.39.3.6 ComputeInverseImportance()	291
6.39.3.7 ComputeScores()	291
6.39.3.8 CurrentNumberCorrect()	291
6.39.3.9 EvaporateWorstAttributes()	291
6.39.3.10 GenerateRandomUniformProbabilities()	291
6.39.3.11 GetKeptRemoved()	292
6.39.3.12 PrintState()	292
6.39.3.13 RemoveImportanceScore()	292
6.39.3.14 UpdateTemperature()	292
6.39.3.15 UsingSimData()	292
6.39.3.16 WriteBestAttributes()	292
6.39.4 Member Data Documentation	293
6.39.4.1 attributeProbability	293

6.39.4.2 cummulativeProbabilities	293
6.39.4.3 currentTemp	293
6.39.4.4 curVarMap	293
6.39.4.5 curVarNames	293
6.39.4.6 datalsSimulated	294
6.39.4.7 deltaQ	294
6.39.4.8 diff	294
6.39.4.9 diffImportance	294
6.39.4.10 diffScores	294
6.39.4.11 engine	294
6.39.4.12 finalTemp	295
6.39.4.13 holdError	295
6.39.4.14 holdout	295
6.39.4.15 holdoutErrors	295
6.39.4.16 holdoutImportance	295
6.39.4.17 holdoutInvImportance	295
6.39.4.18 iteration	296
6.39.4.19 keepAttrs	296
6.39.4.20 MAX_ITERATIONS	296
6.39.4.21 numInstances	296
6.39.4.22 numSignalsInData	296
6.39.4.23 numVariables	296
6.39.4.24 possiblyRemove	297
6.39.4.25 PP	297
6.39.4.26 Q_EPS	297
6.39.4.27 randomForestPredictError	297
6.39.4.28 randUniformProbs	297
6.39.4.29 randUniformValue	297

6.39.4.30 removeAttrs	298
6.39.4.31 scaledProbabilities	298
6.39.4.32 signalNames	298
6.39.4.33 startTemp	298
6.39.4.34 summedProbabilities	298
6.39.4.35 tau	298
6.39.4.36 test	299
6.39.4.37 testError	299
6.39.4.38 testErrors	299
6.39.4.39 threshold	299
6.39.4.40 tolerance	299
6.39.4.41 train	299
6.39.4.42 trainError	300
6.39.4.43 trainErrors	300
6.39.4.44 trainImportance	300
6.39.4.45 trainInvImportance	300
6.39.4.46 update	300
6.39.4.47 updateInterval	301
6.40 ExpressionDataSimulator Class Reference	301
6.40.1 Detailed Description	302
6.40.2 Constructor & Destructor Documentation	302
6.40.2.1 ExpressionDataSimulator()	302
6.40.2.2 ~ExpressionDataSimulator()	302
6.40.3 Member Function Documentation	302
6.40.3.1 CreateDiffCoexpMatrixNoME()	303
6.40.3.2 CreateSimulation()	303
6.40.3.3 GetData()	303
6.40.3.4 SimulateData()	303

6.40.4 Member Data Documentation	304
6.40.4.1 colNames	304
6.40.4.2 dimM	304
6.40.4.3 dimN	304
6.40.4.4 n1	304
6.40.4.5 n2	304
6.40.4.6 phenosBase	305
6.40.4.7 PP	305
6.40.4.8 simulatedDataBase	305
6.40.4.9 simulatedDataHoldout	305
6.40.4.10 simulatedDataTest	305
6.40.4.11 simulatedDataTrain	305
6.40.4.12 sublds	306
6.41 Family Class Reference	306
6.41.1 Detailed Description	307
6.41.2 Constructor & Destructor Documentation	307
6.41.2.1 Family()	307
6.41.3 Member Function Documentation	307
6.41.3.1 copy()	307
6.41.3.2 operator=()	308
6.41.4 Member Data Documentation	308
6.41.4.1 B	308
6.41.4.2 discordant_parents	308
6.41.4.3 include	308
6.41.4.4 kid	308
6.41.4.5 mat	309
6.41.4.6 parents	309
6.41.4.7 pat	309

6.41.4.8	sibship	309
6.41.4.9	singleton	309
6.41.4.10	TDT	309
6.42	FamilyTransmissions Class Reference	310
6.42.1	Detailed Description	310
6.42.2	Member Function Documentation	310
6.42.2.1	operator<()	310
6.42.3	Member Data Documentation	311
6.42.3.1	mt	311
6.42.3.2	mu	311
6.42.3.3	pt	311
6.42.3.4	pu	311
6.43	Forest Class Reference	312
6.43.1	Detailed Description	315
6.43.2	Constructor & Destructor Documentation	315
6.43.2.1	Forest()	316
6.43.2.2	~Forest()	316
6.43.3	Member Function Documentation	316
6.43.3.1	computePermutationImportance()	316
6.43.3.2	computePredictionError()	316
6.43.3.3	computePredictionErrorInternal()	316
6.43.3.4	computeTreePermutationImportanceInThread()	317
6.43.3.5	DISALLOW_COPY_AND_ASSIGN()	317
6.43.3.6	getChildNodeIds()	317
6.43.3.7	getDependentVarId()	317
6.43.3.8	getInbagCounts()	317
6.43.3.9	getIsOrderedVariable()	318
6.43.3.10	getMinNodeSize()	318

6.43.3.11 getMtry()	318
6.43.3.12 getNumIndependentVariables()	318
6.43.3.13 getNumTrees()	318
6.43.3.14 getOverallPredictionError()	318
6.43.3.15 getPredictions()	319
6.43.3.16 getPredictionValues()	319
6.43.3.17 getSplitValues()	319
6.43.3.18 getSplitVarIDs()	319
6.43.3.19 getVariableImportance()	319
6.43.3.20 grow()	319
6.43.3.21 growInternal()	320
6.43.3.22 growTreesInThread()	320
6.43.3.23 init()	320
6.43.3.24 initCpp()	321
6.43.3.25 initInternal()	321
6.43.3.26 initR()	322
6.43.3.27 loadFromFile()	322
6.43.3.28 loadFromFileInternal()	322
6.43.3.29 predict()	323
6.43.3.30 predictInternal()	323
6.43.3.31 predictTreesInThread()	323
6.43.3.32 run()	323
6.43.3.33 saveToFile()	323
6.43.3.34 saveToFileInternal()	324
6.43.3.35 setAlwaysSplitVariables()	324
6.43.3.36 setSplitWeightVector()	324
6.43.3.37 setVerboseOutput()	324
6.43.3.38 showProgress()	324

6.43.3.39 writeConfusionFile()	325
6.43.3.40 writeImportanceFile()	325
6.43.3.41 writeOutput()	325
6.43.3.42 writeOutputInternal()	325
6.43.3.43 writePredictionFile()	325
6.43.4 Member Data Documentation	325
6.43.4.1 alpha	326
6.43.4.2 case_weights	326
6.43.4.3 condition_variable	326
6.43.4.4 data	326
6.43.4.5 dependent_varID	326
6.43.4.6 deterministic_varIDs	326
6.43.4.7 holdout	327
6.43.4.8 importance_mode	327
6.43.4.9 is_ordered_variable	327
6.43.4.10 keep_inbag	327
6.43.4.11 memory_mode	327
6.43.4.12 memory_saving_splitting	327
6.43.4.13 min_node_size	328
6.43.4.14 minprop	328
6.43.4.15 mtry	328
6.43.4.16 mutex	328
6.43.4.17 no_split_variables	328
6.43.4.18 num_independent_variables	328
6.43.4.19 num_samples	329
6.43.4.20 num_threads	329
6.43.4.21 num_trees	329
6.43.4.22 num_variables	329

6.43.4.23 output_prefix	329
6.43.4.24 overall_prediction_error	329
6.43.4.25 predict_all	330
6.43.4.26 prediction_mode	330
6.43.4.27 prediction_type	330
6.43.4.28 predictions	330
6.43.4.29 progress	330
6.43.4.30 random_number_generator	330
6.43.4.31 sample_fraction	331
6.43.4.32 sample_with_replacement	331
6.43.4.33 seed	331
6.43.4.34 split_select_varIDs	331
6.43.4.35 split_select_weights	331
6.43.4.36 splitrule	331
6.43.4.37 thread_ranges	332
6.43.4.38 trees	332
6.43.4.39 variable_importance	332
6.43.4.40 verbose_out	332
6.44 ForestClassification Class Reference	333
6.44.1 Detailed Description	335
6.44.2 Constructor & Destructor Documentation	335
6.44.2.1 ForestClassification()	335
6.44.2.2 ~ForestClassification()	335
6.44.3 Member Function Documentation	335
6.44.3.1 computePredictionErrorInternal()	335
6.44.3.2 DISALLOW_COPY_AND_ASSIGN()	336
6.44.3.3 getClassValues()	336
6.44.3.4 growInternal()	336

6.44.3.5	initInternal()	336
6.44.3.6	loadForest()	336
6.44.3.7	loadFromFileInternal()	337
6.44.3.8	predictInternal()	337
6.44.3.9	saveToFileInternal()	337
6.44.3.10	writeConfusionFile()	337
6.44.3.11	writeOutputInternal()	337
6.44.3.12	writePredictionFile()	338
6.44.4	Member Data Documentation	338
6.44.4.1	class_values	338
6.44.4.2	classification_table	338
6.44.4.3	response_classIDs	338
6.45	ForestProbability Class Reference	339
6.45.1	Detailed Description	341
6.45.2	Constructor & Destructor Documentation	341
6.45.2.1	ForestProbability()	341
6.45.2.2	~ForestProbability()	341
6.45.3	Member Function Documentation	341
6.45.3.1	computePredictionErrorInternal()	341
6.45.3.2	DISALLOW_COPY_AND_ASSIGN()	342
6.45.3.3	getClassValues()	342
6.45.3.4	getTerminalClassCounts()	342
6.45.3.5	growInternal()	342
6.45.3.6	initInternal()	342
6.45.3.7	loadForest()	343
6.45.3.8	loadFromFileInternal()	343
6.45.3.9	predictInternal()	343
6.45.3.10	saveToFileInternal()	343

6.45.3.11 writeConfusionFile()	344
6.45.3.12 writeOutputInternal()	344
6.45.3.13 writePredictionFile()	344
6.45.4 Member Data Documentation	344
6.45.4.1 class_values	344
6.45.4.2 classification_table	344
6.45.4.3 response_classIDs	345
6.46 ForestRegression Class Reference	345
6.46.1 Detailed Description	348
6.46.2 Constructor & Destructor Documentation	348
6.46.2.1 ForestRegression()	348
6.46.2.2 ~ForestRegression()	348
6.46.3 Member Function Documentation	348
6.46.3.1 computePredictionErrorInternal()	348
6.46.3.2 DISALLOW_COPY_AND_ASSIGN()	348
6.46.3.3 growInternal()	349
6.46.3.4 initInternal()	349
6.46.3.5 loadForest()	349
6.46.3.6 loadFromFileInternal()	349
6.46.3.7 predictInternal()	350
6.46.3.8 saveToFileInternal()	350
6.46.3.9 writeConfusionFile()	350
6.46.3.10 writeOutputInternal()	350
6.46.3.11 writePredictionFile()	351
6.47 ForestSurvival Class Reference	351
6.47.1 Detailed Description	354
6.47.2 Constructor & Destructor Documentation	354
6.47.2.1 ForestSurvival()	354

6.47.2.2 ~ForestSurvival()	354
6.47.3 Member Function Documentation	354
6.47.3.1 computePredictionErrorInternal()	354
6.47.3.2 DISALLOW_COPY_AND_ASSIGN()	355
6.47.3.3 getChf()	355
6.47.3.4 getStatusVarId()	355
6.47.3.5 getUniqueTimepoints()	355
6.47.3.6 growInternal()	355
6.47.3.7 initInternal()	356
6.47.3.8 loadForest()	356
6.47.3.9 loadFromFileInternal()	356
6.47.3.10 predictInternal()	356
6.47.3.11 saveToFileInternal()	357
6.47.3.12 writeConfusionFile()	357
6.47.3.13 writeOutputInternal()	357
6.47.3.14 writePredictionFile()	357
6.47.4 Member Data Documentation	357
6.47.4.1 response_timepointIDs	358
6.47.4.2 status_varID	358
6.47.4.3 unique_timepoints	358
6.48 fullGenotype Class Reference	359
6.48.1 Detailed Description	359
6.48.2 Constructor & Destructor Documentation	359
6.48.2.1 fullGenotype()	360
6.48.3 Member Function Documentation	360
6.48.3.1 operator<()	360
6.48.4 Member Data Documentation	360
6.48.4.1 a1	360

6.48.4.2 <code>a2</code>	360
6.49 <code>GSLRandomBase</code> Class Reference	361
6.49.1 Detailed Description	362
6.49.2 Constructor & Destructor Documentation	362
6.49.2.1 <code>GSLRandomBase()</code> [1/2]	363
6.49.2.2 <code>GSLRandomBase()</code> [2/2]	363
6.49.2.3 <code>~GSLRandomBase()</code>	363
6.49.3 Member Function Documentation	363
6.49.3.1 <code>nextRandVal()</code>	363
6.49.3.2 <code>state()</code>	363
6.49.4 Member Data Documentation	363
6.49.4.1 <code>rStatePtr_</code>	364
6.50 <code>GSLRandomFlat</code> Class Reference	364
6.50.1 Detailed Description	365
6.50.2 Constructor & Destructor Documentation	366
6.50.2.1 <code>GSLRandomFlat()</code>	366
6.50.2.2 <code>~GSLRandomFlat()</code>	366
6.50.3 Member Function Documentation	366
6.50.3.1 <code>nextRandVal()</code>	366
6.50.4 Member Data Documentation	366
6.50.4.1 <code>lower_</code>	366
6.50.4.2 <code>upper_</code>	367
6.51 <code>GVariant</code> Class Reference	367
6.51.1 Detailed Description	368
6.51.2 Constructor & Destructor Documentation	368
6.51.2.1 <code>GVariant()</code>	368
6.51.3 Member Data Documentation	368
6.51.3.1 <code>allele1</code>	368

6.51.3.2	allele2	368
6.51.3.3	dosage1	368
6.51.3.4	dosage2	369
6.51.3.5	missing	369
6.52	gzfilebuf Class Reference	369
6.52.1	Detailed Description	373
6.52.2	Constructor & Destructor Documentation	373
6.52.2.1	gzfilebuf()	373
6.52.2.2	~gzfilebuf()	373
6.52.3	Member Function Documentation	373
6.52.3.1	attach()	373
6.52.3.2	close()	374
6.52.3.3	disable_buffer()	374
6.52.3.4	enable_buffer()	374
6.52.3.5	is_open()	375
6.52.3.6	open()	375
6.52.3.7	open_mode()	375
6.52.3.8	overflow()	376
6.52.3.9	setbuf()	376
6.52.3.10	setcompression()	377
6.52.3.11	showmany()	377
6.52.3.12	sync()	378
6.52.3.13	underflow()	378
6.52.4	Member Data Documentation	378
6.52.4.1	buffer	378
6.52.4.2	buffer_size	379
6.52.4.3	file	379
6.52.4.4	io_mode	379

6.52.4.5 <code>own_buffer</code>	379
6.52.4.6 <code>own_fd</code>	380
6.53 <code>gzifstream</code> Class Reference	380
6.53.1 Detailed Description	383
6.53.2 Constructor & Destructor Documentation	383
6.53.2.1 <code>gzifstream()</code> [1/3]	383
6.53.2.2 <code>gzifstream()</code> [2/3]	383
6.53.2.3 <code>gzifstream()</code> [3/3]	384
6.53.3 Member Function Documentation	384
6.53.3.1 <code>attach()</code>	384
6.53.3.2 <code>close()</code>	385
6.53.3.3 <code>is_open()</code>	385
6.53.3.4 <code>open()</code>	385
6.53.3.5 <code>rdbuf()</code>	386
6.53.4 Member Data Documentation	386
6.53.4.1 <code>sb</code>	386
6.54 <code>gzofstream</code> Class Reference	386
6.54.1 Detailed Description	389
6.54.2 Constructor & Destructor Documentation	389
6.54.2.1 <code>gzofstream()</code> [1/3]	389
6.54.2.2 <code>gzofstream()</code> [2/3]	389
6.54.2.3 <code>gzofstream()</code> [3/3]	390
6.54.3 Member Function Documentation	390
6.54.3.1 <code>attach()</code>	390
6.54.3.2 <code>close()</code>	391
6.54.3.3 <code>is_open()</code>	391
6.54.3.4 <code>open()</code>	391
6.54.3.5 <code>rdbuf()</code>	392

6.54.4 Member Data Documentation	392
6.54.4.1 sb	392
6.55 gzomanip2< T1, T2 > Class Template Reference	392
6.55.1 Detailed Description	394
6.55.2 Constructor & Destructor Documentation	394
6.55.2.1 gzomanip2()	394
6.55.3 Friends And Related Function Documentation	394
6.55.3.1 operator<<	395
6.55.4 Member Data Documentation	395
6.55.4.1 func	395
6.55.4.2 val1	395
6.55.4.3 val2	395
6.56 HaploPhase Class Reference	396
6.56.1 Detailed Description	399
6.56.2 Constructor & Destructor Documentation	399
6.56.2.1 HaploPhase()	399
6.56.3 Member Function Documentation	399
6.56.3.1 calculateEmpiricalVariance() [1/2]	399
6.56.3.2 calculateEmpiricalVariance() [2/2]	400
6.56.3.3 calculateHaplotypeFrequencies()	400
6.56.3.4 calculateHaplotypePhases()	400
6.56.3.5 consistentNonfounderMalePhaseGivenXGenotypes()	400
6.56.3.6 consistentNonfounderPhaseGivenGenotypes()	400
6.56.3.7 consistentNonfounderPhaseGivenParents()	401
6.56.3.8 dosage()	401
6.56.3.9 dprime()	401
6.56.3.10 enumerateAllPhases()	401
6.56.3.11 enumerateHaplotypes()	401

6.56.3.12 enumerateNonfounderPhase()	402
6.56.3.13 enumeratePhasedWindows()	402
6.56.3.14 freq()	402
6.56.3.15 getSubHaplotypeName()	402
6.56.3.16 haplotypeName()	403
6.56.3.17 haplotypicCC()	403
6.56.3.18 haplotypicQTL()	403
6.56.3.19 haplotypicTDT()	403
6.56.3.20 haplotypicWeightedCC()	403
6.56.3.21 haplotypicWeightedTDT()	404
6.56.3.22 imputeAllHaplotypes()	404
6.56.3.23 imputeGenotype()	404
6.56.3.24 imputeHaplotypes()	404
6.56.3.25 imputeThisHaplotype()	404
6.56.3.26 includeIndividuals()	405
6.56.3.27 mainImputation()	405
6.56.3.28 makeSetFromMap()	405
6.56.3.29 makeSlidingWindow()	405
6.56.3.30 makeSubHaplotypeSet()	405
6.56.3.31 makeTestSet()	406
6.56.3.32 makeWaplotype()	406
6.56.3.33 name()	406
6.56.3.34 nHap()	406
6.56.3.35 performAlternEM()	406
6.56.3.36 performEM_original()	407
6.56.3.37 performHaplotypeTests()	407
6.56.3.38 phaseAllHaplotypes()	407
6.56.3.39 phaseAndScoreNonfounder()	407

6.56.3.40 prunePhase()	407
6.56.3.41 queryGenotype()	408
6.56.3.42 queryThisGenotype()	408
6.56.3.43 readTagFile()	408
6.56.3.44 reportHaplotypeFrequencies()	408
6.56.3.45 reportPhase()	408
6.56.3.46 reportPhaseWideFormat()	409
6.56.3.47 reset()	409
6.56.3.48 resolveWithKids()	409
6.56.3.49 returnHaplotypes()	409
6.56.3.50 returnHaplotypeSet()	409
6.56.3.51 rsq()	410
6.56.3.52 rsq_internal() [1/2]	410
6.56.3.53 rsq_internal() [2/2]	410
6.56.3.54 scoreTransmissions()	410
6.56.3.55 setSpecificSNPs()	411
6.56.3.56 setTestHaplotype()	411
6.56.3.57 testHaplotypeFreq()	411
6.56.3.58 trackSharedHaplotypes()	411
6.56.3.59 trackThisSegment()	411
6.56.3.60 transmissionCount()	412
6.56.3.61 updateForImputation()	412
6.56.3.62 validateNonfounder()	412
6.56.3.63 verboseDisplayWindows()	412
6.56.4 Member Data Documentation	412
6.56.4.1 actual_map	412
6.56.4.2 actual_nw	413
6.56.4.3 ambig	413

6.56.4.4 calculateDp	413
6.56.4.5 case_freq	413
6.56.4.6 cnt_f	413
6.56.4.7 control_freq	413
6.56.4.8 current	414
6.56.4.9 downcoding	414
6.56.4.10 empiricalVariance	414
6.56.4.11 f	414
6.56.4.12 finishWindow	414
6.56.4.13 hap	414
6.56.4.14 hap1	415
6.56.4.15 hap2	415
6.56.4.16 hapi	415
6.56.4.17 haploid	415
6.56.4.18 haploid_np	415
6.56.4.19 haploid_ph_freq	415
6.56.4.20 haploid_ph_hap1	416
6.56.4.21 hapmap	416
6.56.4.22 hapmapb	416
6.56.4.23 HFRQ	416
6.56.4.24 HIMPUTE	416
6.56.4.25 hname	416
6.56.4.26 homozyg	417
6.56.4.27 HPHASE	417
6.56.4.28 HTEST	417
6.56.4.29 include	417
6.56.4.30 new_map	417
6.56.4.31 new_one	417

6.56.4.32 new_pred_allele	418
6.56.4.33 new_pred_locus	418
6.56.4.34 new_pred_weighted_allele	418
6.56.4.35 new_two	418
6.56.4.36 nh	418
6.56.4.37 nonfounders	418
6.56.4.38 np	419
6.56.4.39 ns	419
6.56.4.40 nsh	419
6.56.4.41 nt	419
6.56.4.42 nw	419
6.56.4.43 odds	419
6.56.4.44 P	420
6.56.4.45 p1	420
6.56.4.46 p2	420
6.56.4.47 ph_freq	420
6.56.4.48 ph_hap1	420
6.56.4.49 ph_hap2	420
6.56.4.50 phasemap	421
6.56.4.51 pp	421
6.56.4.52 pvalue	421
6.56.4.53 ratio	421
6.56.4.54 reference_only	421
6.56.4.55 result	421
6.56.4.56 S	422
6.56.4.57 sets	422
6.56.4.58 startWindow	422
6.56.4.59 subhaplotypes	422

6.56.4.60 test_hap	422
6.56.4.61 testSet	422
6.56.4.62 trackedIBS	423
6.56.4.63 trackedN	423
6.56.4.64 trans	423
6.56.4.65 transmissionTotal	423
6.56.4.66 transmissionX	423
6.56.4.67 transmissionX2	423
6.56.4.68 untrans	424
6.56.4.69 useEmpiricalVariance	424
6.56.4.70 validN	424
6.56.4.71 VPHASE	424
6.56.4.72 WGT	424
6.56.4.73 windows	424
6.56.4.74 X	425
6.57 HaploWindow Class Reference	425
6.57.1 Detailed Description	426
6.57.2 Constructor & Destructor Documentation	427
6.57.2.1 HaploWindow()	427
6.57.2.2 ~HaploWindow()	427
6.57.3 Member Function Documentation	427
6.57.3.1 enumerateGenogroups()	427
6.57.3.2 enumerateHaplotypes()	427
6.57.3.3 enumeratePhase()	428
6.57.3.4 expandGenogroups()	428
6.57.3.5 haplotypeName()	428
6.57.3.6 leftStubFrequency()	428
6.57.3.7 performEM()	428

6.57.3.8 pruneGenogroups()	429
6.57.3.9 prunePhase()	429
6.57.3.10 reportPhase()	429
6.57.3.11 rightStubFrequency()	429
6.57.3.12 setStubCodes()	429
6.57.3.13 tallyUnambiguousCounts()	430
6.57.4 Member Data Documentation	430
6.57.4.1 ambig	430
6.57.4.2 converged	430
6.57.4.3 f	430
6.57.4.4 genoGroup	430
6.57.4.5 genotypes	431
6.57.4.6 hap	431
6.57.4.7 hap1	431
6.57.4.8 hap2	431
6.57.4.9 haplo	431
6.57.4.10 hapmap	431
6.57.4.11 iter	432
6.57.4.12 left_passed	432
6.57.4.13 leftStub	432
6.57.4.14 nh	432
6.57.4.15 np	432
6.57.4.16 ns	432
6.57.4.17 P	433
6.57.4.18 pp	433
6.57.4.19 right_passed	433
6.57.4.20 rightStub	433
6.57.4.21 S	433

6.57.4.22 sampleLogLikelihood	433
6.57.4.23 start	434
6.57.4.24 stop	434
6.57.4.25 uc	434
6.57.4.26 zero	434
6.58 HWindow Class Reference	435
6.58.1 Detailed Description	435
6.58.2 Constructor & Destructor Documentation	436
6.58.2.1 HWindow()	436
6.58.3 Member Function Documentation	436
6.58.3.1 recount()	436
6.58.3.2 set()	436
6.58.3.3 shift()	436
6.58.4 Member Data Documentation	436
6.58.4.1 finished	437
6.58.4.2 hetCount	437
6.58.4.3 homCount	437
6.58.4.4 leftHomozyg	437
6.58.4.5 leftMissing	437
6.58.4.6 misCount	437
6.58.4.7 P	438
6.58.4.8 person	438
6.58.4.9 rightHomozyg	438
6.58.4.10 rightMissing	438
6.58.4.11 start	438
6.58.4.12 stop	438
6.58.4.13 valid	439
6.59 IDField Class Reference	439

6.59.1 Detailed Description	440
6.59.2 Constructor & Destructor Documentation	440
6.59.2.1 IDField()	440
6.59.3 Member Function Documentation	440
6.59.3.1 operator<()	440
6.59.3.2 operator==()	441
6.59.4 Member Data Documentation	441
6.59.4.1 aliasList	441
6.59.4.2 attribute	441
6.59.4.3 eqid	441
6.59.4.4 equiv	441
6.59.4.5 joint	442
6.59.4.6 jointName	442
6.59.4.7 masterList	442
6.59.4.8 name	442
6.59.4.9 null	442
6.59.4.10 width	442
6.60 IDFfile Class Reference	443
6.60.1 Detailed Description	443
6.60.2 Constructor & Destructor Documentation	444
6.60.2.1 IDFfile()	444
6.60.3 Member Data Documentation	444
6.60.3.1 alias_delimit	444
6.60.3.2 delimit	444
6.60.3.3 equiv	444
6.60.3.4 fields	444
6.60.3.5 filename	445
6.60.3.6 hasHeader	445

6.60.3.7	injections	445
6.60.3.8	joint	445
6.60.3.9	missingValues	445
6.60.3.10	uniqFieldCount	445
6.61	IDGroup Class Reference	446
6.61.1	Detailed Description	446
6.61.2	Constructor & Destructor Documentation	446
6.61.2.1	IDGroup()	447
6.61.3	Member Function Documentation	447
6.61.3.1	display()	447
6.61.4	Member Data Documentation	447
6.61.4.1	file	447
6.61.4.2	resolved	447
6.61.4.3	values	447
6.62	IDHelper Class Reference	448
6.62.1	Detailed Description	449
6.62.2	Member Function Documentation	449
6.62.2.1	findAllIndividuals()	449
6.62.2.2	findUniqueIndividual()	449
6.62.2.3	idDump()	450
6.62.2.4	idHelp()	450
6.62.2.5	idListAlias()	450
6.62.2.6	idMatch()	450
6.62.2.7	idReplace()	450
6.62.2.8	matchIndividual()	450
6.62.2.9	parseQuery()	451
6.62.2.10	setAlias()	451
6.62.2.11	setJointValues() [1/2]	451

6.62.2.12 <code>setJointValues()</code> [2/2]	451
6.62.3 Member Data Documentation	451
6.62.3.1 <code>attribFields</code>	451
6.62.3.2 <code>dict</code>	452
6.62.3.3 <code>fieldMap</code>	452
6.62.3.4 <code>fields</code>	452
6.62.3.5 <code>files</code>	452
6.62.3.6 <code>idgroup</code>	452
6.62.3.7 <code>idmap</code>	452
6.62.3.8 <code>iField</code>	453
6.62.3.9 <code>jointField</code>	453
6.62.3.10 <code>jointMap</code>	453
6.62.3.11 <code>jointOrder</code>	453
6.62.3.12 <code>uniqueFields</code>	453
6.63 <code>IDValue</code> Class Reference	454
6.63.1 Detailed Description	455
6.63.2 Constructor & Destructor Documentation	455
6.63.2.1 <code>IDValue()</code>	455
6.63.3 Member Function Documentation	455
6.63.3.1 <code>operator"!="()</code>	455
6.63.3.2 <code>operator<()</code>	455
6.63.3.3 <code>operator==()</code>	456
6.63.3.4 <code>singleMatch()</code>	456
6.63.3.5 <code>updateAlias()</code>	456
6.63.4 Member Data Documentation	456
6.63.4.1 <code>field</code>	456
6.63.4.2 <code>jointValue</code>	456
6.63.4.3 <code>value</code>	457

6.64 Individual Class Reference	457
6.64.1 Detailed Description	458
6.64.2 Constructor & Destructor Documentation	459
6.64.2.1 Individual()	459
6.64.3 Member Function Documentation	459
6.64.3.1 countMeioses()	459
6.64.4 Member Data Documentation	459
6.64.4.1 aff	459
6.64.4.2 B	459
6.64.4.3 bcovar	460
6.64.4.4 bnumeric	460
6.64.4.5 clist	460
6.64.4.6 clistMissing	460
6.64.4.7 covar	460
6.64.4.8 family	460
6.64.4.9 fid	461
6.64.4.10 flag	461
6.64.4.11 founder	461
6.64.4.12 gvar	461
6.64.4.13 i1	461
6.64.4.14 i2	462
6.64.4.15 iid	462
6.64.4.16 ikids	462
6.64.4.17 im	462
6.64.4.18 ip	462
6.64.4.19 kids	462
6.64.4.20 mat	463
6.64.4.21 missing	463

6.64.4.22 missing2	463
6.64.4.23 nlist	463
6.64.4.24 nlistMissing	463
6.64.4.25 one	463
6.64.4.26 pat	464
6.64.4.27 phenotype	464
6.64.4.28 plist	464
6.64.4.29 plistMissing	464
6.64.4.30 pm	464
6.64.4.31 pp	464
6.64.4.32 pperson	465
6.64.4.33 sex	465
6.64.4.34 sexcode	465
6.64.4.35 sol	465
6.64.4.36 T	465
6.64.4.37 two	465
6.64.4.38 W	466
6.64.4.39 wmlocus	466
6.65 indivPair Class Reference	466
6.65.1 Detailed Description	467
6.65.2 Member Function Documentation	467
6.65.2.1 operator<()	467
6.65.3 Member Data Documentation	467
6.65.3.1 p1	467
6.65.3.2 p2	467
6.66 int2 Class Reference	468
6.66.1 Detailed Description	468
6.66.2 Constructor & Destructor Documentation	468

6.66.2.1	int2() [1/2]	469
6.66.2.2	int2() [2/2]	469
6.66.3	Member Function Documentation	469
6.66.3.1	operator<()	469
6.66.3.2	operator==()	469
6.66.4	Member Data Documentation	469
6.66.4.1	p1	470
6.66.4.2	p2	470
6.67	InteractionNetwork Class Reference	470
6.67.1	Detailed Description	472
6.67.2	Constructor & Destructor Documentation	472
6.67.2.1	InteractionNetwork() [1/2]	473
6.67.2.2	InteractionNetwork() [2/2]	473
6.67.2.3	~InteractionNetwork()	473
6.67.3	Member Function Documentation	473
6.67.3.1	AddModule()	473
6.67.3.2	ApplyFisherTransform()	474
6.67.3.3	ApplyPowerTransform()	474
6.67.3.4	CheckIndices()	474
6.67.3.5	CheckMergeResults()	474
6.67.3.6	ComputeQ()	474
6.67.3.7	DebugMessage()	475
6.67.3.8	Deconvolve()	475
6.67.3.9	FlattenModules()	475
6.67.3.10	GetAdjacencyMatrix()	475
6.67.3.11	GetConnectivityMatrix()	475
6.67.3.12	GetModules()	476
6.67.3.13	GetNewmanModules()	476

6.67.3.14 GetNodeNames()	476
6.67.3.15 Homophily()	476
6.67.3.16 Merge()	476
6.67.3.17 MergeSmallModules()	477
6.67.3.18 ModularityBestSplit()	477
6.67.3.19 ModularityLeadingEigenvector()	477
6.67.3.20 NumNodes()	477
6.67.3.21 PrepareConnectivityMatrix()	477
6.67.3.22 PrintAdjacencyMatrix()	478
6.67.3.23 PrintConnectivityMatrix()	478
6.67.3.24 PrintModulesSummary()	478
6.67.3.25 PrintSummary()	478
6.67.3.26 ReadBrainCorr1DFile()	478
6.67.3.27 ReadCsvFile()	478
6.67.3.28 ReadGainFile()	479
6.67.3.29 ReadSifFile()	479
6.67.3.30 RecursiveIndirectPathsModularity()	479
6.67.3.31 ripM()	479
6.67.3.32 SaveModules()	479
6.67.3.33 SetBinaryThresholding()	480
6.67.3.34 SetConnectivityThreshold()	480
6.67.3.35 SetConnectivityThresholdAbs()	480
6.67.3.36 SetConnectivityThresholding()	480
6.67.3.37 SetDebugMode()	480
6.67.3.38 SetModulesFromFile()	481
6.67.3.39 ShowHomophily()	481
6.67.3.40 ShowModuleIndices()	481
6.67.3.41 ShowModules()	481

6.67.3.42 ShowModuleSizes()	481
6.67.3.43 SumMatrixPowerSeries()	482
6.67.3.44 WriteDelimitedFile()	482
6.67.3.45 WriteSifFile()	482
6.67.3.46 WriteToFile()	482
6.67.4 Member Data Documentation	482
6.67.4.1 adjMatrix	483
6.67.4.2 connectivityThreshold	483
6.67.4.3 connectivityThresholdAbs	483
6.67.4.4 connMatrix	483
6.67.4.5 debugMode	483
6.67.4.6 degrees	483
6.67.4.7 inbixEnv	484
6.67.4.8 maxMergeOrder	484
6.67.4.9 maxModuleSize	484
6.67.4.10 minModuleSize	484
6.67.4.11 modules	484
6.67.4.12 networkFile	484
6.67.4.13 nodeNameIndex	485
6.67.4.14 nodeNames	485
6.67.4.15 numEdges	485
6.67.4.16 numNodes	485
6.67.4.17 Q	485
6.67.4.18 ripmMatrix	485
6.67.4.19 ripmResult	486
6.67.4.20 startMergeOrder	486
6.67.4.21 useBinaryThreshold	486
6.67.4.22 useConnectivityThreshold	486

6.68 insilico::is_classified< Type, charT > Class Template Reference	487
6.68.1 Detailed Description	488
6.68.2 Constructor & Destructor Documentation	488
6.68.2.1 is_classified() [1/2]	489
6.68.2.2 is_classified() [2/2]	489
6.68.3 Member Function Documentation	489
6.68.3.1 operator()()	489
6.68.4 Member Data Documentation	489
6.68.4.1 m_ctype	489
6.69 LDPair Class Reference	490
6.69.1 Detailed Description	490
6.69.2 Constructor & Destructor Documentation	490
6.69.2.1 LDPair()	491
6.69.3 Member Function Documentation	491
6.69.3.1 operator<()	491
6.69.4 Friends And Related Function Documentation	491
6.69.4.1 operator<<	491
6.69.5 Member Data Documentation	491
6.69.5.1 dist	491
6.69.5.2 Id	492
6.69.5.3 s1	492
6.69.5.4 s2	492
6.70 std::less< Locus * > Class Template Reference	492
6.70.1 Detailed Description	493
6.70.2 Member Function Documentation	493
6.70.2.1 operator()()	493
6.71 std::less< MultiLocusGenotype * > Class Template Reference	493
6.71.1 Detailed Description	494

6.71.2 Member Function Documentation	494
6.71.2.1 operator()()	494
6.72 std::less< Pool * > Class Template Reference	494
6.72.1 Detailed Description	495
6.72.2 Member Function Documentation	495
6.72.2.1 operator()()	495
6.73 std::less< Segment * > Class Template Reference	495
6.73.1 Detailed Description	496
6.73.2 Member Function Documentation	496
6.73.2.1 operator()()	496
6.74 LinearModel Class Reference	496
6.74.1 Detailed Description	499
6.74.2 Constructor & Destructor Documentation	499
6.74.2.1 LinearModel()	499
6.74.2.2 ~LinearModel()	499
6.74.3 Member Function Documentation	499
6.74.3.1 calculateAdjustedRSquared()	500
6.74.3.2 calculateFTest()	500
6.74.3.3 calculateMallowC()	500
6.74.3.4 calculateRSquared()	500
6.74.3.5 calculateRSS()	500
6.74.3.6 displayResults()	501
6.74.3.7 fitLM()	501
6.74.3.8 fitUnivariateLM()	501
6.74.3.9 function()	501
6.74.3.10 getCoefs()	501
6.74.3.11 getPVals()	502
6.74.3.12 getPValue()	502

6.74.3.13 getSE()	502
6.74.3.14 getVar()	502
6.74.3.15 HuberWhite()	502
6.74.3.16 pruneY()	503
6.74.3.17 reset()	503
6.74.3.18 setDependent()	503
6.74.3.19 setVariance()	503
6.74.3.20 standardise()	503
6.74.4 Member Data Documentation	503
6.74.4.1 C	504
6.74.4.2 chisq	504
6.74.4.3 meanY	504
6.74.4.4 RSS	504
6.74.4.5 se	504
6.74.4.6 sig	504
6.74.4.7 u	505
6.74.4.8 v	505
6.74.4.9 varY	505
6.74.4.10 w	505
6.74.4.11 Y	505
6.75 Locus Class Reference	506
6.75.1 Detailed Description	508
6.75.2 Constructor & Destructor Documentation	508
6.75.2.1 Locus() [1/2]	508
6.75.2.2 Locus() [2/2]	508
6.75.3 Member Function Documentation	508
6.75.3.1 copy()	508
6.75.3.2 operator<()	509

6.75.3.3 <code>operator=()</code>	509
6.75.3.4 <code>operator==(())</code>	509
6.75.4 Member Data Documentation	509
6.75.4.1 <code>allele1</code>	509
6.75.4.2 <code>allele2</code>	509
6.75.4.3 <code>bp</code>	510
6.75.4.4 <code>chr</code>	510
6.75.4.5 <code>freq</code>	510
6.75.4.6 <code>name</code>	510
6.75.4.7 <code>nm</code>	510
6.75.4.8 <code>pos</code>	510
6.76 LogisticModel Class Reference	511
6.76.1 Detailed Description	512
6.76.2 Constructor & Destructor Documentation	513
6.76.2.1 <code>LogisticModel()</code>	513
6.76.2.2 <code>~LogisticModel()</code>	513
6.76.3 Member Function Documentation	513
6.76.3.1 <code>displayResults()</code>	513
6.76.3.2 <code>fitLM()</code>	513
6.76.3.3 <code>fitUnivariateLM()</code>	514
6.76.3.4 <code>getCoefs()</code>	514
6.76.3.5 <code>getLnLk()</code>	514
6.76.3.6 <code>getPVals()</code>	514
6.76.3.7 <code>getPValue()</code>	514
6.76.3.8 <code>getSE()</code>	515
6.76.3.9 <code>getVar()</code>	515
6.76.3.10 <code>HuberWhite()</code>	515
6.76.3.11 <code>pruneY()</code>	515

6.76.3.12 reset()	515
6.76.3.13 setDependent()	516
6.76.4 Member Data Documentation	516
6.76.4.1 chisq	516
6.76.4.2 p	516
6.76.4.3 V	516
6.76.4.4 Y	516
6.77 MainExitException Class Reference	517
6.77.1 Detailed Description	517
6.77.2 Member Function Documentation	517
6.77.2.1 exitMessage()	517
6.78 Model Class Reference	518
6.78.1 Detailed Description	521
6.78.2 Constructor & Destructor Documentation	521
6.78.2.1 Model()	521
6.78.2.2 ~Model()	521
6.78.3 Member Function Documentation	522
6.78.3.1 addAdditiveSNP()	522
6.78.3.2 addCovariate()	522
6.78.3.3 addDominanceSNP()	522
6.78.3.4 addHaplotypeDosage()	522
6.78.3.5 addInteraction()	522
6.78.3.6 addNumeric()	523
6.78.3.7 addSexEffect()	523
6.78.3.8 addTypedInteraction()	523
6.78.3.9 buildAdditive()	523
6.78.3.10 buildCovariate()	523
6.78.3.11 buildDesignMatrix()	524

6.78.3.12 buildDominance()	524
6.78.3.13 buildHaplotype()	524
6.78.3.14 buildInteraction()	524
6.78.3.15 buildIntercept()	524
6.78.3.16 buildNumeric()	525
6.78.3.17 buildQFAM()	525
6.78.3.18 buildSex()	525
6.78.3.19 buildTypedInteraction()	525
6.78.3.20 checkVIF()	525
6.78.3.21 displayResults()	526
6.78.3.22 fitConverged()	526
6.78.3.23 fitLM()	526
6.78.3.24 fitNumIterations()	526
6.78.3.25 fitUnivariateLM()	526
6.78.3.26 getCoefs()	527
6.78.3.27 getMissing()	527
6.78.3.28 getNP()	527
6.78.3.29 getPVals()	527
6.78.3.30 getSE()	527
6.78.3.31 getSimpleSNPValue()	527
6.78.3.32 getStatistic()	528
6.78.3.33 getVar()	528
6.78.3.34 hasSNPs()	528
6.78.3.35 HuberWhite()	528
6.78.3.36 isSexInModel()	528
6.78.3.37 isValid()	528
6.78.3.38 linearHypothesis()	529
6.78.3.39 noCluster()	529

6.78.3.40 pruneY()	529
6.78.3.41 setCluster()	529
6.78.3.42 setDependent()	529
6.78.3.43 setDominant()	530
6.78.3.44 setHaploid()	530
6.78.3.45 setMissing() [1/2]	530
6.78.3.46 setMissing() [2/2]	530
6.78.3.47 setRecessive()	530
6.78.3.48 setValid()	530
6.78.3.49 setX()	531
6.78.3.50 validParameters()	531
6.78.3.51 yokeMissing()	531
6.78.3.52 Ysize()	531
6.78.4 Member Data Documentation	531
6.78.4.1 additive	531
6.78.4.2 all_valid	532
6.78.4.3 clst	532
6.78.4.4 cluster	532
6.78.4.5 coef	532
6.78.4.6 converged	532
6.78.4.7 covariate	532
6.78.4.8 dominance	533
6.78.4.9 haploid	533
6.78.4.10 haplotype	533
6.78.4.11 has_snps	533
6.78.4.12 interaction	533
6.78.4.13 label	533
6.78.4.14 mA	534

6.78.4.15 mAA	534
6.78.4.16 mAB	534
6.78.4.17 mB	534
6.78.4.18 mBB	534
6.78.4.19 miss	534
6.78.4.20 nc	535
6.78.4.21 nind	535
6.78.4.22 np	535
6.78.4.23 numeric	535
6.78.4.24 numIterations	535
6.78.4.25 order	535
6.78.4.26 P	536
6.78.4.27 S	536
6.78.4.28 sex_effect	536
6.78.4.29 skip	536
6.78.4.30 testParameter	536
6.78.4.31 type	536
6.78.4.32 typed_interaction	537
6.78.4.33 valid	537
6.78.4.34 X	537
6.78.4.35 xchr	537
6.79 MultiLocusGenotype Class Reference	538
6.79.1 Detailed Description	538
6.79.2 Member Function Documentation	539
6.79.2.1 operator<()	539
6.79.2.2 operator==(())	539
6.79.3 Member Data Documentation	539
6.79.3.1 count	539

6.79.3.2	g	539
6.79.3.3	reference	540
6.79.3.4	skip	540
6.80	Neighbour Class Reference	540
6.80.1	Detailed Description	541
6.80.2	Member Function Documentation	541
6.80.2.1	operator<()	541
6.80.3	Member Data Documentation	541
6.80.3.1	dist	541
6.80.3.2	neighbour	541
6.81	NList Class Reference	542
6.81.1	Detailed Description	543
6.81.2	Constructor & Destructor Documentation	543
6.81.2.1	NList()	543
6.81.3	Member Function Documentation	543
6.81.3.1	deparseNumberList()	544
6.81.3.2	deparseStringList() [1/2]	544
6.81.3.3	deparseStringList() [2/2]	544
6.81.3.4	expandNumberList()	544
6.81.3.5	setDelimiter()	544
6.81.3.6	setRangeChar()	545
6.81.3.7	tokenize()	545
6.81.4	Member Data Documentation	545
6.81.4.1	delimit_char	545
6.81.4.2	firstWord	545
6.81.4.3	lastWord	545
6.81.4.4	maxcat	546
6.81.4.5	negmode	546

6.81.4.6 <code>range_char</code>	546
6.81.4.7 <code>range_string</code>	546
6.82 Options Class Reference	547
6.82.1 Detailed Description	548
6.82.2 Constructor & Destructor Documentation	548
6.82.2.1 <code>~Options()</code>	548
6.82.3 Member Function Documentation	548
6.82.3.1 <code>addOption()</code>	548
6.82.3.2 <code>getOptions()</code>	548
6.82.4 Member Data Documentation	548
6.82.4.1 <code>opt</code>	549
6.83 OptionSet Class Reference	549
6.83.1 Detailed Description	551
6.83.2 Member Function Documentation	551
6.83.2.1 <code>display()</code>	551
6.83.2.2 <code>getValue()</code>	551
6.83.2.3 <code>getValues()</code>	551
6.83.2.4 <code>isSet()</code>	551
6.83.3 Member Data Documentation	552
6.83.3.1 <code>val</code>	552
6.84 Pair Class Reference	552
6.84.1 Detailed Description	553
6.84.2 Member Function Documentation	553
6.84.2.1 <code>operator<() [1/2]</code>	553
6.84.2.2 <code>operator<() [2/2]</code>	553
6.84.3 Member Data Documentation	553
6.84.3.1 <code> </code>	553
6.84.3.2 <code>p</code>	554

6.84.3.3 p1	554
6.84.3.4 p2	554
6.85 Pair2 Class Reference	554
6.85.1 Detailed Description	555
6.85.2 Member Function Documentation	555
6.85.2.1 operator<()	555
6.85.3 Member Data Documentation	555
6.85.3.1 l	555
6.85.3.2 p	555
6.86 Pair_cmp Struct Reference	556
6.86.1 Detailed Description	556
6.86.2 Member Function Documentation	556
6.86.2.1 operator()()	556
6.87 PairwiseLinkage Class Reference	557
6.87.1 Detailed Description	558
6.87.2 Constructor & Destructor Documentation	558
6.87.2.1 PairwiseLinkage()	558
6.87.3 Member Function Documentation	558
6.87.3.1 calculateCI()	558
6.87.3.2 calculateLD()	558
6.87.4 Member Data Documentation	558
6.87.4.1 a	559
6.87.4.2 b	559
6.87.4.3 dp	559
6.87.4.4 dp_lower	559
6.87.4.5 dp_upper	559
6.87.4.6 knownAA	559
6.87.4.7 knownAB	560

6.87.4.8 knownBA	560
6.87.4.9 knownBB	560
6.87.4.10 lod	560
6.87.4.11 rsq	560
6.87.4.12 unknownDH	560
6.88 par Class Reference	561
6.88.1 Detailed Description	582
6.88.2 Member Data Documentation	582
6.88.2.1 adaptive_alpha	582
6.88.2.2 adaptive_ci	583
6.88.2.3 adaptive_interval	583
6.88.2.4 adaptive_interval2	583
6.88.2.5 adaptive_max	583
6.88.2.6 adaptive_min	583
6.88.2.7 adaptive_perm	583
6.88.2.8 af_count	584
6.88.2.9 af_file	584
6.88.2.10 af_read	584
6.88.2.11 af_write	584
6.88.2.12 affpair	584
6.88.2.13 afni1dFile	584
6.88.2.14 afniNetwork	585
6.88.2.15 algorithm_verbose	585
6.88.2.16 algorithmMode	585
6.88.2.17 algorithmSeqMode	585
6.88.2.18 algorithmSeqS0	585
6.88.2.19 algorithmSnrMode	585
6.88.2.20 algorithmTstatMode	586

6.88.2.21 all_pheno	586
6.88.2.22 alpha	586
6.88.2.23 alwaysplitvars	586
6.88.2.24 analyse_cluster	586
6.88.2.25 annot_file	586
6.88.2.26 annot_filename	587
6.88.2.27 assoc_counts	587
6.88.2.28 assoc_glm	587
6.88.2.29 assoc_glm_without_main_snp	587
6.88.2.30 assoc_gxe	587
6.88.2.31 assoc_test	587
6.88.2.32 assoc_test_alt_perm	588
6.88.2.33 BATCH_SIZE	588
6.88.2.34 bitfilename	588
6.88.2.35 bitfilename_map	588
6.88.2.36 bmatch	588
6.88.2.37 bmatch_direction_filename	588
6.88.2.38 bmatch_filename	589
6.88.2.39 bmatch_usertype	589
6.88.2.40 boot	589
6.88.2.41 bound	589
6.88.2.42 breslowday	589
6.88.2.43 bt	589
6.88.2.44 built_families	590
6.88.2.45 calc_SNPSNP_LD	590
6.88.2.46 calcFst	590
6.88.2.47 caseweights	590
6.88.2.48 catvars	590

6.88.2.49 chap_add_grp_specifics	590
6.88.2.50 chap_drop_snps	591
6.88.2.51 chap_drop_snps_list	591
6.88.2.52 chap_entity	591
6.88.2.53 chap_haplotype_specific	591
6.88.2.54 chap_independent_effect	591
6.88.2.55 chap_model0	591
6.88.2.56 chap_model1	592
6.88.2.57 chap_sole_variant	592
6.88.2.58 chap_sole_variant_specific_allele_list	592
6.88.2.59 chap_sole_variant_specific_alleles	592
6.88.2.60 chap_specified_groups	592
6.88.2.61 chap_specified_snps	592
6.88.2.62 chap_test	593
6.88.2.63 check_sex	593
6.88.2.64 chr_code	593
6.88.2.65 chr_haploid	593
6.88.2.66 chr_map	593
6.88.2.67 chr_sex	593
6.88.2.68 chr_Y	594
6.88.2.69 ci_level	594
6.88.2.70 ci_zt	594
6.88.2.71 cli	594
6.88.2.72 clist	594
6.88.2.73 clist_filename	594
6.88.2.74 clist_number	595
6.88.2.75 clist_selection	595
6.88.2.76 clist_selection_name	595

6.88.2.77 <code>clist_selection_number</code>	595
6.88.2.78 <code>clist_selection_string</code>	595
6.88.2.79 <code>clumpld</code>	595
6.88.2.80 <code>clumpld_annot</code>	596
6.88.2.81 <code>clumpld_annot_fields</code>	596
6.88.2.82 <code>clumpld_best</code>	596
6.88.2.83 <code>clumpld_column</code>	596
6.88.2.84 <code>clumpld_indep</code>	596
6.88.2.85 <code>clumpld_index1</code>	596
6.88.2.86 <code>clumpld_kb</code>	597
6.88.2.87 <code>clumpld_min</code>	597
6.88.2.88 <code>clumpld_only_show_replications</code>	597
6.88.2.89 <code>clumpld_only_show_replications_list</code>	597
6.88.2.90 <code>clumpld_p1</code>	597
6.88.2.91 <code>clumpld_p2</code>	597
6.88.2.92 <code>clumpld_r2</code>	598
6.88.2.93 <code>clumpld_range_annotation</code>	598
6.88.2.94 <code>clumpld_range_file</code>	598
6.88.2.95 <code>clumpld_results</code>	598
6.88.2.96 <code>clumpld_verbose</code>	598
6.88.2.97 <code>cluster</code>	598
6.88.2.98 <code>cluster_euclidean</code>	599
6.88.2.99 <code>cluster_group_avg</code>	599
6.88.2.100 <code>cluster_ibm_constraint</code>	599
6.88.2.101 <code>cluster_ibm_constraint_value</code>	599
6.88.2.102 <code>cluster_mds_dim</code>	599
6.88.2.103 <code>cluster_missing</code>	599
6.88.2.104 <code>cluster_on_mcc</code>	600

6.88.2.105cluster_on_phenotype	600
6.88.2.106cluster_plot	600
6.88.2.107cluster_selcon	600
6.88.2.108cluster_selcon_file	600
6.88.2.109cm_map	600
6.88.2.110CMH_test_1	601
6.88.2.111CMH_test_2	601
6.88.2.112CMH_test_ORD	601
6.88.2.113cnv_check_overlap	601
6.88.2.114cnv_col	601
6.88.2.115cnv_count	601
6.88.2.116cnv_count_baseline	602
6.88.2.117cnv_count_baseline_file	602
6.88.2.118cnv_defined_overlap	602
6.88.2.119cnv_del_only	602
6.88.2.120cnv_disrupt	602
6.88.2.121cnv_drop_no_segment	602
6.88.2.122cnv_dup_only	603
6.88.2.123cnv_en_model	603
6.88.2.124cnv_enrichment_test	603
6.88.2.125cnv_exclude	603
6.88.2.126cnv_freq_include	603
6.88.2.127cnv_freq_include_below	603
6.88.2.128cnv_freq_include_cnt	604
6.88.2.129cnv_freq_include_exact	604
6.88.2.130cnv_freq_include_exact_exclude	604
6.88.2.131cnv_freq_method2	604
6.88.2.132cnv_freq_method2_threshold	604

6.88.2.133cnv_glm	604
6.88.2.134cnv_indiv_perm	605
6.88.2.135cnv_intersect	605
6.88.2.136cnv_intersect_file	605
6.88.2.137cnv_intersect_subset	605
6.88.2.138cnv_intersect_subset_file	605
6.88.2.139cnv_intersect_writeback	605
6.88.2.140cnv_intersect_writeback_verbose	606
6.88.2.141cnv_list	606
6.88.2.142cnv_listname	606
6.88.2.143cnv_makemap	606
6.88.2.144cnv_max_kb	606
6.88.2.145cnv_max_score	606
6.88.2.146cnv_max_sites	607
6.88.2.147cnv_min_kb	607
6.88.2.148cnv_min_score	607
6.88.2.149cnv_min_sites	607
6.88.2.150cnv_overlap	607
6.88.2.151cnv_pos_perm	607
6.88.2.152cnv_region_border	608
6.88.2.153cnv_region_overlap	608
6.88.2.154cnv_type	608
6.88.2.155cnv_union_overlap	608
6.88.2.156cnv_unique	608
6.88.2.157cnv_weighted_gene_test	608
6.88.2.158cnv_write_freq	609
6.88.2.159cnv_writelist	609
6.88.2.160coding01	609

6.88.2.161compound_genotype_code	609
6.88.2.162compress_file	609
6.88.2.163compress_filename	609
6.88.2.164compress_genome	610
6.88.2.165conditioning_snp_name	610
6.88.2.166conditioning_snp_single	610
6.88.2.167conditioning_snps	610
6.88.2.168conditioning_snps_file	610
6.88.2.169covar_file	610
6.88.2.170covar_filename	611
6.88.2.171CP	611
6.88.2.172dcvar_chip_seq_file	611
6.88.2.173dcvar_gene_expression_file	611
6.88.2.174dcvar_genotypes_file	611
6.88.2.175dcvar_pfilter_type	611
6.88.2.176dcvar_pfilter_value	612
6.88.2.177dcvar_radius	612
6.88.2.178dcvar_resume.snp	612
6.88.2.179dcvar_snp_locations_file	612
6.88.2.180dcvar_var_model	612
6.88.2.181debug	612
6.88.2.182deconvolutionAlpha	613
6.88.2.183deconvolutionBeta	613
6.88.2.184deconvolutionControl	613
6.88.2.185depvarname	613
6.88.2.186dfam_sibs	613
6.88.2.187dfam_tdt	613
6.88.2.188dfam_unrelateds	614

6.88.2.189	discordant_parents	614
6.88.2.190	disp_r1	614
6.88.2.191	disp_r2	614
6.88.2.192	disp_r_window	614
6.88.2.193	disp_r_window_kb	614
6.88.2.194	disp_r_window_r2	615
6.88.2.195	disp_r_window.snp	615
6.88.2.196	display_ci	615
6.88.2.197	display_cnv_track	615
6.88.2.198	display_hap_freqs	615
6.88.2.199	display_phase_probs	615
6.88.2.200	display_phase_probs_wide	616
6.88.2.201	display_segment_long	616
6.88.2.202	distance_matrix	616
6.88.2.203	distanceMatrixFilename	616
6.88.2.204	do_coexpression_all	616
6.88.2.205	do_coexpression_casecontrol	616
6.88.2.206	do_covariance_matrix	617
6.88.2.207	do_dcgain_abs	617
6.88.2.208	do_dcvar	617
6.88.2.209	do_dcvar_chipseq	617
6.88.2.210	do_dcvar_pffilter	617
6.88.2.211	do_deconvolution	617
6.88.2.212	do_differential_coexpression	618
6.88.2.213	do_differential_modularity	618
6.88.2.214	do_dmgain_abs	618
6.88.2.215	do_ec	618
6.88.2.216	do_ec_privacy	618

6.88.2.217do_iqtl	618
6.88.2.218do_iqtl_tf	619
6.88.2.219do_iterative_removal	619
6.88.2.220do_iterative_removal_ec	619
6.88.2.221do_iterative_write_scores	619
6.88.2.222do_kopt	619
6.88.2.223do_modularity	619
6.88.2.224do_normalize_scores	620
6.88.2.225do_not_load_snps	620
6.88.2.226do_numeric_extract	620
6.88.2.227do_numeric_lowval_filter	620
6.88.2.228do_numeric_lowvar_filter	620
6.88.2.229do_numeric_standardize	620
6.88.2.230do_numeric_summary	621
6.88.2.231do_randomforest	621
6.88.2.232do_ranker_permutation	621
6.88.2.233do_ranking	621
6.88.2.234do_regain	621
6.88.2.235do_regain_post	621
6.88.2.236do_regain_pvalue_threshold	622
6.88.2.237do_relief	622
6.88.2.238do_rfprobability	622
6.88.2.239do_ripm	622
6.88.2.240do_write_best_k	622
6.88.2.241do_write_each_k_scores	622
6.88.2.242done_global_pihat	623
6.88.2.243dosage_assoc	623
6.88.2.244dosage_file	623

6.88.2.245dosage_hard_call	623
6.88.2.246dosage_hard_call_thresh	623
6.88.2.247dosage_hard_call_thresh2	623
6.88.2.248dosage_hasMap	624
6.88.2.249drop_sets	624
6.88.2.250dummy	624
6.88.2.251dummy_nind	624
6.88.2.252dummy_nsnp	624
6.88.2.253dump_clst	624
6.88.2.254dump_covar	625
6.88.2.255dump_covar_dummy_coding	625
6.88.2.256dump_covar_with_phenotype	625
6.88.2.257dump_gene	625
6.88.2.258dump_genename	625
6.88.2.259dumppped	625
6.88.2.260ecAlgorithmSteps	626
6.88.2.261ecItAlgorithm	626
6.88.2.262ecIterNumToRemove	626
6.88.2.263ecIterPercentToRemove	626
6.88.2.264ecMeAlgorithm	626
6.88.2.265ecNumTarget	626
6.88.2.266ecOptimizeTemp	627
6.88.2.267ecPrivacyFinalTemp	627
6.88.2.268ecPrivacyHoldoutFile	627
6.88.2.269ecPrivacyPercentSignal	627
6.88.2.270ecPrivacyStartTemp	627
6.88.2.271ecPrivacyTau	627
6.88.2.272ecPrivacyTestFile	628

6.88.2.273 <code>ecPrivacyTrainFile</code>	628
6.88.2.274 <code>ecPrivacyUpdateFrequency</code>	628
6.88.2.275 <code>elf_baseline</code>	628
6.88.2.276 <code>elf_pcmode</code>	628
6.88.2.277 <code>elf_pcmode_2sided</code>	628
6.88.2.278 <code>epi_alpha1</code>	629
6.88.2.279 <code>epi_alpha2</code>	629
6.88.2.280 <code>epi_caseonly</code>	629
6.88.2.281 <code>epi_caseonly_kb_gap</code>	629
6.88.2.282 <code>epi_filter</code>	629
6.88.2.283 <code>epi_genebased</code>	629
6.88.2.284 <code>epi_quickscan</code>	630
6.88.2.285 <code>epistasis</code>	630
6.88.2.286 <code>epsilon</code>	630
6.88.2.287 <code>exclude_file</code>	630
6.88.2.288 <code>exclude_set</code>	630
6.88.2.289 <code>exportArff</code>	630
6.88.2.290 <code>exportDelimited</code>	631
6.88.2.291 <code>exportDelimiter</code>	631
6.88.2.292 <code>extract_before_exclude</code>	631
6.88.2.293 <code>extract_file</code>	631
6.88.2.294 <code>extract_set</code>	631
6.88.2.295 <code>fmfile</code>	631
6.88.2.296 <code>fast_binary</code>	632
6.88.2.297 <code>fast_epistasis</code>	632
6.88.2.298 <code>fileroot</code>	632
6.88.2.299 <code>filter_cases</code>	632
6.88.2.300 <code>filter_controls</code>	632

6.88.2.301filter_females	632
6.88.2.302filter_filename	633
6.88.2.303filter_founders	633
6.88.2.304filter_males	633
6.88.2.305filter_nonfounders	633
6.88.2.306filter_on_covar	633
6.88.2.307filter_value	633
6.88.2.308fisher_test	634
6.88.2.309FIX_IBD	634
6.88.2.310fix_lambda	634
6.88.2.311FIX_p	634
6.88.2.312fix_prev	634
6.88.2.313FIXED	634
6.88.2.314FIXED_p	635
6.88.2.315fixed_prev	635
6.88.2.316flag	635
6.88.2.317flip_file	635
6.88.2.318flip_scan	635
6.88.2.319flip_scan_threshold	635
6.88.2.320flip_scan_verbose	636
6.88.2.321flip_strand	636
6.88.2.322flip_subset	636
6.88.2.323flip_subset_file	636
6.88.2.324force_initial_cluster	636
6.88.2.325force_span	636
6.88.2.326fraction	637
6.88.2.327fringe	637
6.88.2.328from_window	637

6.88.2.329full_model_assoc	637
6.88.2.330fuzzy_homo	637
6.88.2.331gainMatrixFilename	637
6.88.2.332geno_qual_file	638
6.88.2.333geno_qual_max	638
6.88.2.334geno_qual_min	638
6.88.2.335genome_2sets	638
6.88.2.336genome_groups	638
6.88.2.337genome_only_check_rels	638
6.88.2.338genome_output	639
6.88.2.339genome_output_full	639
6.88.2.340genome_output_minimal	639
6.88.2.341genome_setlist1	639
6.88.2.342genome_setlist2	639
6.88.2.343genome_test	639
6.88.2.344genome_test_min_snp	640
6.88.2.345genome_test_threshold	640
6.88.2.346gettag_file	640
6.88.2.347gettag_kb	640
6.88.2.348gettag_listall	640
6.88.2.349gettag_mode	640
6.88.2.350gettag_mode1	641
6.88.2.351gettag_mode2	641
6.88.2.352gettag_r2	641
6.88.2.353gfamfile	641
6.88.2.354glm_dominant	641
6.88.2.355glm_no_auto_sex_effect	641
6.88.2.356glm_recessive	642

6.88.2.357glm_sex_effect	642
6.88.2.358glm_user_parameters	642
6.88.2.359glm_user_test	642
6.88.2.360gmapfile	642
6.88.2.361gmulti_output	642
6.88.2.362gplink	643
6.88.2.363greport	643
6.88.2.364greport_display_empty	643
6.88.2.365greport_gene_list	643
6.88.2.366greport_results	643
6.88.2.367greport_subset	643
6.88.2.368greport_subset_file	644
6.88.2.369grid	644
6.88.2.370gvar	644
6.88.2.371gvar_full_report	644
6.88.2.372gvar_include_all_variants	644
6.88.2.373gvar_to_standard	644
6.88.2.374gvar_verbose_association	645
6.88.2.375gvar_write	645
6.88.2.376gvarfile	645
6.88.2.377hap_max_nf_phases	645
6.88.2.378hap_min_phase_prob	645
6.88.2.379hap_missing_geno	645
6.88.2.380hap_post_prob	646
6.88.2.381hap_specific_snps	646
6.88.2.382hap_specific_snps_list	646
6.88.2.383haplo_plem_follow	646
6.88.2.384haplo_plem_follow_fid	646

6.88.2.385haplo_plem_follow_iid	646
6.88.2.386haplo_plem_follow_ind	647
6.88.2.387haplo_plem_iter	647
6.88.2.388haplo_plem_likelihood_iter	647
6.88.2.389haplo_plem_meta_iter	647
6.88.2.390haplo_plem_meta_likelihood_iter	647
6.88.2.391haplo_plem_meta_prune_haplotype	647
6.88.2.392haplo_plem_meta_prune_phase	648
6.88.2.393haplo_plem_meta_tol	648
6.88.2.394haplo_plem_meta_window	648
6.88.2.395haplo_plem_nonzero_threshold	648
6.88.2.396haplo_plem_original_overlap	648
6.88.2.397haplo_plem_overlap	648
6.88.2.398haplo_plem_verbose	649
6.88.2.399haplo_plem_window	649
6.88.2.400haplo_plem_window_prune_phase	649
6.88.2.401haplo_plem_window_tol	649
6.88.2.402haplo_plem_zero_threshold	649
6.88.2.403has_nonfounders	649
6.88.2.404have_numerics	650
6.88.2.405have_snps	650
6.88.2.406holdout	650
6.88.2.407homo_miss_as_hom	650
6.88.2.408homo_run	650
6.88.2.409homo_run_consensus_match	650
6.88.2.410homo_run_density	651
6.88.2.411homo_run_gap	651
6.88.2.412homo_run_het	651

6.88.2.413homo_run_kb	651
6.88.2.414homo_run_length_kb	651
6.88.2.415homo_run_length_snps	651
6.88.2.416homo_run_snps	652
6.88.2.417homo_summary_allelic_match	652
6.88.2.418homo_threshold	652
6.88.2.419homo_windowAllowedHet	652
6.88.2.420homo_windowAllowedMissing	652
6.88.2.421homo_windowKB	652
6.88.2.422homoWindowSize	653
6.88.2.423homozyg_verbose	653
6.88.2.424hotel	653
6.88.2.425HWD_filter_on_all	653
6.88.2.426HWD_limit	653
6.88.2.427HWD_report	653
6.88.2.428HWD_standard	654
6.88.2.429HWD_test	654
6.88.2.430bd_file	654
6.88.2.431ibd_file_list	654
6.88.2.432bd_read	654
6.88.2.433bd_read_list	654
6.88.2.434bd_read_minimal	655
6.88.2.435BD_threshold	655
6.88.2.436BD_within	655
6.88.2.437bs_2only	655
6.88.2.438bs_inner_run_length_kb	655
6.88.2.439bs_inner_run_length.snp	655
6.88.2.440bs_inter.snp_distance	656

6.88.2.441bs_join_kb	656
6.88.2.442bs_join.snp	656
6.88.2.443bs_run	656
6.88.2.444bs_run_0	656
6.88.2.445bs_run_density	656
6.88.2.446bs_run_length_kb	657
6.88.2.447bs_run_length_snps	657
6.88.2.448bs_run_missing	657
6.88.2.449bs_sharing_test	657
6.88.2.450bs_test	657
6.88.2.451bs_test_method2	657
6.88.2.452bs_test_min.snp	658
6.88.2.453bstest_gap	658
6.88.2.454dhelp	658
6.88.2.455dhelp_alias_update	658
6.88.2.456dhelp_auto_alias	658
6.88.2.457dhelp_command	658
6.88.2.458dhelp_dictionary	659
6.88.2.459dhelp_dump_from_dict	659
6.88.2.460dhelp_dump_from_dict_cmd	659
6.88.2.461dhelp_input	659
6.88.2.462dhelp_list_aliases	659
6.88.2.463dhelp_lookup	659
6.88.2.464dhelp_lookup_string	660
6.88.2.465dhelp_match	660
6.88.2.466dhelp_match_string	660
6.88.2.467dhelp_no_dict	660
6.88.2.468dhelp_output_delimit	660

6.88.2.469dhelp_replace	660
6.88.2.470dhelp_replace_string	661
6.88.2.471dhelp_subset	661
6.88.2.472dhelp_subset_string	661
6.88.2.473gnore_missing_sex	661
6.88.2.474gnore_phenotypes	661
6.88.2.475impmeasure	661
6.88.2.476impute_sex	662
6.88.2.477impute_tags	662
6.88.2.478impute_verbose	662
6.88.2.479nbreeding	662
6.88.2.480nc_file	662
6.88.2.481nc_read	662
6.88.2.482nc_write	663
6.88.2.483include_all_pairs	663
6.88.2.484include_all_z1	663
6.88.2.485include_cluster	663
6.88.2.486include_cluster_filename	663
6.88.2.487include_cluster_from_file	663
6.88.2.488nd_attrib_file	664
6.88.2.489nd_attrib_filter	664
6.88.2.490nd_attrib_value	664
6.88.2.491ndiv_report	664
6.88.2.492ndiv_report_fid	664
6.88.2.493ndiv_report_iid	664
6.88.2.494inter_grid	665
6.88.2.495qtl_coord_file	665
6.88.2.496qtl_expression_file	665

6.88.2.497qtl_interaction_full	665
6.88.2.498qtl_local_cis	665
6.88.2.499qtl_pvalue	665
6.88.2.500qtl_radius	666
6.88.2.501qtl_tf_coord_file	666
6.88.2.502qtl_tf_radius	666
6.88.2.503k	666
6.88.2.504keep_file	666
6.88.2.505keep_indiv	666
6.88.2.506keep_indiv_list	667
6.88.2.507koptBegin	667
6.88.2.508koptEnd	667
6.88.2.509koptStep	667
6.88.2.510ambda	667
6.88.2.511d_anchor	667
6.88.2.512d_anchor_list	668
6.88.2.513d_SNP1	668
6.88.2.514d_SNP1_file	668
6.88.2.515d_SNP2	668
6.88.2.516famfile	668
6.88.2.517file_allele_count	668
6.88.2.518file_input	669
6.88.2.519iability	669
6.88.2.520dist_by_allele	669
6.88.2.521list_twolocus	669
6.88.2.522load_gvar	669
6.88.2.523locked	669
6.88.2.524logistic_model_iterations	670

6.88.2.525logistic_tolerance_epsilon	670
6.88.2.526ogscale	670
6.88.2.527ookup	670
6.88.2.528ookup2	670
6.88.2.529ookup2_cmd	670
6.88.2.530ookup_gene	671
6.88.2.531lookup_gene_kb_window	671
6.88.2.532ookup_gene_name	671
6.88.2.533ookup_multiple_genes	671
6.88.2.534lookup_single_snp	671
6.88.2.535ookup_snp	671
6.88.2.536ookup_snp_kb_window	672
6.88.2.537ookup_to_file	672
6.88.2.538oop_counter	672
6.88.2.539oop_over	672
6.88.2.540oop_over_filename	672
6.88.2.541loop_over_label	672
6.88.2.542pedfile	673
6.88.2.543m	673
6.88.2.544m1	673
6.88.2.545m2	673
6.88.2.546make_blocks	673
6.88.2.547make_founders	673
6.88.2.548make_minor_allele	674
6.88.2.549make_missing_parents	674
6.88.2.550make_pheno	674
6.88.2.551make_pheno_filename	674
6.88.2.552make_pheno_present	674

6.88.2.553make_pheno_value	674
6.88.2.554make_set	675
6.88.2.555make_set_border	675
6.88.2.556make_set_collapse	675
6.88.2.557make_set_collapse_label	675
6.88.2.558make_set_complement	675
6.88.2.559make_set_file	675
6.88.2.560make_set_ignore_group	676
6.88.2.561make_tags	676
6.88.2.562map3	676
6.88.2.563mapfile	676
6.88.2.564mapfile_impute	676
6.88.2.565mating_tests	676
6.88.2.566matrix	677
6.88.2.567max_af	677
6.88.2.568max_cluster_case	677
6.88.2.569max_cluster_control	677
6.88.2.570max_cluster_N	677
6.88.2.571max_cluster_size	677
6.88.2.572MAX_CORR_PIHAT_PIHAT_G	678
6.88.2.573MAX_GENO_MISSING	678
6.88.2.574max_hf	678
6.88.2.575MAX_IND_MISSING	678
6.88.2.576MAX_LINE_LENGTH	678
6.88.2.577max_neighbour	678
6.88.2.578MAX_PIHAT	679
6.88.2.579maxMergeOrder	679
6.88.2.580maxModuleSize	679

6.88.2.581mds_by_individual	679
6.88.2.582memmode	679
6.88.2.583MENDEL_ind	679
6.88.2.584MENDEL_report	680
6.88.2.585MENDEL.snp	680
6.88.2.586MENDEL_test	680
6.88.2.587merge_bedfile	680
6.88.2.588merge_bimfile	680
6.88.2.589merge_binary	680
6.88.2.590merge_data	681
6.88.2.591merge_famfile	681
6.88.2.592merge_force_strand	681
6.88.2.593merge_list	681
6.88.2.594merge_list_filename	681
6.88.2.595merge_mapfile	681
6.88.2.596merge_mode	682
6.88.2.597merge_p	682
6.88.2.598merge_pedfile	682
6.88.2.599meta_analysis	682
6.88.2.600meta_files	682
6.88.2.601meta_large_phase	682
6.88.2.602min_af	683
6.88.2.603min_geno_cell	683
6.88.2.604min_hf	683
6.88.2.605min_neighbour	683
6.88.2.606MIN_PIHAT	683
6.88.2.607minModuleSize	683
6.88.2.608minprop	684

6.88.2.609mishap_test	684
6.88.2.610mishap_window	684
6.88.2.611miss_run	684
6.88.2.612miss_run_length	684
6.88.2.613miss_run_length_kb	684
6.88.2.614miss_run_level	685
6.88.2.615missing_genotype	685
6.88.2.616missing_genotype_explicit	685
6.88.2.617missing_phenotype	685
6.88.2.618missing_phenotype_explicit	685
6.88.2.619mk_datfile	685
6.88.2.620modComputeHomophily	686
6.88.2.621modConnectivityThreshold	686
6.88.2.622model_perm_best	686
6.88.2.623model_perm_dom	686
6.88.2.624model_perm_gen	686
6.88.2.625model_perm_rec	686
6.88.2.626model_perm_trend	687
6.88.2.627modEnableConnectivityThreshold	687
6.88.2.628modFisherTransform	687
6.88.2.629modFisherTransformCutoff	687
6.88.2.630modPowerTransform	687
6.88.2.631modPowerTransformExponent	687
6.88.2.632modUseBinaryThreshold	688
6.88.2.633mperm_rank	688
6.88.2.634mperm_save_all	688
6.88.2.635mperm_save_best	688
6.88.2.636mtry	688

6.88.2.637mult_clst	688
6.88.2.638mult_covar	689
6.88.2.639mult_filter	689
6.88.2.640mult_pheno	689
6.88.2.641multi_output	689
6.88.2.642multiple_phenotype_file	689
6.88.2.643multiple_phenotypes	689
6.88.2.644multtest	690
6.88.2.645myfunction	690
6.88.2.646name_pheno	690
6.88.2.647no_show_covar	690
6.88.2.648nrthreads	690
6.88.2.649ntree	690
6.88.2.650nudge	691
6.88.2.651number_list_positive	691
6.88.2.652number_list_string	691
6.88.2.653numDiffMetricName	691
6.88.2.654numeric_extract_file	691
6.88.2.655numeric_file	691
6.88.2.656numeric_filename	692
6.88.2.657numeric_lowval_percentile	692
6.88.2.658numeric_lowvar_percentile	692
6.88.2.659oblig_clusters_filename	692
6.88.2.660oblig_missing	692
6.88.2.661oblig_missing_filename	692
6.88.2.662opt	693
6.88.2.663OR_homog_test	693
6.88.2.664out_missing_genotype	693

6.88.2.665out_missing_phenotype	693
6.88.2.666out_SNP_major	693
6.88.2.667outlier_detection	693
6.88.2.668output_file_name	694
6.88.2.669output_pheno_perm	694
6.88.2.670parameter_list	694
6.88.2.671parent_of_origin	694
6.88.2.672ped_from_stdin	694
6.88.2.673ped_skip_fid	694
6.88.2.674ped_skip_parents	695
6.88.2.675ped_skip_pheno	695
6.88.2.676ped_skip_sex	695
6.88.2.677pedfile	695
6.88.2.678perm_count	695
6.88.2.679perm_genedrop	695
6.88.2.680perm_genedrop_and_swap	696
6.88.2.681perm_genedrop_parents	696
6.88.2.682perm_genedrop_sibships	696
6.88.2.683perm_genedrop_unrel	696
6.88.2.684perm_POO_best	696
6.88.2.685perm_POO_mat	696
6.88.2.686perm_POO_pat	697
6.88.2.687perm_POO_poo	697
6.88.2.688perm_TDT_basic	697
6.88.2.689perm_TDT_parent	697
6.88.2.690permute	697
6.88.2.691permute_within_sol	697
6.88.2.692filter	698

6.88.2.693	pvalue	698
6.88.2.694	phase_hap_all	698
6.88.2.695	phase_snps	698
6.88.2.696	pheno_file	698
6.88.2.697	pheno_filename	698
6.88.2.698	pihat_filter	699
6.88.2.699	plink	699
6.88.2.700	plist	699
6.88.2.701	plist_fid1	699
6.88.2.702	plist_fid2	699
6.88.2.703	plist_iid1	699
6.88.2.704	plist_iid2	700
6.88.2.705	plist_number	700
6.88.2.706	plist_selection	700
6.88.2.707	plist_selection_name	700
6.88.2.708	plist_selection_number	700
6.88.2.709	plist_selection_string	700
6.88.2.710	pool_size_min	701
6.88.2.711	position_window	701
6.88.2.712	pp_maxfid	701
6.88.2.713	pp_maxiid	701
6.88.2.714	pp_maxsnp	701
6.88.2.715	predall	701
6.88.2.716	predict	702
6.88.2.717	preserve_all_genotypes	702
6.88.2.718	preserve_mendel_errors	702
6.88.2.719	profile_sets	702
6.88.2.720	proxy_all	702

6.88.2.721proxy_all_list	702
6.88.2.722proxy_all_list_file	703
6.88.2.723proxy_assoc	703
6.88.2.724proxy_assoc.snp	703
6.88.2.725proxy_CC	703
6.88.2.726proxy_error	703
6.88.2.727proxy_exclude	703
6.88.2.728proxy_exclude_from_file	704
6.88.2.729proxy_exclude_list	704
6.88.2.730proxy_full_report	704
6.88.2.731proxy_genotype	704
6.88.2.732proxy_glm	704
6.88.2.733proxy_impute	704
6.88.2.734proxy_impute_genotypic_concordance	705
6.88.2.735proxy_impute_preserve_genotyped	705
6.88.2.736proxy_impute_replace	705
6.88.2.737proxy_impute_threshold	705
6.88.2.738proxy_include_reference	705
6.88.2.739proxy_info_threshold	705
6.88.2.740proxy_kb	706
6.88.2.741proxy_kb_planA	706
6.88.2.742proxy_kb_planB	706
6.88.2.743proxy_leave_out	706
6.88.2.744proxy_list	706
6.88.2.745proxy_list_file	706
6.88.2.746proxy_list_proxies	707
6.88.2.747proxy_maf	707
6.88.2.748proxy_maxhap	707

6.88.2.749proxy_mhf	707
6.88.2.750proxy_planB_threshold	707
6.88.2.751proxy_r2	707
6.88.2.752proxy_r2_filter	708
6.88.2.753proxy_r2_filter_A	708
6.88.2.754proxy_r2_filter_A_planA	708
6.88.2.755proxy_r2_filter_A_planB	708
6.88.2.756proxy_r2_filter_B	708
6.88.2.757proxy_r2_filter_B_planA	708
6.88.2.758proxy_r2_filter_B_planB	709
6.88.2.759proxy_r2_filter_C	709
6.88.2.760proxy_r2_filter_C_planA	709
6.88.2.761proxy_r2_filter_C_planB	709
6.88.2.762proxy_record dosage	709
6.88.2.763proxy_reference_only	709
6.88.2.764proxy.snp_filter	710
6.88.2.765proxy.snp_filter_planA	710
6.88.2.766proxy.snp_filter_planB	710
6.88.2.767proxy_TDT	710
6.88.2.768proxy_window	710
6.88.2.769proxy_window_planA	710
6.88.2.770proxy_window_planB	711
6.88.2.771prune_id	711
6.88.2.772prune_id_pairwise	711
6.88.2.773prune_id_pairwise_maf	711
6.88.2.774prune_id_r2	711
6.88.2.775prune_id_step	711
6.88.2.776prune_id_vif	712

6.88.2.777prune_id_win	712
6.88.2.778prune_r2_fixed	712
6.88.2.779prune_r2_fixed_list	712
6.88.2.780prune_r2_prefer	712
6.88.2.781prune_r2_prefer_list	712
6.88.2.782QFAM_adaptive	713
6.88.2.783QFAM_between	713
6.88.2.784QFAM_total	713
6.88.2.785QFAM_within1	713
6.88.2.786QFAM_within2	713
6.88.2.787qmatch	713
6.88.2.788qmatch_filename	714
6.88.2.789qmatch_threshold_filename	714
6.88.2.790qq_plot	714
6.88.2.791qt	714
6.88.2.792qt_means	714
6.88.2.793qt_with_covariates	714
6.88.2.794QTDT_test	715
6.88.2.795R_port	715
6.88.2.796R_script	715
6.88.2.797random_seed	715
6.88.2.798range_delimiter	715
6.88.2.799ranker_centrality_gamma	715
6.88.2.800ranker_input_file	716
6.88.2.801ranker_method	716
6.88.2.802ranker_save_data_file	716
6.88.2.803ranker_top_n	716
6.88.2.804rankerPermMethod	716

6.88.2.805rankerPermNum	716
6.88.2.806rankerPermThreshold	717
6.88.2.807rare_test	717
6.88.2.808rare_test_print_details	717
6.88.2.809rare_test_print_details.snp	717
6.88.2.810rare_test_score_range	717
6.88.2.811rare_test_score_range_file	717
6.88.2.812rare_test_score_range_threshold	718
6.88.2.813rare_test_score_results_file	718
6.88.2.814rare_test_summary_controls	718
6.88.2.815rare_test_weight1	718
6.88.2.816rarer_dist_threshold	718
6.88.2.817rarer_interval	718
6.88.2.818rarer_maf_threshold	719
6.88.2.819read_bitfile	719
6.88.2.820read_geno_qual	719
6.88.2.821read_ped	719
6.88.2.822read_segment_file	719
6.88.2.823read_segment_filename	719
6.88.2.824read_set	720
6.88.2.825read_snp_qual	720
6.88.2.826recode	720
6.88.2.827recode_12	720
6.88.2.828recode_1234	720
6.88.2.829recode_ACGT	720
6.88.2.830recode_AD	721
6.88.2.831recode_AD_Aonly	721
6.88.2.832recode_AD_fixed	721

6.88.2.833recode_allele_coding	721
6.88.2.834recode_allele_coding_file	721
6.88.2.835recode_bim bam	721
6.88.2.836recode_delimit	722
6.88.2.837recode_fastphase	722
6.88.2.838recode_HV	722
6.88.2.839recode_indelimit	722
6.88.2.840recode_long	722
6.88.2.841recode_long_ref	722
6.88.2.842recode_mutlist	723
6.88.2.843recode_structure	723
6.88.2.844recode_transpose	723
6.88.2.845recode_whap	723
6.88.2.846ref_file	723
6.88.2.847ref_file_name	723
6.88.2.848regainComponents	724
6.88.2.849regainCompress	724
6.88.2.850regainFailValue	724
6.88.2.851regainFdr	724
6.88.2.852regainFdrPrune	724
6.88.2.853regainFile	724
6.88.2.854regainLargeCoefPvalue	725
6.88.2.855regainLargeCoefTvalue	725
6.88.2.856regainMatrixFormat	725
6.88.2.857regainMatrixThreshold	725
6.88.2.858regainMatrixThresholdValue	725
6.88.2.859regainMatrixToSif	725
6.88.2.860regainMatrixTransform	726

6.88.2.861regainMaxBetaValue	726
6.88.2.862regainPureInteractions	726
6.88.2.863regainPvalueThreshold	726
6.88.2.864regainSifFilter	726
6.88.2.865regainSifThreshold	726
6.88.2.866regainUseBetaValues	727
6.88.2.867relieffilterNumToRemove	727
6.88.2.868relieffilterPercentToRemove	727
6.88.2.869relieffNumTarget	727
6.88.2.870reliefMode	727
6.88.2.871reliefSeqAlgorithmMode	727
6.88.2.872reliefSeqAlgorithmS0	728
6.88.2.873reliefSeqSnrMode	728
6.88.2.874reliefSeqTstatMode	728
6.88.2.875remove_before_keep	728
6.88.2.876remove_file	728
6.88.2.877remove_indiv	728
6.88.2.878remove_indiv_list	729
6.88.2.879remove_unaffected_pairs	729
6.88.2.880replicates	729
6.88.2.881report_missing	729
6.88.2.882return_beta	729
6.88.2.883freplace	729
6.88.2.884run_chr	730
6.88.2.885run_end	730
6.88.2.886run_R_chisq	730
6.88.2.887run_R_nsnp	730
6.88.2.888run_R_script	730

6.88.2.889run_R_write_script	730
6.88.2.890run_R_z	731
6.88.2.891run_start	731
6.88.2.892savemem	731
6.88.2.893score_impute_expected	731
6.88.2.894score_qfile	731
6.88.2.895score_qrange_file	731
6.88.2.896score_risk	732
6.88.2.897score_risk_file	732
6.88.2.898score_risk_on_qrange	732
6.88.2.899score_risk_ranges	732
6.88.2.900score_risk_ranges_file	732
6.88.2.901score_risk_ranges_min	732
6.88.2.902score_test	733
6.88.2.903SD	733
6.88.2.904seg_test_region	733
6.88.2.905seg_test_window	733
6.88.2.906seg_test_window_bp	733
6.88.2.907segment_current_focal_snp	733
6.88.2.908segment_haplotype	734
6.88.2.909segment_haplotype_fid1	734
6.88.2.910segment_haplotype_fid2	734
6.88.2.911segment_haplotype_iid1	734
6.88.2.912segment_haplotype_iid2	734
6.88.2.913segment_inter_snp_distance	734
6.88.2.914segment_length	735
6.88.2.915segment_m1	735
6.88.2.916segment_m2	735

6.88.2.917segment_minimal	735
6.88.2.918segment_output	735
6.88.2.919segment_output_started	735
6.88.2.920segment_overlap	736
6.88.2.921segment_silently_return_groups	736
6.88.2.922segment.snp	736
6.88.2.923segment.snp1	736
6.88.2.924segment.snp2	736
6.88.2.925segment_test_1sided	736
6.88.2.926segment_test_fisher	737
6.88.2.927segment_test_force_1sided	737
6.88.2.928segment_test_ignore_discordant	737
6.88.2.929segment_test_individual	737
6.88.2.930segment_test_specific_segs	737
6.88.2.931segment_threshold_finish	737
6.88.2.932segment_threshold_start	738
6.88.2.933segment_validate	738
6.88.2.934segment_verbose	738
6.88.2.935set_by_set	738
6.88.2.936set_chisq_threshold	738
6.88.2.937set_max	738
6.88.2.938set_min	739
6.88.2.939set_p2	739
6.88.2.940set_r2	739
6.88.2.941set_r2_phase	739
6.88.2.942set_r2_read	739
6.88.2.943set_r2_read_file	739
6.88.2.944set_r2_val	740

6.88.2.945	set_r2_write	740
6.88.2.946	set_reference_allele	740
6.88.2.947	set_reference_allele_file	740
6.88.2.948	set_score	740
6.88.2.949	set_score_p	740
6.88.2.950	set_screen	741
6.88.2.951	set_screen_resultfile	741
6.88.2.952	set_step	741
6.88.2.953	set_step_in	741
6.88.2.954	set_table	741
6.88.2.955	set_test	741
6.88.2.956	setfile	742
6.88.2.957	sex_threshold_female	742
6.88.2.958	sex_threshold_male	742
6.88.2.959	show_impossible_IBD	742
6.88.2.960	sibTDT_test	742
6.88.2.961	sifFile	742
6.88.2.962	sifNetwork	743
6.88.2.963	sifToGain	743
6.88.2.964	silent	743
6.88.2.965	simple_interaction	743
6.88.2.966	simul	743
6.88.2.967	simul_file	743
6.88.2.968	simul_haps	744
6.88.2.969	simul_label	744
6.88.2.970	simul_ncases	744
6.88.2.971	simul_ncontrols	744
6.88.2.972	simul_prevalence	744

6.88.2.973simul_qt	744
6.88.2.974simul_qt_var	745
6.88.2.975simul_tags	745
6.88.2.976singlepoint	745
6.88.2.977sliding_window	745
6.88.2.978sliding_window_size	745
6.88.2.979snp_attrib_file	745
6.88.2.980snp_attrib_filter	746
6.88.2.981snp_attrib_value	746
6.88.2.982snp_include_from_cl	746
6.88.2.983snp_include_range	746
6.88.2.984SNP_major	746
6.88.2.985snp_qual_file	746
6.88.2.986snp_qual_max	747
6.88.2.987snp_qual_min	747
6.88.2.988snp_range_list	747
6.88.2.989snpDiffMetricName	747
6.88.2.990snpNearestNeighborMetricName	747
6.88.2.991sol_family	747
6.88.2.992species_cow	748
6.88.2.993species_dog	748
6.88.2.994species_horse	748
6.88.2.995species_mouse	748
6.88.2.996species_rice	748
6.88.2.997species_sheep	748
6.88.2.998splitrule	749
6.88.2.999splitweights	749
6.88.2.100standard_beta	749

6.88.2.100 startMergeOrder	749
6.88.2.100 statusvarname	749
6.88.2.100 Subsetfile	749
6.88.2.100 summ_nonfounders	750
6.88.2.100 summary_ibd_output	750
6.88.2.100 tagfile	750
6.88.2.100 targetpartitionsize	750
6.88.2.100 BDT_test	750
6.88.2.100 est_full_model	750
6.88.2.101 est_hap_CC	751
6.88.2.101 est_hap_GLM	751
6.88.2.101 est_hap_GLM_omnibus	751
6.88.2.101 est_hap_only	751
6.88.2.101 est_hap_QTL	751
6.88.2.101 est_hap_TDT	751
6.88.2.101 est_list	752
6.88.2.101 est_missing	752
6.88.2.101 bamfile	752
6.88.2.101 file_input	752
6.88.2.102 min_param	752
6.88.2.102 min_snps	752
6.88.2.102 thresholdType	753
6.88.2.102 thresholdValue	753
6.88.2.102 tvFilename	753
6.88.2.102 window	753
6.88.2.102 pedfile	753
6.88.2.102 teetype	753
6.88.2.102 end_only	754

6.88.2.1029cc	754
6.88.2.1030woDFmodel	754
6.88.2.1031woDFmodel_hethom	754
6.88.2.1032volocus_snp1	754
6.88.2.1033volocus_snp2	754
6.88.2.1034uncompress_file	755
6.88.2.1035update_allele_file	755
6.88.2.1036update_alleles	755
6.88.2.1037update_chr	755
6.88.2.1038update_cm	755
6.88.2.1039update_ids	755
6.88.2.1040update_ids_file	756
6.88.2.1041update_map	756
6.88.2.1042update_mapfile	756
6.88.2.1043update_name	756
6.88.2.1044update_parents	756
6.88.2.1045update_parents_file	756
6.88.2.1046update_pheno	757
6.88.2.1047update_pheno_file	757
6.88.2.1048update_sex	757
6.88.2.1049update_sex_file	757
6.88.2.1050use_GC	757
6.88.2.1051use_subset	757
6.88.2.1052useAbs	758
6.88.2.1053useWeighted	758
6.88.2.1054verbose	758
6.88.2.1055f_threshold	758
6.88.2.1056eb_check	758

6.88.2.1057eightByDistanceMethod	758
6.88.2.1058eightByDistanceSigma	759
6.88.2.1059eighted_mm	759
6.88.2.1060indow	759
6.88.2.1061write_bitfile	759
6.88.2.1062rite dosage	759
6.88.2.1063rite_set	759
6.88.2.1064rite_snplist	760
6.88.2.1065riteforest	760
6.88.2.1066chr_model	760
6.88.2.1067ero_cluster	760
6.88.2.1068ero_cluster_filename	760
6.89 Perm Class Reference	761
6.89.1 Detailed Description	762
6.89.2 Constructor & Destructor Documentation	762
6.89.2.1 Perm()	762
6.89.3 Member Function Documentation	763
6.89.3.1 closeDUMP()	763
6.89.3.2 current_reps()	763
6.89.3.3 dropAlleles()	763
6.89.3.4 finished()	763
6.89.3.5 geneDrop()	764
6.89.3.6 max_pvalue()	764
6.89.3.7 nextSNP()	764
6.89.3.8 originalOrder()	764
6.89.3.9 permuteInCluster()	764
6.89.3.10 preGeneDrop()	764
6.89.3.11 pvalue()	765

6.89.3.12 rank()	765
6.89.3.13 report()	765
6.89.3.14 reps_done()	765
6.89.3.15 setAdaptiveSetSNPs()	765
6.89.3.16 setOriginalRanking()	766
6.89.3.17 setPermClusters()	766
6.89.3.18 setTests()	766
6.89.3.19 update()	766
6.89.3.20 updateSNP()	766
6.89.4 Member Data Documentation	767
6.89.4.1 adaptive	767
6.89.4.2 count	767
6.89.4.3 dump_all	767
6.89.4.4 dump_best	767
6.89.4.5 genedrop	767
6.89.4.6 geno	768
6.89.4.7 idmap	768
6.89.4.8 interval	768
6.89.4.9 maxR	768
6.89.4.10 min	768
6.89.4.11 N	768
6.89.4.12 ns	769
6.89.4.13 order	769
6.89.4.14 P	769
6.89.4.15 PDUMP	769
6.89.4.16 performed	769
6.89.4.17 pheno	769
6.89.4.18 R	770

6.89.4.19 reorder	770
6.89.4.20 replicates	770
6.89.4.21 s	770
6.89.4.22.snp_test	770
6.89.4.23 t	770
6.89.4.24 test	771
6.89.4.25 zt	771
6.90 phdr Struct Reference	771
6.90.1 Detailed Description	772
6.90.2 Member Data Documentation	772
6.90.2.1 cmd	772
6.90.2.2 dof	772
6.90.2.3 len	772
6.90.2.4 res	772
6.91 Plink Class Reference	773
6.91.1 Detailed Description	780
6.91.2 Constructor & Destructor Documentation	780
6.91.2.1 Plink()	780
6.91.3 Member Function Documentation	780
6.91.3.1 alleleRecoding()	780
6.91.3.2 annotateFile()	780
6.91.3.3 attribFilterInd()	780
6.91.3.4 attribFilterSNP()	781
6.91.3.5 buildCluster()	781
6.91.3.6 buildT()	781
6.91.3.7 calcAssociationWithBootstrap()	781
6.91.3.8 calcAssociationWithPermutation()	781
6.91.3.9 calcEpistasis()	782

6.91.3.10 calcFlipScan()	782
6.91.3.11 calcFst()	782
6.91.3.12 calcGenomeIBD()	782
6.91.3.13 calcGenomeIBM()	782
6.91.3.14 calcGenomeIBS()	783
6.91.3.15 calcGXE()	783
6.91.3.16 calcHomog()	783
6.91.3.17 calcHotel()	783
6.91.3.18 calcInbreeding()	783
6.91.3.19 calcInformative()	784
6.91.3.20 calcLDStatistics()	784
6.91.3.21 calcLocusIBD()	784
6.91.3.22 calcMantelHaenszel_2x2xK()	784
6.91.3.23 calcMantelHaenszel_IxJxK()	784
6.91.3.24 calcMantelHaenszel_ORD()	785
6.91.3.25 calcMH()	785
6.91.3.26 calcMultiPoint()	785
6.91.3.27 calcPairwiseLD()	785
6.91.3.28 calcPhenotypes()	785
6.91.3.29 calcQTDT()	786
6.91.3.30 calcRegression()	786
6.91.3.31 calcSinglePoint()	786
6.91.3.32 calcStratifiedAlleleFreqs()	786
6.91.3.33 calculateProfile()	787
6.91.3.34 checkMendel()	787
6.91.3.35 cleanUp()	787
6.91.3.36 conditionalHaplotypeTest()	787
6.91.3.37 convertGenericVariantData()	787

6.91.3.38 correlation2SNP()	788
6.91.3.39 countCNVPerRegion()	788
6.91.3.40 deleteIndividuals() [1/2]	788
6.91.3.41 deleteIndividuals() [2/2]	788
6.91.3.42 deleteIndividualsNumeric()	788
6.91.3.43 deleteSNPs() [1/3]	789
6.91.3.44 deleteSNPs() [2/3]	789
6.91.3.45 deleteSNPs() [3/3]	789
6.91.3.46 display_indivReport()	789
6.91.3.47 display_listByAllele()	789
6.91.3.48 display_pairList()	790
6.91.3.49 display_recoded_LONG()	790
6.91.3.50 display_recoded_MUTLIST()	790
6.91.3.51 display_recoded_PEDFILE()	790
6.91.3.52 display_recoded_PEDFILE_AD()	790
6.91.3.53 display_recoded_PEDFILE_transpose()	790
6.91.3.54 display_twolocus()	791
6.91.3.55 displayGeneReport()	791
6.91.3.56 displayGenomePV()	791
6.91.3.57 displayGenomeWideInfo()	791
6.91.3.58 displayGMULTI()	791
6.91.3.59 displayQTMeans()	792
6.91.3.60 displayRareRange()	792
6.91.3.61 displaySegmentsBED()	792
6.91.3.62 displaySegmentsLong()	792
6.91.3.63 doRegression()	792
6.91.3.64 driverSCREEPI()	793
6.91.3.65 dummyLoader()	793

6.91.3.66 elfBaseline()	793
6.91.3.67 extractExcludeSet()	793
6.91.3.68 filterOnCase()	793
6.91.3.69 filterOnControl()	793
6.91.3.70 filterOnCovariate()	794
6.91.3.71 filterOnFemale()	794
6.91.3.72 filterOnFounder()	794
6.91.3.73 filterOnMale()	794
6.91.3.74 filterOnNonFounder()	794
6.91.3.75 filterQualGenotypes()	794
6.91.3.76 filterQualSNPs()	795
6.91.3.77 filterSNPs()	795
6.91.3.78 findAllHomozygousRuns()	795
6.91.3.79 findHomoRuns()	795
6.91.3.80 findHomoWindow()	795
6.91.3.81 findIBSRuns()	796
6.91.3.82 findMissRuns()	796
6.91.3.83 findSegments()	796
6.91.3.84 flipStrand()	796
6.91.3.85 fullModelAssoc()	796
6.91.3.86 generateMDS()	797
6.91.3.87 getSimpleSNPValue()	797
6.91.3.88 glmAssoc()	797
6.91.3.89 glmCNVBurdenModel()	797
6.91.3.90 glmHaplotypeTest()	797
6.91.3.91 groupGenome()	798
6.91.3.92 groupSegmentsSpanning()	798
6.91.3.93 hardyWeinbergCheck()	798

6.91.3.94 homozygousSegmentPermutationTest()	798
6.91.3.95 Ind2SNP()	798
6.91.3.96 indivSegmentSummary()	799
6.91.3.97 indivSegmentSummaryCalc()	799
6.91.3.98 initialiseGeneCountAssociation()	799
6.91.3.99 keep2SetsForGenome()	799
6.91.3.100keepIndividuals()	799
6.91.3.101keepSNPs() [1/2]	800
6.91.3.102keepSNPs() [2/2]	800
6.91.3.103linkRelateds()	800
6.91.3.104lookup()	800
6.91.3.105lookup2()	800
6.91.3.106makeFounders()	801
6.91.3.107makeMissingParents()	801
6.91.3.108makePhenotype()	801
6.91.3.109mergeBinaryData()	801
6.91.3.110mergeData()	801
6.91.3.111mergeList()	801
6.91.3.112metaAnalysis()	802
6.91.3.113missingGenotype()	802
6.91.3.114mkBlks()	802
6.91.3.115multcomp()	802
6.91.3.116obligMissing()	802
6.91.3.117openBinaryFile()	803
6.91.3.118output_bim bam_format()	803
6.91.3.119output_fastphase_format()	803
6.91.3.120output_structure_format()	803
6.91.3.121outputArffFile()	803

6.91.3.122outputDelimitedFile()	804
6.91.3.123outputGenericVariantFile()	804
6.91.3.124outputNumericExtract()	804
6.91.3.125outputNumericFiltered()	804
6.91.3.126outputPermedPhenotypes()	804
6.91.3.127outputSetFile()	805
6.91.3.128outputSifToGain()	805
6.91.3.129parseTrios()	805
6.91.3.130performMisHapTests()	805
6.91.3.131performProxyTests()	805
6.91.3.132perm_segmentIndividualTest()	806
6.91.3.133perm_sharingIBSTest()	806
6.91.3.134perm_testGXE2()	806
6.91.3.135perm_testHotel()	806
6.91.3.136perm_testQTDT()	806
6.91.3.137perm_testTDT()	807
6.91.3.138perm_testTDT_POO()	807
6.91.3.139permTestRareDistribution()	807
6.91.3.140permutationIBSTest()	807
6.91.3.141positionPermuteSegments()	807
6.91.3.142preCalcGenomeIBD()	808
6.91.3.143preCalcMultiPoint()	808
6.91.3.144preCalcPhenotypes()	808
6.91.3.145preCalcRegression_PHENO()	808
6.91.3.146preCalcRegression_PIHAT()	808
6.91.3.147preCalcSinglePoint()	808
6.91.3.148prettyPrintLengths()	809
6.91.3.149printLOG()	809

6.91.3.150processCNVList()	809
6.91.3.151processDosageFile()	809
6.91.3.152processGVAR()	809
6.91.3.153proxyWrapper()	809
6.91.3.154pruneLD()	810
6.91.3.155pseudoCaseControl()	810
6.91.3.156readBinData()	810
6.91.3.157readClusterFile()	810
6.91.3.158readCNVList()	810
6.91.3.159readConditioningList()	810
6.91.3.160readCovariateFile()	811
6.91.3.161readCovListFile()	811
6.91.3.162readData()	811
6.91.3.163readDataLongFormat()	811
6.91.3.164readFamFile()	811
6.91.3.165readGenericVariantData()	811
6.91.3.166readHomozygSegmentFile()	812
6.91.3.167readInformative()	812
6.91.3.168readMapFile()	812
6.91.3.169readMultiplePhenoFile()	812
6.91.3.170readNumericFile()	812
6.91.3.171readPhenoFile()	813
6.91.3.172readSegmentFile()	813
6.91.3.173readSegmentFileMinimal()	813
6.91.3.174readSet()	813
6.91.3.175readStdIn()	813
6.91.3.176readTransposedData()	814
6.91.3.177reconcileMerge()	814

6.91.3.178removeIndividuals()	814
6.91.3.179runTestCNVwithGLM()	814
6.91.3.180runTestCNVwithQT()	814
6.91.3.181scoreIndividuals()	815
6.91.3.182segmentIndividualTest()	815
6.91.3.183segmentPermutationTest()	815
6.91.3.184setAssocSummary()	815
6.91.3.185setFlags()	815
6.91.3.186setFlagToCase()	816
6.91.3.187setFlagToControl()	816
6.91.3.188setMarkerRange()	816
6.91.3.189setObligMissing()	816
6.91.3.190setQtPhenoFromNumericIndex()	816
6.91.3.191setReferenceAllele()	816
6.91.3.192setTable()	817
6.91.3.193setUpForCNVList()	817
6.91.3.194sexCheck()	817
6.91.3.195sharingIBSTest()	817
6.91.3.196simulateSNPs()	817
6.91.3.197simulateSNPs_QT()	817
6.91.3.198SNP2Ind()	818
6.91.3.199summariseHomoRuns()	818
6.91.3.200summaryIBD()	818
6.91.3.201summaryIBDsegments()	818
6.91.3.202summaryIBSsegments()	818
6.91.3.203tagMode()	819
6.91.3.204testAssoc()	819
6.91.3.205testCNVwithGLM()	819

6.91.3.206testCNVwithQT()	819
6.91.3.207testGenomeIBDBByCovariate()	820
6.91.3.208testMiss()	820
6.91.3.209testQAssoc()	820
6.91.3.210testQAssocGXE2()	820
6.91.3.211testRareDistribution()	820
6.91.3.212testSibTDT()	821
6.91.3.213testTDT()	821
6.91.3.214testTDT_POO()	821
6.91.3.215thinSNPs()	821
6.91.3.216updateAlleles()	822
6.91.3.217updateFamFile()	822
6.91.3.218updateMapFile()	822
6.91.3.219validateSegments()	822
6.91.3.220webcheck()	822
6.91.3.221write_BITFILE()	822
6.91.3.222write_clusters()	823
6.91.3.223write_covariates()	823
6.91.3.224write_snplist()	823
6.91.3.225writeInformative()	823
6.91.3.226writeSetFile()	823
6.91.3.227zeroOnCluster()	823
6.91.4 Member Data Documentation	824
6.91.4.1 clistname	824
6.91.4.2 cnt_f	824
6.91.4.3 conditioner	824
6.91.4.4 conditioner_mask	824
6.91.4.5 diff_concordant_overlap	824

6.91.4.6 diff_nonmissing_overlap	825
6.91.4.7 diff_overlap	825
6.91.4.8 dst	825
6.91.4.9 E00	825
6.91.4.10 E01	825
6.91.4.11 E02	825
6.91.4.12 E10	826
6.91.4.13 E11	826
6.91.4.14 E12	826
6.91.4.15 E20	826
6.91.4.16 E21	826
6.91.4.17 E22	826
6.91.4.18 epi1	827
6.91.4.19 epi2	827
6.91.4.20 expectedOverlap	827
6.91.4.21 expectedOverlapBaseline	827
6.91.4.22 family	827
6.91.4.23 gene2segment	827
6.91.4.24 geneList	828
6.91.4.25 gset1	828
6.91.4.26 gset2	828
6.91.4.27 gvar	828
6.91.4.28 haplo	828
6.91.4.29 in_anal	828
6.91.4.30 indivSegmentGroup	829
6.91.4.31 klist	829
6.91.4.32 kmap	829
6.91.4.33 kname	829

6.91.4.34 locus	829
6.91.4.35 m1	829
6.91.4.36 m2	830
6.91.4.37 m_phenotype	830
6.91.4.38 m_pihat	830
6.91.4.39 maxr2	830
6.91.4.40 mdist	830
6.91.4.41 model	830
6.91.4.42 n	831
6.91.4.43 ngvar	831
6.91.4.44 nk	831
6.91.4.45 nl	831
6.91.4.46 nl_all	831
6.91.4.47 nlistname	831
6.91.4.48 np	832
6.91.4.49 npheno	832
6.91.4.50 oblig_missing	832
6.91.4.51 original	832
6.91.4.52 OUTFILE	832
6.91.4.53 pair1	832
6.91.4.54 pair2	833
6.91.4.55 pairid	833
6.91.4.56 phenoLabel	833
6.91.4.57 phenotype	833
6.91.4.58 phenotype_name	833
6.91.4.59 pihat	833
6.91.4.60 pihat_G	834
6.91.4.61 plistname	834

6.91.4.62 pos	834
6.91.4.63 pperm	834
6.91.4.64 prev_bt	834
6.91.4.65 proxyLD	834
6.91.4.66 pS	835
6.91.4.67 pv	835
6.91.4.68 pvlBS0	835
6.91.4.69 pvlBS2het	835
6.91.4.70 related	835
6.91.4.71 sample	835
6.91.4.72 saved_IBDg	836
6.91.4.73 scaffold	836
6.91.4.74 segment	836
6.91.4.75 segmentCount	836
6.91.4.76 segmentCount2	836
6.91.4.77 segmentCount2Baseline	836
6.91.4.78 segmentLength	837
6.91.4.79 setname	837
6.91.4.80 skip_pair	837
6.91.4.81 SNP	837
6.91.4.82 snpset	837
6.91.4.83 T00	837
6.91.4.84 T01	838
6.91.4.85 T02	838
6.91.4.86 T10	838
6.91.4.87 T11	838
6.91.4.88 T12	838
6.91.4.89 T20	838

6.91.4.90 T21	839
6.91.4.91 T22	839
6.91.4.92 tcnt	839
6.91.4.93 transcript	839
6.91.4.94 v_phenotype	839
6.91.4.95 v_pihat	839
6.91.4.96 warnings	840
6.91.4.97 whap	840
6.91.4.98 Zlocus	840
6.91.4.99 ZOUTFILE	840
6.92 PlinkInternalsDataset Class Reference	842
6.92.1 Detailed Description	843
6.92.2 Constructor & Destructor Documentation	844
6.92.2.1 PlinkInternalsDataset()	844
6.92.2.2 ~PlinkInternalsDataset()	844
6.92.3 Member Function Documentation	844
6.92.3.1 LoadDatasetFromPlink()	844
6.92.4 Member Data Documentation	844
6.92.4.1 plinkInternalsPtr	844
6.93 PlinkInternalsDatasetInstance Class Reference	845
6.93.1 Detailed Description	846
6.93.2 Constructor & Destructor Documentation	847
6.93.2.1 PlinkInternalsDatasetInstance()	847
6.93.2.2 ~PlinkInternalsDatasetInstance()	847
6.93.3 Member Function Documentation	847
6.93.3.1 GetAttribute()	848
6.93.3.2 GetNumeric()	848
6.93.3.3 GetSimpleSNPValue()	848

6.93.3.4	NumAttributes()	849
6.93.3.5	NumNumerics()	849
6.93.3.6	Print()	849
6.93.4	Member Data Documentation	849
6.93.4.1	ID	849
6.93.4.2	individual	850
6.93.4.3	plinkInternalsPtr	850
6.94	Pool Class Reference	851
6.94.1	Detailed Description	852
6.94.2	Member Data Documentation	852
6.94.2.1	chr	852
6.94.2.2	group	852
6.94.2.3	index	852
6.94.2.4	match	852
6.94.2.5	matchcount	853
6.94.2.6	max	853
6.94.2.7	min	853
6.94.2.8	ng	853
6.94.2.9	segs	853
6.94.2.10	union_max	853
6.94.2.11	union_min	854
6.95	probabilisticGenotype Class Reference	854
6.95.1	Detailed Description	855
6.95.2	Constructor & Destructor Documentation	855
6.95.2.1	probabilisticGenotype()	855
6.95.3	Member Data Documentation	855
6.95.3.1	AA	855
6.95.3.2	AB	855

6.95.3.3	BA	855
6.95.3.4	BB	856
6.95.3.5	calculated	856
6.95.3.6	genotype	856
6.95.3.7	genotyped	856
6.95.3.8	phased	856
6.95.3.9	phased_genotype	856
6.96	ProxyResult Class Reference	857
6.96.1	Detailed Description	858
6.96.2	Constructor & Destructor Documentation	858
6.96.2.1	ProxyResult()	858
6.96.3	Member Function Documentation	858
6.96.3.1	operator<()	858
6.96.4	Member Data Documentation	858
6.96.4.1	chisq	858
6.96.4.2	f	859
6.96.4.3	name	859
6.96.4.4	odds	859
6.96.4.5	pvalue	859
6.96.4.6	r2	859
6.97	RandomForest Class Reference	860
6.97.1	Detailed Description	862
6.97.2	Constructor & Destructor Documentation	862
6.97.2.1	RandomForest() [1/3]	862
6.97.2.2	RandomForest() [2/3]	863
6.97.2.3	RandomForest() [3/3]	863
6.97.2.4	~RandomForest()	863
6.97.3	Member Function Documentation	864

6.97.3.1	ComputeScores()	864
6.97.3.2	CreateDefaultForestForPheno()	864
6.97.3.3	GetClassificationError()	864
6.97.3.4	InitializeData()	864
6.97.3.5	Predict()	865
6.97.3.6	SaveForest()	865
6.97.3.7	WriteScores()	865
6.97.4	Member Data Documentation	865
6.97.4.1	datasetFile	865
6.97.4.2	forest	866
6.97.4.3	minNodeSize	866
6.97.4.4	PP	866
6.98	Range Class Reference	866
6.98.1	Detailed Description	868
6.98.2	Constructor & Destructor Documentation	868
6.98.2.1	Range() [1/2]	868
6.98.2.2	Range() [2/2]	868
6.98.3	Member Function Documentation	868
6.98.3.1	operator<()	869
6.98.3.2	operator==()	869
6.98.4	Member Data Documentation	869
6.98.4.1	chr	869
6.98.4.2	group	869
6.98.4.3	groupNames	869
6.98.4.4	name	870
6.98.4.5	start	870
6.98.4.6	stop	870
6.99	Rconnection Class Reference	870

6.99.1 Detailed Description	872
6.99.2 Constructor & Destructor Documentation	872
6.99.2.1 Rconnection()	872
6.99.2.2 ~Rconnection()	872
6.99.3 Member Function Documentation	872
6.99.3.1 assign()	873
6.99.3.2 closeFile()	873
6.99.3.3 connect()	873
6.99.3.4 createFile()	873
6.99.3.5 disconnect()	873
6.99.3.6 eval()	874
6.99.3.7 login()	874
6.99.3.8 openFile()	874
6.99.3.9 readFile()	874
6.99.3.10 removeFile()	874
6.99.3.11 request() [1/2]	875
6.99.3.12 request() [2/2]	875
6.99.3.13 shutdown()	875
6.99.3.14 voidEval()	875
6.99.3.15 writeFile()	876
6.99.4 Member Data Documentation	876
6.99.4.1 auth	876
6.99.4.2 family	876
6.99.4.3 host	876
6.99.4.4 port	876
6.99.4.5 s	877
6.99.4.6 salt	877
6.100 RCount Class Reference	877

6.100.1 Detailed Description	878
6.100.2 Constructor & Destructor Documentation	878
6.100.2.1 RCount()	878
6.100.3 Member Function Documentation	878
6.100.3.1 addSNP()	879
6.100.3.2 displayWindow()	879
6.100.3.3 ignorePosition()	879
6.100.3.4 loadCovariate()	879
6.100.3.5 loadPCACovariate()	879
6.100.3.6 mainStats()	879
6.100.3.7 removeSNP()	880
6.100.3.8 setWindow()	880
6.100.4 Member Data Documentation	880
6.100.4.1 acnt	880
6.100.4.2 acnt2	880
6.100.4.3 aval	880
6.100.4.4 awin	881
6.100.4.5 domModel	881
6.100.4.6 gval	881
6.100.4.7 gwin	881
6.100.4.8 nalleles	881
6.100.4.9 npc	881
6.100.4.10nsnps	882
6.100.4.11P	882
6.100.4.12pcMode	882
6.100.4.13rangeLookup	882
6.100.4.14rval	882
6.100.4.15win	882

6.100.4.16ucnt	883
6.100.4.17ucnt2	883
6.101 Rdouble Class Reference	883
6.101.1 Detailed Description	886
6.101.2 Constructor & Destructor Documentation	886
6.101.2.1 Rdouble() [1/3]	886
6.101.2.2 Rdouble() [2/3]	886
6.101.2.3 Rdouble() [3/3]	887
6.101.3 Member Function Documentation	887
6.101.3.1 doubleArray()	887
6.101.3.2 doubleAt()	887
6.101.3.3 fix_content()	887
6.101.3.4 length()	887
6.101.3.5 os_print()	888
6.102 Regain Class Reference	888
6.102.1 Detailed Description	891
6.102.2 Constructor & Destructor Documentation	891
6.102.2.1 Regain() [1/2]	891
6.102.2.2 Regain() [2/2]	891
6.102.2.3 ~Regain()	892
6.102.3 Member Function Documentation	892
6.102.3.1 addCovariates()	892
6.102.3.2 fdrPrune()	892
6.102.3.3 getRawMatrix()	892
6.102.3.4 interactionEffect()	892
6.102.3.5 logMatrixStats()	893
6.102.3.6 logOutputOptions()	893
6.102.3.7 mainEffect()	893

6.102.3.8 mainEffectComparator()	893
6.102.3.9 performPureInteraction()	893
6.102.3.10pureInteractionEffect()	894
6.102.3.11readRegainFromFile()	894
6.102.3.12run()	894
6.102.3.13setFailureValue()	894
6.102.3.14setOutputFormat()	894
6.102.3.15setOutputThreshold()	895
6.102.3.16setOutputTransform()	895
6.102.3.17updateStats()	895
6.102.3.18writeRcomm()	895
6.102.3.19writeRegain()	895
6.102.3.20writeRegainToFile()	896
6.102.3.21writeRegainToSifFile()	896
6.102.4 Member Data Documentation	896
6.102.4.1 attributeNames	896
6.102.4.2 BETAS	896
6.102.4.3 doFdrPrune	896
6.102.4.4 failures	897
6.102.4.5 failureValue	897
6.102.4.6 gainIntPvals	897
6.102.4.7 infCount	897
6.102.4.8 INT_MATRIX	897
6.102.4.9 INT_SIF	897
6.102.4.10IntegratedAttributes	898
6.102.4.11maxInteraction	898
6.102.4.12maxMainEffect	898
6.102.4.13MEBETAS	898

6.102.4.14minInteraction	898
6.102.4.15minMainEffect	898
6.102.4.16nanCount	899
6.102.4.17NUM_MATRIX	899
6.102.4.18NUM_SIF	899
6.102.4.19numAttributes	899
6.102.4.20outputFormat	899
6.102.4.21outputThreshold	899
6.102.4.22outputTransform	900
6.102.4.23pureInteractions	900
6.102.4.24REGAIN_MATRIX	900
6.102.4.25regainMatrix	900
6.102.4.26regainPMatrix	900
6.102.4.27SIF	900
6.102.4.28sifThresh	901
6.102.4.29SNP_MATRIX	901
6.102.4.30SNP_SIF	901
6.102.4.31useOutputThreshold	901
6.102.4.32warnings	901
6.102.4.33writeComponents	901
6.102.4.34writeCompressedFormat	902
6.103RegressionRankResults Class Reference	902
6.103.1 Detailed Description	903
6.103.2 Constructor & Destructor Documentation	903
6.103.2.1 RegressionRankResults()	903
6.103.3 Member Data Documentation	903
6.103.3.1 coefs	903
6.103.3.2 pvals	903

6.103.3.3 stats	903
6.103.3.4 vars	904
6.104 ReliefF Class Reference	904
6.104.1 Detailed Description	908
6.104.2 Constructor & Destructor Documentation	908
6.104.2.1 ReliefF() [1/2]	908
6.104.2.2 ~ReliefF()	909
6.104.2.3 ReliefF() [2/2]	909
6.104.3 Member Function Documentation	909
6.104.3.1 ComputeAttributeScores()	909
6.104.3.2 ComputeAttributeScoresIteratively()	910
6.104.3.3 ComputeAttributeScoresKopt()	910
6.104.3.4 ComputeGRM()	910
6.104.3.5 ComputeScores()	910
6.104.3.6 ComputeWeightByDistanceFactors()	910
6.104.3.7 GetKmax()	911
6.104.3.8 PreComputeDistances()	911
6.104.3.9 PrintAttributeScores()	911
6.104.3.10 PrintBestKs()	911
6.104.3.11 PrintInstancesMask()	912
6.104.3.12 RemoveWorstAttributes()	912
6.104.3.13 ResetForNextIteration()	912
6.104.3.14 SetKoptParameters()	912
6.104.3.15 WriteAttributeScores()	913
6.104.3.16 WriteBestKs()	914
6.104.4 Member Data Documentation	914
6.104.4.1 analysisType	914
6.104.4.2 bestKs	914

6.104.4.3 distanceMatrix	915
6.104.4.4 doRemovePercent	915
6.104.4.5 koptBegin	915
6.104.4.6 koptEnd	915
6.104.4.7 koptStep	915
6.104.4.8 m	916
6.104.4.9 numDiffFuncPtr	916
6.104.4.10numDiffMetricName	916
6.104.4.11numTarget	916
6.104.4.12PP	917
6.104.4.13randomlySelect	917
6.104.4.14removedAttributes	917
6.104.4.15removePercentage	917
6.104.4.16removePerIteration	917
6.104.4.17snpDiffFuncPtr	917
6.104.4.18snpDiffMetricName	918
6.104.4.19W	918
6.104.4.20weightByDistanceMethod	918
6.104.4.21weightByDistanceSigma	919
6.105ReliefFSeq Class Reference	919
6.105.1 Detailed Description	921
6.105.2 Constructor & Destructor Documentation	922
6.105.2.1 ReliefFSeq()	922
6.105.2.2 ~ReliefFSeq()	922
6.105.3 Member Function Documentation	923
6.105.3.1 ComputeAttributeScores()	923
6.105.3.2 GetScores()	923
6.105.3.3 MuDeltaAlphas()	924

6.105.3.4 SigmaDeltaAlphas()	924
6.105.4 Member Data Documentation	924
6.105.4.1 mode	924
6.105.4.2 s0	924
6.105.4.3 snrMode	925
6.105.4.4 tstatMode	925
6.106 ReliefSeqController Class Reference	925
6.106.1 Detailed Description	927
6.106.2 Constructor & Destructor Documentation	927
6.106.2.1 ReliefSeqController()	927
6.106.2.2 ~ReliefSeqController()	928
6.106.3 Member Function Documentation	928
6.106.3.1 ComputeScores()	928
6.106.3.2 ComputeScoresKopt()	928
6.106.3.3 GetAlgorithmMode()	928
6.106.3.4 GetKmax()	929
6.106.3.5 GetScores()	929
6.106.3.6 PrintAttributeScores()	929
6.106.3.7 PrintBestKs()	929
6.106.3.8 PrintScores()	930
6.106.3.9 RemoveWorstAttributes()	930
6.106.3.10 RunReliefF()	930
6.106.3.11 SetKoptParameters()	930
6.106.3.12 WriteAttributeScores()	930
6.106.3.13 WriteBestKs()	931
6.106.4 Member Data Documentation	931
6.106.4.1 algorithmMode	931
6.106.4.2 analysisType	931

6.106.4.3 bestKs	932
6.106.4.4 dataset	932
6.106.4.5 doRemovePercent	932
6.106.4.6 koptBegin	932
6.106.4.7 koptEnd	932
6.106.4.8 koptStep	933
6.106.4.9 numTargetAttributes	933
6.106.4.10numThreads	933
6.106.4.11numToRemoveNextIteration	933
6.106.4.12numToRemovePerIteration	933
6.106.4.13outFilesPrefix	934
6.106.4.14PP	934
6.106.4.15reliefseqAlgorithm	934
6.106.4.16removedAttributes	934
6.106.4.17removePercentage	934
6.106.4.18scores	935
6.107ResultTrio Class Reference	935
6.107.1 Detailed Description	936
6.107.2 Member Function Documentation	936
6.107.2.1 operator<()	936
6.107.3 Member Data Documentation	936
6.107.3.1 annot	936
6.107.3.2 f	937
6.107.3.3 p	937
6.107.3.4 s	937
6.108Rexp Class Reference	937
6.108.1 Detailed Description	939
6.108.2 Constructor & Destructor Documentation	939

6.108.2.1 <code>Rexp()</code> [1/3]	940
6.108.2.2 <code>Rexp()</code> [2/3]	940
6.108.2.3 <code>Rexp()</code> [3/3]	940
6.108.2.4 <code>~Rexp()</code>	940
6.108.3 Member Function Documentation	940
6.108.3.1 <code>attribute()</code>	941
6.108.3.2 <code>attributeNames()</code>	941
6.108.3.3 <code>length()</code>	941
6.108.3.4 <code>os_print()</code>	941
6.108.3.5 <code>parse()</code>	941
6.108.3.6 <code>set_master()</code>	942
6.108.3.7 <code>storageSize()</code>	942
6.108.3.8 <code>store()</code>	942
6.108.4 Friends And Related Function Documentation	942
6.108.4.1 <code>operator<< [1/2]</code>	942
6.108.4.2 <code>operator<< [2/2]</code>	942
6.108.5 Member Data Documentation	943
6.108.5.1 <code>attr</code>	943
6.108.5.2 <code>attribs</code>	943
6.108.5.3 <code>attrnames</code>	943
6.108.5.4 <code>data</code>	943
6.108.5.5 <code>len</code>	943
6.108.5.6 <code>master</code>	944
6.108.5.7 <code>msg</code>	944
6.108.5.8 <code>next</code>	944
6.108.5.9 <code>pos</code>	944
6.108.5.10 <code>count</code>	944
6.108.5.11 <code>type</code>	945

6.109Rinteger Class Reference	945
6.109.1 Detailed Description	948
6.109.2 Constructor & Destructor Documentation	948
6.109.2.1 Rinteger() [1/3]	948
6.109.2.2 Rinteger() [2/3]	948
6.109.2.3 Rinteger() [3/3]	949
6.109.3 Member Function Documentation	949
6.109.3.1 fix_content()	949
6.109.3.2 intArray()	949
6.109.3.3 intAt()	949
6.109.3.4 length()	949
6.109.3.5 os_print()	950
6.110Rlist Class Reference	950
6.110.1 Detailed Description	953
6.110.2 Constructor & Destructor Documentation	953
6.110.2.1 Rlist() [1/3]	953
6.110.2.2 Rlist() [2/3]	953
6.110.2.3 Rlist() [3/3]	954
6.110.2.4 ~Rlist()	954
6.110.3 Member Function Documentation	954
6.110.3.1 entryByTagName()	954
6.110.3.2 fix_content()	954
6.110.3.3 os_print()	954
6.110.4 Member Data Documentation	955
6.110.4.1 head	955
6.110.4.2 tag	955
6.110.4.3 tail	955
6.111Rmessage Class Reference	956

6.111.1 Detailed Description	957
6.111.2 Constructor & Destructor Documentation	957
6.111.2.1 Rmessage() [1/5]	957
6.111.2.2 Rmessage() [2/5]	957
6.111.2.3 Rmessage() [3/5]	958
6.111.2.4 Rmessage() [4/5]	958
6.111.2.5 Rmessage() [5/5]	958
6.111.2.6 ~Rmessage()	958
6.111.3 Member Function Documentation	958
6.111.3.1 command()	959
6.111.3.2 is_complete()	959
6.111.3.3 length()	959
6.111.3.4 parse()	959
6.111.3.5 read()	959
6.111.3.6 send()	959
6.111.4 Member Data Documentation	960
6.111.4.1 complete	960
6.111.4.2 data	960
6.111.4.3 head	960
6.111.4.4 len	960
6.111.4.5 par	960
6.111.4.6 pars	961
6.112 RReliefF Class Reference	961
6.112.1 Detailed Description	963
6.112.2 Constructor & Destructor Documentation	964
6.112.2.1 RReliefF()	964
6.112.2.2 ~RReliefF()	964
6.112.3 Member Function Documentation	964

6.112.3.1 ComputeAttributeScores()	964
6.113 Rstring Class Reference	965
6.113.1 Detailed Description	968
6.113.2 Constructor & Destructor Documentation	968
6.113.2.1 Rstring() [1/3]	968
6.113.2.2 Rstring() [2/3]	968
6.113.2.3 Rstring() [3/3]	968
6.113.3 Member Function Documentation	969
6.113.3.1 os_print()	969
6.113.3.2 string()	969
6.114 Rstrings Class Reference	969
6.114.1 Detailed Description	972
6.114.2 Constructor & Destructor Documentation	972
6.114.2.1 Rstrings() [1/2]	972
6.114.2.2 Rstrings() [2/2]	972
6.114.3 Member Function Documentation	973
6.114.3.1 count()	973
6.114.3.2 decode()	973
6.114.3.3 indexOfString()	973
6.114.3.4 os_print()	973
6.114.3.5 string()	973
6.114.3.6 stringAt()	974
6.114.3.7 strings()	974
6.114.4 Member Data Documentation	974
6.114.4.1 cont	974
6.114.4.2 nel	974
6.115 Rsymbol Class Reference	975
6.115.1 Detailed Description	977

6.115.2 Constructor & Destructor Documentation	977
6.115.2.1 Rsymbol() [1/2]	977
6.115.2.2 Rsymbol() [2/2]	977
6.115.3 Member Function Documentation	977
6.115.3.1 fix_content()	978
6.115.3.2 os_print()	978
6.115.3.3 symbolName()	978
6.115.4 Member Data Documentation	978
6.115.4.1 name	978
6.116 Rvector Class Reference	979
6.116.1 Detailed Description	981
6.116.2 Constructor & Destructor Documentation	981
6.116.2.1 Rvector() [1/2]	981
6.116.2.2 Rvector() [2/2]	982
6.116.2.3 ~Rvector()	982
6.116.3 Member Function Documentation	982
6.116.3.1 byName()	982
6.116.3.2 fix_content()	982
6.116.3.3 indexOf()	982
6.116.3.4 indexOfString()	983
6.116.3.5 os_print()	983
6.116.3.6 stringAt()	983
6.116.3.7 strings()	983
6.116.4 Member Data Documentation	983
6.116.4.1 capacity	983
6.116.4.2 cont	984
6.116.4.3 count	984
6.116.4.4 strs	984

6.117Segment Class Reference	984
6.117.1 Detailed Description	986
6.117.2 Constructor & Destructor Documentation	986
6.117.2.1 Segment()	986
6.117.3 Member Function Documentation	986
6.117.3.1 operator<()	986
6.117.3.2 operator==()	986
6.117.4 Member Data Documentation	986
6.117.4.1 baseline	987
6.117.4.2 count	987
6.117.4.3 finish	987
6.117.4.4 freq	987
6.117.4.5 p1	987
6.117.4.6 p2	987
6.117.4.7 score	988
6.117.4.8 sites	988
6.117.4.9 start	988
6.117.4.10 type	988
6.117.4.11 weightedBaseline	988
6.117.4.12 weightedCount	988
6.118SegmentSizeCmp Class Reference	989
6.118.1 Detailed Description	989
6.118.2 Member Function Documentation	989
6.118.2.1 operator()()	989
6.119Set Class Reference	990
6.119.1 Detailed Description	991
6.119.2 Constructor & Destructor Documentation	991
6.119.2.1 Set()	991

6.119.3 Member Function Documentation	991
6.119.3.1 cumulativeSetSum_WITHLABELS()	991
6.119.3.2 cumulativeSetSum_WITHOUTLABELS()	992
6.119.3.3 dropNotSet()	992
6.119.3.4 empiricalSetPValues()	992
6.119.3.5 fitLDSetTest()	992
6.119.3.6 fitStepwiseModel()	992
6.119.3.7 initialiseSetMapping()	993
6.119.3.8 makeLDSets()	993
6.119.3.9 profileTestInitialise()	993
6.119.3.10profileTestScore()	993
6.119.3.11profileTestSNPInformation()	993
6.119.3.12pruneMC()	994
6.119.3.13pruneSets()	994
6.119.3.14sizeSets()	994
6.119.4 Member Data Documentation	994
6.119.4.1 cur	994
6.119.4.2 IdSet	994
6.119.4.3 numSig	995
6.119.4.4 profileScore	995
6.119.4.5 profileSNPs	995
6.119.4.6 pv_maxE_set	995
6.119.4.7 pv_maxG_set	995
6.119.4.8 pv_set	995
6.119.4.9 s_max	996
6.119.4.10s_min	996
6.119.4.11selectedSNPs	996
6.119.4.12setMapping	996

6.119.4.13	setsort	996
6.119.4.14	snpset	996
6.119.4.15	stat_set	997
6.120	SetSortedSNP Class Reference	997
6.120.1	Detailed Description	997
6.120.2	Member Function Documentation	998
6.120.2.1	operator<()	998
6.120.3	Member Data Documentation	998
6.120.3.1	chisq	998
6.120.3.2	I	998
6.121	SimParameters Class Reference	999
6.121.1	Detailed Description	1000
6.121.2	Constructor & Destructor Documentation	1000
6.121.2.1	SimParameters()	1000
6.121.3	Member Data Documentation	1000
6.121.3.1	dprime	1000
6.121.3.2	hetOdds	1000
6.121.3.3	homOdds	1001
6.121.3.4	lfreq	1001
6.121.3.5	lmarker	1001
6.121.3.6	missing	1001
6.121.3.7	name	1001
6.121.3.8	nsnp	1001
6.121.3.9	ufreq	1002
6.121.3.10	umarker	1002
6.122	SimParametersQT Class Reference	1003
6.122.1	Detailed Description	1004
6.122.2	Constructor & Destructor Documentation	1004

6.122.2.1 SimParametersQT()	1004
6.122.3 Member Data Documentation	1004
6.122.3.1 dom	1004
6.122.3.2 dprime	1005
6.122.3.3 gAA	1005
6.122.3.4 gAB	1005
6.122.3.5 gBB	1005
6.122.3.6 Ifreq	1005
6.122.3.7 Imarker	1005
6.122.3.8 missing	1006
6.122.3.9 name	1006
6.122.3.10 hsnp	1006
6.122.3.11 ufreq	1006
6.122.3.12 umarker	1006
6.122.3.13 variance	1006
6.123 SInfo Class Reference	1007
6.123.1 Detailed Description	1007
6.123.2 Constructor & Destructor Documentation	1007
6.123.2.1 SInfo()	1007
6.123.3 Member Data Documentation	1008
6.123.3.1 d	1008
6.123.3.2 se	1008
6.124 SNP_INFO Struct Reference	1008
6.124.1 Detailed Description	1009
6.124.2 Member Data Documentation	1009
6.124.2.1 chrom	1010
6.124.2.2 position	1010
6.124.2.3 refAllele	1010

6.124.2.4 rsnum	1010
6.125SNReliefF Class Reference	1011
6.125.1 Detailed Description	1012
6.125.2 Constructor & Destructor Documentation	1013
6.125.2.1 SNReliefF()	1013
6.125.2.2 ~SNReliefF()	1013
6.125.3 Member Function Documentation	1013
6.125.3.1 ComputeAttributeScores()	1014
6.125.3.2 ComputeInstanceStats()	1014
6.125.3.3 PreComputeNeighborGeneStats()	1014
6.125.3.4 PrintInstanceAttributeStats()	1015
6.125.3.5 PrintNeighborStats()	1015
6.125.4 Member Data Documentation	1015
6.125.4.1 neighborStats	1015
6.126SortedResult Class Reference	1015
6.126.1 Detailed Description	1016
6.126.2 Member Function Documentation	1016
6.126.2.1 operator<()	1016
6.126.3 Member Data Documentation	1016
6.126.3.1 chisq	1016
6.126.3.2 l	1016
6.126.3.3 p	1017
6.127sortedSegments Class Reference	1017
6.127.1 Detailed Description	1018
6.127.2 Member Function Documentation	1018
6.127.2.1 operator<()	1018
6.127.3 Member Data Documentation	1018
6.127.3.1 s	1018

6.128 Transcript Class Reference	1019
6.128.1 Detailed Description	1020
6.128.2 Constructor & Destructor Documentation	1020
6.128.2.1 Transcript()	1020
6.128.3 Member Data Documentation	1020
6.128.3.1 bpEnd	1020
6.128.3.2 bpStart	1020
6.128.3.3 chr	1020
6.128.3.4 name	1021
6.129 Tree Class Reference	1021
6.129.1 Detailed Description	1024
6.129.2 Constructor & Destructor Documentation	1024
6.129.2.1 Tree() [1/2]	1024
6.129.2.2 Tree() [2/2]	1024
6.129.2.3 ~Tree()	1025
6.129.3 Member Function Documentation	1025
6.129.3.1 appendToFile()	1025
6.129.3.2 appendToFileInternal()	1025
6.129.3.3 bootstrap()	1025
6.129.3.4 bootstrapWeighted()	1025
6.129.3.5 bootstrapWithoutReplacement()	1026
6.129.3.6 bootstrapWithoutReplacementWeighted()	1026
6.129.3.7 cleanUpInternal()	1026
6.129.3.8 computePermutationImportance()	1026
6.129.3.9 computePredictionAccuracyInternal()	1026
6.129.3.10 createEmptyNode()	1027
6.129.3.11 createEmptyNodeInternal()	1027
6.129.3.12 createPossibleSplitVarSubset()	1027

6.129.3.13 <code>DISALLOW_COPY_AND_ASSIGN()</code>	1027
6.129.3.14 <code>dropDownSamplePermuted()</code>	1027
6.129.3.15 <code>getChildNodeIDs()</code>	1028
6.129.3.16 <code>getInbagCounts()</code>	1028
6.129.3.17 <code>getNumSamplesOob()</code>	1028
6.129.3.18 <code>getOobSampleIDs()</code>	1028
6.129.3.19 <code>getSplitValues()</code>	1028
6.129.3.20 <code>getSplitVarIDs()</code>	1028
6.129.3.21 <code>grow()</code>	1029
6.129.3.22 <code>init()</code>	1029
6.129.3.23 <code>initInternal()</code>	1029
6.129.3.24 <code>permuteAndPredictOobSamples()</code>	1030
6.129.3.25 <code>predict()</code>	1030
6.129.3.26 <code>splitNode()</code>	1030
6.129.3.27 <code>splitNodeInternal()</code>	1030
6.129.4 Member Data Documentation	1030
6.129.4.1 <code>alpha</code>	1030
6.129.4.2 <code>case_weights</code>	1031
6.129.4.3 <code>child_nodeIDs</code>	1031
6.129.4.4 <code>data</code>	1031
6.129.4.5 <code>dependent_varID</code>	1031
6.129.4.6 <code>deterministic_varIDs</code>	1031
6.129.4.7 <code>holdout</code>	1031
6.129.4.8 <code>importance_mode</code>	1032
6.129.4.9 <code>inbag_counts</code>	1032
6.129.4.10 <code>s_ordered_variable</code>	1032
6.129.4.11 <code>keep_inbag</code>	1032
6.129.4.12 <code>memory_saving_splitting</code>	1032

6.129.4.13min_node_size	1032
6.129.4.14minprop	1033
6.129.4.15mtry	1033
6.129.4.16no_split_variables	1033
6.129.4.17num_samples	1033
6.129.4.18num_samples_oob	1033
6.129.4.19oob_sampleIDs	1033
6.129.4.20prediction_terminal_nodeIDs	1034
6.129.4.21random_number_generator	1034
6.129.4.22sample_fraction	1034
6.129.4.23sample_with_replacement	1034
6.129.4.24sampleIDs	1034
6.129.4.25split_select_varIDs	1034
6.129.4.26split_select_weights	1035
6.129.4.27split_values	1035
6.129.4.28split_varIDs	1035
6.129.4.29splitrule	1035
6.129.4.30variable_importance	1035
6.130TreeClassification Class Reference	1036
6.130.1 Detailed Description	1038
6.130.2 Constructor & Destructor Documentation	1038
6.130.2.1 TreeClassification() [1/2]	1038
6.130.2.2 TreeClassification() [2/2]	1039
6.130.2.3 ~TreeClassification()	1039
6.130.3 Member Function Documentation	1039
6.130.3.1 addGiniImportance()	1039
6.130.3.2 appendToFileInternal()	1039
6.130.3.3 cleanUpInternal()	1040

6.130.3.4 computePermutationImportanceInternal()	1040
6.130.3.5 computePredictionAccuracyInternal()	1040
6.130.3.6 createEmptyNodeInternal()	1040
6.130.3.7 DISALLOW_COPY_AND_ASSIGN()	1040
6.130.3.8 estimate()	1041
6.130.3.9 findBestSplit()	1041
6.130.3.10 findBestSplitValueLargeQ()	1041
6.130.3.11 findBestSplitValueSmallQ()	1041
6.130.3.12 findBestSplitValueUnordered()	1042
6.130.3.13 getPrediction()	1042
6.130.3.14 getPredictionTerminalNodeID()	1042
6.130.3.15 initInternal()	1042
6.130.3.16 splitNodeInternal()	1043
6.130.4 Member Data Documentation	1043
6.130.4.1 class_values	1043
6.130.4.2 counter	1043
6.130.4.3 counter_per_class	1043
6.130.4.4 response_classIDs	1043
6.131 TreeProbability Class Reference	1044
6.131.1 Detailed Description	1046
6.131.2 Constructor & Destructor Documentation	1046
6.131.2.1 TreeProbability() [1/2]	1046
6.131.2.2 TreeProbability() [2/2]	1047
6.131.2.3 ~TreeProbability()	1047
6.131.3 Member Function Documentation	1047
6.131.3.1 addImpurityImportance()	1047
6.131.3.2 addToTerminalNodes()	1047
6.131.3.3 appendToFileInternal()	1048

6.131.3.4 cleanUpInternal()	1048
6.131.3.5 computePermutationImportanceInternal()	1048
6.131.3.6 computePredictionAccuracyInternal()	1048
6.131.3.7 createEmptyNodeInternal()	1048
6.131.3.8 DISALLOW_COPY_AND_ASSIGN()	1049
6.131.3.9 findBestSplit()	1049
6.131.3.10 findBestSplitValueLargeQ()	1049
6.131.3.11 findBestSplitValueSmallQ()	1049
6.131.3.12 findBestSplitValueUnordered()	1050
6.131.3.13 getPrediction()	1050
6.131.3.14 getPredictionTerminalNodeID()	1050
6.131.3.15 getTerminalClassCounts()	1050
6.131.3.16 initInternal()	1050
6.131.3.17 splitNodeInternal()	1051
6.131.4 Member Data Documentation	1051
6.131.4.1 class_values	1051
6.131.4.2 counter	1051
6.131.4.3 counter_per_class	1051
6.131.4.4 response_classIDs	1051
6.131.4.5 terminal_class_counts	1052
6.132 TreeRegression Class Reference	1052
6.132.1 Detailed Description	1055
6.132.2 Constructor & Destructor Documentation	1055
6.132.2.1 TreeRegression() [1/2]	1055
6.132.2.2 TreeRegression() [2/2]	1056
6.132.2.3 ~TreeRegression()	1056
6.132.3 Member Function Documentation	1056
6.132.3.1 addImpurityImportance()	1056

6.132.3.2 appendToFileInternal()	1056
6.132.3.3 cleanUpInternal()	1057
6.132.3.4 computePermutationImportanceInternal()	1057
6.132.3.5 computePredictionAccuracyInternal()	1057
6.132.3.6 computePredictionMSE()	1057
6.132.3.7 createEmptyNodeInternal()	1057
6.132.3.8 DISALLOW_COPY_AND_ASSIGN()	1058
6.132.3.9 estimate()	1058
6.132.3.10 findBestSplit()	1058
6.132.3.11 findBestSplitMaxstat()	1058
6.132.3.12 findBestSplitValueLargeQ()	1058
6.132.3.13 findBestSplitValueSmallQ()	1059
6.132.3.14 findBestSplitValueUnordered()	1059
6.132.3.15 getPrediction()	1059
6.132.3.16 getPredictionTerminalNodeID()	1059
6.132.3.17 initInternal()	1060
6.132.3.18 splitNodeInternal()	1060
6.132.4 Member Data Documentation	1060
6.132.4.1 counter	1060
6.132.4.2 sums	1060
6.133 TreeSurvival Class Reference	1061
6.133.1 Detailed Description	1063
6.133.2 Constructor & Destructor Documentation	1064
6.133.2.1 TreeSurvival() [1/2]	1064
6.133.2.2 TreeSurvival() [2/2]	1064
6.133.2.3 ~TreeSurvival()	1064
6.133.3 Member Function Documentation	1064
6.133.3.1 appendToFileInternal()	1064

6.133.3.2 cleanUpInternal()	1065
6.133.3.3 computeAucSplit()	1065
6.133.3.4 computeChildDeathCounts()	1065
6.133.3.5 computeDeathCounts()	1065
6.133.3.6 computePermutationImportanceInternal()	1066
6.133.3.7 computePredictionAccuracyInternal()	1066
6.133.3.8 computeSurvival()	1066
6.133.3.9 createEmptyNodeInternal()	1066
6.133.3.10 DISALLOW_COPY_AND_ASSIGN()	1066
6.133.3.11 findBestSplit()	1067
6.133.3.12 findBestSplitMaxstat()	1067
6.133.3.13 findBestSplitValueAUC()	1067
6.133.3.14 findBestSplitValueLogRank() [1/2]	1067
6.133.3.15 findBestSplitValueLogRank() [2/2]	1068
6.133.3.16 findBestSplitValueLogRankUnordered() [1/2]	1068
6.133.3.17 findBestSplitValueLogRankUnordered() [2/2]	1068
6.133.3.18 getChf()	1068
6.133.3.19 getPrediction()	1069
6.133.3.20 getPredictionTerminalNodeID()	1069
6.133.3.21 initInternal()	1069
6.133.3.22 splitNodeInternal()	1069
6.133.4 Member Data Documentation	1069
6.133.4.1 chf	1070
6.133.4.2 num_deaths	1070
6.133.4.3 num_samples_at_risk	1070
6.133.4.4 num_timepoints	1070
6.133.4.5 response_timepointIDs	1070
6.133.4.6 status_varID	1070

6.133.4.7 unique_timepoints	1071
6.134 Var Class Reference	1071
6.134.1 Detailed Description	1072
6.134.2 Member Function Documentation	1072
6.134.2.1 operator<()	1072
6.134.2.2 operator==()	1072
6.134.3 Member Data Documentation	1072
6.134.3.1 a1	1072
6.134.3.2 snp	1073
6.135 Variant Class Reference	1073
6.135.1 Detailed Description	1074
6.135.2 Constructor & Destructor Documentation	1075
6.135.2.1 Variant()	1075
6.135.3 Member Data Documentation	1075
6.135.3.1 acode	1075
6.135.3.2 alleles	1075
6.135.3.3 allelicVariation	1075
6.135.3.4 copyNumberVariation	1075
6.135.3.5 freqs	1076
6.135.3.6 integerDosage	1076
6.135.3.7 nallele	1076
6.136 WMLocus Class Reference	1076
6.136.1 Detailed Description	1077
6.136.2 Constructor & Destructor Documentation	1077
6.136.2.1 WMLocus()	1077
6.136.3 Member Function Documentation	1077
6.136.3.1 reset()	1077
6.136.4 Member Data Documentation	1077

6.136.4.1 allele	1078
6.136.4.2 chr	1078
6.136.4.3 name	1078
6.136.4.4 weight	1078
6.137Z Class Reference	1078
6.137.1 Detailed Description	1079
6.137.2 Constructor & Destructor Documentation	1079
6.137.2.1 Z()	1079
6.137.3 Member Data Documentation	1079
6.137.3.1 z0	1079
6.137.3.2 z1	1079
6.137.3.3 z2	1080
6.138ZInput Class Reference	1080
6.138.1 Detailed Description	1082
6.138.2 Constructor & Destructor Documentation	1082
6.138.2.1 ZInput() [1/2]	1082
6.138.2.2 ZInput() [2/2]	1082
6.138.3 Member Function Documentation	1082
6.138.3.1 close()	1083
6.138.3.2 endOfFile()	1083
6.138.3.3 open()	1083
6.138.3.4 readChar()	1083
6.138.3.5 readLine()	1083
6.138.3.6 tokenizeLine()	1084
6.138.3.7 unbuffered()	1084
6.138.4 Member Data Documentation	1084
6.138.4.1 buf	1084
6.138.4.2 compressed	1084

6.138.4.3 filename	1084
6.138.4.4 inf	1085
6.138.4.5 zinf	1085
6.139 ZOutput Class Reference	1085
6.139.1 Detailed Description	1087
6.139.2 Constructor & Destructor Documentation	1087
6.139.2.1 ZOutput() [1/2]	1087
6.139.2.2 ZOutput() [2/2]	1087
6.139.3 Member Function Documentation	1087
6.139.3.1 close()	1088
6.139.3.2 open()	1088
6.139.3.3 operator<<()	1088
6.139.3.4 unbuffered()	1088
6.139.3.5 write()	1088
6.139.3.6 writeLine()	1089
6.139.4 Member Data Documentation	1089
6.139.4.1 buf	1089
6.139.4.2 compressed	1089
6.139.4.3 filename	1089
6.139.4.4 outf	1089
6.139.4.5 zoutf	1090
6.140 ZZ Class Reference	1090
6.140.1 Detailed Description	1091
6.140.2 Constructor & Destructor Documentation	1091
6.140.2.1 ZZ()	1091
6.140.3 Member Data Documentation	1091
6.140.3.1 z00	1091
6.140.3.2 z01	1091
6.140.3.3 z02	1091
6.140.3.4 z10	1092
6.140.3.5 z11	1092
6.140.3.6 z12	1092
6.140.3.7 z20	1092
6.140.3.8 z21	1092
6.140.3.9 z22	1092

7 File Documentation	1093
7.1 src/annot.cpp File Reference	1093
7.1.1 Function Documentation	1094
7.1.1.1 filterRanges()	1094
7.1.2 Variable Documentation	1094
7.1.2.1 PP	1094
7.2 src/ArgumentHandler.cpp File Reference	1094
7.3 src/ArgumentHandler.h File Reference	1095
7.4 src/ArmadilloFuncs.cpp File Reference	1095
7.4.1 Function Documentation	1096
7.4.1.1 armaComputeCovariance()	1096
7.4.1.2 armaComputeSparseCovariance()	1097
7.4.1.3 armaDcgain()	1097
7.4.1.4 armaGetPlinkNumericToMatrixAll()	1097
7.4.1.5 armaGetPlinkNumericToMatrixCaseControl()	1097
7.4.1.6 armaReadMatrix()	1097
7.4.1.7 armaWriteMatrix()	1098
7.4.1.8 armaWriteSparseMatrix()	1098
7.5 src/ArmadilloFuncs.h File Reference	1098
7.5.1 Function Documentation	1099
7.5.1.1 armaComputeCovariance()	1099
7.5.1.2 armaComputeSparseCovariance()	1099
7.5.1.3 armaDcgain()	1099
7.5.1.4 armaGetPlinkNumericToMatrixAll()	1100
7.5.1.5 armaGetPlinkNumericToMatrixCaseControl()	1100
7.5.1.6 armaReadMatrix()	1100
7.5.1.7 armaWriteMatrix()	1100
7.5.1.8 armaWriteSparseMatrix()	1100

7.6 src/assoc.cpp File Reference	1101
7.6.1 Variable Documentation	1101
7.6.1.1 LOG	1101
7.7 src/AttributeRanker.cpp File Reference	1102
7.8 src/AttributeRanker.h File Reference	1102
7.9 src/BestN.h File Reference	1103
7.9.1 Detailed Description	1104
7.10 src/binput.cpp File Reference	1104
7.10.1 Variable Documentation	1105
7.10.1.1 LOG	1105
7.11 src/BirdseedData.cpp File Reference	1105
7.12 src/BirdseedData.h File Reference	1105
7.13 src/blox.cpp File Reference	1106
7.13.1 Variable Documentation	1107
7.13.1.1 PP	1107
7.14 src/bmerge.cpp File Reference	1108
7.15 src/cdflib.h File Reference	1108
7.15.1 Function Documentation	1110
7.15.1.1 algdiv()	1110
7.15.1.2 alngam()	1111
7.15.1.3 alnrel()	1111
7.15.1.4 apser()	1111
7.15.1.5 basym()	1111
7.15.1.6 bcorr()	1111
7.15.1.7 betaln()	1112
7.15.1.8 bfrac()	1112
7.15.1.9 bgrat()	1112
7.15.1.10 bpser()	1112

7.15.1.11 bratio()	1113
7.15.1.12 brcmp1()	1113
7.15.1.13 brcomp()	1113
7.15.1.14 bup()	1113
7.15.1.15 cdfbet()	1114
7.15.1.16 cdfbin()	1114
7.15.1.17 cdfchi()	1114
7.15.1.18 cdfchn()	1115
7.15.1.19 cdff()	1115
7.15.1.20 cdffnc()	1115
7.15.1.21 cdfgam()	1116
7.15.1.22 cdfnbn()	1116
7.15.1.23 cdfnor()	1116
7.15.1.24 cdfpoi()	1117
7.15.1.25 cdft()	1117
7.15.1.26 cumbet()	1117
7.15.1.27 cumbin()	1118
7.15.1.28 cumchi()	1118
7.15.1.29 cumchn()	1118
7.15.1.30 cumf()	1118
7.15.1.31 cumfnc()	1119
7.15.1.32 cumgam()	1119
7.15.1.33 cumnbn()	1119
7.15.1.34 cumnor()	1119
7.15.1.35 cumpoi()	1120
7.15.1.36 cumt()	1120
7.15.1.37 dbetrm()	1120
7.15.1.38 devpl()	1120

7.15.1.39 <code>dexpm1()</code>	1121
7.15.1.40 <code>dinvnr()</code>	1121
7.15.1.41 <code>dinvr()</code>	1121
7.15.1.42 <code>dlanor()</code>	1121
7.15.1.43 <code>dln1mx()</code>	1121
7.15.1.44 <code>dln1px()</code>	1122
7.15.1.45 <code>dlnbet()</code>	1122
7.15.1.46 <code>dlngam()</code>	1122
7.15.1.47 <code>dstinv()</code>	1122
7.15.1.48 <code>dstrem()</code>	1122
7.15.1.49 <code>dstzr()</code>	1123
7.15.1.50 <code>dt1()</code>	1123
7.15.1.51 <code>dzror()</code>	1123
7.15.1.52 <code>E0000()</code>	1123
7.15.1.53 <code>E0001()</code>	1124
7.15.1.54 <code>erf1()</code>	1124
7.15.1.55 <code>erfc1()</code>	1124
7.15.1.56 <code>esum()</code>	1124
7.15.1.57 <code>exparg()</code>	1125
7.15.1.58 <code>fifdint()</code>	1125
7.15.1.59 <code>fifdmax1()</code>	1125
7.15.1.60 <code>fifdmin1()</code>	1125
7.15.1.61 <code>fifdsign()</code>	1125
7.15.1.62 <code>fifidint()</code>	1126
7.15.1.63 <code>fifmod()</code>	1126
7.15.1.64 <code>fpser()</code>	1126
7.15.1.65 <code>ftnstop()</code>	1126
7.15.1.66 <code>gam1()</code>	1126

7.15.1.67 <code>gaminv()</code>	1127
7.15.1.68 <code>gamln()</code>	1127
7.15.1.69 <code>gamln1()</code>	1127
7.15.1.70 <code>grat1()</code>	1127
7.15.1.71 <code>gratio()</code>	1128
7.15.1.72 <code>gsumln()</code>	1128
7.15.1.73 <code>ipmpar()</code>	1128
7.15.1.74 <code>psi()</code>	1128
7.15.1.75 <code>rcomp()</code>	1128
7.15.1.76 <code>rexp()</code>	1129
7.15.1.77 <code>rlog()</code>	1129
7.15.1.78 <code>rlog1()</code>	1129
7.15.1.79 <code>spmpar()</code>	1129
7.15.1.80 <code>stvaln()</code>	1129
7.15.1.81 <code>Xgamm()</code>	1130
7.16 <code>src/CentralityRanker.cpp</code> File Reference	1130
7.17 <code>src/CentralityRanker.h</code> File Reference	1130
7.17.1 Enumeration Type Documentation	1131
7.17.1.1 <code>SolverMethod</code>	1131
7.18 <code>src/cfamily.cpp</code> File Reference	1132
7.18.1 Function Documentation	1132
7.18.1.1 <code>genrel()</code>	1133
7.18.1.2 <code>isAncestorOf()</code>	1133
7.18.1.3 <code>listAllAncestors()</code>	1133
7.18.1.4 <code>mCount()</code>	1133
7.19 <code>src/cfamily.h</code> File Reference	1134
7.19.1 Function Documentation	1134
7.19.1.1 <code>genrel()</code>	1135

7.20 src/ChiSquared.cpp File Reference	1135
7.21 src/ChiSquared.h File Reference	1135
7.22 src/clumpld.cpp File Reference	1136
7.22.1 Function Documentation	1137
7.22.1.1 returnFullRangeList()	1137
7.23 src/clumpld.h File Reference	1137
7.24 src/cluster.cpp File Reference	1138
7.24.1 Function Documentation	1139
7.24.1.1 cldist()	1140
7.24.1.2 groupAvgLink()	1140
7.24.1.3 homogeneous_clusters()	1140
7.24.1.4 pairable_cluster()	1140
7.24.1.5 selcon_inds()	1141
7.24.1.6 spec_clusters()	1141
7.24.2 Variable Documentation	1141
7.24.2.1 LOG	1141
7.25 src/cnv.cpp File Reference	1141
7.25.1 Function Documentation	1142
7.25.1.1 allSegmentsIntersecting()	1142
7.25.1.2 count_intersects()	1143
7.25.1.3 intersects()	1143
7.25.1.4 probOverlap()	1143
7.25.1.5 segmentCountCaseControls()	1143
7.25.1.6 weighted_count_intersects()	1144
7.25.2 Variable Documentation	1144
7.25.2.1 EPS_OVERLAP	1144
7.25.2.2 PP	1144
7.26 src/cnv.h File Reference	1144

7.26.1 Function Documentation	1145
7.26.1.1 allSegmentsIntersecting()	1145
7.26.1.2 count_intersects()	1145
7.26.1.3 intersects()	1145
7.26.1.4 segmentCountCaseControls()	1146
7.26.1.5 weighted_count_intersects()	1146
7.27 src/cnvqt.cpp File Reference	1146
7.27.1 Variable Documentation	1147
7.27.1.1 PP	1147
7.28 src/config.h File Reference	1147
7.29 src/crandom.cpp File Reference	1148
7.30 src/crandom.h File Reference	1148
7.31 src/Data.cpp File Reference	1149
7.32 src/Data.h File Reference	1149
7.33 src/DataChar.cpp File Reference	1150
7.34 src/DataChar.h File Reference	1150
7.35 src/DataDouble.cpp File Reference	1151
7.36 src/DataDouble.h File Reference	1152
7.37 src/DataFloat.cpp File Reference	1153
7.38 src/DataFloat.h File Reference	1153
7.39 src/Dataset.cpp File Reference	1154
7.39.1 Variable Documentation	1155
7.39.1.1 PP	1155
7.40 src/Dataset.h File Reference	1156
7.41 src/DatasetInstance.cpp File Reference	1156
7.41.1 Typedef Documentation	1157
7.41.1.1 T	1157
7.42 src/DatasetInstance.h File Reference	1157

7.43 src/dcdflib.cpp File Reference	1158
7.43.1 Macro Definition Documentation	1162
7.43.1.1 atol [1/10]	1162
7.43.1.2 atol [2/10]	1162
7.43.1.3 atol [3/10]	1162
7.43.1.4 atol [4/10]	1162
7.43.1.5 atol [5/10]	1162
7.43.1.6 atol [6/10]	1163
7.43.1.7 atol [7/10]	1163
7.43.1.8 atol [8/10]	1163
7.43.1.9 atol [9/10]	1163
7.43.1.10 atol [10/10]	1163
7.43.1.11 dennon	1163
7.43.1.12 dg	1163
7.43.1.13 dlsqpi	1164
7.43.1.14 done [1/2]	1164
7.43.1.15 done [2/2]	1164
7.43.1.16 eps	1164
7.43.1.17 ftol	1164
7.43.1.18 half [1/2]	1164
7.43.1.19 half [2/2]	1164
7.43.1.20 hln2pi [1/2]	1165
7.43.1.21 hln2pi [2/2]	1165
7.43.1.22 inf [1/10]	1165
7.43.1.23 inf [2/10]	1165
7.43.1.24 inf [3/10]	1165
7.43.1.25 inf [4/10]	1165
7.43.1.26 inf [5/10]	1165

7.43.1.27 inf [6/10]	1166
7.43.1.28 inf [7/10]	1166
7.43.1.29 inf [8/10]	1166
7.43.1.30 inf [9/10]	1166
7.43.1.31 inf [10/10]	1166
7.43.1.32 maxdf	1166
7.43.1.33 maxit	1166
7.43.1.34 ncoef	1167
7.43.1.35 nhalf	1167
7.43.1.36 one [1/5]	1167
7.43.1.37 one [2/5]	1167
7.43.1.38 one [3/5]	1167
7.43.1.39 one [4/5]	1167
7.43.1.40 one [5/5]	1167
7.43.1.41 qsmall [1/2]	1168
7.43.1.42 qsmall [2/2]	1168
7.43.1.43 qtired	1168
7.43.1.44 qxmon	1168
7.43.1.45 r2pi	1168
7.43.1.46 tent4 [1/2]	1168
7.43.1.47 tent4 [2/2]	1169
7.43.1.48 tol [1/10]	1169
7.43.1.49 tol [2/10]	1169
7.43.1.50 tol [3/10]	1169
7.43.1.51 tol [4/10]	1169
7.43.1.52 tol [5/10]	1169
7.43.1.53 tol [6/10]	1169
7.43.1.54 tol [7/10]	1170

7.43.1.55 tol [8/10]	1170
7.43.1.56 tol [9/10]	1170
7.43.1.57 tol [10/10]	1170
7.43.1.58 zero [1/8]	1170
7.43.1.59 zero [2/8]	1170
7.43.1.60 zero [3/8]	1170
7.43.1.61 zero [4/8]	1171
7.43.1.62 zero [5/8]	1171
7.43.1.63 zero [6/8]	1171
7.43.1.64 zero [7/8]	1171
7.43.1.65 zero [8/8]	1171
7.43.2 Function Documentation	1171
7.43.2.1 algdiv()	1171
7.43.2.2 alngam()	1172
7.43.2.3 alnrel()	1172
7.43.2.4 apser()	1172
7.43.2.5 basym()	1172
7.43.2.6 bcorr()	1172
7.43.2.7 betaln()	1173
7.43.2.8 bfrac()	1173
7.43.2.9 bgrat()	1173
7.43.2.10 bpser()	1173
7.43.2.11 bratio()	1174
7.43.2.12 brcmp1()	1174
7.43.2.13 brcomp()	1174
7.43.2.14 bup()	1174
7.43.2.15 cdfbet()	1175
7.43.2.16 cdfbin()	1175

7.43.2.17 <code>cdfchi()</code>	1175
7.43.2.18 <code>cdfchn()</code>	1176
7.43.2.19 <code>cdff()</code>	1176
7.43.2.20 <code>cdffnc()</code>	1176
7.43.2.21 <code>cfgam()</code>	1177
7.43.2.22 <code>cdfnbn()</code>	1177
7.43.2.23 <code>cdfnor()</code>	1177
7.43.2.24 <code>cdfpoi()</code>	1178
7.43.2.25 <code>cdft()</code>	1178
7.43.2.26 <code>cumbet()</code>	1178
7.43.2.27 <code>cumbin()</code>	1179
7.43.2.28 <code>cumchi()</code>	1179
7.43.2.29 <code>cumchn()</code>	1179
7.43.2.30 <code>cumf()</code>	1179
7.43.2.31 <code>cumfnc()</code>	1180
7.43.2.32 <code>cumgam()</code>	1180
7.43.2.33 <code>cumnbn()</code>	1180
7.43.2.34 <code>cumnor()</code>	1180
7.43.2.35 <code>cumpoi()</code>	1181
7.43.2.36 <code>cumt()</code>	1181
7.43.2.37 <code>dbetrm()</code>	1181
7.43.2.38 <code>devlpl()</code>	1181
7.43.2.39 <code>dexpm1()</code>	1182
7.43.2.40 <code>dinvnr()</code>	1182
7.43.2.41 <code>dinvr()</code>	1182
7.43.2.42 <code>dlanor()</code>	1182
7.43.2.43 <code>dln1mx()</code>	1182
7.43.2.44 <code>dln1px()</code>	1183

7.43.2.45 dlnbet()	1183
7.43.2.46 dlngam()	1183
7.43.2.47 dstinv()	1183
7.43.2.48 dstrem()	1183
7.43.2.49 dstrzr()	1184
7.43.2.50 dt1()	1184
7.43.2.51 dzror()	1184
7.43.2.52 E0000()	1184
7.43.2.53 E0001()	1185
7.43.2.54 erf1()	1185
7.43.2.55 erfc1()	1185
7.43.2.56 esum()	1185
7.43.2.57 exparg()	1186
7.43.2.58 fifdint()	1186
7.43.2.59 fifdmax1()	1186
7.43.2.60 fifdmin1()	1186
7.43.2.61 fifdsign()	1186
7.43.2.62 fifidint()	1187
7.43.2.63 fifmod()	1187
7.43.2.64 fpser()	1187
7.43.2.65 fnstop()	1187
7.43.2.66 gam1()	1187
7.43.2.67 gaminv()	1188
7.43.2.68 gamln()	1188
7.43.2.69 gamln1()	1188
7.43.2.70 grat1()	1188
7.43.2.71 gratio()	1189
7.43.2.72 gsumln()	1189

7.43.2.73 <code>psi()</code>	1189
7.43.2.74 <code>rcomp()</code>	1189
7.43.2.75 <code>rexp()</code>	1189
7.43.2.76 <code>rlog()</code>	1190
7.43.2.77 <code>rlog1()</code>	1190
7.43.2.78 <code>spmpar()</code>	1190
7.43.2.79 <code>stvaln()</code>	1190
7.43.2.80 <code>Xgamm()</code>	1190
7.44 src/dcdflib.h File Reference	1191
7.44.1 Function Documentation	1191
7.44.1.1 <code>cdfbet()</code>	1191
7.44.1.2 <code>cdfbin()</code>	1192
7.44.1.3 <code>cdfchi()</code>	1192
7.44.1.4 <code>cdfchn()</code>	1192
7.44.1.5 <code>cdff()</code>	1193
7.44.1.6 <code>cdffnc()</code>	1193
7.44.1.7 <code>cdfgam()</code>	1193
7.44.1.8 <code>cdfnbn()</code>	1194
7.44.1.9 <code>cdfnor()</code>	1194
7.44.1.10 <code>cdfpoi()</code>	1194
7.44.1.11 <code>cdft()</code>	1195
7.45 src/DcVar.cpp File Reference	1195
7.45.1 Function Documentation	1196
7.45.1.1 <code>chipseqComparatorAscending()</code>	1196
7.45.1.2 <code>pvalComparatorAscending()</code>	1196
7.45.1.3 <code>snpinfoComparatorAscending()</code>	1196
7.46 src/DcVar.h File Reference	1196
7.46.1 Typedef Documentation	1198

7.46.1.1	CHIP_SEQ_INFO_LIST	1198
7.46.1.2	SNP_INFO_LIST	1198
7.46.1.3	SNP_INFO_LIST_IT	1198
7.46.2	Enumeration Type Documentation	1198
7.46.2.1	CHIP_SEQ_EXTRACT_FIELD_IDX	1198
7.46.2.2	SNP_EXTRACT_FIELD_IDX	1199
7.46.2.3	SNP_INPUT_TYPE	1199
7.46.3	Variable Documentation	1199
7.46.3.1	CHECKPOINT_FILENAME	1199
7.47	src/Dseq.cpp File Reference	1200
7.48	src/Dseq.h File Reference	1200
7.49	src/dfam.cpp File Reference	1201
7.50	src/DgeData.cpp File Reference	1202
7.51	src/DgeData.h File Reference	1202
7.52	src/DistanceMetrics.cpp File Reference	1203
7.52.1	Function Documentation	1204
7.52.1.1	CheckMissing()	1204
7.52.1.2	CheckMissingNumeric()	1204
7.52.1.3	diffAMM()	1205
7.52.1.4	diffEuclidean()	1205
7.52.1.5	diffGMM()	1206
7.52.1.6	diffKM()	1206
7.52.1.7	diffManhattan()	1207
7.52.1.8	diffNCA()	1207
7.52.1.9	diffPredictedValueTau()	1208
7.52.1.10	diffTITV()	1208
7.52.1.11	norm()	1209
7.53	src/DistanceMetrics.h File Reference	1209

7.53.1	Detailed Description	1210
7.53.2	Function Documentation	1210
7.53.2.1	CheckMissing()	1210
7.53.2.2	CheckMissingNumeric()	1211
7.53.2.3	diffAMM()	1211
7.53.2.4	diffEuclidean()	1212
7.53.2.5	diffGMM()	1212
7.53.2.6	diffKM()	1213
7.53.2.7	diffManhattan()	1213
7.53.2.8	diffNCA()	1214
7.53.2.9	diffPredictedValueTau()	1214
7.53.2.10	diffTTIV()	1215
7.53.2.11	norm()	1215
7.54	src/dosage.cpp File Reference	1216
7.54.1	Function Documentation	1217
7.54.1.1	setUpQScoring()	1217
7.54.2	Variable Documentation	1217
7.54.2.1	PP	1217
7.55	src/Edger.cpp File Reference	1217
7.56	src/Edger.h File Reference	1218
7.57	src/elf.cpp File Reference	1218
7.57.1	Function Documentation	1219
7.57.1.1	displayScoresPerson()	1219
7.57.1.2	displayScoresRegion()	1220
7.57.2	Variable Documentation	1220
7.57.2.1	PP	1220
7.58	src/em.cpp File Reference	1220
7.58.1	Variable Documentation	1221

7.58.1.1 LOG	1221
7.59 src/epi.cpp File Reference	1221
7.59.1 Variable Documentation	1222
7.59.1.1 LOG	1222
7.60 src/EpistasisEQtl.cpp File Reference	1222
7.61 src/EpistasisEQtl.h File Reference	1222
7.61.1 Typedef Documentation	1224
7.61.1.1 CoordinateTable	1224
7.61.1.2 CoordinateTableClt	1224
7.61.1.3 TranscriptFactorTable	1224
7.61.1.4 TranscriptFactorTableClt	1224
7.61.2 Enumeration Type Documentation	1224
7.61.2.1 COORD_FIELDS	1224
7.62 src/EvaporativeCooling.cpp File Reference	1225
7.63 src/EvaporativeCooling.h File Reference	1225
7.64 src/EvaporativeCoolingPrivacy.cpp File Reference	1226
7.65 src/EvaporativeCoolingPrivacy.h File Reference	1227
7.65.1 Typedef Documentation	1228
7.65.1.1 ResultsLists	1228
7.65.2 Enumeration Type Documentation	1228
7.65.2.1 DATASET_TYPE	1228
7.66 src/ExpressionDataSimulator.h File Reference	1229
7.66.1 Enumeration Type Documentation	1229
7.66.1.1 SplitType	1229
7.67 src/filters.cpp File Reference	1230
7.67.1 Macro Definition Documentation	1230
7.67.1.1 MISSING	1231
7.68 src/fisher.cpp File Reference	1231

7.68.1 Macro Definition Documentation	1232
7.68.1.1 dwrk	1232
7.68.1.2 i_int	1232
7.68.1.3 i_real	1232
7.68.1.4 iwrk	1233
7.68.1.5 max	1233
7.68.1.6 min	1233
7.68.1.7 rwrk	1233
7.68.1.8 SINT_MAX	1233
7.68.2 Function Documentation	1233
7.68.2.1 alogam()	1234
7.68.2.2 f10act()	1234
7.68.2.3 f11act()	1234
7.68.2.4 f2xact()	1235
7.68.2.5 f3xact()	1235
7.68.2.6 f4xact()	1236
7.68.2.7 f5xact()	1236
7.68.2.8 f6xact()	1237
7.68.2.9 f7xact()	1237
7.68.2.10 f8xact()	1237
7.68.2.11 f9xact()	1237
7.68.2.12 fexact()	1238
7.68.2.13 fisher()	1238
7.68.2.14 gammdd()	1238
7.68.2.15 isort()	1238
7.68.2.16 iwork()	1239
7.68.2.17 prterr()	1239
7.69 src/fisher.h File Reference	1239

7.69.1 Macro Definition Documentation	1240
7.69.1.1 DOUBLE_DIGITS	1240
7.69.1.2 DOUBLE_EPS	1241
7.69.1.3 DOUBLE_XMAX	1241
7.69.1.4 DOUBLE_XMIN	1241
7.69.1.5 M_PI	1241
7.69.1.6 PI	1241
7.69.1.7 R_EXT_BOOLEAN_H_	1241
7.69.1.8 R_EXT_CONSTANTS_H_	1242
7.69.1.9 R_EXT_MEMORY_H_	1242
7.69.1.10 SINGLE_BASE	1242
7.69.1.11 SINGLE_EPS	1242
7.69.1.12 SINGLE_XMAX	1242
7.69.1.13 SINGLE_XMIN	1242
7.69.2 Enumeration Type Documentation	1242
7.69.2.1 Rboolean	1242
7.69.3 Function Documentation	1243
7.69.3.1 fexact()	1243
7.69.3.2 fisher()	1243
7.69.3.3 R_alloc()	1243
7.69.3.4 R_gc()	1244
7.69.3.5 S_alloc()	1244
7.69.3.6 S_realloc()	1244
7.69.3.7 vmaxget()	1244
7.69.3.8 vmaxset()	1244
7.70 src/flip.cpp File Reference	1245
7.71 src/Forest.cpp File Reference	1245
7.72 src/Forest.h File Reference	1246

7.73 src/ForestClassification.cpp File Reference	1247
7.74 src/ForestClassification.h File Reference	1247
7.75 src/ForestProbability.cpp File Reference	1248
7.76 src/ForestProbability.h File Reference	1248
7.77 src/ForestRegression.cpp File Reference	1249
7.78 src/ForestRegression.h File Reference	1250
7.79 src/ForestSurvival.cpp File Reference	1251
7.80 src/ForestSurvival.h File Reference	1251
7.81 src/genedrop.cpp File Reference	1252
7.82 src/genepi.cpp File Reference	1253
7.82.1 Typedef Documentation	1254
7.82.1.1 vector_tld	1254
7.82.2 Function Documentation	1254
7.82.2.1 bartlett()	1254
7.82.2.2 betaln()	1254
7.82.2.3 bico()	1254
7.82.2.4 C()	1255
7.82.2.5 calcGENEPIMeanVariance()	1255
7.82.2.6 CCA_caseonly()	1255
7.82.2.7 CCA_logit()	1255
7.82.2.8 factln()	1256
7.82.2.9 I()	1256
7.82.2.10 pillai()	1256
7.82.2.11 transposeMatrix()	1256
7.83 src/genoerr.cpp File Reference	1257
7.83.1 Variable Documentation	1257
7.83.1.1 LOG	1257
7.84 src/genogroup.cpp File Reference	1258

7.85 src/genogroup.h File Reference	1258
7.86 src/genome.cpp File Reference	1259
7.87 src/glm.cpp File Reference	1259
7.88 src/globals.h File Reference	1260
7.88.1 Macro Definition Documentation	1261
7.88.1.1 DISALLOW_COPY_AND_ASSIGN	1261
7.88.1.2 M_PI	1261
7.88.2 Typedef Documentation	1262
7.88.2.1 uint	1262
7.88.3 Enumeration Type Documentation	1262
7.88.3.1 ImportanceMode	1262
7.88.3.2 MemoryMode	1262
7.88.3.3 PredictionType	1263
7.88.3.4 SplitRule	1263
7.88.3.5 TreeType	1263
7.88.4 Variable Documentation	1263
7.88.4.1 DEFAULT_ALPHA	1264
7.88.4.2 DEFAULT_IMPORTANCE_MODE	1264
7.88.4.3 DEFAULT_MIN_NODE_SIZE_CLASSIFICATION	1264
7.88.4.4 DEFAULT_MIN_NODE_SIZE_PROBABILITY	1264
7.88.4.5 DEFAULT_MIN_NODE_SIZE_REGRESSION	1264
7.88.4.6 DEFAULT_MIN_NODE_SIZE_SURVIVAL	1264
7.88.4.7 DEFAULT_MINPROP	1265
7.88.4.8 DEFAULT_NUM_THREADS	1265
7.88.4.9 DEFAULT_NUM_TREE	1265
7.88.4.10 DEFAULT_PREDICTIONTYPE	1265
7.88.4.11 DEFAULT_SPLITRULE	1265
7.88.4.12 mask	1265

7.88.4.13 MAX_IMP_MODE	1266
7.88.4.14 MAX_MEM_MODE	1266
7.88.4.15 offset	1266
7.88.4.16 Q_THRESHOLD	1266
7.88.4.17 STATUS_INTERVAL	1266
7.89 src/greport.cpp File Reference	1267
7.89.1 Variable Documentation	1267
7.89.1.1 PP	1267
7.90 src/GSLRandomBase.h File Reference	1268
7.91 src/GSLRandomFlat.h File Reference	1268
7.92 src/gvar.cpp File Reference	1269
7.92.1 Function Documentation	1270
7.92.1.1 analyseModel()	1270
7.92.1.2 compareModels()	1270
7.93 src/gvar.h File Reference	1271
7.94 src/gxe.cpp File Reference	1271
7.95 src/hapglm.cpp File Reference	1272
7.96 src/haploCC.cpp File Reference	1273
7.97 src/haplohelper.cpp File Reference	1273
7.97.1 Variable Documentation	1274
7.97.1.1 LOG	1274
7.98 src/haploQTL.cpp File Reference	1274
7.99 src/haploTDT.cpp File Reference	1275
7.100src/haplindow.cpp File Reference	1275
7.100.1 Function Documentation	1276
7.100.1.1 verboseDisplayWindows2()	1276
7.101src/haplindow.h File Reference	1276
7.102src/helper.cpp File Reference	1277

7.102.1 Macro Definition Documentation	1279
7.102.1.1 FPMIN	1279
7.102.1.2 MISSING1	1280
7.102.1.3 MISSING2	1280
7.102.2 Function Documentation	1280
7.102.2.1 affCoding()	1280
7.102.2.2 calcSetCovarianceMatrix()	1280
7.102.2.3 checkDups()	1280
7.102.2.4 checkFileExists() [1/2]	1281
7.102.2.5 checkFileExists() [2/2]	1281
7.102.2.6 chromosomeName()	1281
7.102.2.7 commaParse()	1281
7.102.2.8 compressed()	1281
7.102.2.9 dbl2str()	1282
7.102.2.10 dbl2str_fixed()	1282
7.102.2.11 defineCowChromosomes()	1282
7.102.2.12 defineDogChromosomes()	1282
7.102.2.13 defineHorseChromosomes()	1282
7.102.2.14 defineHumanChromosomes()	1283
7.102.2.15 defineMouseChromosomes()	1283
7.102.2.16 defineRiceChromosomes()	1283
7.102.2.17 defineSheepChromosomes()	1283
7.102.2.18 display() [1/4]	1283
7.102.2.19 display() [2/4]	1284
7.102.2.20 display() [3/4]	1284
7.102.2.21 display() [4/4]	1284
7.102.2.22 displayLine()	1284
7.102.2.23 doesFileExist()	1284

7.102.2.24error()	1285
7.102.2.25filterRanges()	1285
7.102.2.26geno2matrix()	1285
7.102.2.27genotype() [1/2]	1285
7.102.2.28genotype() [2/2]	1285
7.102.2.29genotypeToFile()	1286
7.102.2.30genotypingRate()	1286
7.102.2.31getChromosomeCode()	1286
7.102.2.32getChromosomeMarkerRange()	1286
7.102.2.33getChromosomeRange()	1286
7.102.2.34getDouble()	1287
7.102.2.35getInt()	1287
7.102.2.36getLongUnsignedInt()	1287
7.102.2.37getMarkerChromosome()	1287
7.102.2.38getMarkerNumber()	1287
7.102.2.39getWindowRange()	1288
7.102.2.40identicalSNPs()	1288
7.102.2.41int2str()	1288
7.102.2.42s_rare()	1288
7.102.2.43toa()	1288
7.102.2.44leftWindowEdge()	1289
7.102.2.45listPossibleHaplotypes()	1289
7.102.2.46longint2str()	1289
7.102.2.47makeLocusMap()	1289
7.102.2.48makePersonMap()	1289
7.102.2.49makeScaffold()	1290
7.102.2.50mapRanges2SNP()	1290
7.102.2.51mapRangesToSNPs()	1290

7.102.2.52mapSNPs2Range()	1290
7.102.2.53matrixGetNumericAll()	1290
7.102.2.54matrixGetNumericCaseControl()	1291
7.102.2.55modelComparisonPValue()	1291
7.102.2.56NoMem()	1291
7.102.2.57nvec_bool()	1291
7.102.2.58parse2int()	1291
7.102.2.59parse2str()	1292
7.102.2.60permute() [1/2]	1292
7.102.2.61permute() [2/2]	1292
7.102.2.62angelIntersect()	1292
7.102.2.63readRange()	1292
7.102.2.64readString()	1293
7.102.2.65relType()	1293
7.102.2.66removeMissingPhenotypes()	1293
7.102.2.67rightWindowEdge()	1293
7.102.2.68searchAndReplace()	1293
7.102.2.69seeChromosome()	1294
7.102.2.70shutdown()	1294
7.102.2.71sizefMatrix()	1294
7.102.2.72sizeMatrix()	1294
7.102.2.73sizeTable()	1294
7.102.2.74smoother()	1295
7.102.2.75SNPHWE()	1295
7.102.2.76summaryBasics()	1295
7.102.2.77sw() [1/4]	1295
7.102.2.78sw() [2/4]	1296
7.102.2.79sw() [3/4]	1296

7.102.2.80sw() [4/4]	1296
7.102.2.81tokenizeLine() [1/3]	1296
7.102.2.82tokenizeLine() [2/3]	1296
7.102.2.83tokenizeLine() [3/3]	1297
7.102.2.84two_locus_table()	1297
7.102.2.85vif_prune()	1297
7.102.3 Variable Documentation	1297
7.102.3.1 LOG	1297
7.102.3.2 PP	1297
7.103src/helper.h File Reference	1298
7.103.1 Typedef Documentation	1300
7.103.1.1 matrix_t	1300
7.103.1.2 vector_t	1301
7.103.2 Function Documentation	1301
7.103.2.1 affCoding()	1301
7.103.2.2 calcSetCovarianceMatrix()	1301
7.103.2.3 checkDups()	1301
7.103.2.4 checkFileExists() [1/2]	1301
7.103.2.5 checkFileExists() [2/2]	1302
7.103.2.6 chromosomeName()	1302
7.103.2.7 commaParse()	1302
7.103.2.8 compressed()	1302
7.103.2.9 dbl2str()	1302
7.103.2.10dbl2str_fixed()	1303
7.103.2.11defineCowChromosomes()	1303
7.103.2.12defineDogChromosomes()	1303
7.103.2.13defineHorseChromosomes()	1303
7.103.2.14defineHumanChromosomes()	1303

7.103.2.15defineMouseChromosomes()	1304
7.103.2.16defineRiceChromosomes()	1304
7.103.2.17defineSheepChromosomes()	1304
7.103.2.18display() [1/4]	1304
7.103.2.19display() [2/4]	1304
7.103.2.20display() [3/4]	1305
7.103.2.21display() [4/4]	1305
7.103.2.22displayLine()	1305
7.103.2.23doesFileExist()	1305
7.103.2.24error()	1305
7.103.2.25FDR_BH()	1306
7.103.2.26filterRanges()	1306
7.103.2.27from_string()	1306
7.103.2.28geno2matrix()	1306
7.103.2.29genotype() [1/2]	1306
7.103.2.30genotype() [2/2]	1307
7.103.2.31genotypeToFile()	1307
7.103.2.32genotypingRate()	1307
7.103.2.33getChromosomeCode()	1307
7.103.2.34getChromosomeMarkerRange()	1307
7.103.2.35getChromosomeRange()	1308
7.103.2.36getDouble()	1308
7.103.2.37getInt()	1308
7.103.2.38getLongUnsignedInt()	1308
7.103.2.39getMarkerChromosome()	1308
7.103.2.40getMarkerNumber()	1309
7.103.2.41getWindowRange()	1309
7.103.2.42identicalSNPs()	1309

7.103.2.43int2str()	1309
7.103.2.44itoa()	1309
7.103.2.45leftWindowEdge()	1310
7.103.2.46listPossibleHaplotypes()	1310
7.103.2.47longint2str()	1310
7.103.2.48makeLocusMap()	1310
7.103.2.49makePersonMap()	1310
7.103.2.50makeScaffold()	1311
7.103.2.51mapRanges2SNP()	1311
7.103.2.52mapRangesToSNPs()	1311
7.103.2.53mapSNPs2Range()	1311
7.103.2.54matrixGetNumericAll()	1311
7.103.2.55matrixGetNumericCaseControl()	1312
7.103.2.56MAX()	1312
7.103.2.57MIN()	1312
7.103.2.58modelComparisonPValue()	1312
7.103.2.59NoMem()	1312
7.103.2.60nvec_bool()	1313
7.103.2.61parse2int()	1313
7.103.2.62parse2str()	1313
7.103.2.63permute() [1/2]	1313
7.103.2.64permute() [2/2]	1313
7.103.2.65rangeIntersect()	1314
7.103.2.66readRange()	1314
7.103.2.67readString()	1314
7.103.2.68relType()	1314
7.103.2.69removeMissingPhenotypes()	1314
7.103.2.70rightWindowEdge()	1315

7.103.2.71searchAndReplace()	1315
7.103.2.72seeChromosome()	1315
7.103.2.73shutdown()	1315
7.103.2.74SIGN()	1315
7.103.2.75sizeMatrix()	1316
7.103.2.76sizeMatrix()	1316
7.103.2.77sizeTable()	1316
7.103.2.78smoother()	1316
7.103.2.79SNPHWE()	1317
7.103.2.80SQR()	1317
7.103.2.81summaryBasics()	1317
7.103.2.82sw() [1/4]	1317
7.103.2.83sw() [2/4]	1317
7.103.2.84sw() [3/4]	1318
7.103.2.85sw() [4/4]	1318
7.103.2.86SWAP()	1318
7.103.2.87tokenizeLine() [1/3]	1318
7.103.2.88tokenizeLine() [2/3]	1318
7.103.2.89tokenizeLine() [3/3]	1319
7.103.2.90two_locus_table()	1319
7.103.2.91vif_prune()	1319
7.104src/homozyg.cpp File Reference	1319
7.104.1 Function Documentation	1320
7.104.1.1 compareCNVs()	1321
7.104.1.2 displayPoolVerbose()	1321
7.104.1.3 segsIBDMatch()	1321
7.104.1.4 segsIBDMatchCON()	1321
7.104.1.5 segsMatch()	1322

7.104.1.6 segsMatchCON()	1322
7.104.1.7 segsOverlap()	1322
7.104.1.8 summaryIndivSummaries()	1322
7.104.2 Variable Documentation	1322
7.104.2.1 PP	1323
7.105src/hotel.cpp File Reference	1323
7.105.1 Function Documentation	1323
7.105.1.1 calcHotelSetMeanVariance()	1324
7.106src/idhelp.cpp File Reference	1324
7.106.1 Variable Documentation	1324
7.106.1.1 PP	1325
7.107src/idhelp.h File Reference	1325
7.108src/impute.cpp File Reference	1326
7.108.1 Variable Documentation	1326
7.108.1.1 LOG	1326
7.109src/inbix.cpp File Reference	1327
7.109.1 Function Documentation	1328
7.109.1.1 main()	1328
7.109.2 Variable Documentation	1328
7.109.2.1 LOG	1328
7.109.2.2 PDATE	1329
7.109.2.3 PP	1329
7.109.2.4 PREL	1329
7.109.2.5 PVERSION	1329
7.110src/informative.cpp File Reference	1329
7.111src/input.cpp File Reference	1330
7.111.1 Variable Documentation	1330
7.111.1.1 LOG	1331

7.112src/Insilico.cpp File Reference	1331
7.112.1 Function Documentation	1332
7.112.1.1 ChooseSnpDatasetByType()	1332
7.112.1.2 DetectClassType()	1332
7.112.1.3 GetConfigValue()	1333
7.112.1.4 GetFileBasename()	1333
7.112.1.5 GetFileExtension()	1334
7.112.1.6 GetMatchingIds()	1334
7.112.1.7 LoadNumericIds()	1335
7.112.1.8 LoadPhenoids()	1335
7.112.1.9 ProtectedLog()	1336
7.112.1.10 scoresSortAsc()	1336
7.112.1.11 scoresSortAscByName()	1336
7.112.1.12 scoresSortDesc()	1336
7.112.1.13 timestamp()	1337
7.113src/Insilico.h File Reference	1337
7.113.1 Detailed Description	1340
7.113.2 Typedef Documentation	1340
7.113.2.1 AttributeLevel	1340
7.113.2.2 AttributeScores	1341
7.113.2.3 AttributeScoresClt	1341
7.113.2.4 AttributeScoresIlt	1341
7.113.2.5 ClassLevel	1341
7.113.2.6 ConfigMap	1341
7.113.2.7 DistancePair	1342
7.113.2.8 DistancePairs	1342
7.113.2.9 DistancePairsIlt	1342
7.113.2.10 matrixElement	1342

7.113.2.11NumericLevel	1342
7.113.2.12ScoreVarPair	1343
7.113.3 Enumeration Type Documentation	1343
7.113.3.1 AnalysisType	1343
7.113.3.2 AttributeMutationType	1343
7.113.3.3 AttributeType	1344
7.113.3.4 ClassType	1344
7.113.3.5 CvSetType	1344
7.113.3.6 EcAlgorithmType	1345
7.113.3.7 EcLtAlgorithmType	1345
7.113.3.8 EcMeAlgorithmType	1345
7.113.3.9 OutputDatasetType	1346
7.113.3.10RandomJungleRunMode	1346
7.113.3.11RandomJungleTreeType	1347
7.113.3.12ValueType	1347
7.113.4 Function Documentation	1347
7.113.4.1 ChooseSnpDatasetByType()	1348
7.113.4.2 DetectClassType()	1348
7.113.4.3 DetermineRandomJungleTreeType()	1349
7.113.4.4 GetConfigValue()	1349
7.113.4.5 GetFileBasename()	1349
7.113.4.6 GetFileExtension()	1350
7.113.4.7 GetMatchingIds()	1350
7.113.4.8 LoadNumericIds()	1351
7.113.4.9 LoadPhenoids()	1351
7.113.4.10PrintVector()	1352
7.113.4.11ProtectedLog()	1352
7.113.4.12scoresSortAsc()	1352

7.113.4.13scoresSortAscByName()	1353
7.113.4.14scoresSortDesc()	1353
7.113.4.15Timestamp()	1353
7.113.5 Variable Documentation	1353
7.113.5.1 COMMAND_LINE_ERROR	1353
7.113.5.2 DATASET_LOAD_ERROR	1354
7.113.5.3 datasetTypeToExt	1354
7.113.5.4 DEFAULT_FDR	1354
7.113.5.5 DEFAULT_PVALUE	1354
7.113.5.6 DEFAULT_PVALUE_THRESHOLD	1354
7.113.5.7 DEFAULT_ZVALUE	1355
7.113.5.8 INVALID_ATTRIBUTE_VALUE	1355
7.113.5.9 INVALID_DISCRETE_CLASS_VALUE	1355
7.113.5.10INVALID_DISTANCE	1355
7.113.5.11INVALID_INDEX	1355
7.113.5.12INVALID_INT_VALUE	1356
7.113.5.13INVALID_NUMERIC_CLASS_VALUE	1356
7.113.5.14INVALID_NUMERIC_VALUE	1356
7.113.5.15MIN_NUM_GENES	1356
7.113.5.16MIN_NUM_SUBJ_PER_GROUP	1356
7.113.5.17MISSING_ATTRIBUTE_VALUE	1357
7.113.5.18MISSING_DISCRETE_CLASS_VALUE	1357
7.113.5.19MISSING_NUMERIC_CLASS_VALUE	1357
7.113.5.20MISSING_NUMERIC_VALUE	1357
7.113.5.21PP	1357
7.114src/InteractionNetwork.cpp File Reference	1358
7.115src/InteractionNetwork.h File Reference	1358
7.115.1 Typedef Documentation	1359

7.115.1.1 HomophilyResult	1360
7.115.1.2 Indices	1360
7.115.1.3 ModularityResult	1360
7.115.1.4 ModuleIndices	1360
7.115.1.5 ModuleList	1360
7.115.1.6 RipmResult	1360
7.115.2 Enumeration Type Documentation	1360
7.115.2.1 MatrixFileType	1360
7.115.2.2 NetworkMatrixType	1361
7.115.3 Variable Documentation	1361
7.115.3.1 DEFAULT_CONNECTIVITY_THRESHOLD	1361
7.115.3.2 MODULARITY_THRESHOLD	1361
7.116src/ipmpar.h File Reference	1362
7.116.1 Function Documentation	1362
7.116.1.1 ipmpar()	1362
7.117src/lapackf.cpp File Reference	1362
7.117.1 Function Documentation	1363
7.117.1.1 eigen_lapack()	1363
7.117.1.2 svd_lapack()	1363
7.118src/lapackf.h File Reference	1364
7.118.1 Function Documentation	1364
7.118.1.1 eigen_lapack()	1364
7.118.1.2 svd_lapack()	1364
7.119src/legacy.cpp File Reference	1364
7.120src/linear.cpp File Reference	1365
7.120.1 Function Documentation	1366
7.120.1.1 covsrt()	1366
7.120.1.2 gaussj()	1366

7.120.1.3 Ifit()	1366
7.121src/linear.h File Reference	1367
7.122src/linput.cpp File Reference	1367
7.122.1 Variable Documentation	1368
7.122.1.1 LOG	1368
7.123src/locus.cpp File Reference	1368
7.124src/logistic.cpp File Reference	1369
7.125src/logistic.h File Reference	1369
7.126src/lookup.cpp File Reference	1370
7.126.1 Macro Definition Documentation	1370
7.126.1.1 IP_ADDR	1371
7.126.1.2 PORT_NUM	1371
7.127src/lookup2.cpp File Reference	1371
7.127.1 Macro Definition Documentation	1372
7.127.1.1 IP_ADDR	1372
7.127.1.2 PORT_NUM	1372
7.127.2 Function Documentation	1372
7.127.2.1 convertPosition()	1372
7.128src/mds.cpp File Reference	1373
7.129src/merge.cpp File Reference	1373
7.130src/metaanal.cpp File Reference	1374
7.130.1 Typedef Documentation	1375
7.130.1.1 mymap_t	1375
7.130.2 Variable Documentation	1375
7.130.2.1 PP	1375
7.131src/metaem.cpp File Reference	1375
7.131.1 Variable Documentation	1376
7.131.1.1 LOG	1376

7.132src/mh.cpp File Reference	1376
7.133src/mishap.cpp File Reference	1377
7.133.1 Variable Documentation	1377
7.133.1.1 PP	1378
7.134src/model.cpp File Reference	1378
7.135src/model.h File Reference	1378
7.136src/multi.cpp File Reference	1379
7.136.1 Macro Definition Documentation	1379
7.136.1.1 EPS	1379
7.137src/multiple.cpp File Reference	1380
7.137.1 Function Documentation	1380
7.137.1.1 pprint()	1380
7.138src/nlist.cpp File Reference	1381
7.139src/nlist.h File Reference	1381
7.140src/nonfounderphasing.cpp File Reference	1382
7.140.1 Function Documentation	1383
7.140.1.1 displayFamTran()	1383
7.140.2 Variable Documentation	1383
7.140.2.1 PP	1383
7.141src/options.cpp File Reference	1383
7.142src/options.h File Reference	1384
7.142.1 Function Documentation	1384
7.142.1.1 getOutputFilename()	1384
7.142.1.2 setOptions()	1385
7.143src/output.cpp File Reference	1385
7.143.1 Function Documentation	1386
7.143.1.1 make2LTable()	1386
7.143.2 Variable Documentation	1386

7.143.2.1 LOG	1386
7.144src/parse.cpp File Reference	1386
7.144.1 Function Documentation	1387
7.144.1.1 getOutputFilename()	1387
7.144.1.2 setOptions()	1387
7.145src/pdriver.cpp File Reference	1387
7.146src/perm.cpp File Reference	1388
7.147src/perm.h File Reference	1388
7.148src/phase.cpp File Reference	1389
7.148.1 Variable Documentation	1389
7.148.1.1 LOG	1389
7.149src/phase.h File Reference	1390
7.150src/plink.cpp File Reference	1390
7.151src/plink.h File Reference	1390
7.151.1 Typedef Documentation	1392
7.151.1.1 boolmatrix_t	1392
7.151.1.2 boolevc_t	1392
7.151.1.3 floatvec_t	1392
7.151.1.4 fmatrix_t	1393
7.151.1.5 iAllele	1393
7.151.1.6 iIndividual	1393
7.151.1.7 iLocus	1393
7.151.1.8 interaction_t	1393
7.151.1.9 interactionVar_t	1393
7.151.1.10ntvec_t	1394
7.151.1.11iSNP	1394
7.151.1.12matrix_t	1394
7.151.1.13rankedlist_t	1394

7.151.1.14table_t	1394
7.151.1.15vector_t	1394
7.151.2 Enumeration Type Documentation	1394
7.151.2.1 ModelTermType	1394
7.151.2.2 RegressionRankType	1395
7.152src/PlinkInternalsDataset.cpp File Reference	1395
7.153src/PlinkInternalsDataset.h File Reference	1396
7.154src/PlinkInternalsDatasetInstance.cpp File Reference	1396
7.155src/PlinkInternalsDatasetInstance.h File Reference	1397
7.156src/poo.cpp File Reference	1398
7.157src/prephap.cpp File Reference	1399
7.157.1 Variable Documentation	1399
7.157.1.1 LOG	1399
7.158src/profile.cpp File Reference	1400
7.158.1 Function Documentation	1400
7.158.1.1 scoreRanges()	1400
7.158.2 Variable Documentation	1401
7.158.2.1 LOG	1401
7.158.2.2 PP	1401
7.159src/proxy.cpp File Reference	1401
7.159.1 Function Documentation	1402
7.159.1.1 combinations_recursive()	1402
7.160src/qfam.cpp File Reference	1402
7.160.1 Function Documentation	1403
7.160.1.1 scoreBandW()	1403
7.160.1.2 scoreBetween()	1403
7.160.1.3 setCovariatesForSNP()	1403
7.161src/qualscores.cpp File Reference	1404

7.162src/r.cpp File Reference	1404
7.163src/RandomForest.cpp File Reference	1405
7.164src/RandomForest.h File Reference	1405
7.165src/Rconnection.cpp File Reference	1406
7.165.1 Macro Definition Documentation	1407
7.165.1.1 AF_LOCAL	1407
7.165.2 Function Documentation	1407
7.165.2.1 new_parsed_Rexp()	1408
7.165.2.2 new_parsed_Rexp_from_Msg()	1408
7.165.3 Variable Documentation	1408
7.165.3.1 myID	1408
7.166src/Rconnection.h File Reference	1408
7.166.1 Macro Definition Documentation	1410
7.166.1.1 A_crypt	1410
7.166.1.2 A_plain	1410
7.166.1.3 A_required	1410
7.166.1.4 CERR_auth_unsupported	1410
7.166.1.5 CERR_connect_failed	1411
7.166.1.6 CERR_handshake_failed	1411
7.166.1.7 CERR_invalid_id	1411
7.166.1.8 CERR_io_error	1411
7.166.1.9 CERR_malformed_packet	1411
7.166.1.10CERR_not_connected	1411
7.166.1.11CERR_not_supported	1412
7.166.1.12CERR_out_of_mem	1412
7.166.1.13CERR_peer_closed	1412
7.166.1.14CERR_protocol_not_supp	1412
7.166.1.15CERR_send_error	1412

7.166.2 Typedef Documentation	1412
7.166.2.1 Rsize_t	1413
7.167src/regain.cpp File Reference	1413
7.167.1 Function Documentation	1413
7.167.1.1 pvalComparator()	1414
7.167.2 Variable Documentation	1414
7.167.2.1 PP	1414
7.168src/regain.h File Reference	1414
7.168.1 Enumeration Type Documentation	1415
7.168.1.1 RegainOutputFormat	1415
7.168.1.2 RegainOutputTransform	1416
7.169src/ReliefF.cpp File Reference	1416
7.169.1 Typedef Documentation	1417
7.169.1.1 AttributeIndex	1417
7.169.1.2 AttributeIndexlt	1417
7.169.1.3 ScoresMap	1418
7.169.1.4 ScoresMaplt	1418
7.169.1.5 T	1418
7.169.2 Function Documentation	1418
7.169.2.1 attributeSort()	1418
7.169.2.2 scoreSort()	1419
7.170src/ReliefF.h File Reference	1419
7.171src/ReliefFSeq.cpp File Reference	1420
7.172src/ReliefFSeq.h File Reference	1420
7.173src/ReliefSeqController.cpp File Reference	1421
7.174src/ReliefSeqController.h File Reference	1422
7.175src/RReliefF.cpp File Reference	1423
7.176src/RReliefF.h File Reference	1423

7.177src/Rsrv.h File Reference	1424
7.177.1 Macro Definition Documentation	1427
7.177.1.1 BOOL_FALSE	1427
7.177.1.2 BOOL_NA	1427
7.177.1.3 BOOL_TRUE	1428
7.177.1.4 CMD_assignSEXP	1428
7.177.1.5 CMD_attachSession	1428
7.177.1.6 CMD_closeFile	1428
7.177.1.7 CMD_createFile	1428
7.177.1.8 CMD_detachedVoidEval	1429
7.177.1.9 CMD_detachSession	1429
7.177.1.10CMD_eval	1429
7.177.1.11CMD_login	1429
7.177.1.12CMD_openFile	1429
7.177.1.13CMD_readFile	1430
7.177.1.14CMD_removeFile	1430
7.177.1.15CMD_RESP	1430
7.177.1.16CMD_serAssign	1430
7.177.1.17CMD_serEEval	1430
7.177.1.18CMD_serEval	1431
7.177.1.19CMD_setBufferSize	1431
7.177.1.20CMD_setEncoding	1431
7.177.1.21CMD_setSEXP	1431
7.177.1.22CMD_shutdown	1431
7.177.1.23CMD_SPECIAL_MASK	1432
7.177.1.24CMD_STAT	1432
7.177.1.25CMD_voidEval	1432
7.177.1.26CMD_writeFile	1432

7.177.1.27	default_Rsrv_port	1432
7.177.1.28	DT_ARRAY	1433
7.177.1.29	DT_BYTESTREAM	1433
7.177.1.30	DT_CHAR	1433
7.177.1.31	DT_DOUBLE	1433
7.177.1.32	DT_INT	1433
7.177.1.33	DT_LARGE	1434
7.177.1.34	DT_SEXP	1434
7.177.1.35	DT_STRING	1434
7.177.1.36	dtop	1434
7.177.1.37	ERR_accessDenied	1434
7.177.1.38	ERR_auth_failed	1435
7.177.1.39	ERR_conn_broken	1435
7.177.1.40	ERR_data_overflow	1435
7.177.1.41	ERR_detach_failed	1435
7.177.1.42	ERR_inv_cmd	1436
7.177.1.43	ERR_inv_par	1436
7.177.1.44	ERR_IOerror	1436
7.177.1.45	ERR_notOpen	1436
7.177.1.46	ERR_object_too_big	1436
7.177.1.47	ERR_out_of_mem	1437
7.177.1.48	ERR_Rerror	1437
7.177.1.49	ERR_session_busy	1437
7.177.1.50	ERR_unknownCmd	1437
7.177.1.51	ERR_unsupportedCmd	1438
7.177.1.52	fixdcpy	1438
7.177.1.53	GET_DT	1438
7.177.1.54	GET_XT	1438

7.177.1.55HAS_ATTR	1438
7.177.1.56S_LARGE	1439
7.177.1.57top	1439
7.177.1.58PAR_LEN	1439
7.177.1.59PAR_LENGTH	1439
7.177.1.60PAR_TYPE	1439
7.177.1.61ptod	1440
7.177.1.62ptoi	1440
7.177.1.63RESP_ERR	1440
7.177.1.64RESP_OK	1440
7.177.1.65RSRV_VER	1441
7.177.1.66SET_PAR	1441
7.177.1.67SET_STAT	1441
7.177.1.68XT_ARRAY_BOOL	1441
7.177.1.69XT_ARRAY_BOOL_UA	1441
7.177.1.70XT_ARRAY_CPLX	1442
7.177.1.71XT_ARRAY_DOUBLE	1442
7.177.1.72XT_ARRAY_INT	1442
7.177.1.73XT_ARRAY_STR	1442
7.177.1.74XT_BOOL	1442
7.177.1.75XT_CLOS	1443
7.177.1.76XT_DOUBLE	1443
7.177.1.77XT_HAS_ATTR	1443
7.177.1.78XT_INT	1443
7.177.1.79XT_LANG	1443
7.177.1.80XT_LANG_NOTAG	1444
7.177.1.81XT_LANG_TAG	1444
7.177.1.82XT_LARGE	1444

7.177.1.83XT_LIST	1444
7.177.1.84XT_LIST_NOTAG	1444
7.177.1.85XT_LIST_TAG	1445
7.177.1.86XT_NULL	1445
7.177.1.87XT_RAW	1445
7.177.1.88XT_S4	1445
7.177.1.89XT_STR	1445
7.177.1.90XT_SYM	1445
7.177.1.91XT_SYMNAME	1446
7.177.1.92XT_UNKNOWN	1446
7.177.1.93XT_VECTOR	1446
7.177.1.94XT_VECTOR_EXP	1446
7.177.1.95XT_VECTOR_STR	1446
7.177.2 Function Documentation	1446
7.177.2.1 isByteSexOk()	1447
7.178src/segment.cpp File Reference	1447
7.179src/sets.cpp File Reference	1447
7.179.1 Variable Documentation	1448
7.179.1.1 PP	1448
7.180src/sets.h File Reference	1448
7.181src/setscreen.cpp File Reference	1449
7.181.1 Variable Documentation	1449
7.181.1.1 PP	1449
7.182src/sharing.cpp File Reference	1450
7.183src/simul.cpp File Reference	1450
7.183.1 Function Documentation	1451
7.183.1.1 instanceSNP()	1451
7.183.1.2 instanceSNP_QT()	1451

7.184src/sisocks.h File Reference	1451
7.184.1 Macro Definition Documentation	1453
7.184.1.1 <code>donesocks</code>	1453
7.184.1.2 <code>EACCES</code>	1453
7.184.1.3 <code>EADDRINUSE</code>	1453
7.184.1.4 <code>EAFNOSUPPORT</code>	1453
7.184.1.5 <code>EALREADY</code>	1453
7.184.1.6 <code>EBADF</code>	1454
7.184.1.7 <code>ECONNREFUSED</code>	1454
7.184.1.8 <code>EFAULT</code>	1454
7.184.1.9 <code>EINPROGRESS</code>	1454
7.184.1.10 <code>EINVAL</code>	1454
7.184.1.11 <code>EISCONN</code>	1454
7.184.1.12 <code>ENETUNREACH</code>	1455
7.184.1.13 <code>ENOTSOCK</code>	1455
7.184.1.14 <code>EOPNOTSUPP</code>	1455
7.184.1.15 <code>ETIMEDOUT</code>	1455
7.184.1.16 <code>EWOULDBLOCK</code>	1455
7.184.1.17 <code>net_aton</code>	1455
7.184.1.18 <code>SA</code>	1456
7.184.1.19 <code>SAIN</code>	1456
7.184.1.20 <code>sockerrno</code>	1456
7.184.1.21 <code>windows</code>	1456
7.184.2 Function Documentation	1456
7.184.2.1 <code>build_sin()</code>	1456
7.184.2.2 <code>initsocks()</code>	1457
7.185src/SNReliefF.cpp File Reference	1457
7.186src/SNReliefF.h File Reference	1457

7.186.1 Typedef Documentation	1459
7.186.1.1 InstanceAttributeStats	1459
7.186.1.2 InstanceAttributeStatsIt	1459
7.186.1.3 InstanceHitMissStats	1459
7.186.1.4 NeighborStats	1459
7.186.1.5 NeighborStatsClt	1459
7.186.1.6 NeighborStatsIt	1460
7.187src/sockets.cpp File Reference	1460
7.187.1 Function Documentation	1460
7.187.1.1 socketConnection()	1461
7.187.2 Variable Documentation	1461
7.187.2.1 PVERSION	1461
7.188src/sockets.h File Reference	1461
7.188.1 Function Documentation	1462
7.188.1.1 socketConnection()	1462
7.189src/Statistics.cpp File Reference	1462
7.189.1 Macro Definition Documentation	1464
7.189.1.1 DEBUG_E	1464
7.189.1.2 DEBUG_Z	1464
7.189.2 Function Documentation	1464
7.189.2.1 condentropy()	1464
7.189.2.2 ConditionalEntropy()	1464
7.189.2.3 ConstructAttributeCart()	1465
7.189.2.4 Entropy()	1465
7.189.2.5 hypot()	1466
7.189.2.6 KendallTau() [1/3]	1466
7.189.2.7 KendallTau() [2/3]	1467
7.189.2.8 KendallTau() [3/3]	1467

7.189.2.9 PrintHistogram()	1468
7.189.2.10 SelfEntropy()	1468
7.189.2.11 ZTransform()	1468
7.190 src/Statistics.h File Reference	1469
7.190.1 Detailed Description	1470
7.190.2 Typedef Documentation	1471
7.190.2.1 Histogram	1471
7.190.2.2 HistogramIt	1471
7.190.2.3 VectorDouble	1471
7.190.2.4 VectorDoubleIt	1471
7.190.3 Function Documentation	1471
7.190.3.1 condentropy()	1472
7.190.3.2 ConditionalEntropy()	1472
7.190.3.3 ConstructAttributeCart()	1472
7.190.3.4 Entropy()	1473
7.190.3.5 hypot()	1473
7.190.3.6 KendallTau() [1/3]	1474
7.190.3.7 KendallTau() [2/3]	1474
7.190.3.8 KendallTau() [3/3]	1475
7.190.3.9 PrintHistogram()	1475
7.190.3.10 SelfEntropy()	1475
7.190.3.11 VarStd()	1476
7.190.3.12 ZTransform()	1476
7.191 src/stats.cpp File Reference	1477
7.191.1 Macro Definition Documentation	1479
7.191.1.1 FPMIN	1479
7.191.1.2 HIGH	1480
7.191.1.3 LOW	1480

7.191.2 Function Documentation	1480
7.191.2.1 betacf()	1480
7.191.2.2 betai()	1480
7.191.2.3 chi2x2() [1/3]	1480
7.191.2.4 chi2x2() [2/3]	1481
7.191.2.5 chi2x2() [3/3]	1481
7.191.2.6 chiprobP()	1481
7.191.2.7 chiTable()	1481
7.191.2.8 eigenvalues()	1481
7.191.2.9 eigenvectors()	1482
7.191.2.10EV_tqli()	1482
7.191.2.11EV_tred2()	1482
7.191.2.12factorial()	1482
7.191.2.13fitModel()	1482
7.191.2.14gammln()	1483
7.191.2.15getNumericCaseControl()	1483
7.191.2.16inverse()	1483
7.191.2.17inverse_chiprob()	1483
7.191.2.18tqnorm()	1483
7.191.2.19ubksb()	1484
7.191.2.20udcmp()	1484
7.191.2.21matrixAdd()	1484
7.191.2.22matrixComputeCovariance()	1484
7.191.2.23matrixComputeNodeDegrees()	1485
7.191.2.24matrixConnectivityThreshold()	1485
7.191.2.25matrixDivideScalar()	1485
7.191.2.26matrixElementWiseMultiply()	1485
7.191.2.27matrixExtractRowColIdx()	1486

7.191.2.28matrixFill()	1486
7.191.2.29matrixGetDiag()	1486
7.191.2.30matrixGetTrace()	1486
7.191.2.31matrixMultiplyScalar()	1486
7.191.2.32matrixRead()	1487
7.191.2.33matrixSetDiag()	1487
7.191.2.34matrixSubtract()	1487
7.191.2.35matrixSums()	1487
7.191.2.36matrixTranspose()	1488
7.191.2.37matrixWrite()	1488
7.191.2.38msqrt()	1488
7.191.2.39multMatrix()	1488
7.191.2.40normdist()	1488
7.191.2.41numericLowValueFilter()	1489
7.191.2.42numericMeanCenter()	1489
7.191.2.43numericStandardize()	1489
7.191.2.44numericVarianceFilter()	1489
7.191.2.45pca()	1489
7.191.2.46pF()	1490
7.191.2.47polint()	1490
7.191.2.48pT()	1490
7.191.2.49pythag()	1490
7.191.2.50qromb()	1491
7.191.2.51quantile()	1491
7.191.2.52rankByRegression()	1491
7.191.2.53realnum()	1491
7.191.2.54reportNumericSummaryStats()	1491
7.191.2.55norm()	1492

7.191.2.56SQR()	1492
7.191.2.57svbksb()	1492
7.191.2.58svd()	1492
7.191.2.59svd_inverse()	1492
7.191.2.60svdcmp()	1493
7.191.2.61svdvar()	1493
7.191.2.62symTable()	1493
7.191.2.63qli()	1493
7.191.2.64trapzd()	1494
7.191.2.65red2()	1494
7.191.2.66Test()	1494
7.191.2.67vec2diag()	1494
7.191.2.68vectorSummary()	1494
7.191.2.69zTest()	1495
7.191.3 Variable Documentation	1495
7.191.3.1 a	1495
7.191.3.2 b	1495
7.191.3.3 c	1496
7.191.3.4 d	1496
7.191.3.5 LOG	1496
7.191.3.6 PP	1496
7.192src/stats.h File Reference	1497
7.192.1 Function Documentation	1499
7.192.1.1 betacf()	1499
7.192.1.2 betai()	1499
7.192.1.3 chi2x2() [1/3]	1499
7.192.1.4 chi2x2() [2/3]	1500
7.192.1.5 chi2x2() [3/3]	1500

7.192.1.6 chiprobP()	1500
7.192.1.7 chiTable()	1500
7.192.1.8 eigenvalues()	1500
7.192.1.9 eigenvectors()	1501
7.192.1.10EV_tqli()	1501
7.192.1.11EV_tred2()	1501
7.192.1.12factorial()	1501
7.192.1.13fitModel()	1501
7.192.1.14gammln()	1502
7.192.1.15gammp()	1502
7.192.1.16gcf()	1502
7.192.1.17getNumericCaseControl()	1502
7.192.1.18gser()	1502
7.192.1.19inverse()	1503
7.192.1.20inverse_chiprob()	1503
7.192.1.21ltqnorm()	1503
7.192.1.22ubksb()	1503
7.192.1.23udcmp()	1503
7.192.1.24matrixAdd()	1504
7.192.1.25matrixComputeCovariance()	1504
7.192.1.26matrixComputeNodeDegrees()	1504
7.192.1.27matrixConnectivityThreshold()	1504
7.192.1.28matrixDivideScalar()	1505
7.192.1.29matrixElementWiseMultiply()	1505
7.192.1.30matrixExtractRowColIdx()	1505
7.192.1.31matrixFill()	1505
7.192.1.32matrixGetDiag()	1506
7.192.1.33matrixGetTrace()	1506

7.192.1.34matrixMultiplyScalar()	1506
7.192.1.35matrixRead()	1506
7.192.1.36matrixSetDiag()	1506
7.192.1.37matrixSubtract()	1507
7.192.1.38matrixSums()	1507
7.192.1.39matrixTranspose()	1507
7.192.1.40matrixWrite()	1507
7.192.1.41msqrt()	1507
7.192.1.42multMatrix()	1508
7.192.1.43normdist()	1508
7.192.1.44numericLowValueFilter()	1508
7.192.1.45numericMeanCenter()	1508
7.192.1.46numericStandardize()	1508
7.192.1.47numericVarianceFilter()	1509
7.192.1.48pca()	1509
7.192.1.49pF()	1509
7.192.1.50polint()	1509
7.192.1.51pT()	1510
7.192.1.52pythag()	1510
7.192.1.53qromb()	1510
7.192.1.54quantile()	1510
7.192.1.55rankByRegression()	1510
7.192.1.56realnum()	1511
7.192.1.57reportNumericSummaryStats()	1511
7.192.1.58norm()	1511
7.192.1.59sizeMatrix()	1511
7.192.1.60sizeTable()	1511
7.192.1.61svbksb()	1512

7.192.1.62svd()	1512
7.192.1.63svd_inverse()	1512
7.192.1.64svdcmp()	1512
7.192.1.65svdvar()	1513
7.192.1.66symTable()	1513
7.192.1.67qli()	1513
7.192.1.68rapzd()	1513
7.192.1.69red2()	1514
7.192.1.70Test()	1514
7.192.1.71vec2diag()	1514
7.192.1.72vectorSummary()	1514
7.192.1.73zTest()	1514
7.193src/step.cpp File Reference	1515
7.194src/StringUtils.h File Reference	1515
7.194.1 Detailed Description	1517
7.195src/tag.cpp File Reference	1518
7.196src/tdt.cpp File Reference	1518
7.197src/tinput.cpp File Reference	1519
7.197.1 Variable Documentation	1519
7.197.1.1 LOG	1520
7.198src/Tree.cpp File Reference	1520
7.199src/Tree.h File Reference	1520
7.200src/TreeClassification.cpp File Reference	1521
7.201src/TreeClassification.h File Reference	1521
7.202src/TreeProbability.cpp File Reference	1522
7.203src/TreeProbability.h File Reference	1523
7.204src/TreeRegression.cpp File Reference	1524
7.205src/TreeRegression.h File Reference	1525

7.206src/TreeSurvival.cpp File Reference	1526
7.207src/TreeSurvival.h File Reference	1526
7.208src/trio.cpp File Reference	1527
7.208.1 Function Documentation	1528
7.208.1.1 addParent()	1528
7.208.1.2 addPerson()	1528
7.208.1.3 gprint()	1529
7.208.2 Variable Documentation	1529
7.208.2.1 PP	1529
7.209src/utility.cpp File Reference	1529
7.209.1 Function Documentation	1531
7.209.1.1 adjustPvalues()	1531
7.209.1.2 beautifyTime()	1532
7.209.1.3 checkPositiveIntegers()	1532
7.209.1.4 checkUnorderedVariables()	1532
7.209.1.5 computeConcordanceIndex()	1533
7.209.1.6 drawWithoutReplacementFisherYates()	1533
7.209.1.7 drawWithoutReplacementKnuth()	1534
7.209.1.8 drawWithoutReplacementSimple()	1534
7.209.1.9 drawWithoutReplacementSkip()	1535
7.209.1.10 drawWithoutReplacementWeighted() [1/2]	1535
7.209.1.11 drawWithoutReplacementWeighted() [2/2]	1536
7.209.1.12 dstdnorm()	1536
7.209.1.13 equalSplit()	1537
7.209.1.14 loadDoubleVectorFromFile()	1537
7.209.1.15 logrankScores()	1538
7.209.1.16 logrankScoresData()	1538
7.209.1.17 maxstat()	1539

7.209.1.18maxstatInData()	1539
7.209.1.19maxstatPValueLau92()	1540
7.209.1.20maxstatPValueLau94()	1540
7.209.1.21maxstatPValueUnadjusted()	1541
7.209.1.22mostFrequentValue()	1541
7.209.1.23numSamplesLeftOfCutpoint()	1542
7.209.1.24numSamplesLeftOfCutpointInData()	1542
7.209.1.25orderInData()	1543
7.209.1.26pstdnorm()	1543
7.209.1.27roundToNextMultiple()	1544
7.209.1.28shuffleAndSplit()	1544
7.209.1.29splitString()	1545
7.209.1.30uintToString()	1545
7.210src/utility.h File Reference	1546
7.210.1 Function Documentation	1548
7.210.1.1 adjustPvalues()	1548
7.210.1.2 beautifyTime()	1549
7.210.1.3 checkPositiveIntegers()	1549
7.210.1.4 checkUnorderedVariables()	1550
7.210.1.5 computeConcordanceIndex()	1550
7.210.1.6 drawWithoutReplacementKnuth()	1551
7.210.1.7 drawWithoutReplacementSimple()	1551
7.210.1.8 drawWithoutReplacementSkip()	1552
7.210.1.9 drawWithoutReplacementWeighted() [1/2]	1552
7.210.1.10drawWithoutReplacementWeighted() [2/2]	1553
7.210.1.11dstdnorm()	1553
7.210.1.12equalSplit()	1553
7.210.1.13loadDoubleVectorFromFile()	1554

7.210.1.14logrankScores()	1554
7.210.1.15ogrankScoresData()	1555
7.210.1.16maxstat()	1555
7.210.1.17maxstatInData()	1557
7.210.1.18maxstatPValueLau92()	1558
7.210.1.19maxstatPValueLau94()	1558
7.210.1.20maxstatPValueUnadjusted()	1559
7.210.1.21mostFrequentClass()	1559
7.210.1.22mostFrequentValue()	1560
7.210.1.23numSamplesLeftOfCutpoint()	1560
7.210.1.24numSamplesLeftOfCutpointInData()	1561
7.210.1.25order()	1561
7.210.1.26orderInData()	1562
7.210.1.27pstndnorm()	1562
7.210.1.28rank()	1563
7.210.1.29readVector1D() [1/2]	1563
7.210.1.30readVector1D() [2/2]	1563
7.210.1.31readVector2D()	1564
7.210.1.32roundToNextMultiple()	1564
7.210.1.33saveVector1D() [1/2]	1564
7.210.1.34saveVector1D() [2/2]	1565
7.210.1.35saveVector2D()	1565
7.210.1.36shuffleAndSplit()	1565
7.210.1.37splitString()	1566
7.210.1.38uintToString()	1566
7.211src/version.h File Reference	1567
7.211.1 Macro Definition Documentation	1567
7.211.1.1 RANGER_VERSION	1567

7.212src/webcheck.cpp File Reference	1568
7.212.1 Macro Definition Documentation	1568
7.212.1.1 GET_STRING	1568
7.212.1.2 IP_ADDR	1569
7.212.1.3 PORT_NUM	1569
7.212.2 Variable Documentation	1569
7.212.2.1 PREL	1569
7.212.2.2 PVERSION	1569
7.213src/whap.cpp File Reference	1570
7.213.1 Function Documentation	1570
7.213.1.1 ci()	1570
7.213.1.2 displayHaploGroups()	1571
7.214src/whap.h File Reference	1571
7.215src/zed.cpp File Reference	1572
7.215.1 Function Documentation	1572
7.215.1.1 fileCompress()	1573
7.215.1.2 fileUncompress()	1573
7.215.2 Variable Documentation	1573
7.215.2.1 PP	1573
7.216src/zed.h File Reference	1573
7.216.1 Function Documentation	1574
7.216.1.1 fileCompress()	1574
7.216.1.2 fileUncompress()	1574
7.216.2 Variable Documentation	1574
7.216.2.1 MAX_LINE_LENGTH	1574
7.217src/zfstream.cpp File Reference	1575
7.217.1 Macro Definition Documentation	1575
7.217.1.1 BIGBUFSIZE	1575
7.217.1.2 SMALLBUFSIZE	1575
7.218src/zfstream.h File Reference	1576
7.218.1 Function Documentation	1577
7.218.1.1 operator<<()	1577
7.218.1.2 setcompression() [1/2]	1577
7.218.1.3 setcompression() [2/2]	1577
Index	1579

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

insilico	17
--------------------	----

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

_RipmResult	27
Alleles	29
ArgumentHandler	33
AttributeRanker	42
ChiSquared	82
Deseq	233
Edger	256
RandomForest	860
ReliefF	904
ReliefFSeq	919
RReliefF	961
SNRReliefF	1011
binary_function	
deref_less	230
deref_less_bcw	231
BirdseedData	48
CArgs	61
CentralityRanker	67
Chap	73
ChapModel	76
CHIP_SEQ_INFO	80
ClInfo	90
clump_LD	92
ClumpPair	97
ClumpResults	99
Cluster	101
CNVIndivReport	102
CRandom	106
CSNP	111
Data	112
DataChar	120

DataDouble	125
DataFloat	131
Dataset	135
PlinkInternalsDataset	842
DatasetInstance	198
PlinkInternalsDatasetInstance	845
DcVar	215
DgeData	238
double2	252
DPrime	254
Eigen	261
EpistasisEQtl	263
EvaporativeCooling	273
EvaporativeCoolingPrivacy	287
ExpressionDataSimulator	301
Family	306
FamilyTransmissions	310
Forest	312
ForestClassification	333
ForestProbability	339
ForestRegression	345
ForestSurvival	351
fullGenotype	359
GSLRandomBase	361
GSLRandomFlat	364
GVariant	367
gzomanip2< T1, T2 >	392
HaploPhase	396
HaploWindow	425
HWindow	435
IDField	439
IDFile	443
IDGroup	446
IDHelper	448
IDValue	454
Individual	457
indivPair	466
int2	468
InteractionNetwork	470
std::ios_base	
std::basic_ios	
std::basic_istream	
std::istream	
gzifstream	380
std::basic_ostream	
std::ostream	
gzofstream	386
LDPair	490
std::less< Locus * >	492
std::less< MultiLocusGenotype * >	493
std::less< Pool * >	494
std::less< Segment * >	495
Locus	506

Variant	1073
MainExitException	517
Model	518
LinearModel	496
LogisticModel	511
MultiLocusGenotype	538
Neighbour	540
NList	542
Options	547
OptionSet	549
Pair	552
Pair2	554
Pair_cmp	556
PairwiseLinkage	557
par	561
Perm	761
phdr	771
Plink	773
Pool	851
probabilisticGenotype	854
ProxyResult	857
Range	866
Rconnection	870
RCount	877
Regain	888
RegressionRankResults	902
ReliefSeqController	925
ResultTrio	935
Rexp	937
Rdouble	883
Rinteger	945
Rlist	950
Rstring	965
Rstrings	969
Rsymbol	975
Rvector	979
Rmessage	956
Segment	984
SegmentSizeCmp	989
Set	990
SetSortedSNP	997
SimParameters	999
SimParametersQT	1003
SInfo	1007
SNP_INFO	1008
SortedResult	1015
sortedSegments	1017
streambuf	
gzfilebuf	369
Transcript	1019
Tree	1021
TreeClassification	1036
TreeProbability	1044

TreeRegression	1052
TreeSurvival	1061
unary_function	
insilico::do_to_lower< charT >	246
insilico::do_to_upper< charT >	249
insilico::is_classified< Type, charT >	487
Var	1071
WMLocus	1076
Z	1078
ZInput	1080
ZOutput	1085
ZZ	1090

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

_RipmResult	27
Alleles	29
ArgumentHandler	33
AttributeRanker	
Abstract base class for attribute rankers	42
BirdseedData	
Read Broad's Birdsuite Birdseed-called SNP data	48
CArgs	61
CentralityRanker	67
Chap	73
ChapModel	76
CHIP_SEQ_INFO	80
ChiSquared	
Chi-squared attribute ranking algorithm	82
CInfo	90
clump_LD	92
ClumpPair	97
ClumpResults	99
Cluster	101
CNVIndivReport	102
CRandom	106
CSNP	111
Data	112
DataChar	120
DataDouble	125
DataFloat	131
Dataset	
Base class for collections of instances containing attributes and class	135
DatasetInstance	
Class to hold dataset instances (rows of attributes)	198
DcVar	215

deref_less	230
deref_less_bcw	231
Dseq	
DESeq attribute ranking algorithm	233
DgeData	
Digital gene expression data	238
insilico::do_to_lower< charT >	246
insilico::do_to_upper< charT >	249
double2	252
DPrime	254
Edger	
Edger attribute ranking algorithm	256
Eigen	261
EpistasisEQtl	263
EvaporativeCooling	
Evaporative Cooling attribute ranking algorithm	273
EvaporativeCoolingPrivacy	287
ExpressionDataSimulator	301
Family	306
FamilyTransmissions	310
Forest	312
ForestClassification	333
ForestProbability	339
ForestRegression	345
ForestSurvival	351
fullGenotype	359
GSLRandomBase	
A base class for GNU Scientific Library (GSL) random number functions	361
GSLRandomFlat	
Random numbers in a flat, or uniform distribution	364
GVariant	367
gzfilebuf	
Gzipped file stream buffer class	369
gzifstream	
Gzipped file input stream class	380
gzofstream	
Gzipped file output stream class	386
gzomanip2< T1, T2 >	
Gzipped file output stream manipulator class	392
HaploPhase	396
HaploWindow	425
HWindow	435
IDField	439
IDFile	443
IDGroup	446
IDHelper	448
IDValue	454
Individual	457
indivPair	466
int2	468
InteractionNetwork	470
insilico::is_classified< Type, charT >	487
LDPair	490
std::less< Locus * >	492

std::less< MultiLocusGenotype * >	493
std::less< Pool * >	494
std::less< Segment * >	495
LinearModel	496
Locus	506
LogisticModel	511
MainExitException	517
Model	518
MultiLocusGenotype	538
Neighbour	540
NList	542
Options	547
OptionSet	549
Pair	552
Pair2	554
Pair_cmp	556
PairwiseLinkage	557
par	561
Perm	761
phdr	771
Plink	773
PlinkInternalsDataset	
Plink internals adapter class	842
PlinkInternalsDatasetInstance	
Class to hold dataset instances (rows of attributes)	845
Pool	851
probabilisticGenotype	854
ProxyResult	857
RandomForest	
RandomForest attribute ranking algorithm	860
Range	866
Rconnection	870
RCount	877
Rdouble	883
Regain	888
RegressionRankResults	902
ReliefF	
ReliefF attribute ranking algorithm	904
ReliefFSeq	
ReliefFSeq attribute ranking algorithm	919
ReliefSeqController	
Interface for running ReliefSeq algorithms	925
ResultTrio	935
Rexp	937
Rinteger	945
Rlist	950
Rmessage	956
RReliefF	
Regression ReliefF attribute ranking algorithm	961
Rstring	965
Rstrings	969
Rsymbol	975
Rvector	979
Segment	984

SegmentSizeCmp	989
Set	990
SetSortedSNP	997
SimParameters	999
SimParametersQT	1003
SInfo	1007
SNP_INFO	1008
SNReliefF Signal-to-Noise ReliefF attribute ranking algorithm	1011
SortedResult	1015
sortedSegments	1017
Transcript	1019
Tree	1021
TreeClassification	1036
TreeProbability	1044
TreeRegression	1052
TreeSurvival	1061
Var	1071
Variant	1073
WMLocus	1076
Z	1078
ZInput	1080
ZOutput	1085
ZZ	1090

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

src/annot.cpp	1093
src/ArgumentHandler.cpp	1094
src/ArgumentHandler.h	1095
src/ArmadilloFuncs.cpp	1095
src/ArmadilloFuncs.h	1098
src/assoc.cpp	1101
src/AttributeRanker.cpp	1102
src/AttributeRanker.h	1102
src/BestN.h	
Find the best n keeping original order for ties - stable sort	1103
src/binput.cpp	1104
src/BirdseedData.cpp	1105
src/BirdseedData.h	1105
src/blox.cpp	1106
src/bmerge.cpp	1108
src/cdflib.h	1108
src/CentralityRanker.cpp	1130
src/CentralityRanker.h	1130
src/cfamily.cpp	1132
src/cfamily.h	1134
src/ChiSquared.cpp	1135
src/ChiSquared.h	1135
src/clumpld.cpp	1136
src/clumpld.h	1137
src/cluster.cpp	1138
src/cnv.cpp	1141
src/cnv.h	1144
src/cnvqt.cpp	1146
src/config.h	1147
src/crandom.cpp	1148
src/crandom.h	1148

src/Data.cpp	1149
src/Data.h	1149
src/DataChar.cpp	1150
src/DataChar.h	1150
src/DataDouble.cpp	1151
src/DataDouble.h	1152
src/DataFloat.cpp	1153
src/DataFloat.h	1153
src/Dataset.cpp	1154
src/Dataset.h	1156
src/DatasetInstance.cpp	1156
src/DatasetInstance.h	1157
src/dcdflib.cpp	1158
src/dcdflib.h	1191
src/DcVar.cpp	1195
src/DcVar.h	1196
src/Dseq.cpp	1200
src/Dseq.h	1200
src/dfam.cpp	1201
src/DgeData.cpp	1202
src/DgeData.h	1202
src/DistanceMetrics.cpp	1203
src/DistanceMetrics.h	
Distance metrics for ReliefF	1209
src/dosage.cpp	1216
src/Edger.cpp	1217
src/Edger.h	1218
src/elf.cpp	1218
src/em.cpp	1220
src/epi.cpp	1221
src/EpistasisEQtl.cpp	1222
src/EpistasisEQtl.h	1222
src/EvaporativeCooling.cpp	1225
src/EvaporativeCooling.h	1225
src/EvaporativeCoolingPrivacy.cpp	1226
src/EvaporativeCoolingPrivacy.h	1227
src/ExpressionDataSimulator.h	1229
src/filters.cpp	1230
src/fisher.cpp	1231
src/fisher.h	1239
src/flip.cpp	1245
src/Forest.cpp	1245
src/Forest.h	1246
src/ForestClassification.cpp	1247
src/ForestClassification.h	1247
src/ForestProbability.cpp	1248
src/ForestProbability.h	1248
src/ForestRegression.cpp	1249
src/ForestRegression.h	1250
src/ForestSurvival.cpp	1251
src/ForestSurvival.h	1251
src/genedrop.cpp	1252
src/genepi.cpp	1253
src/genoerr.cpp	1257

src/genogroup.cpp	1258
src/genogroup.h	1258
src/genome.cpp	1259
src/glm.cpp	1259
src/globals.h	1260
src/greport.cpp	1267
src/GSLRandomBase.h	1268
src/GSLRandomFlat.h	1268
src/gvar.cpp	1269
src/gvar.h	1271
src/gxe.cpp	1271
src/hapglm.cpp	1272
src/haploCC.cpp	1273
src/haplohelper.cpp	1273
src/haploQTL.cpp	1274
src/haploTDT.cpp	1275
src/haplowindow.cpp	1275
src/haplowindow.h	1276
src/helper.cpp	1277
src/helper.h	1298
src/homozyg.cpp	1319
src/hotel.cpp	1323
src/idhelp.cpp	1324
src/idhelp.h	1325
src/impute.cpp	1326
src/inbix.cpp	1327
src/informative.cpp	1329
src/input.cpp	1330
src/Insilico.cpp	1331
src/Insilico.h	
Common functions for In silico Lab projects	1337
src/InteractionNetwork.cpp	1358
src/InteractionNetwork.h	1358
src/ipmpar.h	1362
src/lapackf.cpp	1362
src/lapackf.h	1364
src/legacy.cpp	1364
src/linear.cpp	1365
src/linear.h	1367
src/linput.cpp	1367
src/locus.cpp	1368
src/logistic.cpp	1369
src/logistic.h	1369
src/lookup.cpp	1370
src/lookup2.cpp	1371
src/mds.cpp	1373
src/merge.cpp	1373
src/metaanal.cpp	1374
src/metaem.cpp	1375
src/mh.cpp	1376
src/mishap.cpp	1377
src/model.cpp	1378
src/model.h	1378
src/multi.cpp	1379

src/multiple.cpp	1380
src/nlist.cpp	1381
src/nlist.h	1381
src/nonfounderphasing.cpp	1382
src/options.cpp	1383
src/options.h	1384
src/output.cpp	1385
src/parse.cpp	1386
src/pdriver.cpp	1387
src/perm.cpp	1388
src/perm.h	1388
src/phase.cpp	1389
src/phase.h	1390
src/plink.cpp	1390
src/plink.h	1390
src/PlinkInternalsDataset.cpp	1395
src/PlinkInternalsDataset.h	1396
src/PlinkInternalsDatasetInstance.cpp	1396
src/PlinkInternalsDatasetInstance.h	1397
src/poo.cpp	1398
src/prehap.cpp	1399
src/profile.cpp	1400
src/proxy.cpp	1401
src/qfam.cpp	1402
src/qualscores.cpp	1404
src/r.cpp	1404
src/RandomForest.cpp	1405
src/RandomForest.h	1405
src/Rconnection.cpp	1406
src/Rconnection.h	1408
src/regain.cpp	1413
src/regain.h	1414
src/ReliefF.cpp	1416
src/ReliefF.h	1419
src/ReliefFSeq.cpp	1420
src/ReliefFSeq.h	1420
src/ReliefSeqController.cpp	1421
src/ReliefSeqController.h	1422
src/RReliefF.cpp	1423
src/RReliefF.h	1423
src/Rsrv.h	1424
src/segment.cpp	1447
src/sets.cpp	1447
src/sets.h	1448
src/setscreen.cpp	1449
src/sharing.cpp	1450
src/simul.cpp	1450
src/sisocks.h	1451
src/SNReliefF.cpp	1457
src/SNReliefF.h	1457
src/sockets.cpp	1460
src/sockets.h	1461
src/Statistics.cpp	1462

src/ Statistics.h	1469
Statistical utilities	
src/ stats.cpp	1477
src/ stats.h	1497
src/ step.cpp	1515
src/ StringUtil.h	
Various string-related utilities	1515
src/ tag.cpp	1518
src/ tdt.cpp	1518
src/ tinput.cpp	1519
src/ Tree.cpp	1520
src/ Tree.h	1520
src/ TreeClassification.cpp	1521
src/ TreeClassification.h	1521
src/ TreeProbability.cpp	1522
src/ TreeProbability.h	1523
src/ TreeRegression.cpp	1524
src/ TreeRegression.h	1525
src/ TreeSurvival.cpp	1526
src/ TreeSurvival.h	1526
src/ trio.cpp	1527
src/ utility.cpp	1529
src/ utility.h	1546
src/ version.h	1567
src/ webcheck.cpp	1568
src/ whap.cpp	1570
src/ whap.h	1571
src/ zed.cpp	1572
src/ zed.h	1573
src/ zfstream.cpp	1575
src/ zfstream.h	1576

Chapter 5

Namespace Documentation

5.1 insilico Namespace Reference

Classes

- class [do_to_lower](#)
- class [do_to_upper](#)
- class [is_classified](#)

Functions

- template<typename InputIt , typename OutputIt , typename Comp >
void [best_n](#) (InputIt begin, InputIt end, OutputIt out, size_t n, Comp comp)
Get the best n values with ties keeping same original order.
- template<typename stringT >
stringT [trim_left](#) (const stringT &s, const std::locale &loc=std::locale())
- template<typename stringT >
stringT [trim_right](#) (const stringT &s, const std::locale &loc=std::locale())
- template<typename stringT >
stringT [trim](#) (const stringT &s, const std::locale &loc=std::locale())
- template<typename Container , typename stringT >
void [split](#) (Container &cont, const stringT &s, const std::locale &loc=std::locale())
- template<typename Container , typename stringT >
void [split](#) (Container &cont, const stringT &s, const stringT &delim)
- template<typename Container , typename stringT , typename Pred >
void [split_if](#) (Container &cont, const stringT &s, const Pred &pred)
- template<typename It , typename stringT >
stringT [join](#) (const It &begin, const It &end, const stringT &delim)
- template<typename stringT >
stringT [to_upper](#) (const stringT &str, const std::locale &loc=std::locale())
- template<typename stringT >
stringT [to_lower](#) (const stringT &str, const std::locale &loc=std::locale())
- std::string [trim_left](#) (const char *s, const std::locale &loc=std::locale())
- std::wstring [trim_left](#) (const wchar_t *s, const std::locale &loc=std::locale())

- std::string **trim_right** (const char *s, const std::locale &loc=std::locale())
- std::wstring **trim_right** (const wchar_t *s, const std::locale &loc=std::locale())
- std::string **trim** (const char *s, const std::locale &loc=std::locale())
- std::wstring **trim** (const wchar_t *s, const std::locale &loc=std::locale())
- template<typename Container >
void **split** (Container &cont, const char *s, const std::locale &loc=std::locale())
- template<typename Container >
void **split** (Container &cont, const wchar_t *s, const std::locale &loc=std::locale())
- template<typename Container >
void **split** (Container &cont, const std::string &s, const char *delim)
- template<typename Container >
void **split** (Container &cont, const char *s, const std::string &delim)
- template<typename Container >
void **split** (Container &cont, const char *s, const char *delim)
- template<typename Container >
void **split** (Container &cont, const std::wstring &s, const wchar_t *delim)
- template<typename Container >
void **split** (Container &cont, const wchar_t *s, const std::wstring &delim)
- template<typename Container >
void **split** (Container &cont, const wchar_t *s, const wchar_t *delim)
- template<typename Container , typename Pred >
void **split_if** (Container &cont, const char *s, const Pred &pred)
- template<typename Container , typename Pred >
void **split_if** (Container &cont, const wchar_t *s, const Pred &pred)
- template<typename It >
std::string **join** (const It &begin, const It &end, const char *delim)
- template<typename It >
std::wstring **join** (const It &begin, const It &end, const wchar_t *delim)
- std::string **to_upper** (const char *s, const std::locale &loc=std::locale())
- std::wstring **to_upper** (const wchar_t *s, const std::locale &loc=std::locale())
- std::string **to_lower** (const char *s, const std::locale &loc=std::locale())
- std::wstring **to_lower** (const wchar_t *s, const std::locale &loc=std::locale())
- template<typename T >
std::string **get_bits** (T value)
- template<typename T >
std::string **zeroPadNumber** (T num, int padSize)

5.1.1 Function Documentation

5.1.1.1 **best_n()**

```
template<typename InputIt , typename OutputIt , typename Comp >
void insilico::best_n (
    InputIt begin,
    InputIt end,
    OutputIt out,
    size_t n,
    Comp comp )
```

Get the best n values with ties keeping same original order.

Parameters

in	<i>begin</i>	iterator of the beginning of a input container
in	<i>end</i>	iterator of the end of a input container
out	<i>out</i>	iterator of the beginning of a output container
in	<i>size</i>	best n value
in	<i>comp</i>	compare functor

Returns

path/filename without extension

Definition at line 30 of file BestN.h.

5.1.1.2 get_bits()

```
template<typename T >
std::string insilico::get_bits (
    T value )
```

Definition at line 324 of file StringUtils.h.

5.1.1.3 join() [1/3]

```
template<typename It , typename stringT >
stringT insilico::join (
    const It & begin,
    const It & end,
    const stringT & delim )
```

Definition at line 198 of file StringUtils.h.

5.1.1.4 join() [2/3]

```
template<typename It >
std::string insilico::join (
    const It & begin,
    const It & end,
    const char * delim ) [inline]
```

Definition at line 300 of file StringUtils.h.

5.1.1.5 `join()` [3/3]

```
template<typename It >
std::wstring insilico::join (
    const It & begin,
    const It & end,
    const wchar_t * delim ) [inline]
```

Definition at line 304 of file `StringUtils.h`.

5.1.1.6 `split()` [1/10]

```
template<typename Container , typename stringT >
void insilico::split (
    Container & cont,
    const stringT & s,
    const std::locale & loc = std::locale() ) [inline]
```

Definition at line 148 of file `StringUtils.h`.

5.1.1.7 `split()` [2/10]

```
template<typename Container , typename stringT >
void insilico::split (
    Container & cont,
    const stringT & s,
    const stringT & delim )
```

Definition at line 156 of file `StringUtils.h`.

5.1.1.8 `split()` [3/10]

```
template<typename Container >
void insilico::split (
    Container & cont,
    const char * s,
    const std::locale & loc = std::locale() ) [inline]
```

Definition at line 258 of file `StringUtils.h`.

5.1.1.9 split() [4/10]

```
template<typename Container >
void insilico::split (
    Container & cont,
    const wchar_t * s,
    const std::locale & loc = std::locale() ) [inline]
```

Definition at line 263 of file StringUtil.h.

5.1.1.10 split() [5/10]

```
template<typename Container >
void insilico::split (
    Container & cont,
    const std::string & s,
    const char * delim ) [inline]
```

Definition at line 268 of file StringUtil.h.

5.1.1.11 split() [6/10]

```
template<typename Container >
void insilico::split (
    Container & cont,
    const char * s,
    const std::string & delim ) [inline]
```

Definition at line 272 of file StringUtil.h.

5.1.1.12 split() [7/10]

```
template<typename Container >
void insilico::split (
    Container & cont,
    const char * s,
    const char * delim ) [inline]
```

Definition at line 276 of file StringUtil.h.

5.1.1.13 split() [8/10]

```
template<typename Container >
void insilico::split (
    Container & cont,
    const std::wstring & s,
    const wchar_t * delim ) [inline]
```

Definition at line 280 of file StringUtil.h.

5.1.1.14 split() [9/10]

```
template<typename Container >
void insilico::split (
    Container & cont,
    const wchar_t * s,
    const std::wstring & delim ) [inline]
```

Definition at line 284 of file StringUtil.h.

5.1.1.15 split() [10/10]

```
template<typename Container >
void insilico::split (
    Container & cont,
    const wchar_t * s,
    const wchar_t * delim ) [inline]
```

Definition at line 288 of file StringUtil.h.

5.1.1.16 split_if() [1/3]

```
template<typename Container , typename stringT , typename Pred >
void insilico::split_if (
    Container & cont,
    const stringT & s,
    const Pred & pred )
```

Definition at line 178 of file StringUtil.h.

5.1.1.17 split_if() [2/3]

```
template<typename Container , typename Pred >
void insilico::split_if (
    Container & cont,
    const char * s,
    const Pred & pred ) [inline]
```

Definition at line 292 of file StringUtil.h.

5.1.1.18 split_if() [3/3]

```
template<typename Container , typename Pred >
void insilico::split_if (
    Container & cont,
    const wchar_t * s,
    const Pred & pred ) [inline]
```

Definition at line 296 of file StringUtil.h.

5.1.1.19 to_lower() [1/3]

```
template<typename stringT >
stringT insilico::to_lower (
    const stringT & str,
    const std::locale & loc = std::locale() )
```

Definition at line 224 of file StringUtil.h.

5.1.1.20 to_lower() [2/3]

```
std::string insilico::to_lower (
    const char * s,
    const std::locale & loc = std::locale() ) [inline]
```

Definition at line 315 of file StringUtil.h.

5.1.1.21 to_lower() [3/3]

```
std::wstring insilico::to_lower (
    const wchar_t * s,
    const std::locale & loc = std::locale() ) [inline]
```

Definition at line 319 of file StringUtil.h.

5.1.1.22 to_upper() [1/3]

```
template<typename stringT >
stringT insilico::to_upper (
    const stringT & str,
    const std::locale & loc = std::locale() )
```

Definition at line 214 of file StringUtil.h.

5.1.1.23 to_upper() [2/3]

```
std::string insilico::to_upper (
    const char * s,
    const std::locale & loc = std::locale() ) [inline]
```

Definition at line 307 of file StringUtil.h.

5.1.1.24 to_upper() [3/3]

```
std::wstring insilico::to_upper (
    const wchar_t * s,
    const std::locale & loc = std::locale() ) [inline]
```

Definition at line 311 of file StringUtil.h.

5.1.1.25 trim() [1/3]

```
template<typename stringT >
stringT insilico::trim (
    const stringT & s,
    const std::locale & loc = std::locale() )
```

Definition at line 123 of file StringUtil.h.

5.1.1.26 trim() [2/3]

```
std::string insilico::trim (
    const char * s,
    const std::locale & loc = std::locale() ) [inline]
```

Definition at line 249 of file StringUtil.h.

5.1.1.27 trim() [3/3]

```
std::wstring insilico::trim (
    const wchar_t * s,
    const std::locale & loc = std::locale() ) [inline]
```

Definition at line 253 of file StringUtil.h.

5.1.1.28 trim_left() [1/3]

```
template<typename stringT >
stringT insilico::trim_left (
    const stringT & s,
    const std::locale & loc = std::locale() )
```

Definition at line 101 of file StringUtil.h.

5.1.1.29 trim_left() [2/3]

```
std::string insilico::trim_left (
    const char * s,
    const std::locale & loc = std::locale() ) [inline]
```

Definition at line 233 of file StringUtil.h.

5.1.1.30 trim_left() [3/3]

```
std::wstring insilico::trim_left (
    const wchar_t * s,
    const std::locale & loc = std::locale() ) [inline]
```

Definition at line 237 of file StringUtil.h.

5.1.1.31 trim_right() [1/3]

```
template<typename stringT >
stringT insilico::trim_right (
    const stringT & s,
    const std::locale & loc = std::locale() )
```

Definition at line 112 of file StringUtil.h.

5.1.1.32 trim_right() [2/3]

```
std::string insilico::trim_right (
    const char * s,
    const std::locale & loc = std::locale() ) [inline]
```

Definition at line 241 of file StringUtil.h.

5.1.1.33 trim_right() [3/3]

```
std::wstring insilico::trim_right (
    const wchar_t * s,
    const std::locale & loc = std::locale() ) [inline]
```

Definition at line 245 of file StringUtil.h.

5.1.1.34 zeroPadNumber()

```
template<typename T >
std::string insilico::zeroPadNumber (
    T num,
    int padSize )
```

Definition at line 333 of file StringUtil.h.

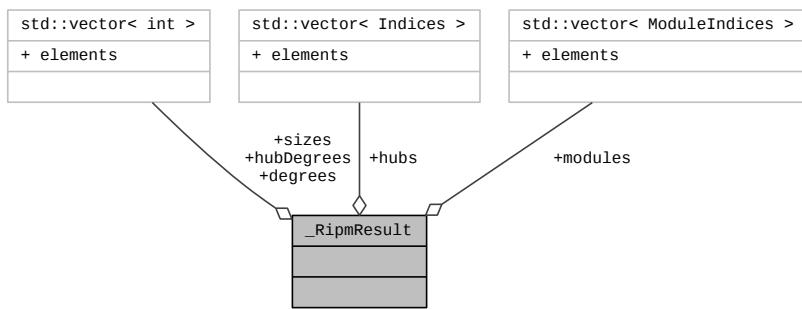
Chapter 6

Class Documentation

6.1 _RipmResult Struct Reference

```
#include <InteractionNetwork.h>
```

Collaboration diagram for _RipmResult:



Public Attributes

- `std::vector< int > degrees`
- `std::vector< int > sizes`
- `std::vector< Indices > hubs`
- `std::vector< int > hubDegrees`
- `ModuleList modules`

6.1.1 Detailed Description

Definition at line 28 of file InteractionNetwork.h.

6.1.2 Member Data Documentation

6.1.2.1 degrees

```
std::vector<int> _RipmResult::degrees
```

Definition at line 29 of file InteractionNetwork.h.

6.1.2.2 hubDegrees

```
std::vector<int> _RipmResult::hubDegrees
```

Definition at line 32 of file InteractionNetwork.h.

6.1.2.3 hubs

```
std::vector<Indices> _RipmResult::hubs
```

Definition at line 31 of file InteractionNetwork.h.

6.1.2.4 modules

```
ModuleList _RipmResult::modules
```

Definition at line 33 of file InteractionNetwork.h.

6.1.2.5 sizes

```
std::vector<int> _RipmResult::sizes
```

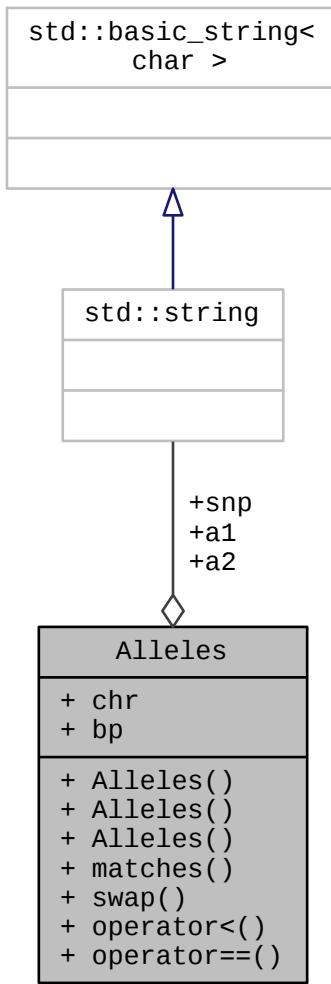
Definition at line 30 of file InteractionNetwork.h.

The documentation for this struct was generated from the following file:

- src/[InteractionNetwork.h](#)

6.2 Alleles Class Reference

Collaboration diagram for Alleles:



Public Member Functions

- `Alleles (string name)`
- `Alleles (string name, int chr, int bp)`
- `Alleles (string name, int chr, int bp, string a1, string a2)`
- `bool matches (string b1, string b2) const`
- `bool swap (string b1) const`
- `bool operator< (const Alleles &b) const`
- `bool operator== (const Alleles &b) const`

Public Attributes

- string `snp`
- string `a1`
- string `a2`
- int `chr`
- int `bp`

6.2.1 Detailed Description

Definition at line 39 of file metaanal.cpp.

6.2.2 Constructor & Destructor Documentation

6.2.2.1 Alleles() [1/3]

```
Alleles::Alleles (
    string name )  [inline]
```

Definition at line 49 of file metaanal.cpp.

6.2.2.2 Alleles() [2/3]

```
Alleles::Alleles (
    string name,
    int chr,
    int bp )  [inline]
```

Definition at line 56 of file metaanal.cpp.

6.2.2.3 Alleles() [3/3]

```
Alleles::Alleles (
    string name,
    int chr,
    int bp,
    string a1,
    string a2 )  [inline]
```

Definition at line 61 of file metaanal.cpp.

6.2.3 Member Function Documentation

6.2.3.1 matches()

```
bool Alleles::matches (
    string b1,
    string b2 ) const [inline]
```

Definition at line 64 of file metaanal.cpp.

6.2.3.2 operator<()

```
bool Alleles::operator< (
    const Alleles & b ) const [inline]
```

Definition at line 73 of file metaanal.cpp.

6.2.3.3 operator==()

```
bool Alleles::operator== (
    const Alleles & b ) const [inline]
```

Definition at line 84 of file metaanal.cpp.

6.2.3.4 swap()

```
bool Alleles::swap (
    string b1 ) const [inline]
```

Definition at line 70 of file metaanal.cpp.

6.2.4 Member Data Documentation

6.2.4.1 a1

```
string Alleles::a1
```

Definition at line 44 of file metaanal.cpp.

6.2.4.2 a2

```
string Alleles::a2
```

Definition at line 45 of file metaanal.cpp.

6.2.4.3 bp

```
int Alleles::bp
```

Definition at line 47 of file metaanal.cpp.

6.2.4.4 chr

```
int Alleles::chr
```

Definition at line 46 of file metaanal.cpp.

6.2.4.5 snp

```
string Alleles::snp
```

Definition at line 43 of file metaanal.cpp.

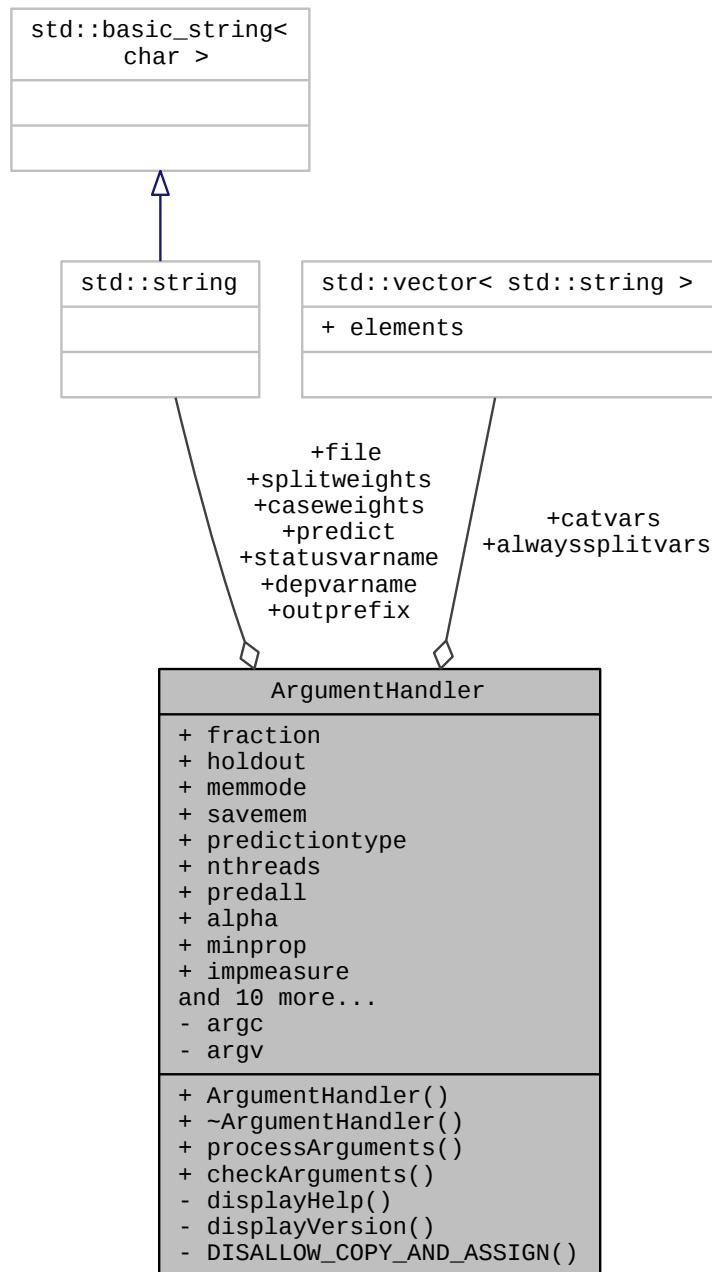
The documentation for this class was generated from the following file:

- [src/metaanal.cpp](#)

6.3 ArgumentHandler Class Reference

```
#include <ArgumentHandler.h>
```

Collaboration diagram for ArgumentHandler:



Public Member Functions

- `ArgumentHandler (int argc, char **argv)`
- `virtual ~ArgumentHandler ()`
- `int processArguments ()`
- `void checkArguments ()`

Public Attributes

- `std::vector< std::string > alwaysplitvars`
- `std::string caseweights`
- `std::string depvarname`
- `double fraction`
- `bool holdout`
- `MemoryMode memmode`
- `bool savemem`
- `std::string predict`
- `PredictionType predictiontype`
- `std::string splitweights`
- `uint nthreads`
- `bool predall`
- `double alpha`
- `double minprop`
- `std::vector< std::string > catvars`
- `std::string file`
- `ImportanceMode impmeasure`
- `uint targetpartitionsize`
- `uint mtry`
- `std::string outprefix`
- `bool probability`
- `SplitRule splitrule`
- `std::string statusvarname`
- `uint ntree`
- `bool replace`
- `bool verbose`
- `bool write`
- `TreeType treetype`
- `uint seed`

Private Member Functions

- `void displayHelp ()`
- `void displayVersion ()`
- `DISALLOW_COPY_AND_ASSIGN (ArgumentHandler)`

Private Attributes

- `int argc`
- `char ** argv`

6.3.1 Detailed Description

Definition at line 51 of file ArgumentHandler.h.

6.3.2 Constructor & Destructor Documentation

6.3.2.1 ArgumentHandler()

```
ArgumentHandler::ArgumentHandler (
    int argc,
    char ** argv )
```

Definition at line 37 of file ArgumentHandler.cpp.

6.3.2.2 ~ArgumentHandler()

```
ArgumentHandler::~ArgumentHandler ( ) [virtual]
```

Definition at line 47 of file ArgumentHandler.cpp.

6.3.3 Member Function Documentation

6.3.3.1 checkArguments()

```
void ArgumentHandler::checkArguments ( )
```

Definition at line 392 of file ArgumentHandler.cpp.

6.3.3.2 DISALLOW_COPY_AND_ASSIGN()

```
ArgumentHandler::DISALLOW_COPY_AND_ASSIGN (
    ArgumentHandler ) [private]
```

6.3.3.3 `displayHelp()`

```
void ArgumentHandler::displayHelp ( ) [private]
```

Definition at line 467 of file ArgumentHandler.cpp.

6.3.3.4 `displayVersion()`

```
void ArgumentHandler::displayVersion ( ) [private]
```

Definition at line 542 of file ArgumentHandler.cpp.

6.3.3.5 `processArguments()`

```
int ArgumentHandler::processArguments ( )
```

Definition at line 50 of file ArgumentHandler.cpp.

6.3.4 Member Data Documentation

6.3.4.1 `alpha`

```
double ArgumentHandler::alpha
```

Definition at line 77 of file ArgumentHandler.h.

6.3.4.2 `alwaysSplitVars`

```
std::vector<std::string> ArgumentHandler::alwaysSplitVars
```

Definition at line 63 of file ArgumentHandler.h.

6.3.4.3 argc

```
int ArgumentHandler::argc [private]
```

Definition at line 100 of file ArgumentHandler.h.

6.3.4.4 argv

```
char** ArgumentHandler::argv [private]
```

Definition at line 101 of file ArgumentHandler.h.

6.3.4.5 caseweights

```
std::string ArgumentHandler::caseweights
```

Definition at line 64 of file ArgumentHandler.h.

6.3.4.6 catvars

```
std::vector<std::string> ArgumentHandler::catvars
```

Definition at line 79 of file ArgumentHandler.h.

6.3.4.7 depvarname

```
std::string ArgumentHandler::depvarname
```

Definition at line 65 of file ArgumentHandler.h.

6.3.4.8 file

```
std::string ArgumentHandler::file
```

Definition at line 80 of file ArgumentHandler.h.

6.3.4.9 fraction

```
double ArgumentHandler::fraction
```

Definition at line 66 of file ArgumentHandler.h.

6.3.4.10 holdout

```
bool ArgumentHandler::holdout
```

Definition at line 67 of file ArgumentHandler.h.

6.3.4.11 impmeasure

```
ImportanceMode ArgumentHandler::impmeasure
```

Definition at line 81 of file ArgumentHandler.h.

6.3.4.12 memmode

```
MemoryMode ArgumentHandler::memmode
```

Definition at line 68 of file ArgumentHandler.h.

6.3.4.13 minprop

```
double ArgumentHandler::minprop
```

Definition at line 78 of file ArgumentHandler.h.

6.3.4.14 mtry

```
uint ArgumentHandler::mtry
```

Definition at line 83 of file ArgumentHandler.h.

6.3.4.15 nthreads

```
uint ArgumentHandler::nthreads
```

Definition at line 73 of file ArgumentHandler.h.

6.3.4.16 ntree

```
uint ArgumentHandler::ntree
```

Definition at line 88 of file ArgumentHandler.h.

6.3.4.17 outprefix

```
std::string ArgumentHandler::outprefix
```

Definition at line 84 of file ArgumentHandler.h.

6.3.4.18 predall

```
bool ArgumentHandler::predall
```

Definition at line 74 of file ArgumentHandler.h.

6.3.4.19 predict

```
std::string ArgumentHandler::predict
```

Definition at line 70 of file ArgumentHandler.h.

6.3.4.20 predictiontype

```
PredictionType ArgumentHandler::predictiontype
```

Definition at line 71 of file ArgumentHandler.h.

6.3.4.21 probability

```
bool ArgumentHandler::probability
```

Definition at line 85 of file ArgumentHandler.h.

6.3.4.22 replace

```
bool ArgumentHandler::replace
```

Definition at line 89 of file ArgumentHandler.h.

6.3.4.23 savemem

```
bool ArgumentHandler::savemem
```

Definition at line 69 of file ArgumentHandler.h.

6.3.4.24 seed

```
uint ArgumentHandler::seed
```

Definition at line 93 of file ArgumentHandler.h.

6.3.4.25 splitrule

```
SplitRule ArgumentHandler::splitrule
```

Definition at line 86 of file ArgumentHandler.h.

6.3.4.26 splitweights

```
std::string ArgumentHandler::splitweights
```

Definition at line 72 of file ArgumentHandler.h.

6.3.4.27 statusvarname

```
std::string ArgumentHandler::statusvarname
```

Definition at line 87 of file ArgumentHandler.h.

6.3.4.28 targetpartitionsize

```
uint ArgumentHandler::targetpartitionsize
```

Definition at line 82 of file ArgumentHandler.h.

6.3.4.29 treetype

```
TreeType ArgumentHandler::treetype
```

Definition at line 92 of file ArgumentHandler.h.

6.3.4.30 verbose

```
bool ArgumentHandler::verbose
```

Definition at line 90 of file ArgumentHandler.h.

6.3.4.31 write

```
bool ArgumentHandler::write
```

Definition at line 91 of file ArgumentHandler.h.

The documentation for this class was generated from the following files:

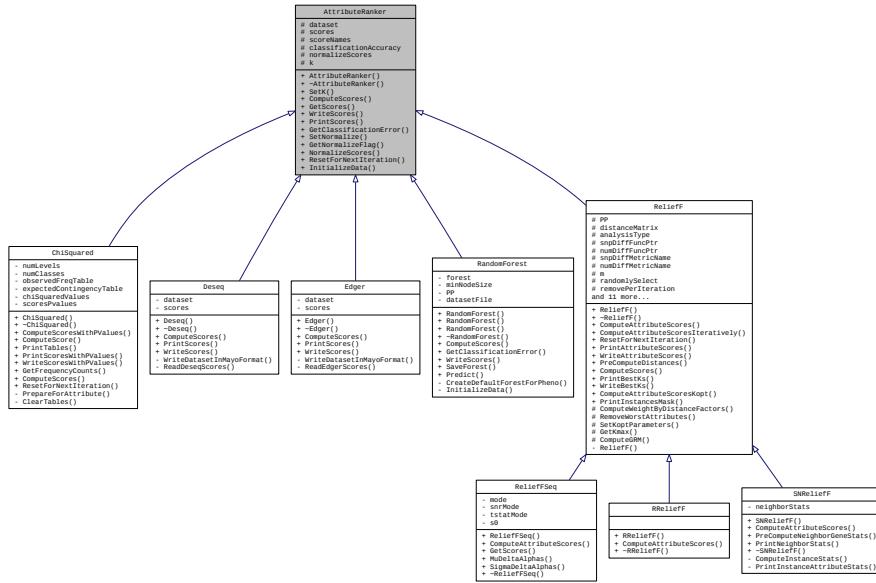
- src/[ArgumentHandler.h](#)
- src/[ArgumentHandler.cpp](#)

6.4 AttributeRanker Class Reference

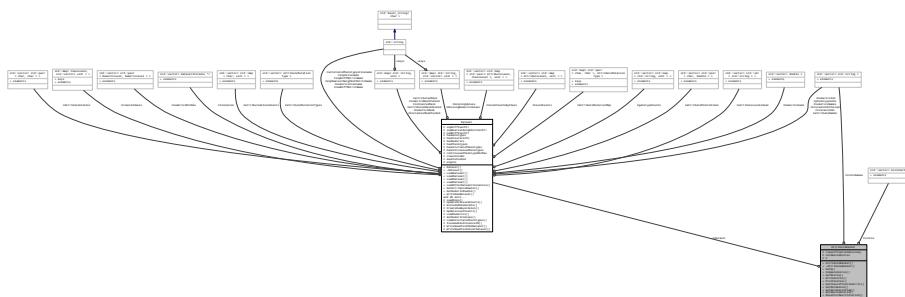
Abstract base class for attribute rankers.

```
#include <AttributeRanker.h>
```

Inheritance diagram for AttributeRanker:



Collaboration diagram for AttributeRanker:



Public Member Functions

- **AttributeRanker (Dataset *ds)**
Construct a default data set.
- virtual **~AttributeRanker ()**
Destruct all dynamically allocated memory.
- virtual bool **SetK (unsigned int newK)**

- **Set k nearest neighbors, with bounds checking.**
- virtual **AttributeScores ComputeScores ()=0**
Compute the attribute scores for the current set of attributes.
- virtual **AttributeScores GetScores ()**
Get the (importance) scores as a vector of pairs: score, attribute name.
- virtual void **WriteScores (std::string baseFilename)**
Write the scores and attribute names to file.
- virtual void **PrintScores (std::ofstream &outStream)**
Write the EC scores and attribute names to stream.
- virtual double **GetClassificationError ()**
Error from using ranked attributes in a classifier.
- void **SetNormalize (bool switchTF=true)**
- bool **GetNormalizeFlag ()**
Will scores be normalized after computing scores.
- bool **NormalizeScores ()**
- virtual bool **ResetForNextIteration ()**
Reset the algorithm.
- virtual bool **InitializeData (bool doPrediction, bool useMask=false, bool dolImportance=true)**

Protected Attributes

- **Dataset * dataset**
*The **Dataset** on which the ranking algorithm is working.*
- **AttributeScores scores**
attribute scores and names
- std::vector< std::string > **scoreNames**
attribute names
- double **classificationAccuracy**
Error from using ranked attributes in a classifier.
- bool **normalizeScores**
Normalize scores 0-1 flag?
- unsigned int **k**
k nearest neighbors

6.4.1 Detailed Description

Abstract base class for attribute rankers.

Author

Bill White

Version

1.0

Contact: bill.c.white@gmail.com Created on: 8/13/12

Definition at line 22 of file AttributeRanker.h.

6.4.2 Constructor & Destructor Documentation

6.4.2.1 AttributeRanker()

```
AttributeRanker::AttributeRanker (
    Dataset * ds )
```

Construct a default data set.

Definition at line 21 of file AttributeRanker.cpp.

6.4.2.2 ~AttributeRanker()

```
AttributeRanker::~AttributeRanker ( ) [virtual]
```

Destruct all dynamically allocated memory.

Definition at line 32 of file AttributeRanker.cpp.

6.4.3 Member Function Documentation

6.4.3.1 ComputeScores()

```
virtual AttributeScores AttributeRanker::ComputeScores ( ) [pure virtual]
```

Compute the attribute scores for the current set of attributes.

Implemented in [ChiSquared](#), [ReliefF](#), [RandomForest](#), [Deseq](#), and [Edger](#).

6.4.3.2 GetClassificationError()

```
double AttributeRanker::GetClassificationError ( ) [virtual]
```

Error from using ranked attributes in a classifier.

Reimplemented in [RandomForest](#).

Definition at line 113 of file AttributeRanker.cpp.

6.4.3.3 GetNormalizeFlag()

```
bool AttributeRanker::GetNormalizeFlag ( )
```

Will scores be normalized after computing scores.

Definition at line 121 of file AttributeRanker.cpp.

6.4.3.4 GetScores()

```
AttributeScores AttributeRanker::GetScores ( ) [virtual]
```

Get the (importance) scores as a vector of pairs: score, attribute name.

Returns

vector of pairs

Reimplemented in [ReliefFSeq](#).

Definition at line 81 of file AttributeRanker.cpp.

6.4.3.5 InitializeData()

```
virtual bool AttributeRanker::InitializeData (
    bool doPrediction,
    bool useMask = false,
    bool doImportance = true ) [inline], [virtual]
```

Reimplemented in [RandomForest](#).

Definition at line 58 of file AttributeRanker.h.

6.4.3.6 NormalizeScores()

```
bool AttributeRanker::NormalizeScores ( )
```

Definition at line 125 of file AttributeRanker.cpp.

6.4.3.7 PrintScores()

```
void AttributeRanker::PrintScores (
    std::ofstream & outStream ) [virtual]
```

Write the EC scores and attribute names to stream.

Parameters

in	<i>outStream</i>	stream to write score-attribute name pairs
----	------------------	--

Definition at line 106 of file AttributeRanker.cpp.

6.4.3.8 ResetForNextIteration()

```
virtual bool AttributeRanker::ResetForNextIteration () [inline], [virtual]
```

Reset the algorithm.

Reimplemented in [ChiSquared](#), and [ReliefF](#).

Definition at line 56 of file AttributeRanker.h.

6.4.3.9 SetK()

```
bool AttributeRanker::SetK (
    unsigned int newK ) [virtual]
```

Set k nearest neighbors, with bounds checking.

Definition at line 35 of file AttributeRanker.cpp.

6.4.3.10 SetNormalize()

```
void AttributeRanker::SetNormalize (
    bool switchTF = true )
```

Definition at line 117 of file AttributeRanker.cpp.

6.4.3.11 WriteScores()

```
void AttributeRanker::WriteScores (
    std::string baseFilename ) [virtual]
```

Write the scores and attribute names to file.

Parameters

in	<i>baseFilename</i>	filename to write score-attribute name pairs
----	---------------------	--

Reimplemented in [RandomForest](#).

Definition at line 93 of file AttributeRanker.cpp.

6.4.4 Member Data Documentation

6.4.4.1 classificationAccuracy

```
double AttributeRanker::classificationAccuracy [protected]
```

Error from using ranked attributes in a classifier.

Definition at line 68 of file AttributeRanker.h.

6.4.4.2 dataset

```
Dataset* AttributeRanker::dataset [protected]
```

The [Dataset](#) on which the ranking algorithm is working.

Definition at line 62 of file AttributeRanker.h.

6.4.4.3 k

```
unsigned int AttributeRanker::k [protected]
```

k nearest neighbors

Definition at line 72 of file AttributeRanker.h.

6.4.4.4 normalizeScores

`bool AttributeRanker::normalizeScores [protected]`

Normalize scores 0-1 flag?

Definition at line 70 of file AttributeRanker.h.

6.4.4.5 scoreNames

`std::vector<std::string> AttributeRanker::scoreNames [protected]`

attribute names

Definition at line 66 of file AttributeRanker.h.

6.4.4.6 scores

`AttributeScores AttributeRanker::scores [protected]`

attribute scores and names

Definition at line 64 of file AttributeRanker.h.

The documentation for this class was generated from the following files:

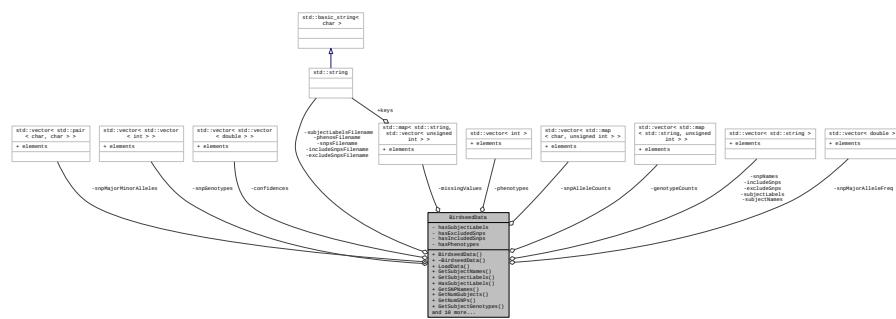
- src/[AttributeRanker.h](#)
- src/[AttributeRanker.cpp](#)

6.5 BirdseedData Class Reference

Read Broad's Birdsuite Birdseed-called SNP data.

```
#include <BirdseedData.h>
```

Collaboration diagram for BirdseedData:



Public Member Functions

- `BirdseedData ()`
- `virtual ~BirdseedData ()`
- `bool LoadData (std::string snpsFile, std::string phenoFile="", std::string subjsFile="", std::string includeSnpsFile="", std::string excludeSnpsFile="")`

Create a new set of Birdseed data with a SNPs file and optional phenotype file and optional subject names file.
- `std::vector< std::string > GetSubjectNames ()`

Get the subject names/IDs.
- `std::vector< std::string > GetSubjectLabels ()`

Get the subject labels.
- `bool HasSubjectLabels ()`

Do the subjects have labels?
- `std::vector< std::string > GetSNPNames ()`

Get the SNP names/IDs.
- `int GetNumSubjects ()`

Get the number of subjects.
- `int GetNumSNPs ()`

Get the number of SNPs.
- `std::vector< int > GetSubjectGenotypes (int subjectIndex)`

Get SNPs for sample at index.
- `std::vector< double > GetSubjectCallConfidences (int subjectIndex)`

Get SNP call confidences for sample at index.
- `int GetSamplePhenotype (int subjectIndex)`

Get the phenotype at sample index.
- `void PrintInfo ()`

Print basic statistics to the console.
- `bool HasPhenotypes ()`

Does this data have phenotypes?
- `std::pair< char, char > GetMajorMinorAlleles (int snpIndex)`

Get the major and minor alleles for a SNP.
- `double GetMajorAlleleFrequency (int snpIndex)`

Get the major allele frequency for a SNP.
- `std::map< char, unsigned int > GetAlleleCounts (int snpIndex)`

Get the allele counts for a SNP.
- `std::map< std::string, unsigned int > GetGenotypeCounts (int snpIndex)`

Get the original string genotype counts for a SNP.
- `bool GetMissingValues (std::string subjectName, std::vector< unsigned int > &missingValueIndices)`

get the missing value indices for the subject name
- `void PrintAlleleCounts ()`

Print the allele counts for each SNP to the console.

Private Attributes

- std::string **snpsFilename**
Filename containing birdseed-called SNPs.
- std::string **subjectLabelsFilename**
Filename containing subject names.
- std::vector< std::string > **subjectLabels**
- bool **hasSubjectLabels**
- std::vector< std::string > **subjectNames**
- std::string **excludeSnpsFilename**
- std::vector< std::string > **excludeSnps**
- bool **hasExcludedSnps**
- std::string **includeSnpsFilename**
- std::vector< std::string > **includeSnps**
- bool **hasIncludedSnps**
- std::vector< std::string > **snpNames**
SNP names.
- std::vector< std::vector< int > > **snpGenotypes**
SNP genotype calls and their corresponding confidences.
- std::vector< std::vector< double > > **confidences**
- std::vector< std::map< std::string, unsigned int > > **genotypeCounts**
SNP genotype->count.
- std::vector< std::pair< char, char > > **snpMajorMinorAlleles**
SNP genotypes alleles.
- std::vector< double > **snpMajorAlleleFreq**
SNP genotypes major allele frequency.
- std::vector< std::map< char, unsigned int > > **snpAlleleCounts**
SNP allele->count.
- std::map< std::string, std::vector< unsigned int > > **missingValues**
subject name -> attribute indices
- std::string **phenosFilename**
Sample phenotypes Filename containing subject phenotypes.
- std::vector< int > **phenotypes**
vector of phenotypes (case-control)
- bool **hasPhenotypes**
has phenotypes?

6.5.1 Detailed Description

Read Broad's Birdsuite Birdseed-called SNP data.

Author

Bill White

Version

1.0

Contact: bill.c.white@gmail.com Created on: 2/12/12

Definition at line 20 of file BirdseedData.h.

6.5.2 Constructor & Destructor Documentation

6.5.2.1 BirdseedData()

```
BirdseedData::BirdseedData ( )
```

Definition at line 24 of file BirdseedData.cpp.

6.5.2.2 ~BirdseedData()

```
BirdseedData::~BirdseedData ( ) [virtual]
```

Definition at line 36 of file BirdseedData.cpp.

6.5.3 Member Function Documentation

6.5.3.1 GetAlleleCounts()

```
map< char, unsigned int > BirdseedData::GetAlleleCounts ( int snpIndex )
```

Get the allele counts for a SNP.

Definition at line 576 of file BirdseedData.cpp.

6.5.3.2 GetGenotypeCounts()

```
map< string, unsigned int > BirdseedData::GetGenotypeCounts ( int snpIndex )
```

Get the original string genotype counts for a SNP.

Definition at line 587 of file BirdseedData.cpp.

6.5.3.3 GetMajorAlleleFrequency()

```
double BirdseedData::GetMajorAlleleFrequency (
    int snpIndex )
```

Get the major allele frequency for a SNP.

Definition at line 565 of file BirdseedData.cpp.

6.5.3.4 GetMajorMinorAlleles()

```
pair< char, char > BirdseedData::GetMajorMinorAlleles (
    int snpIndex )
```

Get the major and minor alleles for a SNP.

Definition at line 554 of file BirdseedData.cpp.

6.5.3.5 GetMissingValues()

```
bool BirdseedData::GetMissingValues (
    std::string subjectName,
    std::vector< unsigned int > & missingValueIndices )
```

get the missing value indices for the subject name

Definition at line 598 of file BirdseedData.cpp.

6.5.3.6 GetNumSNPs()

```
int BirdseedData::GetNumSNPs ( )
```

Get the number of SNPs.

Definition at line 488 of file BirdseedData.cpp.

6.5.3.7 GetNumSubjects()

```
int BirdseedData::GetNumSubjects ( )
```

Get the number of subjects.

Definition at line 484 of file BirdseedData.cpp.

6.5.3.8 GetSamplePhenotype()

```
int BirdseedData::GetSamplePhenotype (
    int subjectIndex )
```

Get the phenotype at sample index.

Definition at line 522 of file BirdseedData.cpp.

6.5.3.9 GetSNPNames()

```
vector< string > BirdseedData::GetSNPNames ( )
```

Get the SNP names/IDs.

Definition at line 480 of file BirdseedData.cpp.

6.5.3.10 GetSubjectCallConfidences()

```
vector< double > BirdseedData::GetSubjectCallConfidences (
    int subjectIndex )
```

Get SNP call confidences for sample at index.

Definition at line 507 of file BirdseedData.cpp.

6.5.3.11 GetSubjectGenotypes()

```
vector< int > BirdseedData::GetSubjectGenotypes (  
    int subjectIndex )
```

Get SNPs for sample at index.

Definition at line 492 of file BirdseedData.cpp.

6.5.3.12 GetSubjectLabels()

```
vector< string > BirdseedData::GetSubjectLabels ( )
```

Get the subject labels.

Definition at line 472 of file BirdseedData.cpp.

6.5.3.13 GetSubjectNames()

```
vector< string > BirdseedData::GetSubjectNames ( )
```

Get the subject names/IDs.

Definition at line 468 of file BirdseedData.cpp.

6.5.3.14 HasPhenotypes()

```
bool BirdseedData::HasPhenotypes ( )
```

Does this data have phenotypes?

Definition at line 550 of file BirdseedData.cpp.

6.5.3.15 HasSubjectLabels()

```
bool BirdseedData::HasSubjectLabels ( )
```

Do the subjects have labels?

Definition at line 476 of file BirdseedData.cpp.

6.5.3.16 LoadData()

```
bool BirdseedData::LoadData (
    std::string snpsFile,
    std::string phenoFile = "",
    std::string subjsFile = "",
    std::string includeSnpsFile = "",
    std::string excludeSnpsFile = "" )
```

Create a new set of Birdseed data with a SNPs file and optional phenotype file and optional subject names file.

read subjects file if specified

read SNP exclusion file

read SNP inclusion file

read SNPs data from the Birdsuite Birdseed SNP call file

skip any header comment lines

assumption: past any comment rows and at the the header row 7 fields per subject

read SNP genotypes across all SNPs and all subjects

split the line into genotypes

first field is the Affymetrix SNP ID

check for the inclusion/exclusion of this snpID

found - skip this SNP

not found - skip this SNP

skip missing genotypes for allele updates

missing data detected

save the attributes for this SNP

for each SNP, map two-allele genotypes to integers using allele frequencies

map genotype string vector to genotype int vector

read phenotypes

Definition at line 39 of file BirdseedData.cpp.

6.5.3.17 PrintAlleleCounts()

```
void BirdseedData::PrintAlleleCounts ( )
```

Print the allele counts for each SNP to the console.

Definition at line 606 of file BirdseedData.cpp.

6.5.3.18 PrintInfo()

```
void BirdseedData::PrintInfo ( )
```

Print basic statistics to the console.

Definition at line 532 of file BirdseedData.cpp.

6.5.4 Member Data Documentation

6.5.4.1 confidences

```
std::vector<std::vector<double>> BirdseedData::confidences [private]
```

Definition at line 84 of file BirdseedData.h.

6.5.4.2 excludeSnp

```
std::vector<std::string> BirdseedData::excludeSnp [private]
```

Definition at line 75 of file BirdseedData.h.

6.5.4.3 excludeSnpFilename

```
std::string BirdseedData::excludeSnpFilename [private]
```

Definition at line 74 of file BirdseedData.h.

6.5.4.4 genotypeCounts

```
std::vector<std::map<std::string, unsigned int>> BirdseedData::genotypeCounts [private]
```

SNP genotype->count.

Definition at line 86 of file BirdseedData.h.

6.5.4.5 hasExcludedSnp

```
bool BirdseedData::hasExcludedSnp [private]
```

Definition at line 76 of file BirdseedData.h.

6.5.4.6 hasIncludedSnp

```
bool BirdseedData::hasIncludedSnp [private]
```

Definition at line 79 of file BirdseedData.h.

6.5.4.7 hasPhenotypes

```
bool BirdseedData::hasPhenotypes [private]
```

has phenotypes?

Definition at line 103 of file BirdseedData.h.

6.5.4.8 hasSubjectLabels

```
bool BirdseedData::hasSubjectLabels [private]
```

Definition at line 71 of file BirdseedData.h.

6.5.4.9 includeSnps

```
std::vector<std::string> BirdseedData::includeSnps [private]
```

Definition at line 78 of file BirdseedData.h.

6.5.4.10 includeSnpsFilename

```
std::string BirdseedData::includeSnpsFilename [private]
```

Definition at line 77 of file BirdseedData.h.

6.5.4.11 missingValues

```
std::map<std::string, std::vector<unsigned int> > BirdseedData::missingValues [private]
```

subject name -> attribute indices

Definition at line 95 of file BirdseedData.h.

6.5.4.12 phenosFilename

```
std::string BirdseedData::phenosFilename [private]
```

Sample phenotypes Filename containing subject phenotypes.

Definition at line 99 of file BirdseedData.h.

6.5.4.13 phenotypes

```
std::vector<int> BirdseedData::phenotypes [private]
```

vector of phenotypes (case-control)

Definition at line 101 of file BirdseedData.h.

6.5.4.14 snpAlleleCounts

```
std::vector<std::map<char, unsigned int>> BirdseedData::snpAlleleCounts [private]
```

SNP allele->count.

Definition at line 93 of file BirdseedData.h.

6.5.4.15 snpGenotypes

```
std::vector<std::vector<int>> BirdseedData::snpGenotypes [private]
```

SNP genotype calls and their corresponding confidences.

Definition at line 83 of file BirdseedData.h.

6.5.4.16 snpMajorAlleleFreq

```
std::vector<double> BirdseedData::snpMajorAlleleFreq [private]
```

SNP genotypes major allele frequency.

Definition at line 91 of file BirdseedData.h.

6.5.4.17 snpMajorMinorAlleles

```
std::vector<std::pair<char, char>> BirdseedData::snpMajorMinorAlleles [private]
```

SNP genotypes alleles.

Definition at line 89 of file BirdseedData.h.

6.5.4.18 snpNames

```
std::vector<std::string> BirdseedData::snpNames [private]
```

SNP names.

Definition at line 81 of file BirdseedData.h.

6.5.4.19 snpsFilename

```
std::string BirdseedData::snpsFilename [private]
```

Filename containing birdseed-called SNPs.

Definition at line 66 of file BirdseedData.h.

6.5.4.20 subjectLabels

```
std::vector<std::string> BirdseedData::subjectLabels [private]
```

Definition at line 70 of file BirdseedData.h.

6.5.4.21 subjectLabelsFilename

```
std::string BirdseedData::subjectLabelsFilename [private]
```

Filename containing subject names.

Definition at line 69 of file BirdseedData.h.

6.5.4.22 subjectNames

```
std::vector<std::string> BirdseedData::subjectNames [private]
```

Definition at line 72 of file BirdseedData.h.

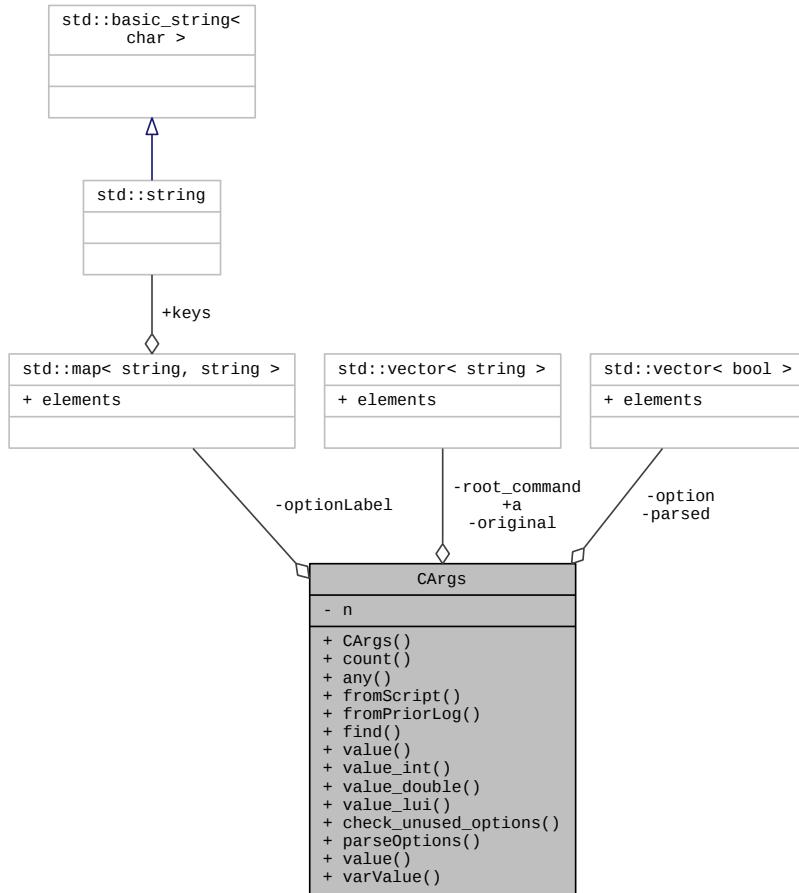
The documentation for this class was generated from the following files:

- src/[BirdseedData.h](#)
- src/[BirdseedData.cpp](#)

6.6 CArgs Class Reference

```
#include <helper.h>
```

Collaboration diagram for CArgs:



Public Member Functions

- `CArgs (int, char **)`
- `int count ()`
- `bool any ()`
- `void fromScript (string)`
- `void fromPriorLog (string)`
- `bool find (string)`
- `string value (string)`
- `int value_int (string)`
- `double value_double (string)`

- long unsigned int [value_lui](#) (string)
- void [check_unused_options](#) (Plink &)
- bool [parseOptions](#) (string, string)
- vector< string > [value](#) (string, int)
- vector< string > [varValue](#) (string)

Public Attributes

- vector< string > [a](#)

Private Attributes

- int [n](#)
- vector< bool > [parsed](#)
- vector< bool > [option](#)
- vector< string > [root_command](#)
- vector< string > [original](#)
- map< string, string > [optionLabel](#)

6.6.1 Detailed Description

Definition at line 56 of file helper.h.

6.6.2 Constructor & Destructor Documentation

6.6.2.1 CArgs()

```
CArgs::CArgs (
    int ,
    char **  )
```

Definition at line 98 of file helper.cpp.

6.6.3 Member Function Documentation

6.6.3.1 any()

```
bool CArgs::any ( ) [inline]
```

Definition at line 64 of file helper.h.

6.6.3.2 check_unused_options()

```
void CArgs::check_unused_options (   
    PLink & P )
```

Definition at line 360 of file helper.cpp.

6.6.3.3 count()

```
int CArgs::count ( ) [inline]
```

Definition at line 61 of file helper.h.

6.6.3.4 find()

```
bool CArgs::find (   
    string s )
```

Definition at line 189 of file helper.cpp.

6.6.3.5 fromPriorLog()

```
void CArgs::fromPriorLog (   
    string f )
```

Definition at line 136 of file helper.cpp.

6.6.3.6 fromScript()

```
void CArgs::fromScript (
    string f )
```

Definition at line 120 of file helper.cpp.

6.6.3.7 parseOptions()

```
bool CArgs::parseOptions (
    string cmd,
    string opt )
```

Definition at line 295 of file helper.cpp.

6.6.3.8 value() [1/2]

```
string CArgs::value (
    string s )
```

Definition at line 198 of file helper.cpp.

6.6.3.9 value() [2/2]

```
vector< string > CArgs::value (
    string s,
    int c )
```

Definition at line 237 of file helper.cpp.

6.6.3.10 value_double()

```
double CArgs::value_double (
    string s )
```

Definition at line 227 of file helper.cpp.

6.6.3.11 value_int()

```
int CArgs::value_int (
    string s )
```

Definition at line 207 of file helper.cpp.

6.6.3.12 value_lui()

```
long unsigned int CArgs::value_lui (
    string s )
```

Definition at line 217 of file helper.cpp.

6.6.3.13 varValue()

```
vector< string > CArgs::varValue (
    string s )
```

Definition at line 255 of file helper.cpp.

6.6.4 Member Data Documentation

6.6.4.1 a

```
vector<string> CArgs::a
```

Definition at line 80 of file helper.h.

6.6.4.2 n

```
int CArgs::n [private]
```

Definition at line 83 of file helper.h.

6.6.4.3 option

```
vector<bool> CArgs::option [private]
```

Definition at line 85 of file helper.h.

6.6.4.4 optionLabel

```
map<string, string> CArgs::optionLabel [private]
```

Definition at line 88 of file helper.h.

6.6.4.5 original

```
vector<string> CArgs::original [private]
```

Definition at line 87 of file helper.h.

6.6.4.6 parsed

```
vector<bool> CArgs::parsed [private]
```

Definition at line 84 of file helper.h.

6.6.4.7 root_command

```
vector<string> CArgs::root_command [private]
```

Definition at line 86 of file helper.h.

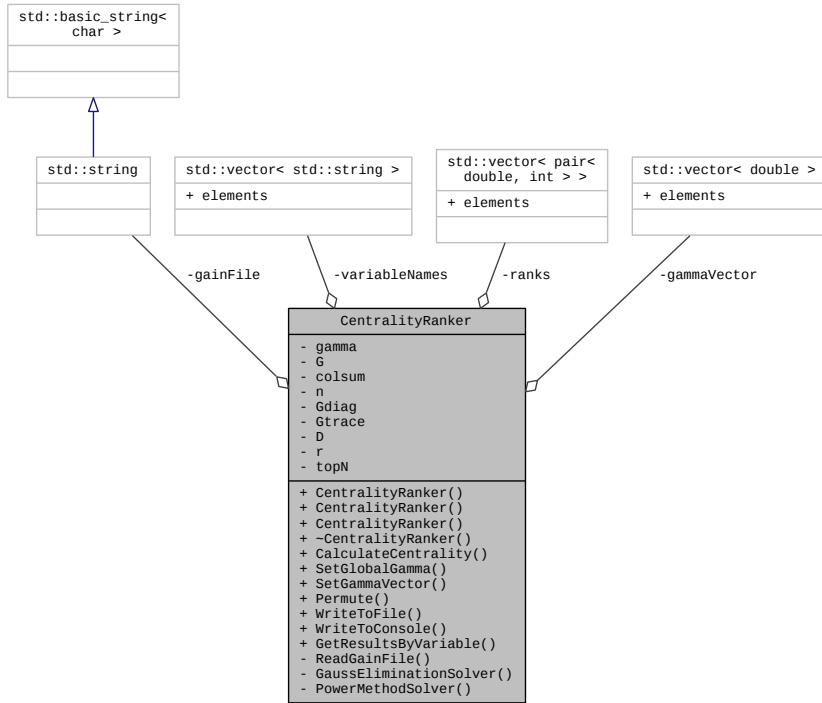
The documentation for this class was generated from the following files:

- src/[helper.h](#)
- src/[helper.cpp](#)

6.7 CentralityRanker Class Reference

```
#include <CentralityRanker.h>
```

Collaboration diagram for CentralityRanker:



Public Member Functions

- `CentralityRanker (std::string gainFileParam, bool isUpperTriangular=false)`
- `CentralityRanker (double **variablesMatrix, unsigned int dim, std::vector< std::string > &variableNames)`
- `CentralityRanker (arma::mat &A, std::vector< std::string > varNames)`
- `virtual ~CentralityRanker ()`
- `bool CalculateCentrality (SolverMethod method)`
- `void SetGlobalGamma (double gammaParam)`
- `bool SetGammaVector (vector_t &gammaVectorValues)`
- `bool Permute (int numPerms)`
- `void WriteToFile (std::string outfile, int topN=-1)`
- `void WriteToConsole (int topN=-1)`
- `arma::vec GetResultsByVariable ()`

Private Member Functions

- `bool ReadGainFile (std::string gainfilename, bool isUpperTriangular=false)`
- `bool GaussEliminationSolver ()`
- `bool PowerMethodSolver ()`

Private Attributes

- std::string `gainFile`
- double `gamma`
- std::vector< double > `gammaVector`
- std::vector< std::string > `variableNames`
- arma::mat `G`
- arma::rowvec `colsum`
- size_t `n`
- arma::mat `Gdiag`
- double `Gtrace`
- arma::mat `D`
- arma::vec `r`
- vector< pair< double, int > > `ranks`
- int `topN`

6.7.1 Detailed Description

Definition at line 18 of file CentralityRanker.h.

6.7.2 Constructor & Destructor Documentation

6.7.2.1 CentralityRanker() [1/3]

```
CentralityRanker::CentralityRanker (
    std::string gainFileParam,
    bool isUpperTriangular = false )
```

Definition at line 29 of file CentralityRanker.cpp.

6.7.2.2 CentralityRanker() [2/3]

```
CentralityRanker::CentralityRanker (
    double ** variablesMatrix,
    unsigned int dim,
    std::vector< std::string > & variableNames )
```

Definition at line 47 of file CentralityRanker.cpp.

6.7.2.3 CentralityRanker() [3/3]

```
CentralityRanker::CentralityRanker (
    arma::mat & A,
    std::vector< std::string > varNames )
```

6.7.2.4 ~CentralityRanker()

```
CentralityRanker::~CentralityRanker ( ) [virtual]
```

Definition at line 91 of file CentralityRanker.cpp.

6.7.3 Member Function Documentation

6.7.3.1 CalculateCentrality()

```
bool CentralityRanker::CalculateCentrality (
    SolverMethod method )
```

Definition at line 94 of file CentralityRanker.cpp.

6.7.3.2 GaussEliminationSolver()

```
bool CentralityRanker::GaussEliminationSolver ( ) [private]
```

Definition at line 292 of file CentralityRanker.cpp.

6.7.3.3 GetResultsByVariable()

```
arma::vec CentralityRanker::GetResultsByVariable ( ) [inline]
```

Definition at line 35 of file CentralityRanker.h.

6.7.3.4 Permute()

```
bool CentralityRanker::Permute (
    int numPerms )
```

Definition at line 153 of file CentralityRanker.cpp.

6.7.3.5 PowerMethodSolver()

```
bool CentralityRanker::PowerMethodSolver ( ) [private]
```

Definition at line 413 of file CentralityRanker.cpp.

6.7.3.6 ReadGainFile()

```
bool CentralityRanker::ReadGainFile (
    std::string gainFilename,
    bool isUpperTriangular = false ) [private]
```

Definition at line 212 of file CentralityRanker.cpp.

6.7.3.7 SetGammaVector()

```
bool CentralityRanker::SetGammaVector (
    vector_t & gammaVectorValues )
```

Definition at line 138 of file CentralityRanker.cpp.

6.7.3.8 SetGlobalGamma()

```
void CentralityRanker::SetGlobalGamma (
    double gammaParam )
```

Definition at line 133 of file CentralityRanker.cpp.

6.7.3.9 WriteToConsole()

```
void CentralityRanker::WriteToConsole (
    int topN = -1 )
```

Definition at line 192 of file CentralityRanker.cpp.

6.7.3.10 WriteToFile()

```
void CentralityRanker::WriteToFile (
    std::string outfile,
    int topN = -1 )
```

Definition at line 165 of file CentralityRanker.cpp.

6.7.4 Member Data Documentation

6.7.4.1 colsum

```
arma::rowvec CentralityRanker::colsum [private]
```

Definition at line 57 of file CentralityRanker.h.

6.7.4.2 D

```
arma::mat CentralityRanker::D [private]
```

Definition at line 61 of file CentralityRanker.h.

6.7.4.3 G

```
arma::mat CentralityRanker::G [private]
```

Definition at line 55 of file CentralityRanker.h.

6.7.4.4 gainFile

```
std::string CentralityRanker::gainFile [private]
```

Definition at line 46 of file CentralityRanker.h.

6.7.4.5 gamma

```
double CentralityRanker::gamma [private]
```

Definition at line 49 of file CentralityRanker.h.

6.7.4.6 gammaVector

```
std::vector<double> CentralityRanker::gammaVector [private]
```

Definition at line 50 of file CentralityRanker.h.

6.7.4.7 Gdiag

```
arma::mat CentralityRanker::Gdiag [private]
```

Definition at line 59 of file CentralityRanker.h.

6.7.4.8 Gtrace

```
double CentralityRanker::Gtrace [private]
```

Definition at line 60 of file CentralityRanker.h.

6.7.4.9 n

```
size_t CentralityRanker::n [private]
```

Definition at line 58 of file CentralityRanker.h.

6.7.4.10 r

```
arma::vec CentralityRanker::r [private]
```

Definition at line 64 of file CentralityRanker.h.

6.7.4.11 ranks

```
vector<pair<double, int> > CentralityRanker::ranks [private]
```

Definition at line 65 of file CentralityRanker.h.

6.7.4.12 topN

```
int CentralityRanker::topN [private]
```

Definition at line 67 of file CentralityRanker.h.

6.7.4.13 variableNames

```
std::vector<std::string> CentralityRanker::variableNames [private]
```

Definition at line 53 of file CentralityRanker.h.

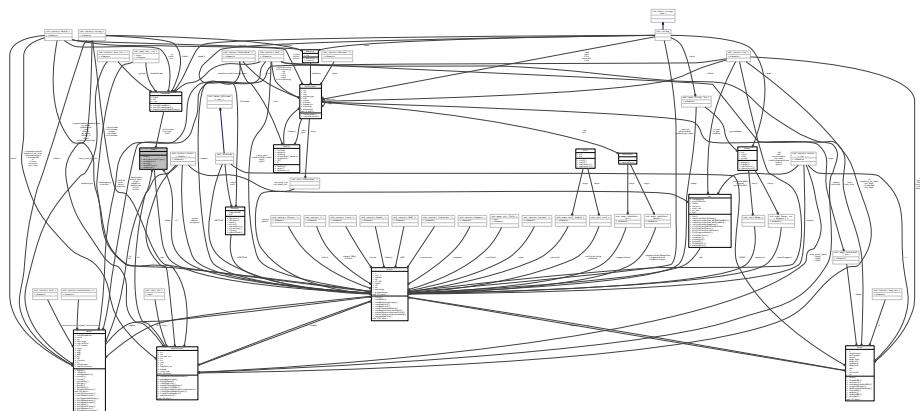
The documentation for this class was generated from the following files:

- src/[CentralityRanker.h](#)
- src/[CentralityRanker.cpp](#)

6.8 Chap Class Reference

```
#include <whap.h>
```

Collaboration diagram for Chap:



Public Member Functions

- `Chap (Plink *p_, HaploPhase *h_)`
- `void determineTestType ()`
- `void setModels (ChapModel &, ChapModel &)`
- `void build (ChapModel &)`
- `void setSNPList (vector< int > &, ChapModel &)`
- `bool isNested ()`

Public Attributes

- `ChapModel * alternate`
- `ChapModel * null`
- `ChapModel * current`
- `Plink * P`
- `HaploPhase * H`

6.8.1 Detailed Description

Definition at line 57 of file whap.h.

6.8.2 Constructor & Destructor Documentation

6.8.2.1 Chap()

```
Chap::Chap (
    Plink * p_,
    HaploPhase * h_ ) [inline]
```

Definition at line 70 of file whap.h.

6.8.3 Member Function Documentation

6.8.3.1 build()

```
void Chap::build (
    ChapModel & model )
```

Definition at line 922 of file whap.cpp.

6.8.3.2 determineTestType()

```
void Chap::determineTestType ( )
```

Definition at line 916 of file whap.cpp.

6.8.3.3 isNested()

```
bool Chap::isNested ( )
```

Definition at line 1350 of file whap.cpp.

6.8.3.4 setModels()

```
void Chap::setModels (
    ChapModel & a,
    ChapModel & n )
```

Definition at line 1344 of file whap.cpp.

6.8.3.5 setSNPList()

```
void Chap::setSNPList (
    vector< int > & snplist,
    ChapModel & model )
```

Definition at line 1385 of file whap.cpp.

6.8.4 Member Data Documentation

6.8.4.1 alternate

```
ChapModel* Chap::alternate
```

Definition at line 62 of file whap.h.

6.8.4.2 current

`ChapModel* Chap::current`

Definition at line 65 of file whap.h.

6.8.4.3 H

`HaploPhase* Chap::H`

Definition at line 68 of file whap.h.

6.8.4.4 null

`ChapModel* Chap::null`

Definition at line 63 of file whap.h.

6.8.4.5 P

`Plink* Chap::P`

Definition at line 67 of file whap.h.

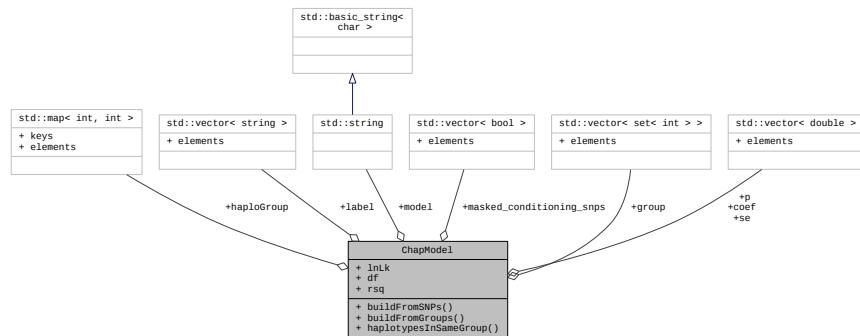
The documentation for this class was generated from the following files:

- src/whap.h
- src/whap.cpp

6.9 ChapModel Class Reference

#include <whap.h>

Collaboration diagram for ChapModel:



Public Member Functions

- void `buildFromSNPs` (string)
- void `buildFromGroups` (string)
- bool `haplotypesInSameGroup` (int, int)

Public Attributes

- string `model`
- map< int, int > `haploGroup`
- vector< bool > `masked_conditioning_snps`
- `vector_t` `coef`
- `vector_t` `se`
- `vector_t` `p`
- vector< string > `label`
- vector< set< int > > `group`
- double `lnLk`
- int `df`
- double `rsq`

6.9.1 Detailed Description

Definition at line 29 of file whap.h.

6.9.2 Member Function Documentation

6.9.2.1 buildFromGroups()

```
void ChapModel::buildFromGroups (
    string )
```

6.9.2.2 buildFromSNPs()

```
void ChapModel::buildFromSNPs (
    string )
```

6.9.2.3 `haplotypesInSameGroup()`

```
bool ChapModel::haplotypesInSameGroup (
    int h1,
    int h2 )
```

Definition at line 1367 of file whap.cpp.

6.9.3 Member Data Documentation

6.9.3.1 `coef`

```
vector_t ChapModel::coef
```

Definition at line 39 of file whap.h.

6.9.3.2 `df`

```
int ChapModel::df
```

Definition at line 47 of file whap.h.

6.9.3.3 `group`

```
vector< set<int> > ChapModel::group
```

Definition at line 44 of file whap.h.

6.9.3.4 `haploGroup`

```
map<int,int> ChapModel::haploGroup
```

Definition at line 36 of file whap.h.

6.9.3.5 label

```
vector<string> ChapModel::label
```

Definition at line 42 of file whap.h.

6.9.3.6 lnLk

```
double ChapModel::lnLk
```

Definition at line 46 of file whap.h.

6.9.3.7 masked_conditioning_snps

```
vector<bool> ChapModel::masked_conditioning_snps
```

Definition at line 37 of file whap.h.

6.9.3.8 model

```
string ChapModel::model
```

Definition at line 34 of file whap.h.

6.9.3.9 p

```
vector_t ChapModel::p
```

Definition at line 41 of file whap.h.

6.9.3.10 rsq

```
double ChapModel::rsq
```

Definition at line 48 of file whap.h.

6.9.3.11 se

```
vector_t ChapModel::se
```

Definition at line 40 of file whap.h.

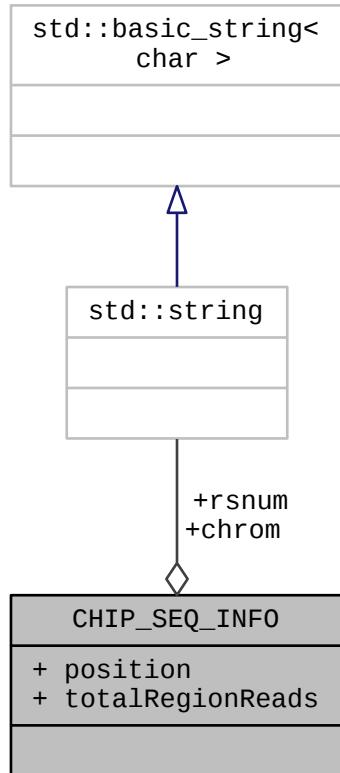
The documentation for this class was generated from the following files:

- src/[whap.h](#)
- src/[whap.cpp](#)

6.10 CHIP_SEQ_INFO Struct Reference

```
#include <DcVar.h>
```

Collaboration diagram for CHIP_SEQ_INFO:



Public Attributes

- std::string [chrom](#)
- uint [position](#)
- std::string [rsnum](#)
- uint [totalRegionReads](#)

6.10.1 Detailed Description

Definition at line 52 of file DcVar.h.

6.10.2 Member Data Documentation

6.10.2.1 chrom

```
std::string CHIP_SEQ_INFO::chrom
```

Definition at line 53 of file DcVar.h.

6.10.2.2 position

```
uint CHIP_SEQ_INFO::position
```

Definition at line 54 of file DcVar.h.

6.10.2.3 rsnum

```
std::string CHIP_SEQ_INFO::rsnum
```

Definition at line 55 of file DcVar.h.

6.10.2.4 totalRegionReads

```
uint CHIP_SEQ_INFO::totalRegionReads
```

Definition at line 56 of file DcVar.h.

The documentation for this struct was generated from the following file:

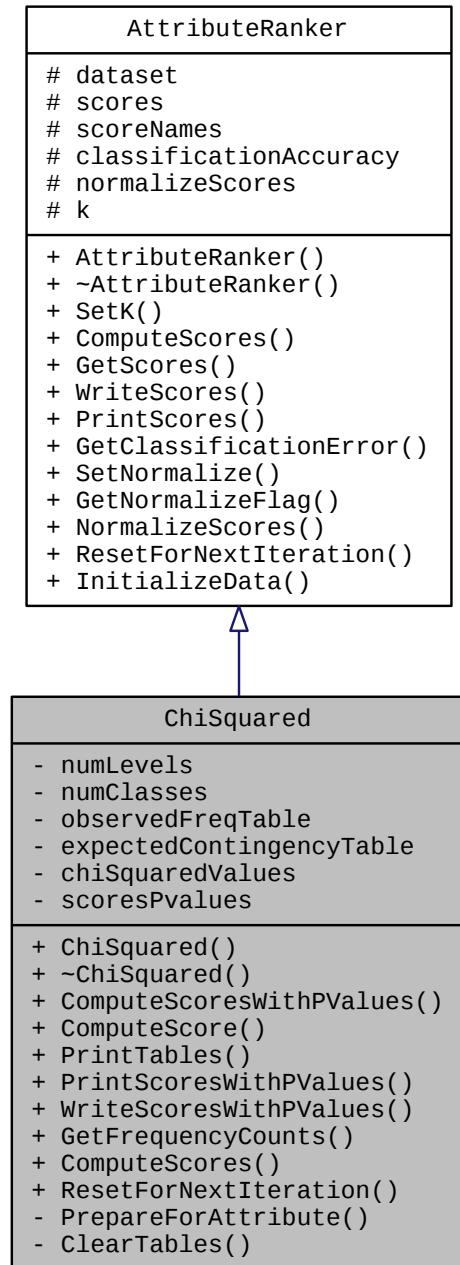
- src/[DcVar.h](#)

6.11 ChiSquared Class Reference

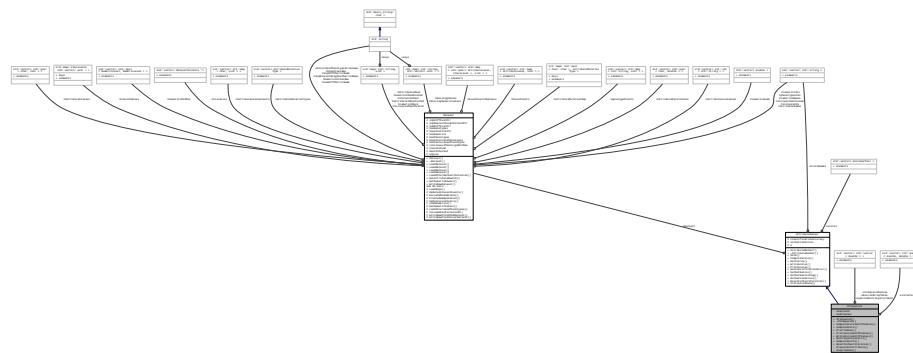
Chi-squared attribute ranking algorithm.

```
#include <ChiSquared.h>
```

Inheritance diagram for ChiSquared:



Collaboration diagram for ChiSquared:



Public Member Functions

- **ChiSquared (Dataset *ds)**
Construct an chi-squared algorithm object.
- **~ChiSquared ()**
- **const std::vector< std::pair< double, double > > & ComputeScoresWithPValues ()**
For each attribute, calculate chi-squared and associated p-value.
- **std::pair< double, double > ComputeScore (unsigned int index)**
For the attribute at the specified index, calculate the chi-squared and associated p-value.
- **void PrintTables ()**
Print calculation tables.
- **void PrintScoresWithPValues (std::ostream &outStream, unsigned int topN=0)**
Print the scores to a stream.
- **void WriteScoresWithPValues (std::string outFilename, unsigned int topN=0)**
Print the scores to a stream.
- **std::vector< std::vector< double > > GetFrequencyCounts ()**
Get the observed frequencies table as a vector of vector of doubles.
- **AttributeScores ComputeScores ()**
Compute the attribute scores for the current set of attributes.
- **bool ResetForNextIteration ()**
Reset the algorithm.

Private Member Functions

- **void PrepareForAttribute (unsigned int attributeIndex)**
Private method to setup the chi-squared contingency tables for a particular attribute.
- **void ClearTables ()**
Clear calculation tables.

Private Attributes

- unsigned int `numLevels`
number of levels in the attributes
- unsigned int `numClasses`
number of classes in the instances
- std::vector< std::vector< double > > `observedFreqTable`
observed frequencies
- std::vector< std::vector< double > > `expectedContingencyTable`
- std::vector< std::vector< double > > `chiSquaredValues`
chi squared computed values
- std::vector< std::pair< double, double > > `scoresPvalues`
chi-squared value, p-value for each attribute

Additional Inherited Members

6.11.1 Detailed Description

Chi-squared attribute ranking algorithm.

[ChiSquared](#) algorithm interface. For performing chi-squared tests of association between an attribute and its class across all instances in a data set.

Author

Bill White

Version

1.0

Contact: bill.c.white@gmail.com Created on: 6/15/05

Modified to implement new [AttributeRanker](#) interface.

Definition at line 28 of file ChiSquared.h.

6.11.2 Constructor & Destructor Documentation

6.11.2.1 ChiSquared()

```
ChiSquared::ChiSquared (
```

```
    Dataset * ds )
```

Construct an chi-squared algorithm object.

Parameters

in	<i>ds</i>	pointer to a Dataset object
----	-----------	---

Definition at line 22 of file ChiSquared.cpp.

6.11.2.2 ~ChiSquared()

```
ChiSquared::~ChiSquared ( )
```

Definition at line 32 of file ChiSquared.cpp.

6.11.3 Member Function Documentation**6.11.3.1 ClearTables()**

```
void ChiSquared::ClearTables ( ) [private]
```

Clear calculation tables.

Definition at line 231 of file ChiSquared.cpp.

6.11.3.2 ComputeScore()

```
pair< double, double > ChiSquared::ComputeScore (  
    unsigned int index )
```

For the attribute at the specified index, calculate the chi-squared and associated p-value.

Return as a pair.

Parameters

in	<i>index</i>	index into the attributes of the data set
----	--------------	---

Returns

pairs of chi-squared score and associated p-value for the attribute

Definition at line 47 of file ChiSquared.cpp.

6.11.3.3 ComputeScores()

```
AttributeScores ChiSquared::ComputeScores () [virtual]
```

Compute the attribute scores for the current set of attributes.

Implements [AttributeRanker](#).

Definition at line 244 of file ChiSquared.cpp.

6.11.3.4 ComputeScoresWithPValues()

```
const vector< pair< double, double > > & ChiSquared::ComputeScoresWithPValues ()
```

For each attribute, calculate chi-squared and associated p-value.

Return in a vector of pairs indexed by attribute index.

Returns

vector of pairs of chi-squared scores and associated p-values

Definition at line 35 of file ChiSquared.cpp.

6.11.3.5 GetFrequencyCounts()

```
std::vector<std::vector<double> > ChiSquared::GetFrequencyCounts () [inline]
```

Get the observed frequencies table as a vector of vector of doubles.

Definition at line 65 of file ChiSquared.h.

6.11.3.6 PrepareForAttribute()

```
void ChiSquared::PrepareForAttribute (
    unsigned int attributeIndex ) [private]
```

Private method to setup the chi-squared contingency tables for a particular attribute.

Parameters

<code>in</code>	<code>attributeIndex</code>	attribute index
-----------------	-----------------------------	-----------------

Definition at line 210 of file ChiSquared.cpp.

6.11.3.7 PrintScoresWithPValues()

```
void ChiSquared::PrintScoresWithPValues (
    std::ofstream & outStream,
    unsigned int topN = 0 )
```

Print the scores to a stream.

Parameters

<code>in</code>	<code>outStream</code>	reference to an output stream
<code>in</code>	<code>topN</code>	top number of attributes to print

Definition at line 177 of file ChiSquared.cpp.

6.11.3.8 PrintTables()

```
void ChiSquared::PrintTables ( )
```

Print calculation tables.

Definition at line 146 of file ChiSquared.cpp.

6.11.3.9 ResetForNextIteration()

```
bool ChiSquared::ResetForNextIteration ( ) [inline], [virtual]
```

Reset the algorithm.

Reimplemented from [AttributeRanker](#).

Definition at line 70 of file ChiSquared.h.

6.11.3.10 WriteScoresWithPValues()

```
void ChiSquared::WriteScoresWithPValues (
    std::string outfilename,
    unsigned int topN = 0 )
```

Print the scores to a stream.

Parameters

in	<i>outFilename</i>	filename to write scores to
in	<i>topN</i>	top number of attributes to print

Definition at line 194 of file ChiSquared.cpp.

6.11.4 Member Data Documentation

6.11.4.1 chiSquaredValues

```
std::vector<std::vector<double>> ChiSquared::chiSquaredValues [private]
```

chi squared computed values

Definition at line 90 of file ChiSquared.h.

6.11.4.2 expectedContingencyTable

```
std::vector<std::vector<double>> ChiSquared::expectedContingencyTable [private]
```

Definition at line 88 of file ChiSquared.h.

6.11.4.3 numClasses

```
unsigned int ChiSquared::numClasses [private]
```

number of classes in the instances

Definition at line 84 of file ChiSquared.h.

6.11.4.4 numLevels

```
unsigned int ChiSquared::numLevels [private]
```

number of levels in the attributes

Definition at line 82 of file ChiSquared.h.

6.11.4.5 observedFreqTable

```
std::vector<std::vector<double> > ChiSquared::observedFreqTable [private]
```

observed frequencies

Definition at line 86 of file ChiSquared.h.

6.11.4.6 scoresPvalues

```
std::vector<std::pair<double, double> > ChiSquared::scoresPvalues [private]
```

chi-squared value, p-value for each attribute

Definition at line 92 of file ChiSquared.h.

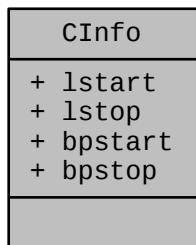
The documentation for this class was generated from the following files:

- src/[ChiSquared.h](#)
- src/[ChiSquared.cpp](#)

6.12 CInfo Class Reference

```
#include <plink.h>
```

Collaboration diagram for CInfo:



Public Attributes

- int `Istart`
- int `Istop`
- int `bpstart`
- int `bpstop`

6.12.1 Detailed Description

Definition at line 503 of file plink.h.

6.12.2 Member Data Documentation

6.12.2.1 `bpstart`

```
int CInfo::bpstart
```

Definition at line 507 of file plink.h.

6.12.2.2 `bpstop`

```
int CInfo::bpstop
```

Definition at line 508 of file plink.h.

6.12.2.3 `Istart`

```
int CInfo::lstart
```

Definition at line 505 of file plink.h.

6.12.2.4 lstop

```
int CInfo::lstop
```

Definition at line 506 of file plink.h.

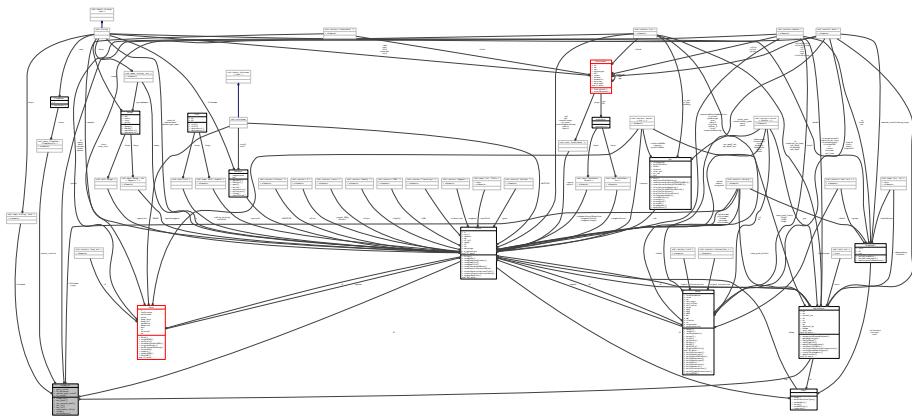
The documentation for this class was generated from the following file:

- [src/plink.h](#)

6.13 clump_LD Class Reference

```
#include <clumpld.h>
```

Collaboration diagram for clump_LD:



Public Member Functions

- `clump_LD (Plink *, HaploPhase *, double, double, double, float)`
- `void set_pval (double)`
- `void set_second_pval (double)`
- `void set_id (double)`
- `void set_r2 (double)`
- `vector< ResultTrio > read_assoc_file (string)`
- `void clump ()`
- `string allelePairs (int, int)`

Public Attributes

- `Plink * P`
- `HaploPhase * hp`
- double `pval_cutoff`
- double `ld_distance`
- double `second_pval_cutoff`
- float `r2_cutoff`
- map< string, bool > `clumped`
- vector< string > `snps`
- vector< double > `pvals`
- map< `ClumpPair`, `ClumpResults` > `assoc_results`
- vector< string > `filename`

6.13.1 Detailed Description

Definition at line 66 of file `clumpld.h`.

6.13.2 Constructor & Destructor Documentation

6.13.2.1 `clump_LD()`

```
clump_LD::clump_LD (
    Plink * pp,
    HaploPhase * hp_,
    double sig,
    double dist,
    double secondp,
    float r2c )
```

Definition at line 29 of file `clumpld.cpp`.

6.13.3 Member Function Documentation

6.13.3.1 `allelePairs()`

```
string clump_LD::allelePairs (
    int l1,
    int l2 )
```

Definition at line 1181 of file `clumpld.cpp`.

6.13.3.2 clump()

```
void clump_LD::clump ( )
```

Definition at line 227 of file clumpld.cpp.

6.13.3.3 read_assoc_file()

```
vector< ResultTrio > clump_LD::read_assoc_file (
    string fileList )
```

Definition at line 75 of file clumpld.cpp.

6.13.3.4 set_ld()

```
void clump_LD::set_ld (
    double dist )
```

Definition at line 62 of file clumpld.cpp.

6.13.3.5 set_pval()

```
void clump_LD::set_pval (
    double sig )
```

Definition at line 54 of file clumpld.cpp.

6.13.3.6 set_r2()

```
void clump_LD::set_r2 (
    double r2c )
```

Definition at line 66 of file clumpld.cpp.

6.13.3.7 set_second_pval()

```
void clump_LD::set_second_pval (
    double secondp )
```

Definition at line 58 of file clumpld.cpp.

6.13.4 Member Data Documentation

6.13.4.1 assoc_results

```
map<ClumpPair, ClumpResults> clump_LD::assoc_results
```

Definition at line 83 of file clumpld.h.

6.13.4.2 clumped

```
map<string, bool> clump_LD::clumped
```

Definition at line 80 of file clumpld.h.

6.13.4.3 filename

```
vector<string> clump_LD::filename
```

Definition at line 85 of file clumpld.h.

6.13.4.4 hp

```
HaploPhase* clump_LD::hp
```

Definition at line 71 of file clumpld.h.

6.13.4.5 ld_distance

```
double clump_LD::ld_distance
```

Definition at line 76 of file clumpld.h.

6.13.4.6 P

```
Plink* clump_LD::P
```

Definition at line 70 of file clumpld.h.

6.13.4.7 pval_cutoff

```
double clump_LD::pval_cutoff
```

Definition at line 75 of file clumpld.h.

6.13.4.8 pvals

```
vector<double> clump_LD::pvals
```

Definition at line 82 of file clumpld.h.

6.13.4.9 r2_cutoff

```
float clump_LD::r2_cutoff
```

Definition at line 78 of file clumpld.h.

6.13.4.10 second_pval_cutoff

```
double clump_LD::second_pval_cutoff
```

Definition at line 77 of file clumpld.h.

6.13.4.11 snps

```
vector<string> clump_LD::snps
```

Definition at line 81 of file clumpld.h.

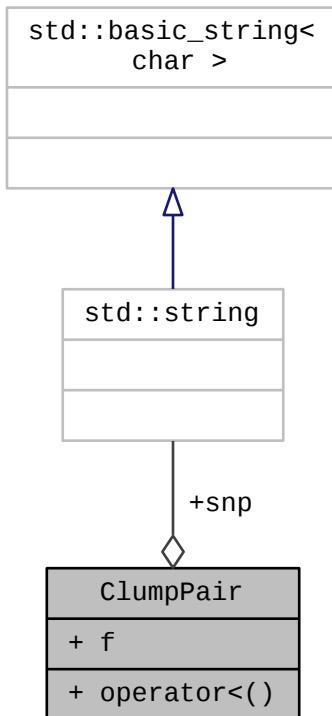
The documentation for this class was generated from the following files:

- src/[clumpld.h](#)
- src/[clumpld.cpp](#)

6.14 ClumpPair Class Reference

```
#include <clumpld.h>
```

Collaboration diagram for ClumpPair:



Public Member Functions

- bool [operator<](#) (const ClumpPair &p2) const

Public Attributes

- string [snp](#)
- int [f](#)

6.14.1 Detailed Description

Definition at line 42 of file [clumpld.h](#).

6.14.2 Member Function Documentation

6.14.2.1 operator<()

```
bool ClumpPair::operator< (
    const ClumpPair & p2 ) const [inline]
```

Definition at line 47 of file [clumpld.h](#).

6.14.3 Member Data Documentation

6.14.3.1 f

```
int ClumpPair::f
```

Definition at line 45 of file [clumpld.h](#).

6.14.3.2 snp

```
string ClumpPair::snp
```

Definition at line 44 of file [clumpld.h](#).

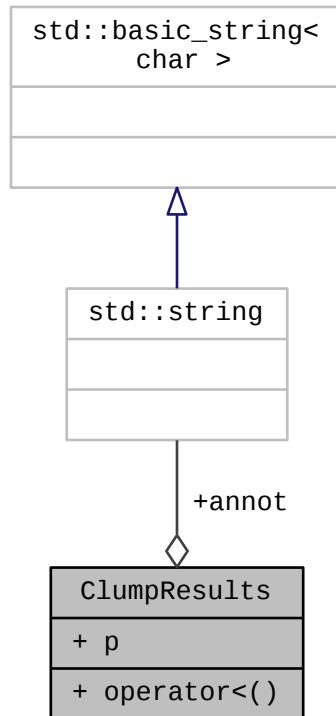
The documentation for this class was generated from the following file:

- src/[clumpld.h](#)

6.15 ClumpResults Class Reference

```
#include <clumpld.h>
```

Collaboration diagram for ClumpResults:



Public Member Functions

- `bool operator< (const ClumpResults &p2) const`

Public Attributes

- `double p`
- `string annot`

6.15.1 Detailed Description

Definition at line 55 of file `clumpld.h`.

6.15.2 Member Function Documentation

6.15.2.1 operator<()

```
bool ClumpResults::operator< (
    const ClumpResults & p2 ) const [inline]
```

Definition at line 60 of file clumpld.h.

6.15.3 Member Data Documentation

6.15.3.1 annot

```
string ClumpResults::annot
```

Definition at line 58 of file clumpld.h.

6.15.3.2 p

```
double ClumpResults::p
```

Definition at line 57 of file clumpld.h.

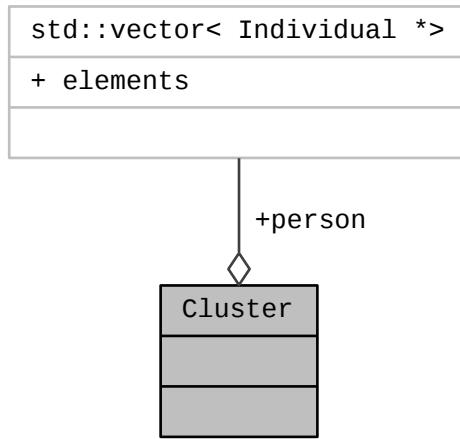
The documentation for this class was generated from the following file:

- src/[clumpld.h](#)

6.16 Cluster Class Reference

```
#include <plink.h>
```

Collaboration diagram for Cluster:



Public Attributes

- `vector< Individual * > person`

6.16.1 Detailed Description

Definition at line 331 of file plink.h.

6.16.2 Member Data Documentation

6.16.2.1 person

```
vector<Individual*> Cluster::person
```

Definition at line 333 of file plink.h.

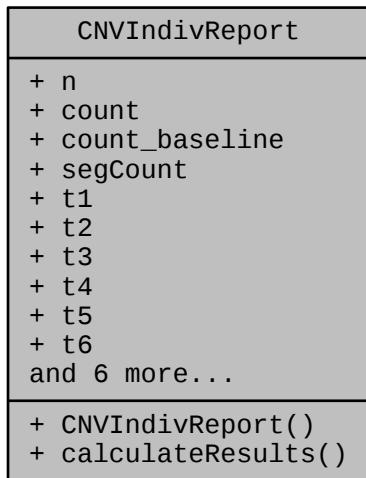
The documentation for this class was generated from the following file:

- `src/plink.h`

6.17 CNVIndivReport Class Reference

```
#include <cnv.h>
```

Collaboration diagram for CNVIndivReport:



Public Member Functions

- `CNVIndivReport ()`
- void `calculateResults ()`

Public Attributes

- int `n`
- int `count`
- int `count_baseline`
- int `segCount`
- double `t1`
- double `t2`
- double `t3`
- double `t4`
- double `t5`
- double `t6`
- double `t7`
- double `t8`
- double `t9`
- double `t10`
- double `t11`
- double `t12`

6.17.1 Detailed Description

Definition at line 17 of file cnv.h.

6.17.2 Constructor & Destructor Documentation

6.17.2.1 CNVIndivReport()

```
CNVIndivReport::CNVIndivReport ( ) [inline]
```

Definition at line 20 of file cnv.h.

6.17.3 Member Function Documentation

6.17.3.1 calculateResults()

```
void CNVIndivReport::calculateResults ( )
```

Definition at line 74 of file cnv.cpp.

6.17.4 Member Data Documentation

6.17.4.1 count

```
int CNVIndivReport::count
```

Definition at line 33 of file cnv.h.

6.17.4.2 count_baseline

```
int CNVIndivReport::count_baseline
```

Definition at line 36 of file cnv.h.

6.17.4.3 n

```
int CNVIndivReport::n
```

Definition at line 30 of file cnv.h.

6.17.4.4 segCount

```
int CNVIndivReport::segCount
```

Definition at line 39 of file cnv.h.

6.17.4.5 t1

```
double CNVIndivReport::t1
```

Definition at line 45 of file cnv.h.

6.17.4.6 t10

```
double CNVIndivReport::t10
```

Definition at line 56 of file cnv.h.

6.17.4.7 t11

```
double CNVIndivReport::t11
```

Definition at line 57 of file cnv.h.

6.17.4.8 t12

```
double CNVIndivReport::t12
```

Definition at line 58 of file cnv.h.

6.17.4.9 t2

```
double CNVIndivReport::t2
```

Definition at line 46 of file cnv.h.

6.17.4.10 t3

```
double CNVIndivReport::t3
```

Definition at line 47 of file cnv.h.

6.17.4.11 t4

```
double CNVIndivReport::t4
```

Definition at line 48 of file cnv.h.

6.17.4.12 t5

```
double CNVIndivReport::t5
```

Definition at line 49 of file cnv.h.

6.17.4.13 t6

```
double CNVIndivReport::t6
```

Definition at line 50 of file cnv.h.

6.17.4.14 t7

```
double CNVIndivReport::t7
```

Definition at line 51 of file cnv.h.

6.17.4.15 t8

```
double CNVIndivReport::t8
```

Definition at line 52 of file cnv.h.

6.17.4.16 t9

```
double CNVIndivReport::t9
```

Definition at line 55 of file cnv.h.

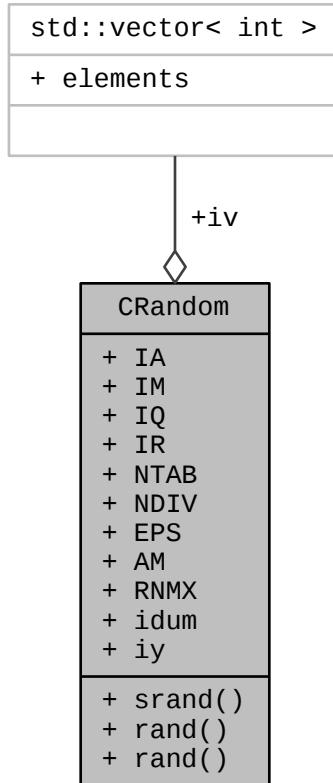
The documentation for this class was generated from the following files:

- src/cnv.h
- src/cnv.cpp

6.18 CRandom Class Reference

```
#include <crandom.h>
```

Collaboration diagram for CRandom:



Static Public Member Functions

- static void **srand** (long unsigned iseed=0)
- static double **rand** ()
- static int **rand** (int)

Static Public Attributes

- static const int **IA** =16807
- static const int **IM** =2147483647
- static const int **IQ** =127773
- static const int **IR** =2836
- static const int **NTAB** =32
- static const int **NDIV** =(1+(IM-1)/NTAB)
- static const double **EPS** =3.0e-16

- static const double **AM** =1.0/**IM**
- static const double **RNMX** =(1.0-**EPS**)
- static int **idum** =0
- static int **iy** =0
- static vector< int > **iv**

6.18.1 Detailed Description

Definition at line 21 of file crandom.h.

6.18.2 Member Function Documentation

6.18.2.1 **rand()** [1/2]

```
double CRandom::rand ( )  [static]
```

Definition at line 63 of file crandom.cpp.

6.18.2.2 **rand()** [2/2]

```
int CRandom::rand (
    int n )  [static]
```

Definition at line 79 of file crandom.cpp.

6.18.2.3 **srand()**

```
void CRandom::srand (
    long unsigned iseed = 0 )  [static]
```

Definition at line 40 of file crandom.cpp.

6.18.3 Member Data Documentation

6.18.3.1 AM

```
const double CRandom::AM =1.0/IM [static]
```

Definition at line 33 of file crandom.h.

6.18.3.2 EPS

```
const double CRandom::EPS =3.0e-16 [static]
```

Definition at line 32 of file crandom.h.

6.18.3.3 IA

```
const int CRandom::IA =16807 [static]
```

Definition at line 25 of file crandom.h.

6.18.3.4 idum

```
int CRandom::idum =0 [static]
```

Definition at line 37 of file crandom.h.

6.18.3.5 IM

```
const int CRandom::IM =2147483647 [static]
```

Definition at line 26 of file crandom.h.

6.18.3.6 IQ

```
const int CRandom::IQ =127773 [static]
```

Definition at line 27 of file crandom.h.

6.18.3.7 IR

```
const int CRandom::IR =2836 [static]
```

Definition at line 28 of file crandom.h.

6.18.3.8 iv

```
vector< int > CRandom::iv [static]
```

Definition at line 40 of file crandom.h.

6.18.3.9 iy

```
int CRandom::iy =0 [static]
```

Definition at line 39 of file crandom.h.

6.18.3.10 NDIV

```
const int CRandom::NDIV =(1+(IM-1)/NTAB) [static]
```

Definition at line 30 of file crandom.h.

6.18.3.11 NTAB

```
const int CRandom::NTAB =32 [static]
```

Definition at line 29 of file crandom.h.

6.18.3.12 RNMX

```
const double CRandom::RNMX =(1.0-EPS) [static]
```

Definition at line 34 of file crandom.h.

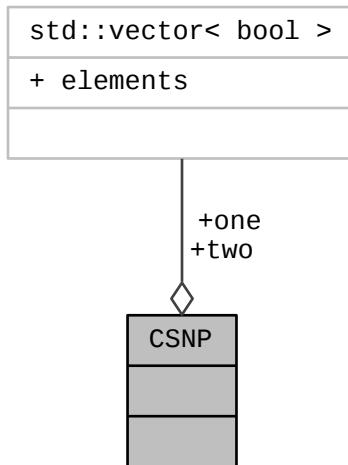
The documentation for this class was generated from the following files:

- [src/crandom.h](#)
- [src/crandom.cpp](#)

6.19 CSNP Class Reference

```
#include <plink.h>
```

Collaboration diagram for CSNP:



Public Attributes

- `vector< bool > one`
- `vector< bool > two`

6.19.1 Detailed Description

Definition at line 325 of file plink.h.

6.19.2 Member Data Documentation

6.19.2.1 one

```
vector<bool> CSNP::one
```

Definition at line 327 of file plink.h.

6.19.2.2 two

```
vector<bool> CSNP::two
```

Definition at line 328 of file plink.h.

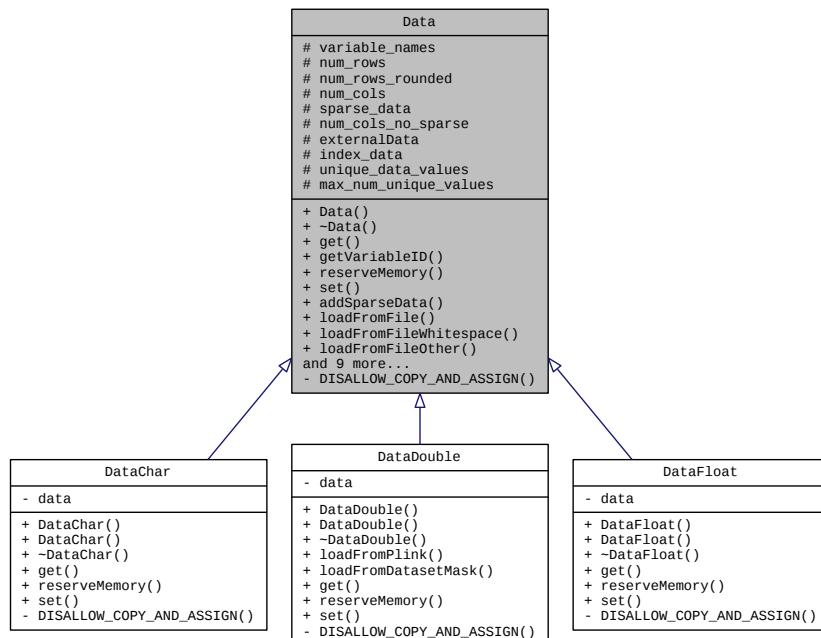
The documentation for this class was generated from the following file:

- [src/plink.h](#)

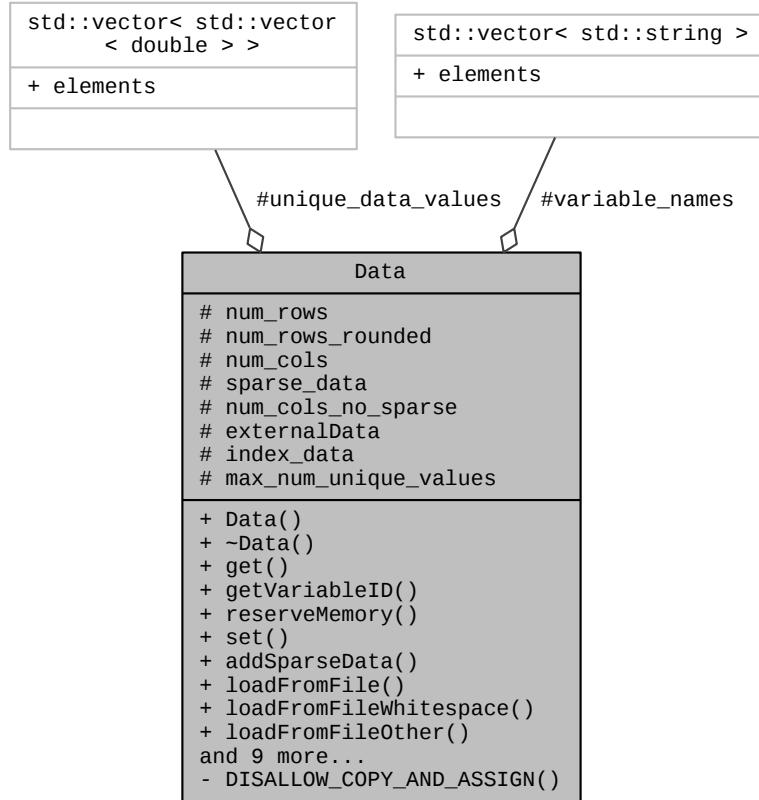
6.20 Data Class Reference

```
#include <Data.h>
```

Inheritance diagram for Data:



Collaboration diagram for Data:



Public Member Functions

- [Data \(\)](#)
- [virtual ~Data \(\)](#)
- [virtual double get \(size_t row, size_t col\) const =0](#)
- [size_t getVariableID \(std::string variable_name\)](#)
- [virtual void reserveMemory \(\)=0](#)
- [virtual void set \(size_t col, size_t row, double value, bool &error\)=0](#)
- [void addSparseData \(unsigned char *sparse_data, size_t num_cols_sparse\)](#)
- [bool loadFromFile \(std::string filename\)](#)
- [bool loadFromFileWhitespace \(std::ifstream &input_file, std::string header_line\)](#)
- [bool loadFromFileOther \(std::ifstream &input_file, std::string header_line, char seperator\)](#)
- [void getAllValues \(std::vector< double > &all_values, std::vector< size_t > &sampleIDs, size_t varID\)](#)
- [size_t getIndex \(size_t row, size_t col\) const](#)
- [double getUniqueDataValue \(size_t varID, size_t index\) const](#)
- [size_t getNumUniqueDataValues \(size_t varID\) const](#)
- [void sort \(\)](#)

- const std::vector< std::string > & [getVariableNames \(\) const](#)
- size_t [getNumCols \(\) const](#)
- size_t [getNumRows \(\) const](#)
- size_t [getMaxNumUniqueValues \(\) const](#)

Protected Attributes

- std::vector< std::string > [variable_names](#)
- size_t [num_rows](#)
- size_t [num_rows_rounded](#)
- size_t [num_cols](#)
- unsigned char * [sparse_data](#)
- size_t [num_cols_no_sparse](#)
- bool [externalData](#)
- size_t * [index_data](#)
- std::vector< std::vector< double > > [unique_data_values](#)
- size_t [max_num_unique_values](#)

Private Member Functions

- [DISALLOW_COPY_AND_ASSIGN \(Data\)](#)

6.20.1 Detailed Description

Definition at line 38 of file Data.h.

6.20.2 Constructor & Destructor Documentation

6.20.2.1 Data()

Data::Data ()

Definition at line 39 of file Data.cpp.

6.20.2.2 ~Data()

Data::~Data () [virtual]

Definition at line 44 of file Data.cpp.

6.20.3 Member Function Documentation

6.20.3.1 addSparseData()

```
void Data::addSparseData (
    unsigned char * sparse_data,
    size_t num_cols_sparse )
```

Definition at line 58 of file Data.cpp.

6.20.3.2 DISALLOW_COPY_AND_ASSIGN()

```
Data::DISALLOW_COPY_AND_ASSIGN (
    Data ) [private]
```

6.20.3.3 get()

```
virtual double Data::get (
    size_t row,
    size_t col ) const [pure virtual]
```

Implemented in [DataDouble](#), [DataChar](#), and [DataFloat](#).

6.20.3.4 getAllValues()

```
void Data::getAllValues (
    std::vector< double > & all_values,
    std::vector< size_t > & sampleIDs,
    size_t varID )
```

Definition at line 171 of file Data.cpp.

6.20.3.5 getIndex()

```
size_t Data::getIndex (
    size_t row,
    size_t col ) const [inline]
```

Definition at line 58 of file Data.h.

6.20.3.6 getMaxNumUniqueValues()

```
size_t Data::getMaxNumUniqueValues ( ) const [inline]
```

Definition at line 105 of file Data.h.

6.20.3.7 getNumCols()

```
size_t Data::getNumCols ( ) const [inline]
```

Definition at line 98 of file Data.h.

6.20.3.8 getNumRows()

```
size_t Data::getNumRows ( ) const [inline]
```

Definition at line 101 of file Data.h.

6.20.3.9 getNumUniqueDataValues()

```
size_t Data::getNumUniqueDataValues (
    size_t varID ) const [inline]
```

Definition at line 84 of file Data.h.

6.20.3.10 getUniqueDataValue()

```
double Data::getUniqueDataValue (
    size_t varID,
    size_t index ) const [inline]
```

Definition at line 75 of file Data.h.

6.20.3.11 getVariableID()

```
size_t Data::getVariableID (
    std::string variable_name )
```

Definition at line 50 of file Data.cpp.

6.20.3.12 getVariableNames()

```
const std::vector<std::string>& Data::getVariableNames () const [inline]
```

Definition at line 95 of file Data.h.

6.20.3.13 loadFromFile()

```
bool Data::loadFromFile (
    std::string filename )
```

Definition at line 64 of file Data.cpp.

6.20.3.14 loadFromFileOther()

```
bool Data::loadFromFileOther (
    std::ifstream & input_file,
    std::string header_line,
    char separator )
```

Definition at line 138 of file Data.cpp.

6.20.3.15 loadFromFileWhitespace()

```
bool Data::loadFromFileWhitespace (
    std::ifstream & input_file,
    std::string header_line )
```

Definition at line 103 of file Data.cpp.

6.20.3.16 reserveMemory()

```
virtual void Data::reserveMemory ( ) [pure virtual]
```

Implemented in [DataDouble](#), [DataChar](#), and [DataFloat](#).

6.20.3.17 set()

```
virtual void Data::set (
    size_t col,
    size_t row,
    double value,
    bool & error ) [pure virtual]
```

Implemented in [DataDouble](#), [DataChar](#), and [DataFloat](#).

6.20.3.18 sort()

```
void Data::sort ( )
```

Definition at line 189 of file Data.cpp.

6.20.4 Member Data Documentation

6.20.4.1 externalData

```
bool Data::externalData [protected]
```

Definition at line 124 of file Data.h.

6.20.4.2 index_data

```
size_t* Data::index_data [protected]
```

Definition at line 126 of file Data.h.

6.20.4.3 max_num_unique_values

```
size_t Data::max_num_unique_values [protected]
```

Definition at line 128 of file Data.h.

6.20.4.4 num_cols

```
size_t Data::num_cols [protected]
```

Definition at line 119 of file Data.h.

6.20.4.5 num_cols_no_sparse

```
size_t Data::num_cols_no_sparse [protected]
```

Definition at line 122 of file Data.h.

6.20.4.6 num_rows

```
size_t Data::num_rows [protected]
```

Definition at line 117 of file Data.h.

6.20.4.7 num_rows_rounded

```
size_t Data::num_rows_rounded [protected]
```

Definition at line 118 of file Data.h.

6.20.4.8 sparse_data

```
unsigned char* Data::sparse_data [protected]
```

Definition at line 121 of file Data.h.

6.20.4.9 unique_data_values

```
std::vector<std::vector<double> > Data::unique_data_values [protected]
```

Definition at line 127 of file Data.h.

6.20.4.10 variable_names

```
std::vector<std::string> Data::variable_names [protected]
```

Definition at line 116 of file Data.h.

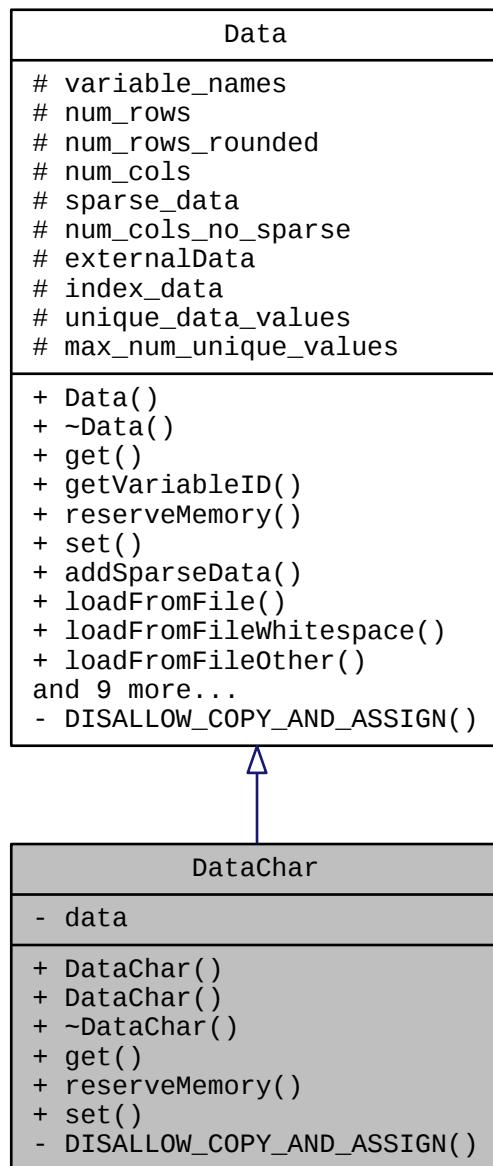
The documentation for this class was generated from the following files:

- src/[Data.h](#)
- src/[Data.cpp](#)

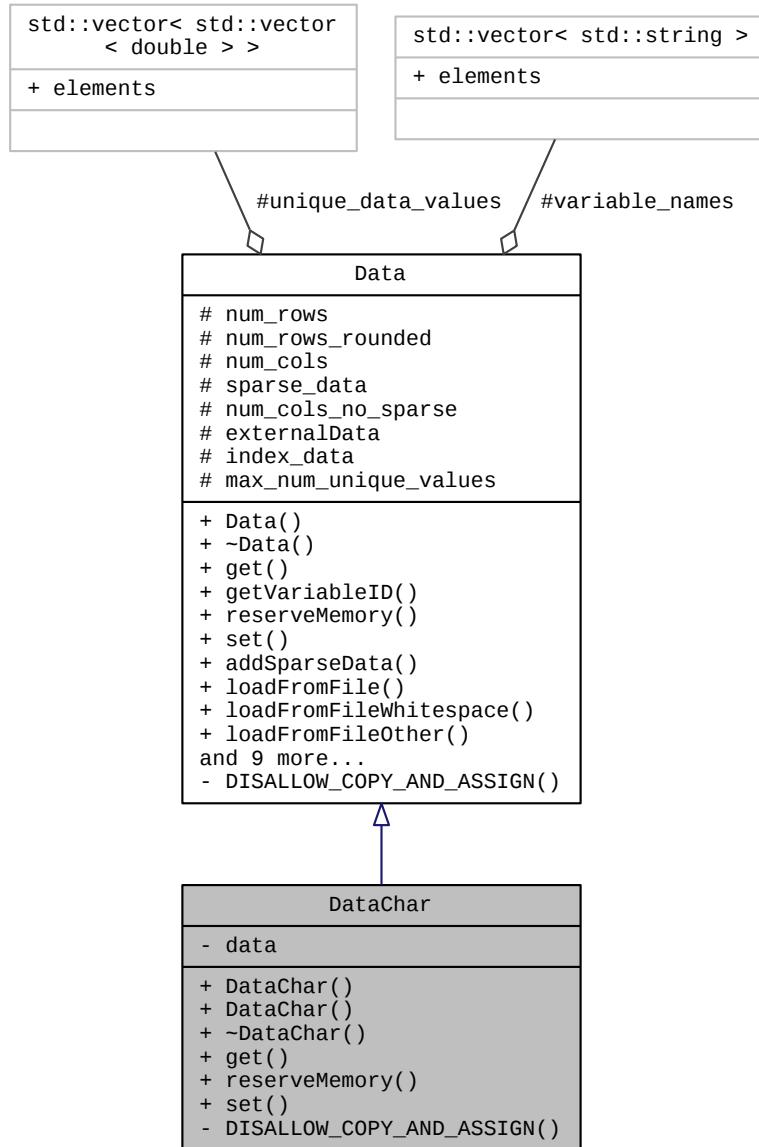
6.21 DataChar Class Reference

```
#include <DataChar.h>
```

Inheritance diagram for DataChar:



Collaboration diagram for DataChar:



Public Member Functions

- `DataChar ()`
- `DataChar (double *data_double, std::vector< std::string > variable_names, size_t num_rows, size_t num_cols, bool &error)`
- `virtual ~DataChar ()`
- `double get (size_t row, size_t col) const override`
- `void reserveMemory () override`
- `void set (size_t col, size_t row, double value, bool &error) override`

Private Member Functions

- [DISALLOW_COPY_AND_ASSIGN \(DataChar\)](#)

Private Attributes

- `char * data`

Additional Inherited Members

6.21.1 Detailed Description

Definition at line 37 of file DataChar.h.

6.21.2 Constructor & Destructor Documentation

6.21.2.1 DataChar() [1/2]

```
DataChar::DataChar ( )
```

Definition at line 36 of file DataChar.cpp.

6.21.2.2 DataChar() [2/2]

```
DataChar::DataChar (
    double * data_double,
    std::vector< std::string > variable_names,
    size_t num_rows,
    size_t num_cols,
    bool & error )
```

Definition at line 40 of file DataChar.cpp.

6.21.2.3 ~DataChar()

```
DataChar::~DataChar ( ) [virtual]
```

Definition at line 64 of file DataChar.cpp.

6.21.3 Member Function Documentation

6.21.3.1 DISALLOW_COPY_AND_ASSIGN()

```
DataChar::DISALLOW_COPY_AND_ASSIGN (   
    DataChar ) [private]
```

6.21.3.2 get()

```
double DataChar::get (   
    size_t row,   
    size_t col ) const [inline], [override], [virtual]
```

Implements [Data](#).

Definition at line 43 of file DataChar.h.

6.21.3.3 reserveMemory()

```
void DataChar::reserveMemory ( ) [inline], [override], [virtual]
```

Implements [Data](#).

Definition at line 53 of file DataChar.h.

6.21.3.4 set()

```
void DataChar::set (   
    size_t col,   
    size_t row,   
    double value,   
    bool & error ) [inline], [override], [virtual]
```

Implements [Data](#).

Definition at line 57 of file DataChar.h.

6.21.4 Member Data Documentation

6.21.4.1 data

```
char* DataChar::data [private]
```

Definition at line 68 of file DataChar.h.

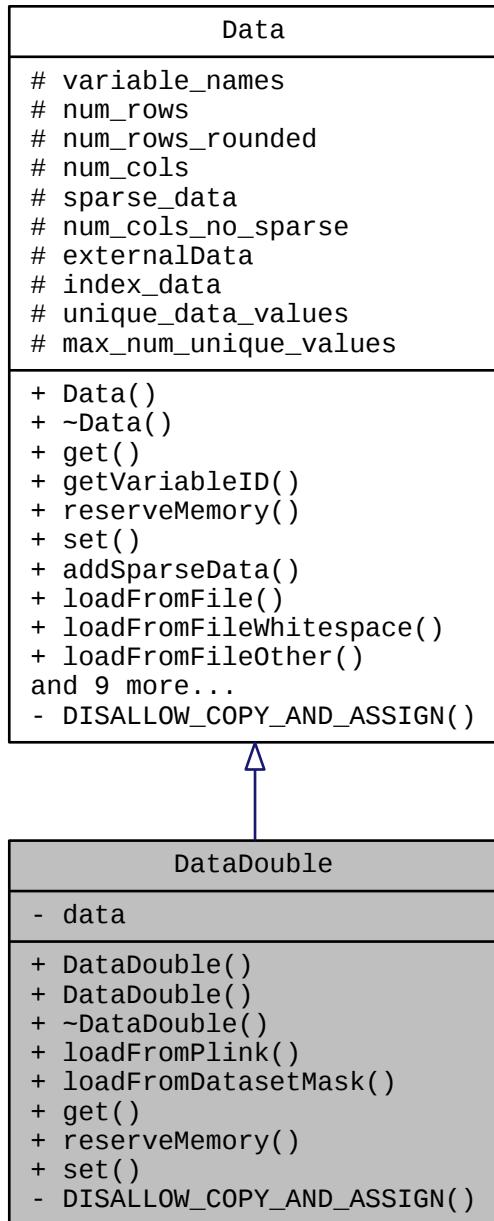
The documentation for this class was generated from the following files:

- src/[DataChar.h](#)
- src/[DataChar.cpp](#)

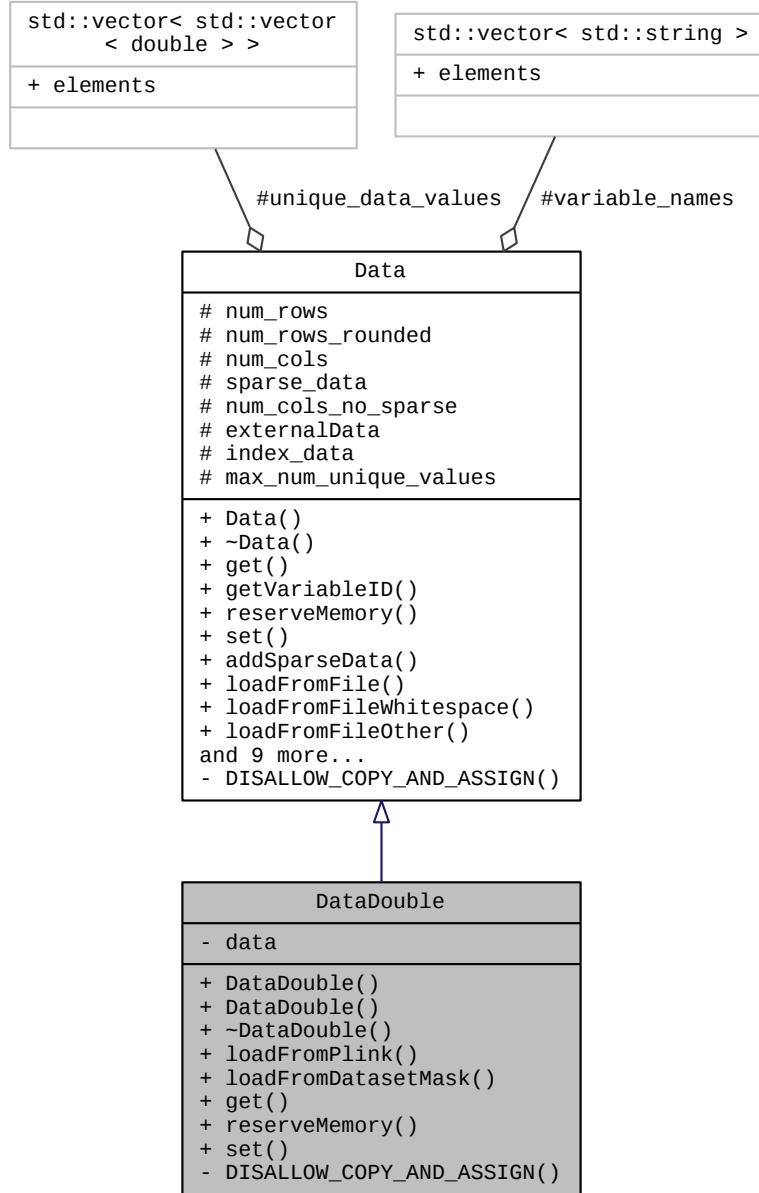
6.22 DataDouble Class Reference

```
#include <DataDouble.h>
```

Inheritance diagram for DataDouble:



Collaboration diagram for DataDouble:



Public Member Functions

- [DataDouble \(\)](#)
- [DataDouble \(double *data, std::vector< std::string > variable_names, size_t num_rows, size_t num_cols\)](#)
- [virtual ~DataDouble \(\)](#)
- [bool loadFromPlink \(Plink *PP\)](#)

- bool `loadFromDatasetMask (Dataset *ds, std::vector< std::string > bestAttributes)`
- double `get (size_t row, size_t col) const override`
- void `reserveMemory () override`
- void `set (size_t col, size_t row, double value, bool &error) override`

Private Member Functions

- `DISALLOW_COPY_AND_ASSIGN (DataDouble)`

Private Attributes

- double * `data`

Additional Inherited Members

6.22.1 Detailed Description

Definition at line 39 of file DataDouble.h.

6.22.2 Constructor & Destructor Documentation

6.22.2.1 DataDouble() [1/2]

`DataDouble::DataDouble ()`

Definition at line 41 of file DataDouble.cpp.

6.22.2.2 DataDouble() [2/2]

```
DataDouble::DataDouble (
    double * data,
    std::vector< std::string > variable_names,
    size_t num_rows,
    size_t num_cols ) [inline]
```

Definition at line 42 of file DataDouble.h.

6.22.2.3 ~DataDouble()

```
DataDouble::~DataDouble ( ) [virtual]
```

Definition at line 45 of file DataDouble.cpp.

6.22.3 Member Function Documentation

6.22.3.1 DISALLOW_COPY_AND_ASSIGN()

```
DataDouble::DISALLOW_COPY_AND_ASSIGN ( DataDouble ) [private]
```

6.22.3.2 get()

```
double DataDouble::get ( size_t row, size_t col ) const [inline], [override], [virtual]
```

Implements [Data](#).

Definition at line 56 of file DataDouble.h.

6.22.3.3 loadFromDatasetMask()

```
bool DataDouble::loadFromDatasetMask ( Dataset * ds, std::vector< std::string > bestAttributes )
```

Definition at line 96 of file DataDouble.cpp.

6.22.3.4 loadFromPlink()

```
bool DataDouble::loadFromPlink ( Plink * PP )
```

Definition at line 51 of file DataDouble.cpp.

6.22.3.5 reserveMemory()

```
void DataDouble::reserveMemory ( ) [inline], [override], [virtual]
```

Implements [Data](#).

Definition at line 67 of file [DataDouble.h](#).

6.22.3.6 set()

```
void DataDouble::set (
    size_t col,
    size_t row,
    double value,
    bool & error ) [inline], [override], [virtual]
```

Implements [Data](#).

Definition at line 71 of file [DataDouble.h](#).

6.22.4 Member Data Documentation

6.22.4.1 data

```
double* DataDouble::data [private]
```

Definition at line 76 of file [DataDouble.h](#).

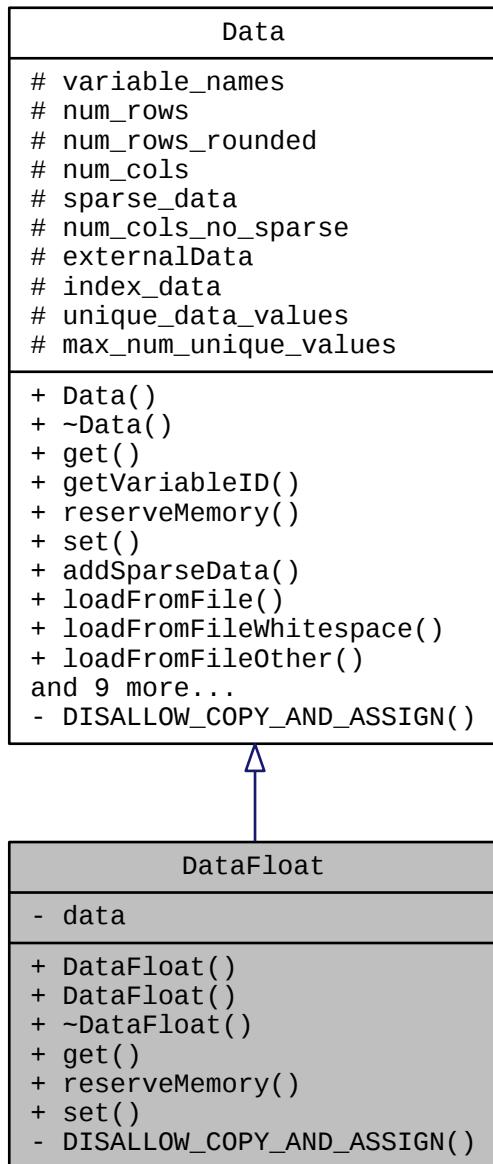
The documentation for this class was generated from the following files:

- src/[DataDouble.h](#)
- src/[DataDouble.cpp](#)

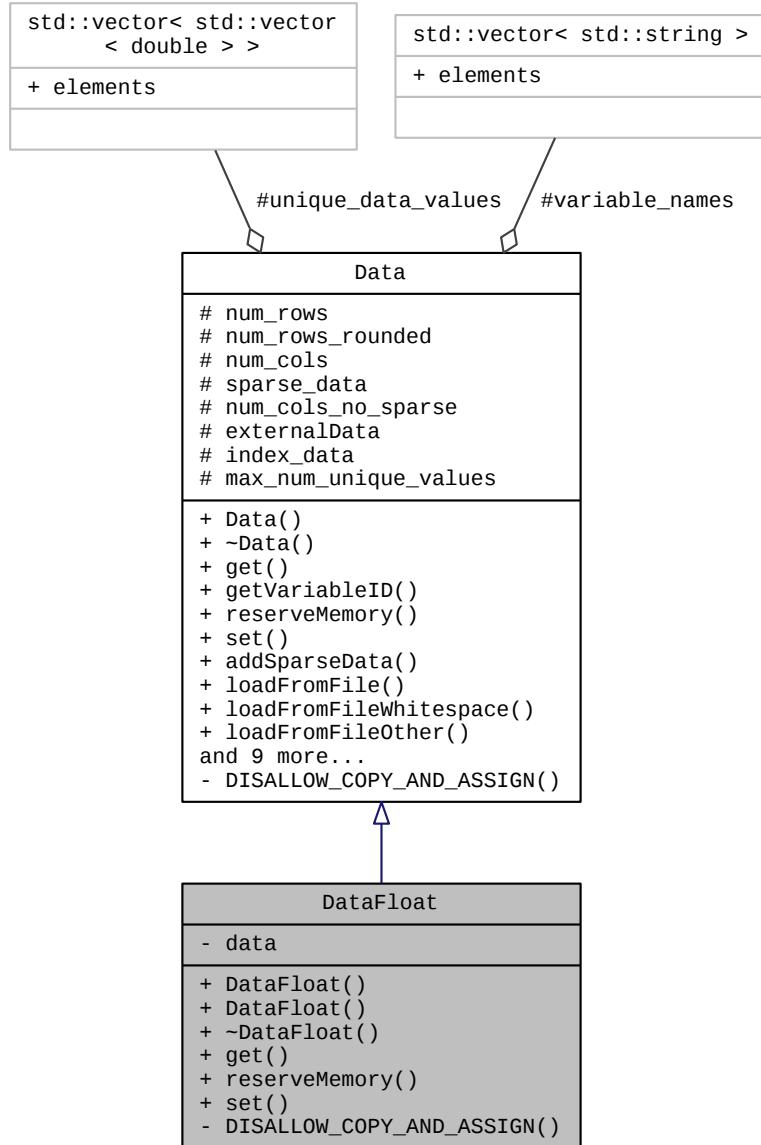
6.23 DataFloat Class Reference

```
#include <DataFloat.h>
```

Inheritance diagram for DataFloat:



Collaboration diagram for DataFloat:



Public Member Functions

- `DataFloat ()`
- `DataFloat (double *data_double, std::vector< std::string > variable_names, size_t num_rows, size_t num_cols)`
- `virtual ~DataFloat ()`
- `double get (size_t row, size_t col) const override`
- `void reserveMemory () override`
- `void set (size_t col, size_t row, double value, bool &error) override`

Private Member Functions

- [DISALLOW_COPY_AND_ASSIGN \(DataFloat\)](#)

Private Attributes

- float * [data](#)

Additional Inherited Members

6.23.1 Detailed Description

Definition at line 35 of file DataFloat.h.

6.23.2 Constructor & Destructor Documentation

6.23.2.1 DataFloat() [1/2]

```
DataFloat::DataFloat ( )
```

Definition at line 34 of file DataFloat.cpp.

6.23.2.2 DataFloat() [2/2]

```
DataFloat::DataFloat (
    double * data_double,
    std::vector< std::string > variable_names,
    size_t num_rows,
    size_t num_cols )
```

Definition at line 38 of file DataFloat.cpp.

6.23.2.3 ~DataFloat()

```
DataFloat::~DataFloat ( ) [virtual]
```

Definition at line 52 of file DataFloat.cpp.

6.23.3 Member Function Documentation

6.23.3.1 DISALLOW_COPY_AND_ASSIGN()

```
DataFloat::DISALLOW_COPY_AND_ASSIGN (
    DataFloat ) [private]
```

6.23.3.2 get()

```
double DataFloat::get (
    size_t row,
    size_t col ) const [inline], [override], [virtual]
```

Implements [Data](#).

Definition at line 41 of file DataFloat.h.

6.23.3.3 reserveMemory()

```
void DataFloat::reserveMemory () [inline], [override], [virtual]
```

Implements [Data](#).

Definition at line 51 of file DataFloat.h.

6.23.3.4 set()

```
void DataFloat::set (
    size_t col,
    size_t row,
    double value,
    bool & error ) [inline], [override], [virtual]
```

Implements [Data](#).

Definition at line 55 of file DataFloat.h.

6.23.4 Member Data Documentation

6.23.4.1 data

```
float* DataFloat::data [private]
```

Definition at line 60 of file DataFloat.h.

The documentation for this class was generated from the following files:

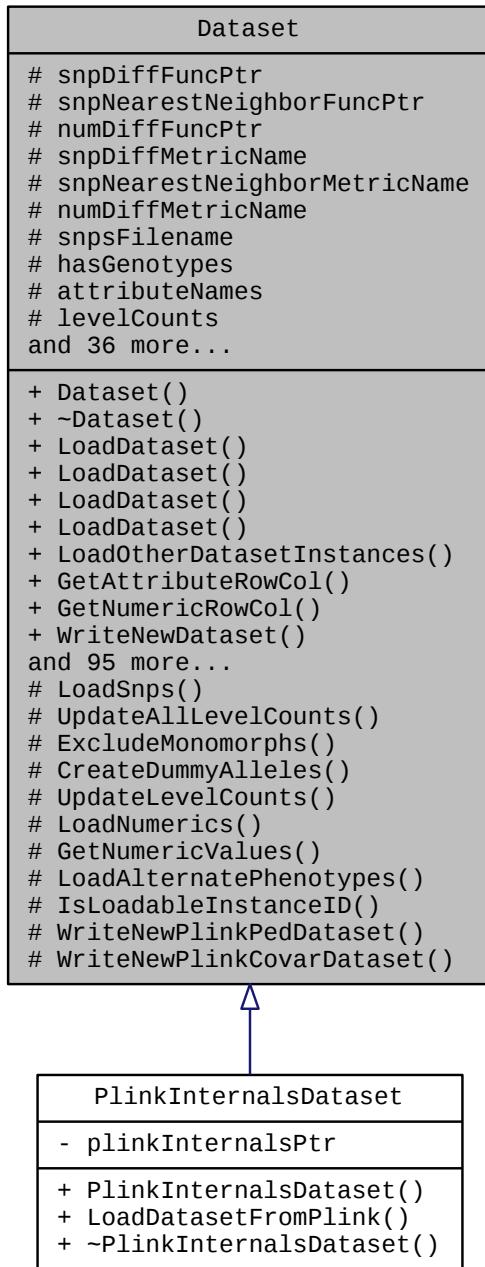
- src/[DataFloat.h](#)
- src/[DataFloat.cpp](#)

6.24 Dataset Class Reference

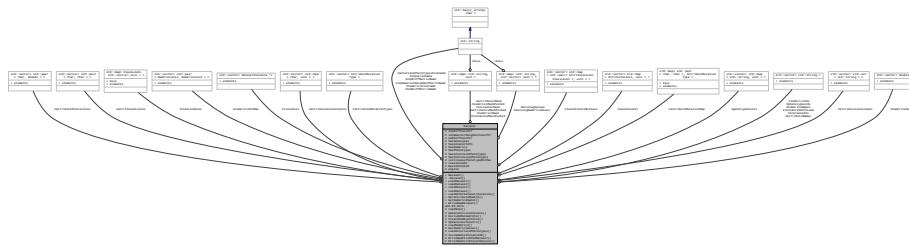
Base class for collections of instances containing attributes and class.

```
#include <Dataset.h>
```

Inheritance diagram for Dataset:



Collaboration diagram for Dataset:



Public Member Functions

- **Dataset ()**
Construct a default data set.
 - **virtual ~Dataset ()**
Destruct all dynamically allocated memory.
 - **bool LoadDataset (std::vector< std::vector< int > > &dataMatrix, std::vector< int > &classLabels, std::vector< std::string > &attrNames)**
Load the data set from "raw data".
 - **bool LoadDataset (std::string snpsFilename, std::string numericsFilename, std::string altPhenoFilename, std::vector< std::string > ids)**
Load the data set from files passed as parameters.
 - **bool LoadDataset (DgeData *dgeData)**
Load the data set from DGE data.
 - **bool LoadDataset (BirdseedData *birdseedData)**
Load the data set from Birdseed data.
 - **bool LoadOtherDatasetInstances (Dataset *otherDs, std::vector< uint > instIdx)**
Load the data set from another data set's instances.
 - **bool GetAttributeRowCol (uint row, uint col, AttributeLevel &attrVal)**
Get the attribute value at row, column.
 - **bool GetNumericRowCol (uint row, uint col, NumericLevel &numVal)**
Get the numeric value at row, column.
 - **bool WriteNewDataset (std::string newDatasetFilename, OutputDatasetType outputDatasetType)**
Write the data set to a new filename, respecting masked attributes and numerics and class/phenotype data type.
 - **bool WriteNewDataset (std::string newDatasetFilename, std::vector< std::string > attributes, OutputDatasetType outputDatasetType)**
Write the data set to a new filename, writing only the names in the passed attributes list and also respecting masked attributes and numerics and class/phenotype data type.
 - **bool ExtractAttributes (std::string scoresFilename, uint topN, std::string newDatasetFilename)**
Extracts top N attributes based on a file of attribute scores and writes a new dataset.
 - **bool SwapAttributes (uint a1, uint a2)**
Swap two attributes/columns in the dataset.
 - **uint NumVariables ()**
Return the number of discrete plus continuous variables in the data set.
 - **std::vector< std::string > GetVariableNames ()**
Returns the names of discrete and continuous variables in the data set.

- virtual `uint NumInstances ()`
Returns the number of instances in the data set.
- `DatasetInstance * GetInstance (uint index)`
Returns a pointer to a dataset instance selected by index.
- `DatasetInstance * GetRandomInstance ()`
Returns a pointer to a randomly chosen data set instance.
- `std::vector< std::string > GetInstanceIDs ()`
Get all instance IDs.
- `bool GetInstanceIndexForID (std::string ID, uint &instanceIndex)`
Get the instance index from the instance ID.
- virtual `uint NumAttributes ()`
Return the number of unmasked discrete attributes in the data set.
- `std::vector< std::string > GetAttributeNames ()`
Return the discrete (SNP) attribute names.
- `std::vector< std::string > GetFileAttributeNames ()`
Return the discrete (SNP) attribute names as read from file originally.
- `bool GetAttributeValues (uint attributeIndex, std::vector< AttributeLevel > &attributeValues)`
Loads the referenced vector with an attribute's values (column).
- `bool GetAttributeValues (std::string attributeName, std::vector< AttributeLevel > &attributeValues)`
Loads the referenced vector with an attribute's values (column) from the dataset.
- `std::string GetSnpFilename ()`
Get the filename SNPs were read from.
- `uint GetAttributeIndexFromName (std::string attributeName)`
Looks up original attribute index from attribute name.
- `bool HasGenotypes ()`
Does the data set have genotype variables?
- `bool HasAllelicInfo ()`
Does the data set have allelic information for genotypes?
- `AttributeLevel GetAttribute (unsigned instanceIndex, std::string name)`
Get attribute value for attribute name at instance index.
- `std::pair< char, char > GetAttributeAlleles (uint attributeIndex)`
Get attribute major and minor alleles.
- virtual `std::pair< char, double > GetAttributeMAF (uint attributeIndex)`
Get attribute minor allele and frequency.
- `bool ProcessExclusionFile (std::string exclusionFilename)`
Remove file of attribute names from consideration in analyses.
- virtual `AttributeMutationType GetAttributeMutationType (uint attributeIndex)`
Get attribute mutation type.
- `std::vector< double > GetMAFs ()`
- `double GetJukesCantorDistance (DatasetInstance *dsi1, DatasetInstance *dsi2)`
Apply Jukes-Cantor distance.
- `double GetKimuraDistance (DatasetInstance *dsi1, DatasetInstance *dsi2)`
Get Kimura Two-Parameter distance.
- `bool GetIntForGenotype (std::string genotype, AttributeLevel &newAttr)`
Get integer value for string genotype.
- `uint NumLevels (uint index)`
Returns the number of levels in a given attribute index.

- `uint NumNumerics ()`
Return the number of unmasked discrete attributes in the data set.
- `std::vector< std::string > GetNumericsNames ()`
Return the numeric attribute names.
- `std::vector< std::string > GetFileNumericsNames ()`
Return the numeric attribute names as originally read from file.
- `std::pair< double, double > GetMinMaxForNumeric (uint numericIdx)`
Get the minimum and maximum values for a numeric at index.
- `double GetMeanForNumeric (uint numericIdx)`
Get the mean/average of numeric at index.
- `bool HasNumerics ()`
Does the data set have numeric variables? setter/getter.
- `void HasNumerics (bool setHasNumerics)`
- `NumericLevel GetNumeric (uint instanceIndex, std::string name)`
Get numeric value for numeric name at instance index.
- `bool GetNumericValues (std::string numericName, std::vector< NumericLevel > &numericValues)`
Loads the referenced vector with a numeric's values (column) from the dataset.
- `std::string GetNumericsFilename ()`
Get the filename numerics were read from.
- `uint GetNumericIndexFromName (std::string numericName)`
Looks up original numeric index from numeric name.
- `arma::mat GetNumericMatrix ()`
- `uint NumClasses ()`
Normalize numerics by column sums. Get the number of classes in the data set.
- `uint GetClassColumn ()`
Get the class column as read from the file.
- `bool GetClassValues (std::vector< ClassLevel > &classValues)`
Loads the referenced vector with the dataset's class labels.
- `const std::map< ClassLevel, std::vector< uint > > & GetClassIndexes ()`
Get a map from class levels to a vector of instance indices.
- `bool HasAlternatePhenotypes ()`
Does the data set have alternate phenotypes loaded?
- `void HasAlternatePhenotypes (bool setHasAlternatePhenotypes)`
- `std::string GetAlternatePhenotypesFilename ()`
Get the alternate phenotype filename.
- `bool HasContinuousPhenotypes ()`
Does the data set have continuous phenotypes?
- `bool HasPhenotypes ()`
Does the data set have any valid phenotypes?
- `std::pair< double, double > GetMinMaxForContinuousPhenotype ()`
Get the minimum and maximum values for the continuous phenotype.
- `void Print ()`
Print the entire data set in compact format.
- `void PrintStats ()`
Print basic statistics about the data set - discrete/SNPs only.
- `void PrintNumericsStats ()`
Print statistics about the data set including numerics.

- void **PrintStatsSimple** (std::ostream &outStream=std::cout)
Print very simple statistics about the data set with no formatting.
- void **PrintClassIndexInfo** (std::ostream &outStream=std::cout)
Print class index information.
- void **PrintMissingValuesStats** ()
Print missing value statistics.
- void **PrintLevelCounts** ()
Print attribute level counts.
- void **WriteLevelCounts** (std::string levelsFilename)
Write attribute level counts to a text file.
- void **PrintAttributeLevelsSeen** ()
Print unique attribute levels seen.
- bool **MaskRemoveVariable** (std::string variableName)
Removes the variable name from consideration in any data set operations.
- bool **MaskRemoveVariableType** (std::string variableName, **AttributeType** varType)
Removes the attribute name from consideration in any data set operations.
- bool **MaskSearchVariableType** (std::string variableName, **AttributeType** attrType)
Determines if the named variable is in the current masked data set.
- bool **MaskIncludeAllAttributes** (**AttributeType** attrType)
Mark all attributes for inclusion in data set operations.
- std::vector< uint > **MaskGetAttributeIndices** (**AttributeType** attrType)
Return a vector of all the attribute indices under consideration.
- const std::map< std::string, uint > & **MaskGetAttributeMask** (**AttributeType** attrType)
Return a map of attribute name to attribute index of attributes to include.
- std::vector< std::string > **MaskGetAllVariableNames** ()
Return a vector of all the variable names under consideration.
- bool **MaskRemoveInstance** (std::string instanceId)
Removes the instance from consideration in any data set operations.
- bool **MaskSearchInstance** (std::string instanceId)
Determines if the names Instance is in the current masked dataset.
- bool **MaskIncludeAllInstances** ()
Mark all instances for inclusion in algorithms.
- std::vector< uint > **MaskGetInstanceIndices** ()
Return a vector of all the instance indices under consideration.
- std::vector< std::string > **MaskGetInstanceIds** ()
Return a vector of all the instance ids under consideration.
- const std::map< std::string, uint > & **MaskGetInstanceMask** ()
Return a map of instance name to instance index of instances to include.
- bool **MaskPushAll** ()
Save the current masks for later restore.
- bool **MaskPopAll** ()
Restore the masks previously pushed.
- bool **MaskWriteNewDataset** (std::string newDatasetFilename)
Saved the unmasked attributes as a tab-delimited text file.
- void **PrintMaskStats** ()
Print mask statistics.

- void `RunSnpDiagnosticTests` (std::string logFilename, double globalGenotypeThreshold=0.01, uint cellThreshold=5)

Perform and report SNP diagnostic test information.
- bool `CheckHardyWeinbergEquilibrium` (std::vector< uint > &chkGenotypeCounts)

Calculate whether passed genotype counts are in HWE.
- double `SNPHWE` (int obs_hets, int obs_hom1, int obs_hom2)

This code implements an exact SNP test of Hardy-Weinberg Equilibrium.
- double `GetClassProbability` (ClassLevel thisClass)

Get the probability of a class value in the data set.
- double `GetProbabilityValueGivenClass` (uint attributeIndex, AttributeLevel A, ClassLevel classValue)

Get the probability of an attribute value at an attribute index.
- void `AttributeInteractionInformation` ()

Calculate and display interaction information for all attribute combinations.
- void `CalculateInteractionInformation` (std::map< std::pair< int, int >, std::map< std::string, double > > &results)

Calculate all the information needed to construct the interaction diagram.
- bool `CalculateGainMatrix` (double **gainMatrix, std::string matrixFilename="")

Calculate the GAIN matrix to run snprank on this data set.
- bool `CalculateRegainMatrix` (double **gainMatrix, std::string matrixFilename="")

Calculate the regression GAIN matrix to run snprank on this data set.
- bool `CalculateDistanceMatrix` (double **distanceMatrix, std::string matrixFilename="")

Calculate the instance-to-instance distance matrix for this data set.
- bool `CalculateDistanceMatrix` (std::vector< std::vector< double > > &distanceMatrix)

Calculate the instance-to-instance distance matrix for this data set.
- double `ComputeInstanceToInstanceDistance` (DatasetInstance *dsi1, DatasetInstance *dsi2)

Compute the distance between two DatasetInstances.
- bool `SetDistanceMetrics` (std::string newSnpDiffMetricName, std::string newSnpNNMetricName, std::string newNumMetricName="manhattan")

Set the the distance metrics used to compute instance-to-instance distances.
- std::vector< std::string > `GetDistanceMetrics` ()

Get the the distance metrics used to compute instance-to-instance distances.
- std::pair< uint, uint > `GetAttributeTiTvCounts` ()

Get the the mutation transition and transversion counts.
- bool `WriteSnpTiTvInfo` (std::string titvFilename)

Dump the SNP transition/transversion information to file.
- bool `ResetNearestNeighbors` ()

Reset instances nearest neighbor information.
- bool `LoadPrivacySim` (std::string filename)

Load numerics (continuous attributes) with pheno from a simulation file.

Protected Member Functions

- virtual bool `LoadSnps` (std::string filename)

Load SNPs from file using the data set filename.
- void `UpdateAllLevelCounts` ()

Update level counts for all instances by calling UpdateLevelCounts(inst)
- void `ExcludeMonomorphs` ()

- Exclude any monomorphic SNPs, since they add no information about class.
- void `CreateDummyAlleles ()`
Create dummy alleles from genotypes for data sets that have no allele info.
- void `UpdateLevelCounts (DatasetInstance *dsi)`
Update all attribute level counts from one data set instance.
- bool `LoadNumerics (std::string filename)`
Load numerics (continuous attributes) from a file set in the constructor.
- bool `GetNumericValues (uint numericIndex, std::vector< NumericLevel > &numericValues)`
Loads the referenced vector with an numeric's values (column).
- bool `LoadAlternatePhenotypes (std::string filename)`
Load alternate phenotype/class values from a plink covariate .cov file.
- bool `IsLoadableInstanceID (std::string ID)`
Is the passed instance ID loadable (not filtered).
- bool `WriteNewPlinkPedDataset (std::string baseDatasetFilename)`
Write the dataset to a new PLINK PED/MAP format, respecting masked attributes class/phenotype data type.
- bool `WriteNewPlinkCovarDataset (std::string baseDatasetFilename)`
Write the dataset to a new PLINK covar format, respecting masked attributes class/phenotype data type.

Protected Attributes

- double(* `snpDiffFuncPtr`)(uint attributeIndex, `DatasetInstance *dsi1`, `DatasetInstance *dsi2`)
Compute the discrete difference in an attribute between two instances for Relief-F scoring.
- double(* `snpNearestNeighborFuncPtr`)(uint attributeIndex, `DatasetInstance *dsi1`, `DatasetInstance *dsi2`)
Compute the discrete difference in an attribute between two instances for determining nearest neighbors.
- double(* `numDiffFuncPtr`)(uint attributeIndex, `DatasetInstance *dsi1`, `DatasetInstance *dsi2`)
Compute the continuous difference in an attribute between two instances.
- std::string `snpDiffMetricName`
the name of discrete diff(erence) function
- std::string `snpNearestNeighborMetricName`
the name of discrete distance function for nearest neighbors
- std::string `numDiffMetricName`
the name of continuous diff(erence) function
- std::string `snpsFilename`
file from which the discrete attributes (SNPs) were read
- bool `hasGenotypes`
does the data set contain any genotypes?
- std::vector< std::string > `attributeNames`
discrete attribute names read from file
- std::vector< std::map< `AttributeLevel`, uint > > `levelCounts`
attribute values/levels counts
- std::vector< std::map< std::pair< `AttributeLevel`, `ClassLevel` >, uint > > `levelCountsByClass`
attribute values/levels counts by discrete class
- std::vector< std::set< std::string > > `attributeLevelsSeen`
unique attribute values/levels read from file
- std::vector< std::pair< char, char > > `attributeAlleles`
allele1, allele2

- std::vector< std::map< char, uint > > attributeAlleleCounts
 - allele->count*
- std::vector< std::pair< char, double > > attributeMinorAllele
 - minor allele, minor allele frequency*
- bool hasAllelicInfo
 - Does this data set have allelic information?*
- std::vector< std::map< std::string, uint > > genotypeCounts
 - genotype->count*
- std::vector< AttributeMutationType > attributeMutationTypes
 - Keep mutation type for all attributes.*
- std::map< std::pair< char, char >, AttributeMutationType > attributeMutationMap
 - Lookup table for mutation type.*
- std::string numericsFilename
 - file from which the continuous attributes were read*
- bool hasNumerics
 - does the data set contain any continuous attributes?*
- std::vector< std::string > numericsIds
 - IDs associated with the numerics read from file.*
- std::vector< std::pair< NumericLevel, NumericLevel > > numericsMinMax
 - the minimum and maximum value for each continuous attribute*
- std::vector< double > numericsSums
 - the column sum for each continuous attribute*
- std::vector< std::string > numericsNames
 - continuous attribute names read from file*
- bool hasPhenotypes
 - Does the data set contain phenotypes?*
- std::string alternatePhenotypesFilename
 - file from which the alternate phenotypes (class labels) were read*
- bool hasAlternatePhenotypes
 - does the data set contain alternate phenotypes?*
- std::vector< std::string > phenotypeIds
 - IDs associated with the phenotypes/classes read from file.*
- bool hasContinuousPhenotypes
 - does the data set contain continuous phenotypes?*
- std::pair< NumericLevel, NumericLevel > continuousPhenotypeMinMax
 - the minimum and maximum value for each continuous phenotype*
- std::vector< DatasetInstance * > instances
 - vector of pointers to all instances in the data set*
- std::vector< std::string > instanceIds
 - IDs associated with the instances read from file.*
- std::vector< std::string > instanceIdsToLoad
 - IDs of instances to load from numeric and/or phenotype files.*
- std::map< std::string, std::vector< uint > > missingValues
 - missing discrete values and their instance indices*
- std::map< std::string, std::vector< uint > > missingNumericValues
 - missing continuous values and their instance indices*
- uint classColumn

- *class column from the original data set*
- std::map< [ClassLevel](#), std::vector< [uint](#) > > [classIndexes](#)
 - *class values mapped to instance indices*
 - std::map< std::string, [uint](#) > [attributesMask](#)
 - std::map< std::string, [uint](#) > [numericsMask](#)
 - std::map< std::string, [uint](#) > [instancesMask](#)
 - std::map< std::string, [uint](#) > [attributesMaskPushed](#)
 - *masks can be temporarily pushed and popped*
 - std::map< std::string, [uint](#) > [numericsMaskPushed](#)
 - std::map< std::string, [uint](#) > [instancesMaskPushed](#)
 - bool [maskIsPushed](#)
 - std::mt19937_64 [engine](#)

6.24.1 Detailed Description

Base class for collections of instances containing attributes and class.

Added interaction information week of 4/18-26/06 Totally redone for McKinney Lab. February 2011.

Author

Bill White

Version

1.0

Contact: bill.c.white@gmail.com Created on: 6/14/05

Definition at line 39 of file Dataset.h.

6.24.2 Constructor & Destructor Documentation

6.24.2.1 Dataset()

Dataset::Dataset ()

Construct a default data set.

[Set defaults.](#)

Load attribute mutation map for transitions/transversions.

metric defaults

Definition at line 60 of file Dataset.cpp.

6.24.2.2 ~Dataset()

```
Dataset::~Dataset ( ) [virtual]
```

Destruct all dynamically allocated memory.

Definition at line 115 of file Dataset.cpp.

6.24.3 Member Function Documentation

6.24.3.1 AttributeInteractionInformation()

```
void Dataset::AttributeInteractionInformation ( )
```

Calculate and display interaction information for all attribute combinations.

call CalculateInteractionInformation

display results of interaction calculations

display results header line

get the column sum

display results detail; $I(A;B|C)$ column as percentage

Definition at line 2415 of file Dataset.cpp.

6.24.3.2 CalculateDistanceMatrix() [1/2]

```
bool Dataset::CalculateDistanceMatrix (
    double ** distanceMatrix,
    std::string matrixFilename = "" )
```

Calculate the instance-to-instance distance matrix for this data set.

Uses OpenMP to calculate matrix entries in parallel threads.

Parameters

out	<i>distanceMatrix</i>	pointer to an allocated $m \times m$ matrix, m = number of instances
in	<i>distanceMatrixFilename</i>	filename to write matrix

Returns

success

write header

write all n-by-n matrix entries

write phenotypes for the instances

Definition at line 2941 of file Dataset.cpp.

6.24.3.3 CalculateDistanceMatrix() [2/2]

```
bool Dataset::CalculateDistanceMatrix (
    std::vector< std::vector< double > > & distanceMatrix )
```

Calculate the instance-to-instance distance matrix for this data set.

Uses OpenMP to calculate matrix entries in parallel threads.

Parameters

out	<i>distanceMatrix</i>	vector of vectors of double: m x m matrix, m = number of instances
-----	-----------------------	--

Returns

success

Definition at line 3034 of file Dataset.cpp.

6.24.3.4 CalculateGainMatrix()

```
bool Dataset::CalculateGainMatrix (
    double ** gainMatrix,
    std::string matrixFilename = "" )
```

Calculate the GAIN matrix to run snprank on this data set.

Uses OpenMP to calculate matrix entries in parallel threads.

Parameters

out	<i>gainMatrix</i>	pointer to an allocated n x n matrix, n = number of attributes
-----	-------------------	--

Returns

success

Calculate the interaction information from entropies

Populate the GAIN matrix

write header

write all m-by-m matrix entries

Definition at line 2607 of file Dataset.cpp.

6.24.3.5 CalculateInteractionInformation()

```
void Dataset::CalculateInteractionInformation (
    std::map< std::pair< int, int >, std::map< std::string, double > > & results )
```

Calculate all the information needed to construct the interaction diagram.

Parameters

out	<i>results</i>	map of attribute combinations to results
-----	----------------	--

so many way to fail before getting started

vectors to hold sequences for attributes a and b with class c ab is an attribute constructed from the cartesian product of a and b

Get the class values once

for all possible (unique) interactions, ie nCk

use OpenMP to run in parallel the construction of the attribute interaction matrix

THREAD STARTS

load attribute values (columns) into vectors for entropy routines

construct a new attribute that is the cartesian product of a and b

compute all entropies, calculate information theoretic quantities and save the results

inner loop over j ends THREAD ENDS

outer loop over i ends

end - no return value

Definition at line 2475 of file Dataset.cpp.

6.24.3.6 CalculateRegainMatrix()

```
bool Dataset::CalculateRegainMatrix (
    double ** gainMatrix,
    std::string matrixFilename = "" )
```

Calculate the regression GAIN matrix to run snprank on this data set.

Uses OpenMP to calculate matrix entries in parallel threads.

Parameters

<code>out</code>	<code>gainMatrix</code>	pointer to an allocated n x n matrix, n = number of attributes
------------------	-------------------------	--

Returns

`success`

write header

write all m-by-m matrix entries

Definition at line 2673 of file Dataset.cpp.

6.24.3.7 CheckHardyWeinbergEquilibrium()

```
bool Dataset::CheckHardyWeinbergEquilibrium (
    std::vector< uint > & chkGenotypeCounts )
```

Calculate whether passed genotype counts are in HWE.

Parameters

<code>genotypeCounts</code>	vector of genotype counts: AA, Aa, aa
-----------------------------	---------------------------------------

Returns

`counts are in HWE?`

observed counts

HWE probabilities

expected values

perform Pearson's chi-squared test

one degree of freedom (# genotypes - # alleles), 5% significance level

Definition at line 2230 of file Dataset.cpp.

6.24.3.8 ComputeInstanceToInstanceDistance()

```
double Dataset::ComputeInstanceToInstanceDistance (
    DatasetInstance * dsi1,
    DatasetInstance * dsi2 )
```

Compute the distance between two DatasetInstances.

Parameters

in	<i>dsi1</i>	pointer to DatasetInstance 1
in	<i>dsi2</i>	pointer to DatasetInstance 2

Returns

distance

Definition at line 2735 of file Dataset.cpp.

6.24.3.9 CreateDummyAlleles()

```
void Dataset::CreateDummyAlleles ( ) [protected]
```

Create dummy alleles from genotypes for data sets that have no allele info.

assign major and minor alleles

Definition at line 3346 of file Dataset.cpp.

6.24.3.10 ExcludeMonomorphs()

```
void Dataset::ExcludeMonomorphs ( ) [protected]
```

Exclude any monomorphic SNPs, since they add no information about class.

Definition at line 3328 of file Dataset.cpp.

6.24.3.11 ExtractAttributes()

```
bool Dataset::ExtractAttributes (
    std::string scoresFilename,
    uint topN,
    std::string newDatasetFilename )
```

Extracts top N attributes based on a file of attribute scores and writes a new dataset.

Revised 10/3/11 for numerics and continuous class/phenotypes.

Parameters

in	<i>scoresFilename</i>	filename of attribute scores and names
in	<i>topN</i>	top N attributes
in	<i>newDatasetFilename</i>	filename of new dataset

Returns

success

Definition at line 928 of file Dataset.cpp.

6.24.3.12 GetAlternatePhenotypesFilename()

```
string Dataset::GetAlternatePhenotypesFilename ( )
```

Get the alternate phenotype filename.

Definition at line 1452 of file Dataset.cpp.

6.24.3.13 GetAttribute()

```
AttributeLevel Dataset::GetAttribute (
    unsigned instanceIndex,
    std::string name )
```

Get attribute value for attribute name at instance index.

Parameters

in	<i>instanceIndex</i>	instance index
in	<i>name</i>	attribute name

Returns

attributevalue

Definition at line 1146 of file Dataset.cpp.

6.24.3.14 GetAttributeAlleles()

```
pair< char, char > Dataset::GetAttributeAlleles (  
    uint attributeIndex )
```

Get attribute major and minor alleles.

Parameters

in	<i>attribute</i>	index
----	------------------	-------

Returns

pair (major allele, minor allele frequency)

Definition at line 1160 of file Dataset.cpp.

6.24.3.15 GetAttributeIndexFromName()

```
uint Dataset::GetAttributeIndexFromName (  
    std::string attributeName )
```

Looks up original attribute index from attribute name.

Parameters

in	<i>attributeName</i>	attribute name
----	----------------------	----------------

Returns

attribute index or INVALID_INDEX

Definition at line 1295 of file Dataset.cpp.

6.24.3.16 GetAttributeMAF()

```
pair< char, double > Dataset::GetAttributeMAF (   
    uint attributeIndex ) [virtual]
```

Get attribute minor allele and frequency.

Parameters

in	<i>attribute</i>	index
----	------------------	-------

Returns

pair (minor allele, minor allele frequency)

An Introduction to Genetic Analysis by Griffiths, Miller, Suzuki, Lewontin and Gelbart, 2000, page 715.

Definition at line 1175 of file Dataset.cpp.

6.24.3.17 GetAttributeMutationType()

```
AttributeMutationType Dataset::GetAttributeMutationType (   
    uint attributeIndex ) [virtual]
```

Get attribute mutation type.

Parameters

in	<i>attribute</i>	index
----	------------------	-------

Returns

mutation type (transition, transversion, unknown)

Definition at line 1228 of file Dataset.cpp.

6.24.3.18 GetAttributeNames()

```
vector< string > Dataset::GetAttributeNames ( )
```

Return the discrete (SNP) attribute names.

Returns

vector of attribute names

Definition at line 1078 of file Dataset.cpp.

6.24.3.19 GetAttributeRowCol()

```
bool Dataset::GetAttributeRowCol (
    uint row,
    uint col,
    AttributeLevel & attrVal )
```

Get the attribute value at row, column.

Same as instance index, attribute index.

Parameters

in	<i>row</i>	instance row
in	<i>col</i>	attribute column
out	<i>attrVal</i>	attribute value

Returns

success

Definition at line 508 of file Dataset.cpp.

6.24.3.20 GetAttributeTiTvCounts()

```
pair< uint, uint > Dataset::GetAttributeTiTvCounts ( )
```

Get the the mutation transition and transversion counts.

Returns

pair<number of transitions, number of transversions>

Definition at line 2893 of file Dataset.cpp.

6.24.3.21 GetAttributeValues() [1/2]

```
bool Dataset::GetAttributeValues (
    uint attributeIndex,
    std::vector< AttributeLevel > & attributeValues )
```

Loads the referenced vector with an attribute's values (column).

from the dataset

Parameters

in	<i>attributeIndex</i>	attribute index
out	<i>attributeValues</i>	reference to a a vector allocated by the caller

Returns

success

Definition at line 1098 of file Dataset.cpp.

6.24.3.22 GetAttributeValues() [2/2]

```
bool Dataset::GetAttributeValues (
    std::string attributeName,
    std::vector< AttributeLevel > & attributeValues )
```

Loads the referenced vector with an attribute's values (column) from the dataset.

Parameters

in	<i>attributeName</i>	attribute name
out	<i>attributeValues</i>	reference to a a vector allocated by the caller

Returns

success

Definition at line 1121 of file Dataset.cpp.

6.24.3.23 GetClassColumn()

```
uint Dataset::GetClassColumn ( )
```

Get the class column as read from the file.

Definition at line 1418 of file Dataset.cpp.

6.24.3.24 GetClassIndexes()

```
const std::map< ClassLevel, std::vector< uint > > & Dataset::GetClassIndexes ( )
```

Get a map from class levels to a vector of instance indices.

Returns

map of class => instance indices

Definition at line 1440 of file Dataset.cpp.

6.24.3.25 GetClassProbability()

```
double Dataset::GetClassProbability (   
    ClassLevel thisClass )
```

Get the probability of a class value in the data set.

Parameters

<i>thisClass</i>	class value
------------------	-------------

Returns

probability

Definition at line 2391 of file Dataset.cpp.

6.24.3.26 GetClassValues()

```
bool Dataset::GetClassValues (
    std::vector< ClassLevel > & classValues )
```

Loads the referenced vector with the dataset's class labels.

Parameters

<code>out</code>	<code>classValues</code>	reference to a vector allocated by the caller
------------------	--------------------------	---

Returns`success`

Definition at line 1426 of file Dataset.cpp.

6.24.3.27 GetDistanceMetrics()

```
vector< string > Dataset::GetDistanceMetrics ( )
```

Get the distance metrics used to compute instance-to-instance distances.

Returns`pair<snp distance metric name, numeric distance metric name>`

Definition at line 2884 of file Dataset.cpp.

6.24.3.28 GetFileAttributeNames()

```
vector< string > Dataset::GetFileAttributeNames ( )
```

Return the discrete (SNP) attribute names as read from file originally.

Returns`vector of attribute names`

Definition at line 1087 of file Dataset.cpp.

6.24.3.29 GetFileNumericsNames()

```
vector< string > Dataset::GetFileNumericsNames ( )
```

Return the numeric attribute names as originally read from file.

Returns`vector of attribute names`

Definition at line 1317 of file Dataset.cpp.

6.24.3.30 GetInstance()

```
DatasetInstance * Dataset::GetInstance (   
    uint index )
```

Returns a pointer to a dataset instance selected by index.

Parameters

in	<i>index</i>	index of instance
----	--------------	-------------------

Returns

pointer to an instance

Definition at line 1042 of file Dataset.cpp.

6.24.3.31 GetInstanceIDs()

```
vector< string > Dataset::GetInstanceIDs ( )
```

Get all instance IDs.

Returns

vector of instance IDs

Definition at line 1054 of file Dataset.cpp.

6.24.3.32 GetInstanceIndexForID()

```
bool Dataset::GetInstanceIndexForID (
    std::string ID,
    uint & instanceIndex )
```

Get the instance index from the instance ID.

Parameters

in	<i>ID</i>	string ID
out	<i>instanceIndex</i>	instance index

Returns

success

Definition at line 1063 of file Dataset.cpp.

6.24.3.33 GetIntForGenotype()

```
bool Dataset::GetIntForGenotype (
    std::string genotype,
    AttributeLevel & newAttr )
```

Get integer value for string genotype.

Parameters

in	<i>genotype</i>	genotype string
out	<i>newAttr</i>	new attribute value

Returns

success

6.24.3.34 GetJukesCantorDistance()

```
double Dataset::GetJukesCantorDistance (
    DatasetInstance * dsil,
    DatasetInstance * dsi2 )
```

Apply Jukes-Cantor distance.

Definition at line 1236 of file Dataset.cpp.

6.24.3.35 GetKimuraDistance()

```
double Dataset::GetKimuraDistance (
    DatasetInstance * dsil,
    DatasetInstance * dsi2 )
```

Get Kimura Two-Parameter distance.

Definition at line 1255 of file Dataset.cpp.

6.24.3.36 GetMAFs()

```
vector< double > Dataset::GetMAFs ( )
```

Definition at line 2727 of file Dataset.cpp.

6.24.3.37 GetMeanForNumeric()

```
double Dataset::GetMeanForNumeric (
    uint numericIdx )
```

Get the mean/average of numeric at index.

Parameters

in	<i>numericIdx</i>	numeric index
----	-------------------	---------------

Returns

average value of numeric attribute at index

Definition at line 1333 of file Dataset.cpp.

6.24.3.38 GetMinMaxForContinuousPhenotype()

```
pair< double, double > Dataset::GetMinMaxForContinuousPhenotype ( )
```

Get the minumum and maximum values for the continuous phenotype.

Parameters

minimum/maximum pair

Definition at line 1464 of file Dataset.cpp.

6.24.3.39 GetMinMaxForNumeric()

```
pair< NumericLevel, NumericLevel > Dataset::GetMinMaxForNumeric (
    uint numericIdx )
```

Get the minimum and maximum values for a numeric at index.

Parameters

in	<i>numericIdx</i>	numeric index
----	-------------------	---------------

Returns

minimum/maximum pair

Definition at line 1328 of file Dataset.cpp.

6.24.3.40 GetNumeric()

```
NumericLevel Dataset::GetNumeric (
    uint instanceIndex,
    std::string name )
```

Get numeric value for numeric name at instance index.

Parameters

in	<i>instanceIndex</i>	instance index
in	<i>name</i>	numeric name

Returns

numeric value at index

Definition at line 1351 of file Dataset.cpp.

6.24.3.41 GetNumericIndexFromName()

```
uint Dataset::GetNumericIndexFromName (
    std::string numericName )
```

Looks up original numeric index from numeric name.

Parameters

in	<i>numericName</i>	numeric name
----	--------------------	--------------

Returns

attribute index or INVALID_INDEX

Definition at line 1384 of file Dataset.cpp.

6.24.3.42 GetNumericMatrix()

```
mat Dataset::GetNumericMatrix ( )
```

write continuous attribute values

Definition at line 1394 of file Dataset.cpp.

6.24.3.43 GetNumericRowCol()

```
bool Dataset::GetNumericRowCol (
    uint row,
    uint col,
    NumericLevel & numVal )
```

Get the numeric value at row, column.

Same as instance index, numeric index.

Parameters

<i>in</i>	<i>row</i>	instance row
<i>in</i>	<i>col</i>	numeric column
<i>out</i>	<i>numVal</i>	numeric value

Returns

success

Definition at line 521 of file Dataset.cpp.

6.24.3.44 GetNumericsFilename()

```
std::string Dataset::GetNumericsFilename ( )
```

Get the filename numerics were read from.

Definition at line 1380 of file Dataset.cpp.

6.24.3.45 GetNumericsNames()

```
vector< string > Dataset::GetNumericsNames ( )
```

Return the numeric attribute names.

Returns

vector of attribute names

Definition at line 1308 of file Dataset.cpp.

6.24.3.46 GetNumericValues() [1/2]

```
bool Dataset::GetNumericValues (
    std::string numericName,
    std::vector< NumericLevel > & numericValues )
```

Loads the referenced vector with a numeric's values (column) from the dataset.

Parameters

in	<i>numericName</i>	numeric name
out	<i>numericValues</i>	reference to a a vector allocated by the caller

Returns

success

Definition at line 1366 of file Dataset.cpp.

6.24.3.47 GetNumericValues() [2/2]

```
bool Dataset::GetNumericValues (
    uint numericIndex,
    std::vector< NumericLevel > & numericValues ) [protected]
```

Loads the referenced vector with an numeric's values (column).

from the dataset

Parameters

in	<i>numericIndex</i>	numeric index
out	<i>numericValues</i>	reference to a a vector allocated by the caller

Returns

success

Definition at line 3689 of file Dataset.cpp.

6.24.3.48 GetProbabilityValueGivenClass()

```
double Dataset::GetProbabilityValueGivenClass (
    uint attributeIndex,
    AttributeLevel A,
    ClassLevel classValue )
```

Get the probability of an attribute value at an attribute index.

Parameters

in	<i>attributeIndex</i>	attribute index
in	<i>A</i>	attribute value
in	<i>classValue</i>	class value

Returns

probability of the value in attribute given class

Definition at line 2399 of file Dataset.cpp.

6.24.3.49 GetRandomInstance()

```
DatasetInstance * Dataset::GetRandomInstance ( )
```

Returns a pointer to a randomly chosen data set instance.

The random number generator is set to give values in range of instance indexes.

Returns

pointer to a data set instance

Definition at line 1049 of file Dataset.cpp.

6.24.3.50 GetSnpsFilename()

```
std::string Dataset::GetSnpsFilename ( )
```

Get the filename SNPs were read from.

Definition at line 1134 of file Dataset.cpp.

6.24.3.51 GetVariableNames()

```
vector< string > Dataset::GetVariableNames ( )
```

Returns the names of discrete and continuous variables in the data set.

Returns

vector of names as strings

Definition at line 1026 of file Dataset.cpp.

6.24.3.52 HasAllelicInfo()

```
bool Dataset::HasAllelicInfo ( )
```

Does the data set have allelic information for genotypes?

Definition at line 1142 of file Dataset.cpp.

6.24.3.53 HasAlternatePhenotypes() [1/2]

```
bool Dataset::HasAlternatePhenotypes ( )
```

Does the data set have alternate phenotypes loaded?

Definition at line 1444 of file Dataset.cpp.

6.24.3.54 HasAlternatePhenotypes() [2/2]

```
void Dataset::HasAlternatePhenotypes ( 
    bool setHasAlternatePhenotypes )
```

Definition at line 1448 of file Dataset.cpp.

6.24.3.55 HasContinuousPhenotypes()

```
bool Dataset::HasContinuousPhenotypes ( )
```

Does the data set have continuous phenotypes?

Definition at line 1456 of file Dataset.cpp.

6.24.3.56 HasGenotypes()

```
bool Dataset::HasGenotypes ( )
```

Does the data set have genotype variables?

Definition at line 1138 of file Dataset.cpp.

6.24.3.57 HasNumerics() [1/2]

```
bool Dataset::HasNumerics ( )
```

Does the data set have numeric variables? setter/getter.

Definition at line 1343 of file Dataset.cpp.

6.24.3.58 HasNumerics() [2/2]

```
void Dataset::HasNumerics (
    bool setHasNumerics )
```

Definition at line 1347 of file Dataset.cpp.

6.24.3.59 HasPhenotypes()

```
bool Dataset::HasPhenotypes ( )
```

Does the data set have any valid phenotypes?

Definition at line 1460 of file Dataset.cpp.

6.24.3.60 IsLoadableInstanceID()

```
bool Dataset::IsLoadableInstanceID (
    std::string ID ) [protected]
```

Is the passed instance ID loadable (not filtered).

Parameters

in	ID	instance ID
----	----	-------------

Returns

[out] success

Definition at line 3926 of file Dataset.cpp.

6.24.3.61 LoadAlternatePhenotypes()

```
bool Dataset::LoadAlternatePhenotypes (
    std::string filename ) [protected]
```

Load alternate phenotype/class values from a plink covariate .cov file.

Format described here: <http://pngu.mgh.harvard.edu/~purcell/plink/data.shtml#covar> Major changes for continuous phenotypes/class - 9/29/11

Parameters

in	<i>filename</i>	alternate phenotype data filename in PLINK covar format
----	-----------------	---

Returns

success

Detect the class type

Definition at line 3736 of file Dataset.cpp.

6.24.3.62 LoadDataset() [1/4]

```
bool Dataset::LoadDataset (
    std::vector< std::vector< int > > & dataMatrix,
    std::vector< int > & classLabels,
    std::vector< std::string > & attrNames )
```

Load the data set from "raw data".

Parameters

in	<i>dataMatrix</i>	reference to a matrix of SNP values 0-1-2-missing
in	<i>classLabels</i>	reference to a vector of case-control labels
in	<i>attrNames</i>	reference to a vector of attribute names

Returns

success

Definition at line 124 of file Dataset.cpp.

6.24.3.63 LoadDataset() [2/4]

```
bool Dataset::LoadDataset (
    std::string snpsFilename,
    std::string numericsFilename,
    std::string altPhenoFilename,
    std::vector< std::string > ids )
```

Load the data set from files passed as parameters.

Parameters

in	<i>snpFilename</i>	discrete values (SNPs) filename
in	<i>doRecodeA</i>	perform recodeA encoding after reading
in	<i>numericsFilename</i>	continuous values (numerics) filename or empty string
in	<i>altPhenoFilename</i>	alternate class (phenotype) filename or empty string
in	<i>ids</i>	vector of possibly empty IDs to match in auxiliary files

Returns

success

Definition at line 201 of file Dataset.cpp.

6.24.3.64 LoadDataset() [3/4]

```
bool Dataset::LoadDataset (
    DgeData * dgeData )
```

Load the data set from DGE data.

Parameters

in	<i>dgeData</i>	pointer to a digital gene expression (DGE) data object
----	----------------	--

Returns

success

Definition at line 291 of file Dataset.cpp.

6.24.3.65 LoadDataset() [4/4]

```
bool Dataset::LoadDataset (
    BirdseedData * birdseedData )
```

Load the data set from Birdseed data.

Parameters

in	<i>birdseedData</i>	pointer to a Birdseed-called SNP data object
----	---------------------	--

Returns

success

add allelic info

Definition at line 338 of file Dataset.cpp.

6.24.3.66 LoadNumerics()

```
bool Dataset::LoadNumerics (
    std::string filename ) [protected]
```

Load numerics (continuous attributes) from a file set in the constructor.

Parameters

in	<i>filename</i>	numerics data filename in PLINK covar format
----	-----------------	--

Returns

success

Definition at line 3397 of file Dataset.cpp.

6.24.3.67 LoadOtherDatasetInstances()

```
bool Dataset::LoadOtherDatasetInstances (
    Dataset * otherDS,
    std::vector< uint > instIdx )
```

Load the data set from another data set's instances.

Parameters

in	<i>pointer</i>	to a Dataset
in	<i>vector</i>	of instance indicies

Returns

success

Definition at line 443 of file [Dataset.cpp](#).**6.24.3.68 LoadPrivacySim()**

```
bool Dataset::LoadPrivacySim (
    std::string filename )
```

Load numerics (continuous attributes) with pheno from a simulation file.

Parameters

in	<i>filename</i>	numerics data filename in Privacy EC format
----	-----------------	---

Returns

success

Definition at line 3546 of file [Dataset.cpp](#).**6.24.3.69 LoadSnp()**

```
bool Dataset::LoadSnp (
    std::string filename ) [protected], [virtual]
```

Load SNPs from file using the data set filename.

----- Beginning of private methods -----

Parameters

in	<i>filename</i>	SNPs filename
in	<i>deRecodeA</i>	perform a recodeA operation after reading raw data?

Returns

success

Open the data file and read line-by-line

Detect the class type

Definition at line 3058 of file Dataset.cpp.

6.24.3.70 MaskGetAllVariableNames()

```
vector< string > Dataset::MaskGetAllVariableNames ( )
```

Return a vector of all the variable names under consideration.

Returns

vector of discrete and numeric variable

Definition at line 1867 of file Dataset.cpp.

6.24.3.71 MaskGetAttributeIndices()

```
vector< uint > Dataset::MaskGetAttributeIndices (   
     AttributeType attrType )
```

Return a vector of all the attribute indices under consideration.

Parameters

<i>attrType</i>	attribute type
-----------------	----------------

Returns

vector of indices into currently considered discrete attributes

Definition at line 1842 of file Dataset.cpp.

6.24.3.72 MaskGetAttributeMask()

```
const map< string, uint > & Dataset::MaskGetAttributeMask (   
    AttributeType attrType )
```

Return a map of attribute name to attribute index of attributes to include.

Parameters

in	<i>attrType</i>	attribute type
----	-----------------	----------------

Returns

attributes mask: name->index

Definition at line 1859 of file Dataset.cpp.

6.24.3.73 MaskGetInstanceIds()

```
vector< string > Dataset::MaskGetInstanceIds ( )
```

Return a vector of all the instance ids under consideration.

Returns

vector of ids of currently included instances

Definition at line 1924 of file Dataset.cpp.

6.24.3.74 MaskGetInstanceIndices()

```
vector< uint > Dataset::MaskGetInstanceIndices ( )
```

Return a vector of all the instance indices under consideration.

vector of indices into current instances

Definition at line 1915 of file Dataset.cpp.

6.24.3.75 MaskGetInstanceMask()

```
const map< string, uint > & Dataset::MaskGetInstanceMask ( )
```

Return a map of instance name to instance index of instances to include.

Returns

instances mask: instance ID=>vector of instance indices

Definition at line 1933 of file Dataset.cpp.

6.24.3.76 MaskIncludeAllAttributes()

```
bool Dataset::MaskIncludeAllAttributes (
    AttributeType attrType )
```

Mark all attributes for inclusion in data set operations.

Parameters

in	<i>attrType</i>	attribute type
----	-----------------	----------------

Returns

success

Definition at line 1820 of file Dataset.cpp.

6.24.3.77 MaskIncludeAllInstances()

```
bool Dataset::MaskIncludeAllInstances ( )
```

Mark all instances for inclusion in algorithms.

Returns

success

Definition at line 1901 of file Dataset.cpp.

6.24.3.78 MaskPopAll()

```
bool Dataset::MaskPopAll ( )
```

Restore the masks previously pushed.

Returns

success

Definition at line 1951 of file Dataset.cpp.

6.24.3.79 MaskPushAll()

```
bool Dataset::MaskPushAll ( )
```

Save the current masks for later restore.

Returns

success

Definition at line 1937 of file Dataset.cpp.

6.24.3.80 MaskRemoveInstance()

```
bool Dataset::MaskRemoveInstance (
    std::string instanceId )
```

Removes the instance from consideration in any data set operations.

Parameters

in	<i>instanceId</i>	instance ID
----	-------------------	-------------

Returns

success

Definition at line 1880 of file Dataset.cpp.

6.24.3.81 MaskRemoveVariable()

```
bool Dataset::MaskRemoveVariable (
    std::string variableName )
```

Removes the variable name from consideration in any data set operations.

Parameters

in	<i>variableName</i>	variable name
----	---------------------	---------------

Returns

success

Definition at line 1765 of file Dataset.cpp.

6.24.3.82 MaskRemoveVariableType()

```
bool Dataset::MaskRemoveVariableType (
    std::string variableName,
    AttributeType varType )
```

Removes the attribute name from consideration in any data set operations.

Parameters

in	<i>attributeName</i>	attribute name
in	<i>attrType</i>	attribute type

Returns

success

Definition at line 1775 of file Dataset.cpp.

6.24.3.83 MaskSearchInstance()

```
bool Dataset::MaskSearchInstance (
    std::string instanceId )
```

Determines if the names Instance is in the current masked dataset.

Parameters

in	<i>instanceID</i>	instance ID
----	-------------------	-------------

Returns

true if instance ID is in the dataset, considering instance mask

Definition at line 1892 of file Dataset.cpp.

6.24.3.84 MaskSearchVariableType()

```
bool Dataset::MaskSearchVariableType (
    std::string variableName,
    AttributeType attrType )
```

Determines if the named variable is in the current masked data set.

Parameters

in	<i>attributeName</i>	attribute name
in	<i>attributeType</i>	attribute type

Returns

true if discrete attribute name is being considered in operations.

Definition at line 1800 of file Dataset.cpp.

6.24.3.85 MaskWriteNewDataset()

```
bool Dataset::MaskWriteNewDataset (
    std::string newDatasetFilename )
```

Saved the unmasked attributes as a tab-delimited text file.

Parameters

in	<i>newDatasetFilename</i>	new data set filename
----	---------------------------	-----------------------

Returns

success

Definition at line 1964 of file Dataset.cpp.

6.24.3.86 NumAttributes()

```
uint Dataset::NumAttributes () [virtual]
```

Return the number of unmasked discrete attributes in the data set.

Definition at line 1074 of file Dataset.cpp.

6.24.3.87 NumClasses()

```
uint Dataset::NumClasses ()
```

Normalize numerics by column sums. Get the number of classes in the data set.

Definition at line 1411 of file Dataset.cpp.

6.24.3.88 NumInstances()

```
uint Dataset::NumInstances () [virtual]
```

Returns the number of instances in the data set.

Definition at line 1038 of file Dataset.cpp.

6.24.3.89 NumLevels()

```
uint Dataset::NumLevels (
    uint index )
```

Returns the number of levels in a given attribute index.

Parameters

in	<i>index</i>	attribute index
----	--------------	-----------------

Returns

number of levels

Definition at line 1285 of file Dataset.cpp.

6.24.3.90 NumNumerics()

```
uint Dataset::NumNumerics ()
```

Return the number of unmasked discrete attributes in the data set.

Definition at line 1304 of file Dataset.cpp.

6.24.3.91 NumVariables()

```
uint Dataset::NumVariables ()
```

Return the number of discrete plus continuous variables in the data set.

The number does not include masked variables removed.

Returns

number of discrete plus continuous variables

Definition at line 1022 of file Dataset.cpp.

6.24.3.92 Print()

```
void Dataset::Print ()
```

Print the entire data set in compact format.

Definition at line 1468 of file Dataset.cpp.

6.24.3.93 PrintAttributeLevelsSeen()

```
void Dataset::PrintAttributeLevelsSeen ( )
```

Print unique attribute levels seen.

Definition at line 1750 of file Dataset.cpp.

6.24.3.94 PrintClassIndexInfo()

```
void Dataset::PrintClassIndexInfo ( std::ostream & outStream = std::cout )
```

Print class index information.

Definition at line 1614 of file Dataset.cpp.

6.24.3.95 PrintLevelCounts()

```
void Dataset::PrintLevelCounts ( )
```

Print attribute level counts.

Definition at line 1663 of file Dataset.cpp.

6.24.3.96 PrintMaskStats()

```
void Dataset::PrintMaskStats ( )
```

Print mask statistics.

Definition at line 2014 of file Dataset.cpp.

6.24.3.97 PrintMissingValuesStats()

```
void Dataset::PrintMissingValuesStats ( )
```

Print missing value statistics.

Definition at line 1630 of file Dataset.cpp.

6.24.3.98 PrintNumericsStats()

```
void Dataset::PrintNumericsStats ( )
```

Print statistics about the data set including numerics.

Definition at line 1536 of file Dataset.cpp.

6.24.3.99 PrintStats()

```
void Dataset::PrintStats ( )
```

Print basic statistics about the data set - discrete/SNPs only.

Definition at line 1496 of file Dataset.cpp.

6.24.3.100 PrintStatsSimple()

```
void Dataset::PrintStatsSimple ( std::ostream & outStream = std::cout )
```

Print very simple statistics about the data set with no formatting.

Definition at line 1578 of file Dataset.cpp.

6.24.3.101 ProcessExclusionFile()

```
bool Dataset::ProcessExclusionFile ( std::string exclusionFilename )
```

Remove file of attribute names from consideration in analyses.

Parameters

in	<i>exclusionFilename</i>	filename of attributes to exclude
----	--------------------------	-----------------------------------

Returns

SUCCESS

Definition at line 1205 of file Dataset.cpp.

6.24.3.102 ResetNearestNeighbors()

```
bool Dataset::ResetNearestNeighbors ( )
```

Reset instances nearest neighbor information.

Definition at line 2928 of file Dataset.cpp.

6.24.3.103 RunSnpDiagnosticTests()

```
void Dataset::RunSnpDiagnosticTests (
    std::string logFilename,
    double globalGenotypeThreshold = 0.01,
    uint cellThreshold = 5 )
```

Perform and report SNP diagnostic test information.

Parameters

in	<i>logFilename</i>	log filename
in	<i>globalGenotypeThreshold</i>	genotype count threshold
in	<i>cellThreshold</i>	x^2 cell count threshold

open the diagnostic log file

write diagnostic log information collected in screwySnps to file

Definition at line 2023 of file Dataset.cpp.

6.24.3.104 SetDistanceMetrics()

```
bool Dataset::SetDistanceMetrics (
    std::string newSnpDiffMetricName,
    std::string newSnpNNMetricName,
    std::string newNumMetricName = "manhattan" )
```

Set the the distance metrics used to compute instance-to-instance distances.

Parameters

in	<i>newSnpWeightMetric</i>	name of SNP metric for diff
in	<i>newSnpNNMetric</i>	name of SNP metric for nearest neighbors
in	<i>newNumMetric</i>	name of the numeric metric

Returns

distance

set the SNP metric function pointer

set the nearest neighbors metric

Definition at line 2793 of file Dataset.cpp.

6.24.3.105 SNPHWE()

```
double Dataset::SNPHWE (
    int obs_hets,
    int obs_hom1,
    int obs_hom2 )
```

This code implements an exact SNP test of Hardy-Weinberg Equilibrium.

As described in Wigginton, JE, Cutler, DJ, and Abecasis, GR (2005) A Note on Exact Tests of Hardy-Weinberg Equilibrium. American Journal of Human Genetics: 76. Written by Jan Wigginton.

Parameters

in	<i>obs_hets</i>	observed heterozygotes
in	<i>obs_hom1</i>	observed homozygotes type 1
in	<i>obs_hom2</i>	homozygotes type 2

Returns

HWE value

Definition at line 2296 of file Dataset.cpp.

6.24.3.106 SwapAttributes()

```
bool Dataset::SwapAttributes (
    uint a1,
    uint a2 )
```

Swap two attributes/columns in the dataset.

Parameters

in	a1	attribute index 1
in	a2	attribute index 2

Returns

success

Definition at line 1013 of file Dataset.cpp.

6.24.3.107 UpdateAllLevelCounts()

```
void Dataset::UpdateAllLevelCounts ( ) [protected]
```

Update level counts for all instances by calling UpdateLevelCounts(inst)

initialize level count maps to contain at least three levels

Definition at line 3277 of file Dataset.cpp.

6.24.3.108 UpdateLevelCounts()

```
void Dataset::UpdateLevelCounts (
    DatasetInstance * dsi ) [protected]
```

Update all attribute level counts from one data set instance.

Updates levelCountsByClass.

Parameters

in	dsi	pointer to a data set instance
----	-----	--------------------------------

Definition at line 3312 of file Dataset.cpp.

6.24.3.109 WriteLevelCounts()

```
void Dataset::WriteLevelCounts (
    std::string levelsFilename )
```

Write attribute level counts to a text file.

Parameters

in	<i>levelsFilename</i>	filename to write levels to
----	-----------------------	-----------------------------

Definition at line 1679 of file Dataset.cpp.

6.24.3.110 WriteNewDataset() [1/2]

```
bool Dataset::WriteNewDataset (
    std::string newDatasetFilename,
    OutputDatasetType outputDatasetType )
```

Write the data set to a new filename, respecting masked attributes and numerics and class/phenotype data type.

Parameters

in	<i>newDatasetFilename</i>	new data set filename
in	<i>outputDatasetType</i>	type of file to write

Returns

success

write the attribute names header

write the data, respecting the masked attributes, numerics and masked instances - 10/28/11 write the attribute names header

write continuous attribute values

Definition at line 534 of file Dataset.cpp.

6.24.3.111 WriteNewDataset() [2/2]

```
bool Dataset::WriteNewDataset (
    std::string newDatasetFilename,
    std::vector< std::string > attributes,
    OutputDatasetType outputDatasetType )
```

Write the data set to a new filename, writing only the names in the passed attributes list and also respecting masked attributes and numerics and class/phenotype data type.

Parameters

in	<i>newDatasetFilename</i>	new data set filename
in	<i>attributes</i>	list of attribute names to write
in	<i>outputDatasetType</i>	type of file to write

Returns

success

write the attribute names header

is this attribute in the list passed in as a parameter

write the data, respecting the masked attributes, numerics and masked instances - 10/28/11 write the attribute names header

write continuous attribute values

Definition at line 729 of file Dataset.cpp.

6.24.3.112 WriteNewPlinkCovarDataset()

```
bool Dataset::WriteNewPlinkCovarDataset (
    std::string baseDatasetFilename ) [protected]
```

Write the dataset to a new PLINK covar format, respecting masked attributes class/phenotype data type.

Parameters

in	<i>baseDatasetFilename</i>	base data set filename without extension
----	----------------------------	--

Returns

success

Definition at line 4033 of file Dataset.cpp.

6.24.3.113 WriteNewPlinkPedDataset()

```
bool Dataset::WriteNewPlinkPedDataset (
    std::string baseDatasetFilename ) [protected]
```

Write the dataset to a new PLINK PED/MAP format, respecting masked attributes class/phenotype data type.

Parameters

in	<i>baseDatasetFilename</i>	base data set filename without extension
----	----------------------------	--

Returns

success

Definition at line 3938 of file Dataset.cpp.

6.24.3.114 WriteSnpTiTvInfo()

```
bool Dataset::WriteSnpTiTvInfo (
    std::string titvFilename )
```

Dump the SNP transition/transversion information to file.

Definition at line 2907 of file Dataset.cpp.

6.24.4 Member Data Documentation

6.24.4.1 alternatePhenotypesFilename

```
std::string Dataset::alternatePhenotypesFilename [protected]
```

file from which the alternate phenotypes (class labels) were read

Definition at line 751 of file Dataset.h.

6.24.4.2 attributeAlleleCounts

```
std::vector<std::map<char, uint>> Dataset::attributeAlleleCounts [protected]
```

allele->count

Definition at line 723 of file Dataset.h.

6.24.4.3 attributeAlleles

```
std::vector<std::pair<char, char>> Dataset::attributeAlleles [protected]
```

allele1, allele2

Definition at line 721 of file Dataset.h.

6.24.4.4 attributeLevelsSeen

```
std::vector<std::set<std::string>> Dataset::attributeLevelsSeen [protected]
```

unique attribute values/levels read from file

Definition at line 719 of file Dataset.h.

6.24.4.5 attributeMinorAllele

```
std::vector<std::pair<char, double>> Dataset::attributeMinorAllele [protected]
```

minor allele, minor allele frequency

Definition at line 725 of file Dataset.h.

6.24.4.6 attributeMutationMap

```
std::map<std::pair<char, char>, AttributeMutationType> Dataset::attributeMutationMap [protected]
```

Lookup table for mutation type.

Definition at line 733 of file Dataset.h.

6.24.4.7 attributeMutationTypes

```
std::vector<AttributeMutationType> Dataset::attributeMutationTypes [protected]
```

Keep mutation type for all attributes.

Definition at line 731 of file Dataset.h.

6.24.4.8 attributeNames

```
std::vector<std::string> Dataset::attributeNames [protected]
```

discrete attribute names read from file

Definition at line 713 of file Dataset.h.

6.24.4.9 attributesMask

```
std::map<std::string, uint> Dataset::attributesMask [protected]
```

Definition at line 782 of file Dataset.h.

6.24.4.10 attributesMaskPushed

```
std::map<std::string, uint> Dataset::attributesMaskPushed [protected]
```

masks can be temporarily pushed and popped

Definition at line 786 of file Dataset.h.

6.24.4.11 classColumn

```
uint Dataset::classColumn [protected]
```

class column from the original data set

Definition at line 773 of file Dataset.h.

6.24.4.12 classIndexes

```
std::map<ClassLevel, std::vector<uint> > Dataset::classIndexes [protected]
```

class values mapped to instance indices

Definition at line 775 of file Dataset.h.

6.24.4.13 continuousPhenotypeMinMax

```
std::pair<NumericLevel, NumericLevel> Dataset::continuousPhenotypeMinMax [protected]
```

the minimum and maximum value for each continuous phenotype

Definition at line 759 of file Dataset.h.

6.24.4.14 engine

```
std::mt19937_64 Dataset::engine [protected]
```

Definition at line 792 of file Dataset.h.

6.24.4.15 genotypeCounts

```
std::vector<std::map<std::string, uint> > Dataset::genotypeCounts [protected]
```

genotype->count

Definition at line 729 of file Dataset.h.

6.24.4.16 hasAllelicInfo

```
bool Dataset::hasAllelicInfo [protected]
```

Does this data set have allelic information?

Definition at line 727 of file Dataset.h.

6.24.4.17 hasAlternatePhenotypes

```
bool Dataset::hasAlternatePhenotypes [protected]
```

does the data set contain alternate phenotypes?

Definition at line 753 of file Dataset.h.

6.24.4.18 hasContinuousPhenotypes

```
bool Dataset::hasContinuousPhenotypes [protected]
```

does the data set contain continuous phenotypes?

Definition at line 757 of file Dataset.h.

6.24.4.19 hasGenotypes

```
bool Dataset::hasGenotypes [protected]
```

does the data set contain any genotypes?

Definition at line 711 of file Dataset.h.

6.24.4.20 hasNumerics

```
bool Dataset::hasNumerics [protected]
```

does the data set contain any continuous attributes?

Definition at line 738 of file Dataset.h.

6.24.4.21 hasPhenotypes

```
bool Dataset::hasPhenotypes [protected]
```

Does the data set contain phenotypes?

Definition at line 749 of file Dataset.h.

6.24.4.22 instanceIds

```
std::vector<std::string> Dataset::instanceIds [protected]
```

IDs associated with the instances read from file.

Definition at line 764 of file Dataset.h.

6.24.4.23 instanceIdsToLoad

```
std::vector<std::string> Dataset::instanceIdsToLoad [protected]
```

IDs of instances to load from numeric and/or phenotype files.

Definition at line 766 of file Dataset.h.

6.24.4.24 instances

```
std::vector<DatasetInstance*> Dataset::instances [protected]
```

vector of pointers to all instances in the data set

Definition at line 762 of file Dataset.h.

6.24.4.25 instancesMask

```
std::map<std::string, uint> Dataset::instancesMask [protected]
```

Definition at line 784 of file Dataset.h.

6.24.4.26 instancesMaskPushed

```
std::map<std::string, uint> Dataset::instancesMaskPushed [protected]
```

Definition at line 788 of file Dataset.h.

6.24.4.27 levelCounts

```
std::vector<std::map<AttributeLevel, uint>> Dataset::levelCounts [protected]
```

attribute values/levels counts

Definition at line 715 of file Dataset.h.

6.24.4.28 levelCountsByClass

```
std::vector<std::map<std::pair<AttributeLevel, ClassLevel>, uint>> Dataset::levelCountsByClass [protected]
```

attribute values/levels counts by discrete class

Definition at line 717 of file Dataset.h.

6.24.4.29 maskIsPushed

```
bool Dataset::maskIsPushed [protected]
```

Definition at line 789 of file Dataset.h.

6.24.4.30 missingNumericValues

```
std::map<std::string, std::vector<uint>> Dataset::missingNumericValues [protected]
```

missing continuous values and their instance indices

Definition at line 770 of file Dataset.h.

6.24.4.31 missingValues

```
std::map<std::string, std::vector<uint>> Dataset::missingValues [protected]
```

missing discrete values and their instance indices

Definition at line 768 of file Dataset.h.

6.24.4.32 numDiffFuncPtr

```
double(* Dataset::numDiffFuncPtr) (uint attributeIndex, DatasetInstance *dsi1, DatasetInstance *dsi2) [protected]
```

Compute the continuous difference in an attribute between two instances.

Parameters

in	<i>attributeIndex</i>	index into vector of all attributes
in	<i>dsi1</i>	pointer to DatasetInstance 1
in	<i>dsi2</i>	pointer to DatasetInstance 2

Returns

`diff(erence)`

Definition at line 697 of file `Dataset.h`.

6.24.4.33 numDiffMetricName

`std::string Dataset::numDiffMetricName [protected]`

the name of continuous `diff(erence)` function

Definition at line 706 of file `Dataset.h`.

6.24.4.34 numericsFilename

`std::string Dataset::numericsFilename [protected]`

file from which the continuous attributes were read

Definition at line 736 of file `Dataset.h`.

6.24.4.35 numericsIds

`std::vector<std::string> Dataset::numericsIds [protected]`

IDs associated with the numerics read from file.

Definition at line 740 of file `Dataset.h`.

6.24.4.36 numericsMask

```
std::map<std::string, uint> Dataset::numericsMask [protected]
```

Definition at line 783 of file Dataset.h.

6.24.4.37 numericsMaskPushed

```
std::map<std::string, uint> Dataset::numericsMaskPushed [protected]
```

Definition at line 787 of file Dataset.h.

6.24.4.38 numericsMinMax

```
std::vector< std::pair<NumericLevel, NumericLevel> > Dataset::numericsMinMax [protected]
```

the minimum and maximum value for each continuous attribute

Definition at line 742 of file Dataset.h.

6.24.4.39 numericsNames

```
std::vector<std::string> Dataset::numericsNames [protected]
```

continuous attribute names read from file

Definition at line 746 of file Dataset.h.

6.24.4.40 numericsSums

```
std::vector<double> Dataset::numericsSums [protected]
```

the column sum for each continuous attribute

Definition at line 744 of file Dataset.h.

6.24.4.41 phenotypesIds

```
std::vector<std::string> Dataset::phenotypesIds [protected]
```

IDs associated with the phenotypes/classes read from file.

Definition at line 755 of file Dataset.h.

6.24.4.42 snpDiffFuncPtr

```
double(* Dataset::snpDiffFuncPtr) (uint attributeIndex, DatasetInstance *dsi1, DatasetInstance *dsi2) [protected]
```

Compute the discrete difference in an attribute between two instances for Relief-F scoring.

Parameters

in	<i>attributeIndex</i>	index into vector of all attributes
in	<i>dsi1</i>	pointer to DatasetInstance 1
in	<i>dsi2</i>	pointer to DatasetInstance 2

Returns`diff(erence)`

Definition at line 676 of file `Dataset.h`.

6.24.4.43 `snpDiffMetricName`

```
std::string Dataset::snpDiffMetricName [protected]
```

the name of discrete `diff(erence)` function

Definition at line 702 of file `Dataset.h`.

6.24.4.44 `snpNearestNeighborFuncPtr`

```
double(* Dataset::snpNearestNeighborFuncPtr) (uint attributeIndex, DatasetInstance *dsi1, DatasetInstance *dsi2) [protected]
```

Compute the discrete difference in an attribute between two instances for determining nearest neighbors.

Parameters

in	<i>attributeIndex</i>	index into vector of all attributes
in	<i>dsi1</i>	pointer to DatasetInstance 1
in	<i>dsi2</i>	pointer to DatasetInstance 2

Returns`diff(erence)`

Definition at line 687 of file `Dataset.h`.

6.24.4.45 snpNearestNeighborMetricName

```
std::string Dataset::snpNearestNeighborMetricName [protected]
```

the name of discrete distance function for nearest neighbors

Definition at line 704 of file Dataset.h.

6.24.4.46 snpsFilename

```
std::string Dataset::snpsFilename [protected]
```

file from which the discrete attributes (SNPs) were read

Definition at line 709 of file Dataset.h.

The documentation for this class was generated from the following files:

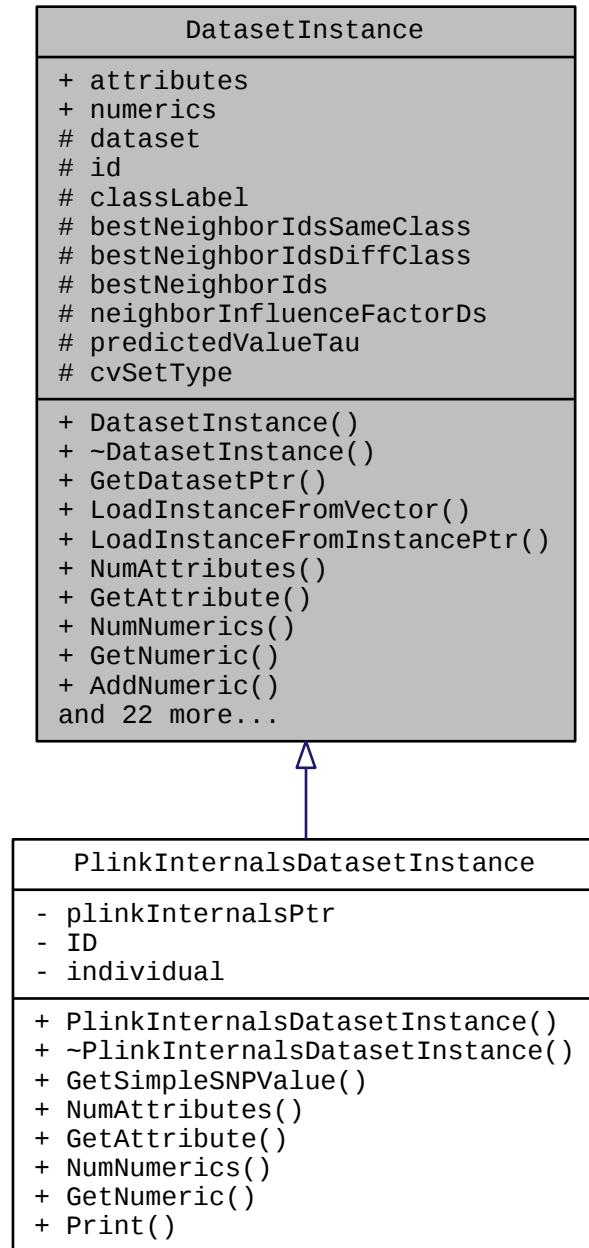
- src/[Dataset.h](#)
- src/[Dataset.cpp](#)

6.25 DatasetInstance Class Reference

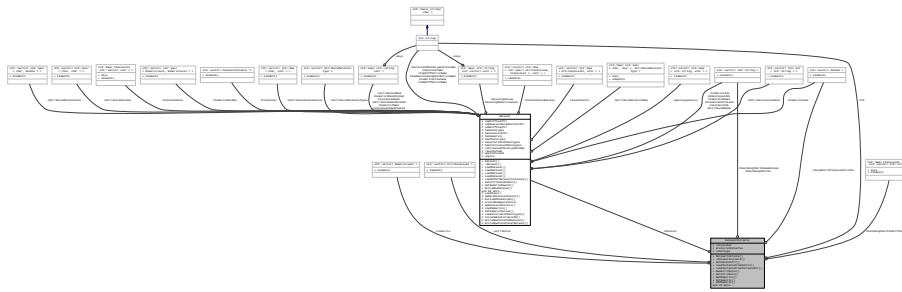
Class to hold dataset instances (rows of attributes).

```
#include <DatasetInstance.h>
```

Inheritance diagram for DatasetInstance:



Collaboration diagram for DatasetInstance:



Public Member Functions

- **`DatasetInstance (Dataset *ds, std::string newId)`**
Construct an data set instance object.
- **`virtual ~DatasetInstance ()`**
- **`Dataset * GetDatasetPtr ()`**
return the `Dataset` pointer associated with this instance
- **`bool LoadInstanceFromVector (std::vector< AttributeLevel > newAttributes)`**
Load this instance with the attributes and class value from the newAttributes vector.
- **`bool LoadInstanceFromInstancePtr (Dataset *srcDs, DatasetInstance *srcInstance)`**
- **`virtual unsigned int NumAttributes ()`**
return the number of discrete attributes
- **`virtual AttributeLevel GetAttribute (unsigned int index)`**
Get and return an attribute value at index.
- **`virtual unsigned int NumNumerics ()`**
return the number of continuous attributes
- **`virtual NumericLevel GetNumeric (unsigned int index)`**
Get and return numeric value at index.
- **`bool AddNumeric (NumericLevel newNum)`**
Add a numeric value to the instance's numerics vector.
- **`ClassLevel GetClass ()`**
Get the discrete class value.
- **`void SetClass (ClassLevel classValue)`**
Set the discrete class value.
- **`double GetPredictedValueTau ()`**
Get the continuous class value.
- **`void SetPredictedValueTau (double newValue)`**
Set the continuous class value.
- **`double GetInfluenceFactorD (unsigned int neighborIndex)`**
Get the nearest neighbor value at neighborIndex.
- **`void ClearInfluenceFactors ()`**
Clear all nearest neighbor values.
- **`bool AddInfluenceFactorD (double factor)`**
Add the next nearest neighbor influence factor.

- virtual void [Print \(\)](#)
Print the attributes, numerics and class name of this instance to stdout.
- bool [SwapAttributes \(unsigned int a1, unsigned int a2\)](#)
Swap attribute/column values in this instance.
- void [SetDistanceSums \(unsigned int kNearestNeighbors, DistancePairs &sameClassSums, std::map< ClassLevel, DistancePairs > &diffClassSums\)](#)
Set the best kNearestNeighbors from the same and different classes SIDE_EFFECT: Sorts and loads class the variables: sameSums and diffSums from the neighbors.
- void [SetDistanceSums \(unsigned int kNearestNeighbors, DistancePairs instancesSums\)](#)
Set the best kNearestNeighbors from all other instances/neighbors.
- void [PrintDistancePairs \(const DistancePairs &distPairs\)](#)
Prints passed distance pairs.
- bool [GetNNearestInstances \(unsigned int n, std::vector< unsigned int > &sameClassInstances, std::vector< unsigned int > &diffClassInstances\)](#)
Returns N closest instances using the sameSums and diffSums class variables.
- bool [GetNNearestInstances \(unsigned int n, std::vector< unsigned int > &sameClassInstances, std::map< ClassLevel, std::vector< unsigned int > > &diffClassInstances\)](#)
Returns N closest instances using the sameSums and diffSums class variables.
- bool [GetNNearestInstances \(unsigned int n, std::vector< unsigned int > &closestInstances\)](#)
Returns N closest instances to this instance.
- bool [ResetNearestNeighbors \(\)](#)
Reset nearest neighbor information.
- bool [SetCvSetType \(CvSetType newType\)](#)
Set cross-validation set type.
- [CvSetType GetCvSetType \(\)](#)
Get cross-validation set type.
- std::vector< AttributeLevel > [GetAttributes \(\)](#)
- std::vector< NumericLevel > [GetNumerics \(\)](#)
- bool [SetId \(std::string newId\)](#)
- std::string [GetId \(\)](#)

Public Attributes

- std::vector< AttributeLevel > [attributes](#)
discrete attributes
- std::vector< NumericLevel > [numerics](#)
continuous attributes

Protected Attributes

- [Dataset * dataset](#)
pointer to a [Dataset](#) object
- std::string [id](#)
unique identifier
- [ClassLevel classLabel](#)
the class value for this instance

- std::vector< std::string > **bestNeighborhoodsSameClass**
vector of instance IDs for the best neighbors in this instance's class
- std::map< **ClassLevel**, std::vector< std::string > > **bestNeighborhoodsDiffClass**
vector of instance IDs for the best neighbors of different class(es)
- std::vector< std::string > **bestNeighbors**
best neighbor IDs for continuous class
- std::vector< double > **neighborInfluenceFactorDs**
nearest neighbor weighting factors
- double **predictedValueTau**
continuous value for this class
- CvSetType **cvsetType**
classification dataset type: train | holdout | test

6.25.1 Detailed Description

Class to hold dataset instances (rows of attributes).

Reworked entirely for McKinney Lab work - 2/28/11

Author

Bill White

Version

1.0

Contact: bill.c.white@gmail.com Created on: 6/14/05

Definition at line 28 of file DatasetInstance.h.

6.25.2 Constructor & Destructor Documentation

6.25.2.1 DatasetInstance()

```
DatasetInstance::DatasetInstance (
    Dataset * ds,
    std::string newId )
```

Construct an data set instance object.

Parameters

in	<i>ds</i>	pointer to a Dataset object
----	-----------	---

Definition at line 34 of file [DatasetInstance.cpp](#).

6.25.2.2 ~DatasetInstance()

```
DatasetInstance::~DatasetInstance ( ) [virtual]
```

Definition at line 42 of file [DatasetInstance.cpp](#).

6.25.3 Member Function Documentation**6.25.3.1 AddInfluenceFactorD()**

```
bool DatasetInstance::AddInfluenceFactorD (
    double factor )
```

Add the next nearest neighbor influence factor.

Definition at line 150 of file [DatasetInstance.cpp](#).

6.25.3.2 AddNumeric()

```
bool DatasetInstance::AddNumeric (
    NumericLevel newNum )
```

Add a numeric value to the instance's numerics vector.

Parameters

in	<i>newNum</i>	new numeric value
----	---------------	-------------------

Returns

success

Definition at line 120 of file DatasetInstance.cpp.

6.25.3.3 ClearInfluenceFactors()

```
void DatasetInstance::ClearInfluenceFactors ( )
```

Clear all nearest neighbor values.

Definition at line 146 of file DatasetInstance.cpp.

6.25.3.4 GetAttribute()

```
AttributeLevel DatasetInstance::GetAttribute (
    unsigned int index ) [virtual]
```

Get and return an attribute value at index.

Parameters

in	<i>index</i>	attribute index
----	--------------	-----------------

Returns

attribute value at index

Reimplemented in [PlinkInternalsDatasetInstance](#).

Definition at line 85 of file DatasetInstance.cpp.

6.25.3.5 GetAttributes()

```
std::vector<AttributeLevel> DatasetInstance::GetAttributes ( ) [inline]
```

Definition at line 154 of file DatasetInstance.h.

6.25.3.6 GetClass()

```
ClassLevel DatasetInstance::GetClass ( )
```

Get the discrete class value.

Definition at line 126 of file DatasetInstance.cpp.

6.25.3.7 GetCvSetType()

```
CvSetType DatasetInstance::GetCvSetType ( )
```

Get cross-validation set type.

Definition at line 368 of file DatasetInstance.cpp.

6.25.3.8 GetDatasetPtr()

```
Dataset * DatasetInstance::GetDatasetPtr ( )
```

return the [Dataset](#) pointer associated with this instance

Definition at line 45 of file DatasetInstance.cpp.

6.25.3.9 GetId()

```
std::string DatasetInstance::GetId ( ) [inline]
```

Definition at line 157 of file DatasetInstance.h.

6.25.3.10 GetInfluenceFactorD()

```
double DatasetInstance::GetInfluenceFactorD ( unsigned int neighborIndex )
```

Get the nearest neighbor value at neighborIndex.

Definition at line 142 of file DatasetInstance.cpp.

6.25.3.11 GetNNearlestInstances() [1/3]

```
bool DatasetInstance::GetNNearlestInstances ( unsigned int n, std::vector< unsigned int > & sameClassInstances, std::vector< unsigned int > & diffClassInstances )
```

Returns N closest instances using the sameSums and diffSums class variables.

Parameters

in	<i>n</i>	n nearest neighbors
in	<i>sameClassInstances</i>	vector of same class instances indices
in	<i>diffClassInstances</i>	vector of different class instance indices

Returns

success

Definition at line 260 of file DatasetInstance.cpp.

6.25.3.12 GetNNeariestInstances() [2/3]

```
bool DatasetInstance::GetNNeariestInstances (
    unsigned int n,
    std::vector< unsigned int > & sameClassInstances,
    std::map< ClassLevel, std::vector< unsigned int > > & diffClassInstances )
```

Returns N closest instances using the sameSums and diffSums class variables.

Parameters

in	<i>n</i>	n nearest neighbors
in	<i>sameClassInstances</i>	vector of same class instances indices
in	<i>diffClassInstances</i>	vector of different classes instance indices

Returns

success

Definition at line 285 of file DatasetInstance.cpp.

6.25.3.13 GetNNeariestInstances() [3/3]

```
bool DatasetInstance::GetNNeariestInstances (
    unsigned int n,
    std::vector< unsigned int > & closestInstances )
```

Returns N closest instances to this instance.

Parameters

in	<i>n</i>	n nearest neighbors
in	<i>closestInstances</i>	reference to a vector of instance indices

Returns

success

Definition at line 324 of file DatasetInstance.cpp.

6.25.3.14 GetNumeric()

```
double DatasetInstance::GetNumeric (
    unsigned int index ) [virtual]
```

Get and return numeric value at index.

Parameters

in	<i>index</i>	numeric index
----	--------------	---------------

Returns

numeric value at index

Reimplemented in [PlinkInternalsDatasetInstance](#).

Definition at line 105 of file DatasetInstance.cpp.

6.25.3.15 GetNumerics()

```
std::vector<NumericLevel> DatasetInstance::GetNumerics ( ) [inline]
```

Definition at line 155 of file DatasetInstance.h.

6.25.3.16 GetPredictedValueTau()

```
double DatasetInstance::GetPredictedValueTau ( )
```

Get the continuous class value.

Definition at line 134 of file DatasetInstance.cpp.

6.25.3.17 LoadInstanceFromInstancePtr()

```
bool DatasetInstance::LoadInstanceFromInstancePtr (
    Dataset * srcDs,
    DatasetInstance * srcInstance )
```

Definition at line 64 of file DatasetInstance.cpp.

6.25.3.18 LoadInstanceFromVector()

```
bool DatasetInstance::LoadInstanceFromVector (
    std::vector< AttributeLevel > newAttributes )
```

Load this instance with the attributes and class value from the newAttributes vector.

Parameters

in	<i>newAttributes</i>	vector of new attribute values
----	----------------------	--------------------------------

Returns

success

Definition at line 50 of file DatasetInstance.cpp.

6.25.3.19 NumAttributes()

```
unsigned int DatasetInstance::NumAttributes ( ) [virtual]
```

return the number of discrete attributes

Reimplemented in [PlinkInternalsDatasetInstance](#).

Definition at line 81 of file DatasetInstance.cpp.

6.25.3.20 NumNumerics()

```
unsigned int DatasetInstance::NumNumerics ( ) [virtual]
```

return the number of continuous attributes

Reimplemented in [PlinkInternalsDatasetInstance](#).

Definition at line 101 of file DatasetInstance.cpp.

6.25.3.21 Print()

```
void DatasetInstance::Print ( ) [virtual]
```

Print the attributes, numerics and class name of this instance to stdout.

Reimplemented in [PlinkInternalsDatasetInstance](#).

Definition at line 155 of file DatasetInstance.cpp.

6.25.3.22 PrintDistancePairs()

```
void DatasetInstance::PrintDistancePairs ( const DistancePairs & distPairs )
```

Prints passed distance pairs.

Parameters

in	<i>distPairs</i>	distance pairs
----	------------------	----------------

Definition at line 254 of file DatasetInstance.cpp.

6.25.3.23 ResetNearestNeighbors()

```
bool DatasetInstance::ResetNearestNeighbors ( )
```

Reset nearest neighbor information.

Definition at line 346 of file DatasetInstance.cpp.

6.25.3.24 SetClass()

```
void DatasetInstance::SetClass (
    ClassLevel classValue )
```

Set the discrete class value.

Definition at line 130 of file DatasetInstance.cpp.

6.25.3.25 SetCvSetType()

```
bool DatasetInstance::SetCvSetType (
    CvSetType newType )
```

Set cross-validation set type.

Definition at line 354 of file DatasetInstance.cpp.

6.25.3.26 SetDistanceSums() [1/2]

```
void DatasetInstance::SetDistanceSums (
    unsigned int kNearestNeighbors,
    DistancePairs & sameClassSums,
    std::map< ClassLevel, DistancePairs > & diffClassSums )
```

Set the best kNearestNeighbors from the same and different classes **SIDE_EFFECT:** Sorts and loads class the variables: sameSums and diffSums from the neighbors.

Parameters

in	<i>kNearestNeighbors</i>	k nearest neighbors,
in	<i>sameClassSums</i>	vectors of pairs <instance, sum> of same class
in	<i>diffClassSums</i>	vectors of pairs <instance, sum> of other classes

Returns

nothing

Definition at line 194 of file DatasetInstance.cpp.

6.25.3.27 SetDistanceSums() [2/2]

```
void DatasetInstance::SetDistanceSums (
    unsigned int kNearestNeighbors,
    DistancePairs instancesSums )
```

Set the best kNearestNeighbors from all other instances/neighbors.

SIDE_EFFECT: Sorts and loads neighborSums from the instanceSums

Parameters

in	<i>kNearestNeighbors</i>	k nearest neighbors
in	<i>instanceSums</i>	vectors of k pairs <instance, sum> for neighbors

Returns

nothing

Definition at line 229 of file DatasetInstance.cpp.

6.25.3.28 SetId()

```
bool DatasetInstance::SetId (
    std::string newId )
```

Definition at line 76 of file DatasetInstance.cpp.

6.25.3.29 SetPredictedValueTau()

```
void DatasetInstance::SetPredictedValueTau (
    double newValue )
```

Set the continuous class value.

Definition at line 138 of file DatasetInstance.cpp.

6.25.3.30 SwapAttributes()

```
bool DatasetInstance::SwapAttributes (
    unsigned int a1,
    unsigned int a2 )
```

Swap attribute/column values in this instance.

Parameters

in	a1	attribute index 1
in	a2	attribute index 2

Returns

bool success

Definition at line 175 of file DatasetInstance.cpp.

6.25.4 Member Data Documentation

6.25.4.1 attributes

`std::vector<AttributeLevel> DatasetInstance::attributes`

discrete attributes

Definition at line 159 of file DatasetInstance.h.

6.25.4.2 bestNeighborhoods

`std::vector<std::string> DatasetInstance::bestNeighborIds [protected]`

best neighbor IDs for continuous class

Definition at line 178 of file DatasetInstance.h.

6.25.4.3 bestNeighborhoodsDiffClass

`std::map<ClassLevel, std::vector<std::string> > DatasetInstance::bestNeighborIdsDiffClass [protected]`

vector of instance IDs for the best neighbors of different class(es)

Definition at line 176 of file DatasetInstance.h.

6.25.4.4 bestNeighborhoodsSameClass

`std::vector<std::string> DatasetInstance::bestNeighborIdsSameClass [protected]`

vector of instance IDs for the best neighbors in this instance's class

Definition at line 174 of file DatasetInstance.h.

6.25.4.5 classLabel

`ClassLevel DatasetInstance::classLabel [protected]`

the class value for this instance

Definition at line 168 of file DatasetInstance.h.

6.25.4.6 cvSetType

`CvsetType DatasetInstance::cvSetType [protected]`

classification dataset type: train | holdout | test

Definition at line 184 of file DatasetInstance.h.

6.25.4.7 dataset

`Dataset* DatasetInstance::dataset [protected]`

pointer to a [Dataset](#) object

Definition at line 164 of file DatasetInstance.h.

6.25.4.8 id

`std::string DatasetInstance::id [protected]`

unique identifier

Definition at line 166 of file DatasetInstance.h.

6.25.4.9 neighborInfluenceFactorDs

```
std::vector<double> DatasetInstance::neighborInfluenceFactorDs [protected]
```

nearest neighbor weighting factors

Definition at line 180 of file DatasetInstance.h.

6.25.4.10 numerics

```
std::vector<NumericLevel> DatasetInstance::numerics
```

continuous attributes

Definition at line 161 of file DatasetInstance.h.

6.25.4.11 predictedValueTau

```
double DatasetInstance::predictedValueTau [protected]
```

continuous value for this class

Definition at line 182 of file DatasetInstance.h.

The documentation for this class was generated from the following files:

- src/[DatasetInstance.h](#)
- src/[DatasetInstance.cpp](#)

6.26 DcVar Class Reference

```
#include <DcVar.h>
```

Collaboration diagram for DcVar:



Public Member Functions

- [DcVar \(SNP_INPUT_TYPE snpInputTypeParam=SNP_SRC_PLINK\)](#)
- [bool Run \(\)](#)
- [bool ReadTranscriptCoordinates \(std::string coordinatesFile\)](#)
- [bool ReadTranscriptFactorCoordinates \(std::string coordinatesFile\)](#)
- [bool SetRadius \(int newRadius\)](#)
- [int GetRadius \(\)](#)
- [bool SetLocalCis \(bool localCisFlag\)](#)
- [bool GetLocalCis \(\)](#)
- [bool SetTFRadius \(int newRadius\)](#)
- [int GetTFRadius \(\)](#)
- [bool SetTF \(bool tfFlag\)](#)
- [bool GetTF \(\)](#)
- [bool GetTFInfo \(std::string tf, std::vector<int> &tflInfo\)](#)
- [virtual ~DcVar \(\)](#)

Private Member Functions

- bool `RunPlink ()`
- bool `RunOMRF ()`
- bool `RunOMRFChipSeq ()`
- void `PrintState ()`
- bool `ReadGenotypesFile (uint chrom)`
- bool `ReadSnpLocationsFile (uint chrom)`
- bool `ReadGeneExpressionFile ()`
- bool `ReadChipSeqFile ()`
- bool `FindSnps (uint pos, std::vector< uint > &inRadius)`
- bool `MapPhenosToModel (std::vector< uint > phenos, std::string varModel, std::vector< uint > &mappedPhenos)`
- std::pair< `uint, uint > MapSnpIndexToPlinkPhenos (uint snpIndex, std::string varModel)`
- bool `SplitExpressionCaseControl (arma::mat &caseMatrix, arma::mat &ctrlMatrix)`
- bool `ComputeDifferentialCorrelationZsparse (std::string.snp, arma::mat &cases, arma::mat &ctrls)`
- bool `ComputeDifferentialCorrelationZ (std::string.snp, arma::mat &cases, arma::mat &ctrls, double correctedP)`
- bool `ComputeDifferentialCorrelationZvals (std::string.snp, arma::mat &cases, arma::mat &ctrls)`
- bool `FlattenPvals (vector_t &retPvals)`
- bool `FilterPvalues (uint &numFiltered)`
- double `CalculateFdrBHThreshold ()`
- bool `WriteCheckpoint (uint snpIndex, std::string.snpName)`
- bool `ReadCheckpoint (std::pair< uint, string > &lastSnp)`
- bool `WriteResults (std::string.filename, std::string.curSnp)`
- bool `CheckInputs ()`
- bool `GetSnpsForTranscript (std::string.transcript, std::vector< int > &snpIndices)`
- bool `GetSnpsForTFs (std::vector< int > &snpIndices, std::vector< std::string > &tf)`
- bool `LoadDefaultTranscriptionFactorLUT ()`
- bool `IsSnpInTFs (int chr, int bp, std::string &tf)`

Private Attributes

- bool `chipSeqMode`
- `CHIP_SEQ_INFO_LIST chipSeqExpression`
- `SNP_INPUT_TYPE snpInputType`
- `std::vector< std::string > snpNames`
- `SNP_INFO_LIST snpLocations`
- `std::vector< std::string > genotypeSubjects`
- `matrix_t genotypeMatrix`
- `std::vector< std::string > geneExprNames`
- `std::vector< std::string > geneExprSubjects`
- `matrix_t expressionMatrix`
- double `numCombs`
- `uint totalTests`
- `std::vector< uint > caseIdxCol`
- `std::vector< uint > ctrlIdxCol`
- `arma::sp_mat zVals`
- `arma::mat pVals`
- `uint radius`
- bool `localCis`

- CoordinateTable coordinates
- bool tfTableLoaded
- bool tfMode
- int tfRadius
- TranscriptFactorTable transcriptFactorLUT
- std::vector< int > thisTranscriptSnpIndices
- uint nOuterLoop
- uint nInnerLoop
- std::vector< int > thisTFSnpIndices

6.26.1 Detailed Description

Definition at line 62 of file DcVar.h.

6.26.2 Constructor & Destructor Documentation

6.26.2.1 DcVar()

```
DcVar::DcVar (   
    SNP_INPUT_TYPE snpInputTypeParam = SNP_SRC_PLINK )
```

Definition at line 54 of file DcVar.cpp.

6.26.2.2 ~DcVar()

```
DcVar::~DcVar ( ) [virtual]
```

Definition at line 83 of file DcVar.cpp.

6.26.3 Member Function Documentation

6.26.3.1 CalculateFdrBHThreshold()

```
double DcVar::CalculateFdrBHThreshold ( ) [private]
```

Definition at line 1077 of file DcVar.cpp.

6.26.3.2 CheckInputs()

```
bool DcVar::CheckInputs ( ) [private]
```

Definition at line 535 of file DcVar.cpp.

6.26.3.3 ComputeDifferentialCorrelationZ()

```
bool DcVar::ComputeDifferentialCorrelationZ (
    std::string snp,
    arma::mat & cases,
    arma::mat & ctrls,
    double correctedP ) [private]
```

Definition at line 867 of file DcVar.cpp.

6.26.3.4 ComputeDifferentialCorrelationZsparse()

```
bool DcVar::ComputeDifferentialCorrelationZsparse (
    std::string snp,
    arma::mat & cases,
    arma::mat & ctrls ) [private]
```

Definition at line 933 of file DcVar.cpp.

6.26.3.5 ComputeDifferentialCorrelationZvals()

```
bool DcVar::ComputeDifferentialCorrelationZvals (
    std::string snp,
    arma::mat & cases,
    arma::mat & ctrls ) [private]
```

Definition at line 790 of file DcVar.cpp.

6.26.3.6 FilterPvalues()

```
bool DcVar::FilterPvalues (
    uint & numFiltered ) [private]
```

Definition at line 1033 of file DcVar.cpp.

6.26.3.7 FindSnps()

```
bool DcVar::FindSnps (
    uint pos,
    std::vector< uint > & inRadius ) [private]
```

Definition at line 539 of file DcVar.cpp.

6.26.3.8 FlattenPvals()

```
bool DcVar::FlattenPvals (
    vector_t & retPvals ) [private]
```

Definition at line 1021 of file DcVar.cpp.

6.26.3.9 GetLocalCis()

```
bool DcVar::GetLocalCis ( ) [inline]
```

Definition at line 71 of file DcVar.h.

6.26.3.10 GetRadius()

```
int DcVar::GetRadius ( ) [inline]
```

Definition at line 69 of file DcVar.h.

6.26.3.11 GetSnpsForTFs()

```
bool DcVar::GetSnpsForTFs (
    std::vector< int > & snpIndices,
    std::vector< std::string > & tfs ) [private]
```

Definition at line 1252 of file DcVar.cpp.

6.26.3.12 GetSnpsForTranscript()

```
bool DcVar::GetSnpsForTranscript (
    std::string transcript,
    std::vector< int > & snpIndices ) [private]
```

Definition at line 1212 of file DcVar.cpp.

6.26.3.13 GetTF()

```
bool DcVar::GetTF ( ) [inline]
```

Definition at line 76 of file DcVar.h.

6.26.3.14 GetTFIInfo()

```
bool DcVar::GetTFIInfo (
    std::string tf,
    std::vector< int > & tfInfo )
```

Definition at line 1296 of file DcVar.cpp.

6.26.3.15 GetTFRadius()

```
int DcVar::GetTFRadius ( ) [inline]
```

Definition at line 74 of file DcVar.h.

6.26.3.16 IsSnpInTFs()

```
bool DcVar::IsSnpInTFs (
    int chr,
    int bp,
    std::string & tf ) [private]
```

Definition at line 1275 of file DcVar.cpp.

6.26.3.17 LoadDefaultTranscriptionFactorLUT()

```
bool DcVar::LoadDefaultTranscriptionFactorLUT ( ) [private]
```

6.26.3.18 MapPhenosToModel()

```
bool DcVar::MapPhenosToModel (
    std::vector< uint > phenos,
    std::string varModel,
    std::vector< uint > & mappedPhenos ) [private]
```

Definition at line 741 of file DcVar.cpp.

6.26.3.19 MapSnpIndexToPlinkPhenos()

```
pair< uint, uint > DcVar::MapSnpIndexToPlinkPhenos (
    uint snpIndex,
    std::string varModel ) [private]
```

Definition at line 101 of file DcVar.cpp.

6.26.3.20 PrintState()

```
void DcVar::PrintState ( ) [private]
```

Definition at line 569 of file DcVar.cpp.

6.26.3.21 ReadCheckpoint()

```
bool DcVar::ReadCheckpoint (
    std::pair< uint, string > & lastSnp ) [private]
```

Definition at line 1122 of file DcVar.cpp.

6.26.3.22 ReadChipSeqFile()

```
bool DcVar::ReadChipSeqFile ( ) [private]
```

Definition at line 703 of file DcVar.cpp.

6.26.3.23 ReadGeneExpressionFile()

```
bool DcVar::ReadGeneExpressionFile ( ) [private]
```

Definition at line 655 of file DcVar.cpp.

6.26.3.24 ReadGenotypesFile()

```
bool DcVar::ReadGenotypesFile (
    uint chrom ) [private]
```

Definition at line 580 of file DcVar.cpp.

6.26.3.25 ReadSnpLocationsFile()

```
bool DcVar::ReadSnpLocationsFile (
    uint chrom ) [private]
```

Definition at line 621 of file DcVar.cpp.

6.26.3.26 ReadTranscriptCoordinates()

```
bool DcVar::ReadTranscriptCoordinates (
    std::string coordinatesFile )
```

6.26.3.27 ReadTranscriptFactorCoordinates()

```
bool DcVar::ReadTranscriptFactorCoordinates (
    std::string coordinatesFile )
```

6.26.3.28 Run()

```
bool DcVar::Run ( )
```

Definition at line 86 of file DcVar.cpp.

6.26.3.29 RunOMRF()

```
bool DcVar::RunOMRF ( ) [private]
```

Definition at line 278 of file DcVar.cpp.

6.26.3.30 RunOMRFChipSeq()

```
bool DcVar::RunOMRFChipSeq ( ) [private]
```

Definition at line 393 of file DcVar.cpp.

6.26.3.31 RunPlink()

```
bool DcVar::RunPlink ( ) [private]
```

Definition at line 175 of file DcVar.cpp.

6.26.3.32 SetLocalCis()

```
bool DcVar::SetLocalCis ( bool localCisFlag )
```

Definition at line 1192 of file DcVar.cpp.

6.26.3.33 SetRadius()

```
bool DcVar::SetRadius (
    int newRadius )
```

Definition at line 1182 of file DcVar.cpp.

6.26.3.34 SetTF()

```
bool DcVar::SetTF (
    bool tfFlag )
```

Definition at line 1207 of file DcVar.cpp.

6.26.3.35 SetTFRadius()

```
bool DcVar::SetTFRadius (
    int newRadius )
```

Definition at line 1197 of file DcVar.cpp.

6.26.3.36 SplitExpressionCaseControl()

```
bool DcVar::SplitExpressionCaseControl (
    arma::mat & caseMatrix,
    arma::mat & ctrlMatrix ) [private]
```

Definition at line 769 of file DcVar.cpp.

6.26.3.37 WriteCheckpoint()

```
bool DcVar::WriteCheckpoint (
    uint snpIndex,
    std::string snpName ) [private]
```

Definition at line 1114 of file DcVar.cpp.

6.26.3.38 WriteResults()

```
bool DcVar::WriteResults (
    std::string filename,
    std::string curSnp ) [private]
```

Definition at line 1137 of file DcVar.cpp.

6.26.4 Member Data Documentation

6.26.4.1 caseIdxCol

```
std::vector<uint> DcVar::caseIdxCol [private]
```

Definition at line 128 of file DcVar.h.

6.26.4.2 chipSeqExpression

```
CHIP_SEQ_INFO_LIST DcVar::chipSeqExpression [private]
```

Definition at line 113 of file DcVar.h.

6.26.4.3 chipSeqMode

```
bool DcVar::chipSeqMode [private]
```

Definition at line 112 of file DcVar.h.

6.26.4.4 coordinates

```
CoordinateTable DcVar::coordinates [private]
```

Definition at line 143 of file DcVar.h.

6.26.4.5 ctrlIdxCol

```
std::vector<uint> DcVar::ctrlIdxCol [private]
```

Definition at line 129 of file DcVar.h.

6.26.4.6 expressionMatrix

```
matrix_t DcVar::expressionMatrix [private]
```

Definition at line 124 of file DcVar.h.

6.26.4.7 geneExprNames

```
std::vector<std::string> DcVar::geneExprNames [private]
```

Definition at line 122 of file DcVar.h.

6.26.4.8 geneExprSubjects

```
std::vector<std::string> DcVar::geneExprSubjects [private]
```

Definition at line 123 of file DcVar.h.

6.26.4.9 genotypeMatrix

```
matrix_t DcVar::genotypeMatrix [private]
```

Definition at line 120 of file DcVar.h.

6.26.4.10 genotypeSubjects

```
std::vector<std::string> DcVar::genotypeSubjects [private]
```

Definition at line 119 of file DcVar.h.

6.26.4.11 localCis

```
bool DcVar::localCis [private]
```

Definition at line 142 of file DcVar.h.

6.26.4.12 nInnerLoop

```
uint DcVar::nInnerLoop [private]
```

Definition at line 151 of file DcVar.h.

6.26.4.13 nOuterLoop

```
uint DcVar::nOuterLoop [private]
```

Definition at line 150 of file DcVar.h.

6.26.4.14 numCombs

```
double DcVar::numCombs [private]
```

Definition at line 126 of file DcVar.h.

6.26.4.15 pVals

```
arma::mat DcVar::pVals [private]
```

Definition at line 132 of file DcVar.h.

6.26.4.16 radius

```
uint DcVar::radius [private]
```

Definition at line 141 of file DcVar.h.

6.26.4.17 snpInputType

```
SNP_INPUT_TYPE DcVar::snpInputType [private]
```

Definition at line 115 of file DcVar.h.

6.26.4.18 snpLocations

```
SNP_INFO_LIST DcVar::snpLocations [private]
```

Definition at line 117 of file DcVar.h.

6.26.4.19 snpNames

```
std::vector<std::string> DcVar::snpNames [private]
```

Definition at line 116 of file DcVar.h.

6.26.4.20 tfMode

```
bool DcVar::tfMode [private]
```

Definition at line 146 of file DcVar.h.

6.26.4.21 tfRadius

```
int DcVar::tfRadius [private]
```

Definition at line 147 of file DcVar.h.

6.26.4.22 tfTableLoaded

```
bool DcVar::tfTableLoaded [private]
```

Definition at line 145 of file DcVar.h.

6.26.4.23 thisTFSnpIndices

```
std::vector<int> DcVar::thisTFSnpIndices [private]
```

Definition at line 152 of file DcVar.h.

6.26.4.24 thisTranscriptSnpIndices

```
std::vector<int> DcVar::thisTranscriptSnpIndices [private]
```

Definition at line 149 of file DcVar.h.

6.26.4.25 totalTests

```
uint DcVar::totalTests [private]
```

Definition at line 127 of file DcVar.h.

6.26.4.26 transcriptFactorLUT

```
TranscriptFactorTable DcVar::transcriptFactorLUT [private]
```

Definition at line 148 of file DcVar.h.

6.26.4.27 zVals

```
arma::sp_mat DcVar::zVals [private]
```

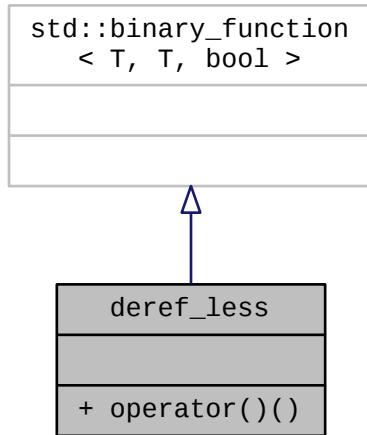
Definition at line 131 of file DcVar.h.

The documentation for this class was generated from the following files:

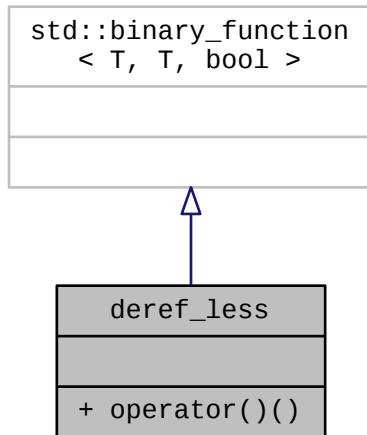
- src/[DcVar.h](#)
- src/[DcVar.cpp](#)

6.27 deref_less Class Reference

Inheritance diagram for deref_less:



Collaboration diagram for deref_less:



Public Member Functions

- bool `operator()` (const `T a`, const `T b`) const

6.27.1 Detailed Description

Definition at line 57 of file ReliefF.cpp.

6.27.2 Member Function Documentation

6.27.2.1 operator()

```
bool deref_less::operator() (
    const T a,
    const T b ) const [inline]
```

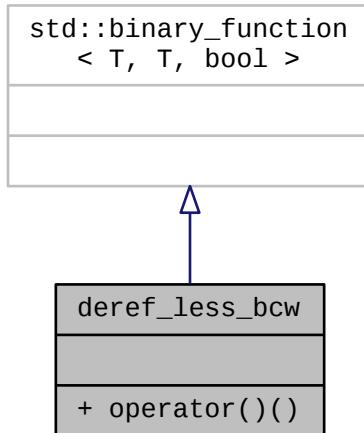
Definition at line 59 of file ReliefF.cpp.

The documentation for this class was generated from the following file:

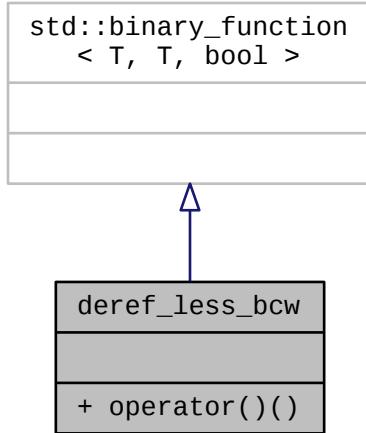
- src/[ReliefF.cpp](#)

6.28 deref_less_bcw Class Reference

Inheritance diagram for deref_less_bcw:



Collaboration diagram for deref_less_bcw:



Public Member Functions

- `bool operator()(const T a, const T b) const`

6.28.1 Detailed Description

Definition at line 25 of file DatasetInstance.cpp.

6.28.2 Member Function Documentation

6.28.2.1 operator()

```
bool deref_less_bcw::operator() (
    const T a,
    const T b ) const [inline]
```

Definition at line 29 of file DatasetInstance.cpp.

The documentation for this class was generated from the following file:

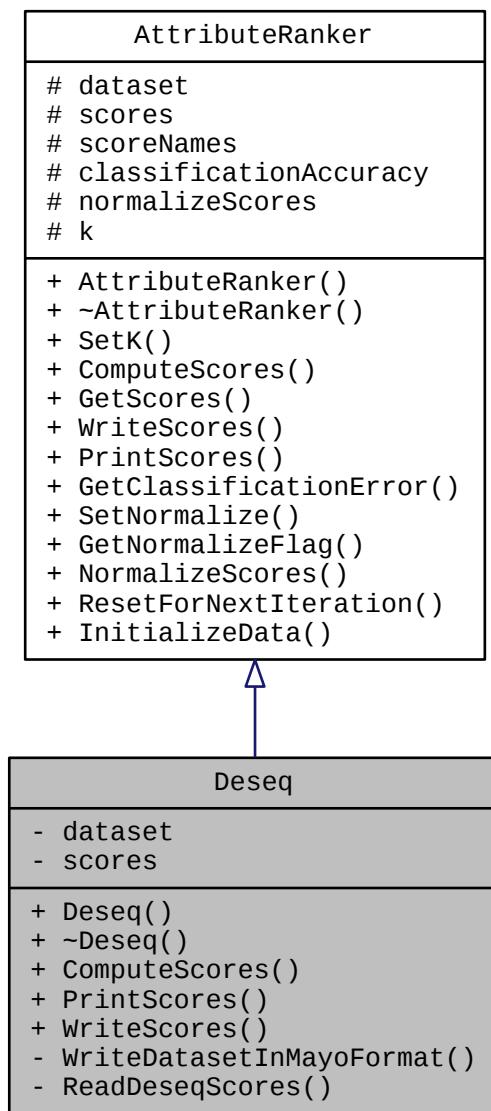
- `src/DatasetInstance.cpp`

6.29 Deseq Class Reference

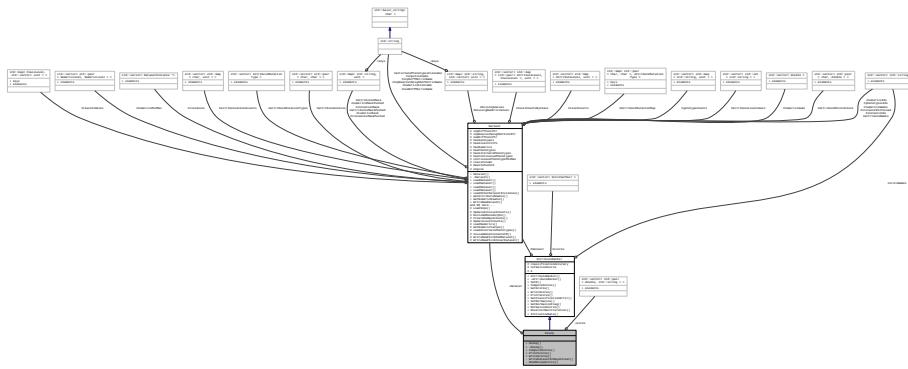
DESeq attribute ranking algorithm.

```
#include <Deseq.h>
```

Inheritance diagram for Deseq:



Collaboration diagram for Deseq:



Public Member Functions

- [Deseq \(Dataset *ds\)](#)
Construct an deseq algorithm object.
- [~Deseq \(\)](#)
- [std::vector< std::pair< double, std::string > > ComputeScores \(\)](#) override
For each attribute, calculate deseq and associated p-value.
- [void PrintScores \(std::ofstream &outStream, unsigned int topN=0\)](#)
Print the scores to a stream.
- [void WriteScores \(std::string outFilename, unsigned int topN=0\)](#)
Print the scores to a file.

Private Member Functions

- [void WriteDatasetInMayoFormat \(std::string filename\)](#)
Write the data set in Mayo GEO format for reading into my DESeq script.
- [bool ReadDeseqScores \(std::string resultsFilename\)](#)
Read DESeq scores into scores map from DESeq results file.

Private Attributes

- [Dataset * dataset](#)
- [std::vector< std::pair< double, std::string > > scores](#)
vector of pairs: scores, attribute names deseq 1.0 - pvalue for each attribute

Additional Inherited Members

6.29.1 Detailed Description

DESeq attribute ranking algorithm.

[Deseq](#) algorithm interface.

Author

Bill White

Version

1.0

Contact: bill.c.white@gmail.com Created on: 8/8/12

Definition at line 24 of file Deseq.h.

6.29.2 Constructor & Destructor Documentation

6.29.2.1 Deseq()

```
Deseq::Deseq (   
    Dataset * ds )
```

Construct an deseq algorithm object.

Parameters

in	ds	pointer to a Dataset object
----	----	---

Definition at line 26 of file Deseq.cpp.

6.29.2.2 ~Deseq()

```
Deseq::~Deseq ( )
```

Definition at line 40 of file Deseq.cpp.

6.29.3 Member Function Documentation

6.29.3.1 ComputeScores()

```
vector< pair< double, string > > Deseq::ComputeScores ( ) [override], [virtual]
```

For each attribute, calculate deseq and associated p-value.

Return in a vector of pairs indexed by attribute index.

Returns

vector of pairs of deseq scores and associated p-values

save the current data set to a temporary file for DESeq

run DESeq through a system call to the shell

loads DESeq scores map from the output file

remove the temporary file

Implements [AttributeRanker](#).

Definition at line 43 of file Deseq.cpp.

6.29.3.2 PrintScores()

```
void Deseq::PrintScores (
    std::ofstream & outStream,
    unsigned int topN = 0 )
```

Print the scores to a stream.

Parameters

in	<i>outStream</i>	reference to an output stream
in	<i>topN</i>	top number of attributes to print

Definition at line 86 of file Deseq.cpp.

6.29.3.3 ReadDeseqScores()

```
bool Deseq::ReadDeseqScores (
    std::string resultsFilename ) [private]
```

Read DESeq scores into scores map from DESeq results file.

Definition at line 151 of file Deseq.cpp.

6.29.3.4 WriteDatasetInMayoFormat()

```
void Deseq::WriteDatasetInMayoFormat (
    std::string filename ) [private]
```

Write the data set in Mayo GEO format for reading into my DESeq script.

Definition at line 117 of file Deseq.cpp.

6.29.3.5 WriteScores()

```
void Deseq::WriteScores (
    std::string outfilename,
    unsigned int topN = 0 )
```

Print the scores to a file.

Parameters

in	<i>outfilename</i>	filename to write scores to
in	<i>topN</i>	top number of attributes to print

Definition at line 101 of file Deseq.cpp.

6.29.4 Member Data Documentation

6.29.4.1 dataset

```
Dataset* Deseq::dataset [private]
```

Definition at line 52 of file Deseq.h.

6.29.4.2 scores

```
std::vector<std::pair<double, std::string> > Deseq::scores [private]
```

vector of pairs: scores, attribute names deseq 1.0 - pvalue for each attribute

Definition at line 55 of file Deseq.h.

The documentation for this class was generated from the following files:

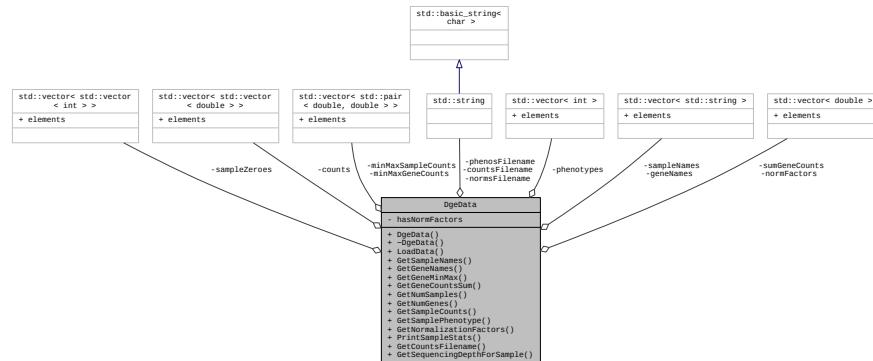
- src/[Deseq.h](#)
- src/[Deseq.cpp](#)

6.30 DgeData Class Reference

Digital gene expression data.

```
#include <DgeData.h>
```

Collaboration diagram for DgeData:



Public Member Functions

- [DgeData \(\)](#)
- virtual [~DgeData \(\)](#)
- bool [LoadData](#) (std::string countsFile, std::string normsFile="")
Create a new set of DGE data with a counts file and a phenotype file.
- std::vector< std::string > [GetSampleNames \(\)](#)
Get the sample names/IDs.
- std::vector< std::string > [GetGeneNames \(\)](#)
Get the gene names/IDs.
- std::pair< double, double > [GetGeneMinMax](#) (int geneIndex)
Get the min and max values for gene at index.

- double [GetGeneCountsSum](#) (int geneIndex)
Get the sum of raw counts for the gene at index.
- int [GetNumSamples](#) ()
Get the number of samples.
- int [GetNumGenes](#) ()
Get the number of genes.
- std::vector< double > [GetSampleCounts](#) (int sampleIndex)
Get sample counts for sample at index.
- int [GetSamplePhenotype](#) (int sampleIndex)
Get the phenotype at sample index.
- std::vector< double > [GetNormalizationFactors](#) ()
Get the normalization factors.
- void [PrintSampleStats](#) ()
Print the Sample statistics to the console.
- std::string [GetCountsFilename](#) ()
Get the original data set filename.
- unsigned int [GetSequencingDepthForSample](#) (unsigned int sampleIndex)

Private Attributes

- std::string [countsFilename](#)
Filename containing DGE counts.
- std::string [phenosFilename](#)
Filename containing DGE phenotypes.
- std::string [normsFilename](#)
Filename containing DGE normalization factors.
- bool [hasNormFactors](#)
Are we using normalization?
- std::vector< double > [normFactors](#)
Vector of (optional) normalization factors for each sample.
- std::vector< std::string > [geneNames](#)
Gene names.
- std::vector< std::vector< double > > [counts](#)
Digital gene expression counts.
- std::vector< std::string > [sampleNames](#)
Sample names.
- std::vector< int > [phenotypes](#)
Sample phenotypes.
- std::vector< std::pair< double, double > > [minMaxGeneCounts](#)
Min and max count for genes.
- std::vector< std::pair< double, double > > [minMaxSampleCounts](#)
Min and max values for samples.
- std::vector< double > [sumGeneCounts](#)
Gene count sums.
- std::vector< std::vector< int > > [sampleZeroes](#)
Zero count sample indices.

6.30.1 Detailed Description

Digital gene expression data.

Author

Bill White

Version

1.0

Contact: bill.c.white@gmail.com Created on: 1/18/12

Definition at line 16 of file DgeData.h.

6.30.2 Constructor & Destructor Documentation

6.30.2.1 DgeData()

DgeData::DgeData ()

Definition at line 23 of file DgeData.cpp.

6.30.2.2 ~DgeData()

DgeData::~DgeData () [virtual]

Definition at line 27 of file DgeData.cpp.

6.30.3 Member Function Documentation

6.30.3.1 GetCountsFilename()

string DgeData::GetCountsFilename ()

Get the original data set filename.

Definition at line 296 of file DgeData.cpp.

6.30.3.2 GetGeneCountsSum()

```
double DgeData::GetGeneCountsSum ( int geneIndex )
```

Get the sum of raw counts for the gene at index.

Definition at line 235 of file DgeData.cpp.

6.30.3.3 GetGeneMinMax()

```
pair< double, double > DgeData::GetGeneMinMax ( int geneIndex )
```

Get the min and max values for gene at index.

Definition at line 224 of file DgeData.cpp.

6.30.3.4 GetGeneNames()

```
vector< string > DgeData::GetGeneNames ( )
```

Get the gene names/IDs.

Definition at line 220 of file DgeData.cpp.

6.30.3.5 GetNormalizationFactors()

```
vector< double > DgeData::GetNormalizationFactors ( )
```

Get the normalization factors.

Definition at line 279 of file DgeData.cpp.

6.30.3.6 GetNumGenes()

```
int DgeData::GetNumGenes ( )
```

Get the number of genes.

Definition at line 250 of file DgeData.cpp.

6.30.3.7 GetNumSamples()

```
int DgeData::GetNumSamples ( )
```

Get the number of samples.

Definition at line 246 of file DgeData.cpp.

6.30.3.8 GetSampleCounts()

```
vector< double > DgeData::GetSampleCounts (  
    int sampleIndex )
```

Get sample counts for sample at index.

Definition at line 254 of file DgeData.cpp.

6.30.3.9 GetSampleNames()

```
vector< string > DgeData::GetSampleNames ( )
```

Get the sample names/IDs.

Definition at line 216 of file DgeData.cpp.

6.30.3.10 GetSamplePhenotype()

```
int DgeData::GetSamplePhenotype (  
    int sampleIndex )
```

Get the phenotype at sample index.

Definition at line 269 of file DgeData.cpp.

6.30.3.11 GetSequencingDepthForSample()

```
unsigned int DgeData::GetSequencingDepthForSample (   
    unsigned int sampleIndex )
```

Definition at line 300 of file DgeData.cpp.

6.30.3.12 LoadData()

```
bool DgeData::LoadData (
    std::string countsFile,
    std::string normsFile = "")
```

Create a new set of DGE data with a counts file and a phenotype file.

read gene counts

collect minimum, maximum and sum of raw counts for each gene

save this gene's counts to the counts class member variable

get min and max sample counts, and sample zeroes

Definition at line 30 of file DgeData.cpp.

6.30.3.13 PrintSampleStats()

```
void DgeData::PrintSampleStats ( )
```

Print the Sample statistics to the console.

Definition at line 283 of file DgeData.cpp.

6.30.4 Member Data Documentation

6.30.4.1 counts

```
std::vector<std::vector<double>> DgeData::counts [private]
```

Digital gene expression counts.

Definition at line 60 of file DgeData.h.

6.30.4.2 countsFilename

```
std::string DgeData::countsFilename [private]
```

Filename containing DGE counts.

Definition at line 48 of file DgeData.h.

6.30.4.3 geneNames

```
std::vector<std::string> DgeData::geneNames [private]
```

Gene names.

Definition at line 58 of file DgeData.h.

6.30.4.4 hasNormFactors

```
bool DgeData::hasNormFactors [private]
```

Are we using normalization?

Definition at line 54 of file DgeData.h.

6.30.4.5 minMaxGeneCounts

```
std::vector<std::pair<double, double> > DgeData::minMaxGeneCounts [private]
```

Min and max count for genes.

Definition at line 66 of file DgeData.h.

6.30.4.6 minMaxSampleCounts

```
std::vector<std::pair<double, double> > DgeData::minMaxSampleCounts [private]
```

Min and max values for samples.

Definition at line 68 of file DgeData.h.

6.30.4.7 normFactors

```
std::vector<double> DgeData::normFactors [private]
```

Vector of (optional) normalization factors for each sample.

Definition at line 56 of file DgeData.h.

6.30.4.8 normsFilename

```
std::string DgeData::normsFilename [private]
```

Filename containing DGE normalization factors.

Definition at line 52 of file DgeData.h.

6.30.4.9 phenosFilename

```
std::string DgeData::phenosFilename [private]
```

Filename containing DGE phenotypes.

Definition at line 50 of file DgeData.h.

6.30.4.10 phenotypes

```
std::vector<int> DgeData::phenotypes [private]
```

Sample phenotypes.

Definition at line 64 of file DgeData.h.

6.30.4.11 sampleNames

```
std::vector<std::string> DgeData::sampleNames [private]
```

Sample names.

Definition at line 62 of file DgeData.h.

6.30.4.12 sampleZeroes

```
std::vector<std::vector<int> > DgeData::sampleZeroes [private]
```

Zero count sample indices.

Definition at line 72 of file DgeData.h.

6.30.4.13 sumGeneCounts

```
std::vector<double> DgeData::sumGeneCounts [private]
```

Gene count sums.

Definition at line 70 of file DgeData.h.

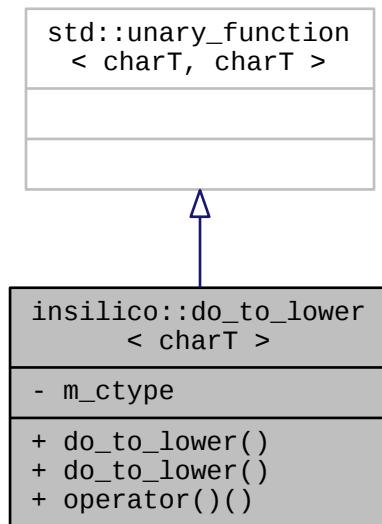
The documentation for this class was generated from the following files:

- src/[DgeData.h](#)
- src/[DgeData.cpp](#)

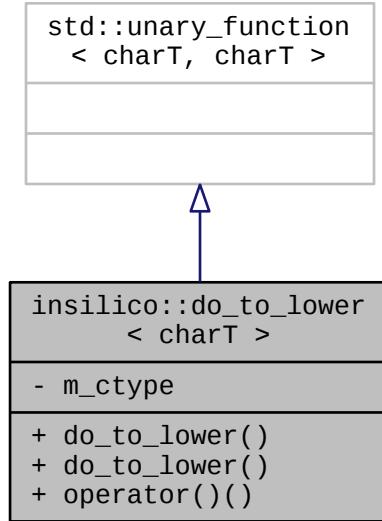
6.31 insilico::do_to_lower< charT > Class Template Reference

```
#include <StringUtils.h>
```

Inheritance diagram for insilico::do_to_lower< charT >:



Collaboration diagram for insilico::do_to_lower< charT >:



Public Member Functions

- `do_to_lower (std::ctype< charT > &ct)`
- `do_to_lower (const std::locale &loc=std::locale())`
- `charT operator() (charT c) const`

Private Attributes

- `std::ctype< charT > const & m_ctype`

6.31.1 Detailed Description

```
template<class charT = char>
class insilico::do_to_lower< charT >
```

Definition at line 79 of file StringUtil.h.

6.31.2 Constructor & Destructor Documentation

6.31.2.1 `do_to_lower()` [1/2]

```
template<class charT = char>
insilico::do_to_lower<charT>::do_to_lower (
    std::ctype<charT> & ct) [inline]
```

Definition at line 83 of file `StringUtil.h`.

6.31.2.2 `do_to_lower()` [2/2]

```
template<class charT = char>
insilico::do_to_lower<charT>::do_to_lower (
    const std::locale & loc = std::locale() ) [inline]
```

Definition at line 86 of file `StringUtil.h`.

6.31.3 Member Function Documentation

6.31.3.1 `operator()()`

```
template<class charT = char>
charT insilico::do_to_lower<charT>::operator() (
    charT c) const [inline]
```

Definition at line 89 of file `StringUtil.h`.

6.31.4 Member Data Documentation

6.31.4.1 `m_ctype`

```
template<class charT = char>
std::ctype<charT> const& insilico::do_to_lower<charT>::m_ctype [private]
```

Definition at line 93 of file `StringUtil.h`.

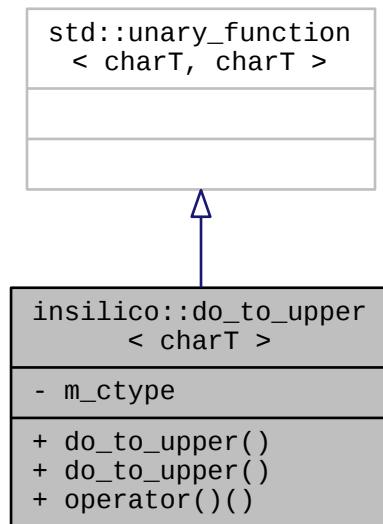
The documentation for this class was generated from the following file:

- `src/StringUtil.h`

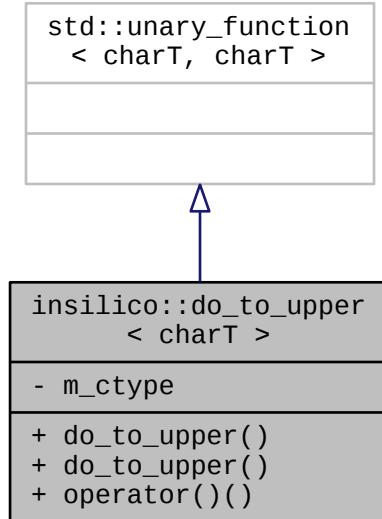
6.32 insilico::do_to_upper< charT > Class Template Reference

```
#include <StringUtils.h>
```

Inheritance diagram for insilico::do_to_upper< charT >:



Collaboration diagram for insilico::do_to_upper< charT >:



Public Member Functions

- `do_to_upper (std::ctype< charT > &ct)`
- `do_to_upper (const std::locale &loc=std::locale())`
- `charT operator() (charT c) const`

Private Attributes

- `std::ctype< charT > const & m_ctype`

6.32.1 Detailed Description

```
template<class charT = char>
class insilico::do_to_upper<charT>
```

Definition at line 59 of file StringUtil.h.

6.32.2 Constructor & Destructor Documentation

6.32.2.1 do_to_upper() [1/2]

```
template<class charT = char>
insilico::do_to_upper< charT >::do_to_upper (
    std::ctype< charT > & ct )  [inline]
```

Definition at line 63 of file StringUtil.h.

6.32.2.2 do_to_upper() [2/2]

```
template<class charT = char>
insilico::do_to_upper< charT >::do_to_upper (
    const std::locale & loc = std::locale() )  [inline]
```

Definition at line 66 of file StringUtil.h.

6.32.3 Member Function Documentation

6.32.3.1 operator()()

```
template<class charT = char>
charT insilico::do_to_upper< charT >::operator() (
    charT c ) const  [inline]
```

Definition at line 69 of file StringUtil.h.

6.32.4 Member Data Documentation

6.32.4.1 m_ctype

```
template<class charT = char>
std::ctype<charT> const& insilico::do_to_upper< charT >::m_ctype  [private]
```

Definition at line 73 of file StringUtil.h.

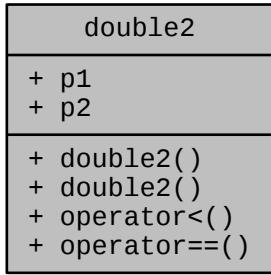
The documentation for this class was generated from the following file:

- src/StringUtil.h

6.33 double2 Class Reference

```
#include <plink.h>
```

Collaboration diagram for double2:



Public Member Functions

- [double2 \(\)](#)
- [double2 \(double a, double b\)](#)
- [bool operator< \(const double2 &b\) const](#)
- [bool operator== \(const double2 &b\) const](#)

Public Attributes

- [double p1](#)
- [double p2](#)

6.33.1 Detailed Description

Definition at line 116 of file plink.h.

6.33.2 Constructor & Destructor Documentation

6.33.2.1 double2() [1/2]

```
double2::double2 ( ) [inline]
```

Definition at line 121 of file plink.h.

6.33.2.2 double2() [2/2]

```
double2::double2 (
    double a,
    double b ) [inline]
```

Definition at line 125 of file plink.h.

6.33.3 Member Function Documentation

6.33.3.1 operator<()

```
bool double2::operator< (
    const double2 & b ) const [inline]
```

Definition at line 130 of file plink.h.

6.33.3.2 operator==()

```
bool double2::operator== (
    const double2 & b ) const [inline]
```

Definition at line 134 of file plink.h.

6.33.4 Member Data Documentation

6.33.4.1 p1

```
double double2::p1
```

Definition at line 118 of file plink.h.

6.33.4.2 p2

```
double double2::p2
```

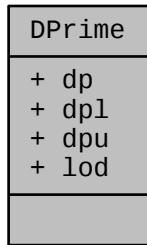
Definition at line 119 of file plink.h.

The documentation for this class was generated from the following file:

- src/[plink.h](#)

6.34 DPrime Class Reference

Collaboration diagram for DPrime:



Public Attributes

- double [dp](#)
- double [dpl](#)
- double [dpu](#)
- double [lod](#)

6.34.1 Detailed Description

Definition at line 74 of file blox.cpp.

6.34.2 Member Data Documentation

6.34.2.1 dp

```
double DPrime::dp
```

Definition at line 77 of file blox.cpp.

6.34.2.2 dpl

```
double DPrime::dpl
```

Definition at line 78 of file blox.cpp.

6.34.2.3 dpu

```
double DPrime::dpu
```

Definition at line 79 of file blox.cpp.

6.34.2.4 lod

```
double DPrime::lod
```

Definition at line 80 of file blox.cpp.

The documentation for this class was generated from the following file:

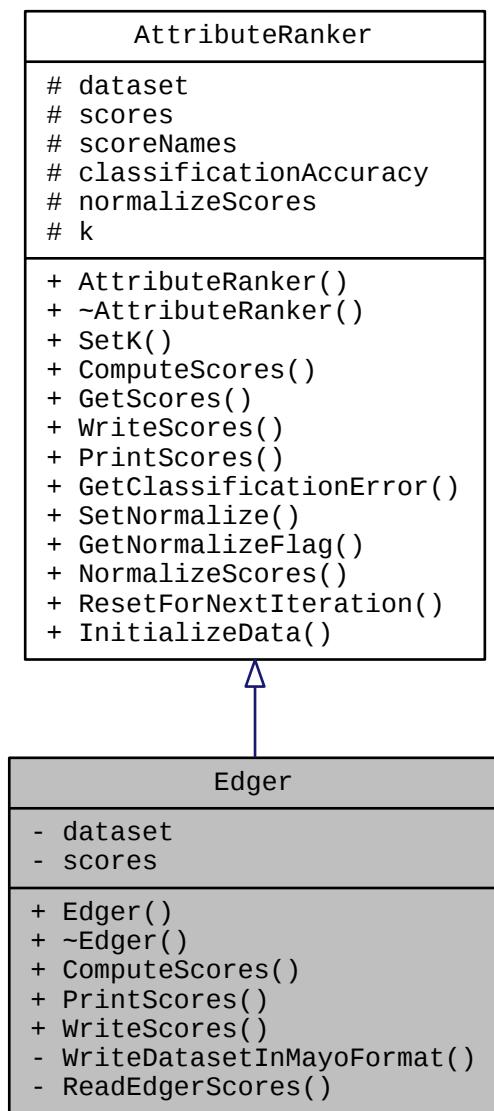
- [src/blox.cpp](#)

6.35 Edger Class Reference

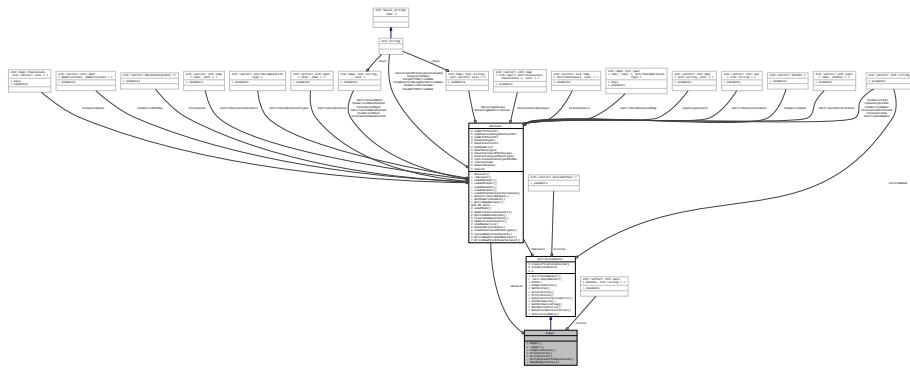
[Edger](#) attribute ranking algorithm.

```
#include <Edger.h>
```

Inheritance diagram for Edger:



Collaboration diagram for Edger:



Public Member Functions

- `Edger` (*Dataset* *ds)
Construct an `Edger` algorithm object.
 - `~Edger` ()
 - `std::vector< std::pair< double, std::string > > ComputeScores ()` override
For each attribute, calculate `Edger` and associated p-value.
 - void `PrintScores` (`std::ofstream` &*outStream*, `unsigned int topN=0`)
Print the scores to a stream.
 - void `WriteScores` (`std::string` *outFilename*, `unsigned int topN=0`)
Print the scores to a file.

Private Member Functions

- void `WriteDatasetInMayoFormat` (std::string filename)
Write the data set in Mayo GEO format for reading into my `Edger` script.
 - bool `ReadEdgerScores` (std::string resultsFilename)
Read `Edger` scores into scores map from `Edger` results file.

Private Attributes

- Dataset * dataset
 - std::vector< std::pair< double, std::string > > scores
 - vector of pairs: scores, attribute names Edger 1.0 - pvalue for each attribute

Additional Inherited Members

6.35.1 Detailed Description

[Edger](#) attribute ranking algorithm.

[Edger](#) algorithm interface.

Author

Bill White

Version

1.0

Contact: bill.c.white@gmail.com Created on: 10/2/12

Definition at line 24 of file Edger.h.

6.35.2 Constructor & Destructor Documentation

6.35.2.1 Edger()

```
Edger::Edger (   
    Dataset * ds )
```

Construct an [Edger](#) algorithm object.

Parameters

in	ds	pointer to a Dataset object
----	----	---

Definition at line 26 of file Edger.cpp.

6.35.2.2 ~Edger()

```
Edger::~Edger ( )
```

Definition at line 40 of file Edger.cpp.

6.35.3 Member Function Documentation

6.35.3.1 ComputeScores()

```
vector< pair< double, string > > Edger::ComputeScores ( ) [override], [virtual]
```

For each attribute, calculate [Edger](#) and associated p-value.

Return in a vector of pairs indexed by attribute index.

Returns

vector of pairs of [Edger](#) scores and associated p-values

save the current data set to a temporary file for [Edger](#)

run edgeR through a system call to the shell

loads edgeR scores map from the output file

remove the temporary files

Implements [AttributeRanker](#).

Definition at line 43 of file Edger.cpp.

6.35.3.2 PrintScores()

```
void Edger::PrintScores (  
    std::ofstream & outStream,  
    unsigned int topN = 0 )
```

Print the scores to a stream.

Parameters

in	<i>outStream</i>	reference to an output stream
in	<i>topN</i>	top number of attributes to print

Definition at line 91 of file Edger.cpp.

6.35.3.3 `ReadEdgerScores()`

```
bool Edger::ReadEdgerScores (
    std::string resultsFilename ) [private]
```

Read [Edger](#) scores into scores map from [Edger](#) results file.

Definition at line 156 of file [Edger.cpp](#).

6.35.3.4 `WriteDatasetInMayoFormat()`

```
void Edger::WriteDatasetInMayoFormat (
    std::string filename ) [private]
```

Write the data set in Mayo GEO format for reading into my [Edger](#) script.

Definition at line 122 of file [Edger.cpp](#).

6.35.3.5 `WriteScores()`

```
void Edger::WriteScores (
    std::string outfilename,
    unsigned int topN = 0 )
```

Print the scores to a file.

Parameters

in	<i>outfilename</i>	filename to write scores to
in	<i>topN</i>	top number of attributes to print

Definition at line 106 of file [Edger.cpp](#).

6.35.4 Member Data Documentation

6.35.4.1 `dataset`

```
Dataset* Edger::dataset [private]
```

Definition at line 52 of file [Edger.h](#).

6.35.4.2 scores

```
std::vector<std::pair<double, std::string> > Edger::scores [private]
```

vector of pairs: scores, attribute names [Edger](#) 1.0 - pvalue for each attribute

Definition at line 55 of file [Edger.h](#).

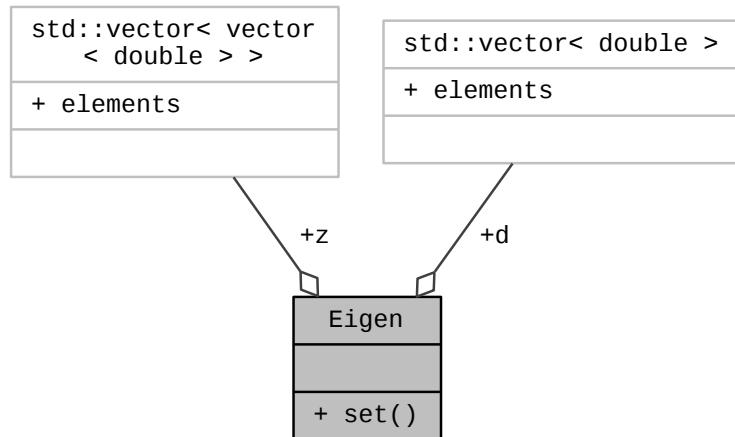
The documentation for this class was generated from the following files:

- [src/Edger.h](#)
- [src/Edger.cpp](#)

6.36 Eigen Class Reference

```
#include <stats.h>
```

Collaboration diagram for Eigen:



Public Member Functions

- [void set \(int n\)](#)

Public Attributes

- [vector_t d](#)
- [matrix_t z](#)

6.36.1 Detailed Description

Definition at line 106 of file stats.h.

6.36.2 Member Function Documentation

6.36.2.1 set()

```
void Eigen::set (
    int n ) [inline]
```

Definition at line 109 of file stats.h.

6.36.3 Member Data Documentation

6.36.3.1 d

```
vector_t Eigen::d
```

Definition at line 114 of file stats.h.

6.36.3.2 z

```
matrix_t Eigen::z
```

Definition at line 115 of file stats.h.

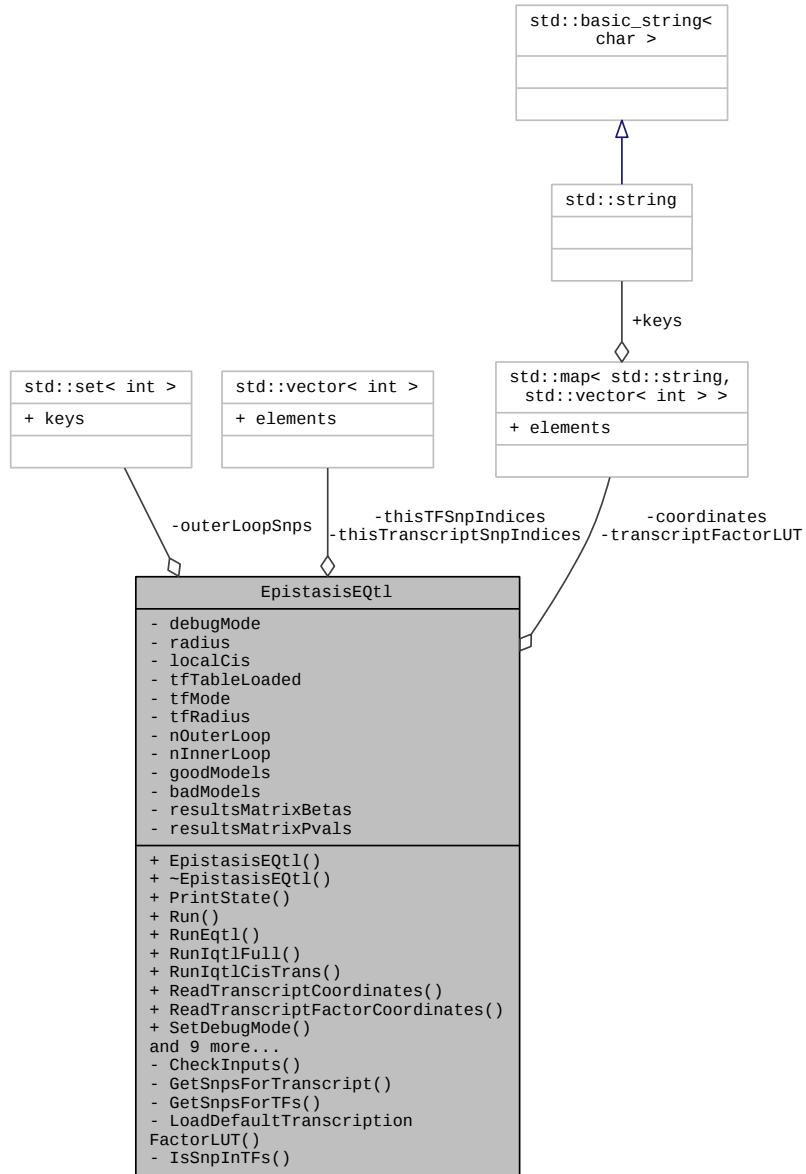
The documentation for this class was generated from the following file:

- [src/stats.h](#)

6.37 EpistasisEQtl Class Reference

```
#include <EpistasisEQtl.h>
```

Collaboration diagram for EpistasisEQtl:



Public Member Functions

- [EpistasisEQtl \(\)](#)

- virtual ~EpistasisEQtl ()
- void PrintState ()
- bool Run (bool debug=false)
- bool RunEqtl (std::string transcript)
- bool RunIqtlFull ()
- bool RunIqtlCisTrans ()
- bool ReadTranscriptCoordinates (std::string coordinatesFile)
- bool ReadTranscriptFactorCoordinates (std::string coordinatesFile)
- bool SetDebugMode (bool debugFlag=true)
- bool SetRadius (int newRadius)
- int GetRadius ()
- bool SetLocalCis (bool localCisFlag)
- bool GetLocalCis ()
- bool SetTFRadius (int newRadius)
- int GetTFRadius ()
- bool SetTF (bool tfFlag)
- bool GetTF ()
- bool GetTFInfo (std::string tf, std::vector< int > &tflInfo)

Private Member Functions

- bool CheckInputs ()
- bool GetSnpsForTranscript (std::string transcript, std::vector< int > &snpIndices)
- bool GetSnpsForTFs (std::vector< int > &snpIndices, std::vector< std::string > &tf)
- bool LoadDefaultTranscriptionFactorLUT ()
- bool IsSnpInTFs (int chr, int bp, std::string &tf)

Private Attributes

- bool debugMode
- int radius
- bool localCis
- CoordinateTable coordinates
- bool tfTableLoaded
- bool tfMode
- int tfRadius
- TranscriptFactorTable transcriptFactorLUT
- std::vector< int > thisTranscriptSnpIndices
- uint nOuterLoop
- uint nInnerLoop
- std::vector< int > thisTFSnpIndices
- uint goodModels
- uint badModels
- std::set< int > outerLoopSnps
- arma::mat resultsMatrixBetas
- arma::mat resultsMatrixPvals

6.37.1 Detailed Description

Definition at line 29 of file EpistasisEQtl.h.

6.37.2 Constructor & Destructor Documentation

6.37.2.1 EpistasisEQtl()

```
EpistasisEQtl::EpistasisEQtl ( )
```

Definition at line 29 of file EpistasisEQtl.cpp.

6.37.2.2 ~EpistasisEQtl()

```
EpistasisEQtl::~EpistasisEQtl ( ) [virtual]
```

Definition at line 40 of file EpistasisEQtl.cpp.

6.37.3 Member Function Documentation

6.37.3.1 CheckInputs()

```
bool EpistasisEQtl::CheckInputs ( ) [private]
```

Definition at line 48 of file EpistasisEQtl.cpp.

6.37.3.2 GetLocalCis()

```
bool EpistasisEQtl::GetLocalCis ( ) [inline]
```

Definition at line 44 of file EpistasisEQtl.h.

6.37.3.3 GetRadius()

```
int EpistasisEQtl::GetRadius ( ) [inline]
```

Definition at line 42 of file EpistasisEQtl.h.

6.37.3.4 GetSnpForTFs()

```
bool EpistasisEQtl::GetSnpForTFs (
    std::vector< int > & snpIndices,
    std::vector< std::string > & tfs ) [private]
```

Definition at line 717 of file EpistasisEQtl.cpp.

6.37.3.5 GetSnpForTranscript()

```
bool EpistasisEQtl::GetSnpForTranscript (
    std::string transcript,
    std::vector< int > & snpIndices ) [private]
```

Definition at line 677 of file EpistasisEQtl.cpp.

6.37.3.6 GetTF()

```
bool EpistasisEQtl::GetTF ( ) [inline]
```

Definition at line 49 of file EpistasisEQtl.h.

6.37.3.7 GetTFInfo()

```
bool EpistasisEQtl::GetTFInfo (
    std::string tf,
    std::vector< int > & tfInfo )
```

Definition at line 761 of file EpistasisEQtl.cpp.

6.37.3.8 GetTFRadius()

```
int EpistasisEQtl::GetTFRadius ( ) [inline]
```

Definition at line 47 of file EpistasisEQtl.h.

6.37.3.9 IsSnpInTFs()

```
bool EpistasisEQtl::IsSnpInTFs (
    int chr,
    int bp,
    std::string & tf ) [private]
```

Definition at line 740 of file EpistasisEQtl.cpp.

6.37.3.10 LoadDefaultTranscriptionFactorLUT()

```
bool EpistasisEQtl::LoadDefaultTranscriptionFactorLUT ( ) [private]
```

Definition at line 773 of file EpistasisEQtl.cpp.

6.37.3.11 PrintState()

```
void EpistasisEQtl::PrintState ( )
```

Definition at line 84 of file EpistasisEQtl.cpp.

6.37.3.12 ReadTranscriptCoordinates()

```
bool EpistasisEQtl::ReadTranscriptCoordinates (
    std::string coordinatesFile )
```

Definition at line 498 of file EpistasisEQtl.cpp.

6.37.3.13 ReadTranscriptFactorCoordinates()

```
bool EpistasisEQtl::ReadTranscriptFactorCoordinates (
    std::string coordinatesFile )
```

Definition at line 569 of file EpistasisEQtl.cpp.

6.37.3.14 Run()

```
bool EpistasisEQtl::Run (
    bool debug = false )
```

Definition at line 120 of file EpistasisEQtl.cpp.

6.37.3.15 RunEqtl()

```
bool EpistasisEQtl::RunEqtl (
    std::string transcript )
```

Definition at line 447 of file EpistasisEQtl.cpp.

6.37.3.16 RunIqtlCisTrans()

```
bool EpistasisEQtl::RunIqtlCisTrans ( )
```

Definition at line 299 of file EpistasisEQtl.cpp.

6.37.3.17 RunIqtlFull()

```
bool EpistasisEQtl::RunIqtlFull ( )
```

Definition at line 394 of file EpistasisEQtl.cpp.

6.37.3.18 SetDebugMode()

```
bool EpistasisEQtl::SetDebugMode (
    bool debugFlag = true )
```

Definition at line 43 of file EpistasisEQtl.cpp.

6.37.3.19 SetLocalCis()

```
bool EpistasisEQtl::SetLocalCis (
    bool localCisFlag )
```

Definition at line 652 of file EpistasisEQtl.cpp.

6.37.3.20 SetRadius()

```
bool EpistasisEQtl::SetRadius (
    int newRadius )
```

Definition at line 642 of file EpistasisEQtl.cpp.

6.37.3.21 SetTF()

```
bool EpistasisEQtl::SetTF (
    bool tfFlag )
```

Definition at line 668 of file EpistasisEQtl.cpp.

6.37.3.22 SetTFRadius()

```
bool EpistasisEQtl::SetTFRadius (
    int newRadius )
```

Definition at line 658 of file EpistasisEQtl.cpp.

6.37.4 Member Data Documentation

6.37.4.1 badModels

```
uint EpistasisEQtl::badModels [private]
```

Definition at line 72 of file EpistasisEQtl.h.

6.37.4.2 coordinates

```
CoordinateTable EpistasisEQtl::coordinates [private]
```

Definition at line 61 of file EpistasisEQtl.h.

6.37.4.3 debugMode

```
bool EpistasisEQtl::debugMode [private]
```

Definition at line 58 of file EpistasisEQtl.h.

6.37.4.4 goodModels

```
uint EpistasisEQtl::goodModels [private]
```

Definition at line 71 of file EpistasisEQtl.h.

6.37.4.5 localCis

```
bool EpistasisEQtl::localCis [private]
```

Definition at line 60 of file EpistasisEQtl.h.

6.37.4.6 nInnerLoop

```
uint EpistasisEQtl::nInnerLoop [private]
```

Definition at line 69 of file EpistasisEQtl.h.

6.37.4.7 nOuterLoop

```
uint EpistasisEQtl::nOuterLoop [private]
```

Definition at line 68 of file EpistasisEQtl.h.

6.37.4.8 outerLoopSnps

```
std::set<int> EpistasisEQtl::outerLoopSnps [private]
```

Definition at line 73 of file EpistasisEQtl.h.

6.37.4.9 radius

```
int EpistasisEQtl::radius [private]
```

Definition at line 59 of file EpistasisEQtl.h.

6.37.4.10 resultsMatrixBetas

```
arma::mat EpistasisEQtl::resultsMatrixBetas [private]
```

Definition at line 74 of file EpistasisEQtl.h.

6.37.4.11 resultsMatrixPvals

```
arma::mat EpistasisEQtl::resultsMatrixPvals [private]
```

Definition at line 75 of file EpistasisEQtl.h.

6.37.4.12 tfMode

```
bool EpistasisEQtl::tfMode [private]
```

Definition at line 64 of file EpistasisEQtl.h.

6.37.4.13 tfRadius

```
int EpistasisEqtl::tfRadius [private]
```

Definition at line 65 of file EpistasisEqtl.h.

6.37.4.14 tfTableLoaded

```
bool EpistasisEqtl::tfTableLoaded [private]
```

Definition at line 63 of file EpistasisEqtl.h.

6.37.4.15 thisTFSnpIndices

```
std::vector<int> EpistasisEqtl::thisTFSnpIndices [private]
```

Definition at line 70 of file EpistasisEqtl.h.

6.37.4.16 thisTranscriptSnpIndices

```
std::vector<int> EpistasisEqtl::thisTranscriptSnpIndices [private]
```

Definition at line 67 of file EpistasisEqtl.h.

6.37.4.17 transcriptFactorLUT

```
TranscriptFactorTable EpistasisEqtl::transcriptFactorLUT [private]
```

Definition at line 66 of file EpistasisEqtl.h.

The documentation for this class was generated from the following files:

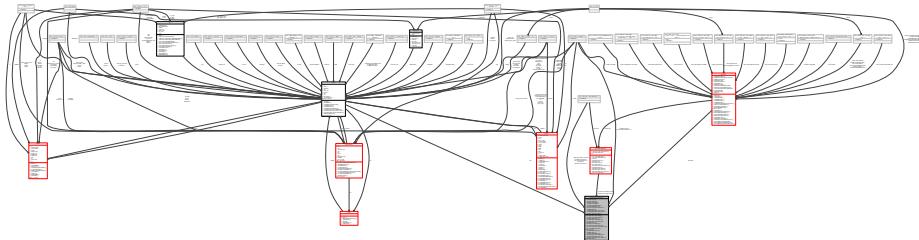
- src/[EpistasisEqtl.h](#)
- src/[EpistasisEqtl.cpp](#)

6.38 EvaporativeCooling Class Reference

Evaporative Cooling attribute ranking algorithm.

```
#include <EvaporativeCooling.h>
```

Collaboration diagram for EvaporativeCooling:



Public Member Functions

- **EvaporativeCooling (Dataset *ds, Plink *plinkPtr, AnalysisType anaType=SNP_ONLY_ANALYSIS)**
Construct an EC algorithm object.
- **virtual ~EvaporativeCooling ()**
- **bool ComputeECScores ()**
Compute the EC scores based on the current set of attributes.
- **AttributeScores & GetMainEffectScores ()**
Get the last computed RandomJungle scores.
- **AttributeScores & GetInteractionScores ()**
Get the last computed ReliefF scores.
- **AttributeScores & GetECScores ()**
Get the last computed EC scores.
- **EcAlgorithmType GetAlgorithmType ()**
Return the algorithm type.
- **void WriteAttributeScores (std::string baseFilename)**
Write the scores and attribute names to file.
- **void WriteClassificationErrors (std::string filename)**
Write the classification errors to file.
- **void WriteTemperatures (std::string filename)**
Write the temperatures to file.
- **void PrintAttributeScores (std::ofstream &outStream)**
Write the EC scores and attribute names to stream.
- **void PrintAllAttributeScores (std::ofstream &outStream)**
Write the EC scores and attribute names to stream, includes evaporated attributes with score of zero.
- **void PrintMainEffectAttributeScores (std::ofstream &outStream)**
Write the main effect scores and attribute names to stream.
- **void PrintInteractionAttributeScores (std::ofstream &outStream)**
Write the interaction scores and attribute names to stream.
- **bool PrintAllScoresTabular ()**
Print the current attributes scores to std::cout in tab-delimited format.
- **bool PrintKendallTaus ()**
Print the Kendall taus between the main effects and interactions scores.

Private Member Functions

- bool [RunRandomForest \(\)](#)
Run the random forest algorithm.
- bool [RunReliefF \(\)](#)
Run the [ReliefF](#) algorithm.
- bool [ComputeFreeEnergy \(double temperature\)](#)
Compute the attribute free energy using the couple temperature.
- bool [RemoveWorstAttributes \(unsigned int numToRemove=1\)](#)
Remove the worst attribute based on free energy scores.
- double [OptimizeTemperature \(std::vector< double > deltas\)](#)
optimize the temperature coupling constant
- double [ComputeClassificationErrorRandomForest \(\)](#)
use Random Jungle to compute the classification error of the current set of attributes with numToRemovePerIteration attributes removed

Private Attributes

- [Dataset * dataset](#)
pointer to a [Dataset](#) object
- [Plink * PP](#)
command line parameters map
- [AnalysisType analysisType](#)
type of analysis to perform
- [EcAlgorithmType algorithmType](#)
algorithm steps to perform
- [EcMeAlgorithmType meAlgorithmType](#)
main effects algorithm
- [EcIAlgorithmType itAlgorithmType](#)
interactions algorithm
- [AttributeRanker * maineffectAlgorithm](#)
pointer to a main effects algorithm object
- [AttributeRanker * interactionAlgorithm](#)
pointer to an interaction ranker algorithm object
- bool [optimizeTemperature](#)
- double [optimalTemperature](#)
- std::vector< double > [temperatures](#)
- double [bestClassificationError](#)
- std::vector< double > [classificationErrors](#)
classification errors used by the T optimization algorithm
- [AttributeScores maineffectScores](#)
current random jungle scores
- [AttributeScores interactionScores](#)
current interaction scores
- [AttributeScores freeEnergyScores](#)
current free energy scores
- unsigned int [numRFTThreads](#)

- `unsigned int numToRemovePerIteration`
number of attributes to remove per iteration
- `unsigned int numToRemoveNextIteration`
number of attributes to remove next iteration
- `unsigned int numTargetAttributes`
number of target attributes
- `AttributeScores evaporatedAttributes`
attributes that have been evaporated so far
- `AttributeScores ecScores`
current set of ec scores

6.38.1 Detailed Description

Evaporative Cooling attribute ranking algorithm.

Implements the Evaporative Cooling algorithm in: McKinney, et. al. "Capturing the Spectrum of Interaction Effects in Genetic Association Studies by Simulated Evaporative Cooling Network Analysis." PLoS Genetics, Vol 5, Issue 3, 2009.

See also

[ReliefF](#)
[RReliefF](#)

Author

Bill White

Version

2.0

Contact: bill.c.white@gmail.com Created on: 7/14/11

Made even more generic with main effects and interaction effects algorithms in a class hierarchy from a [AttributeRanker](#) base. 8/12/12

Modified for inbix. 9/29/16

Definition at line 36 of file EvaporativeCooling.h.

6.38.2 Constructor & Destructor Documentation

6.38.2.1 EvaporativeCooling()

```
EvaporativeCooling::EvaporativeCooling (
    Dataset * ds,
    Plink * plinkPtr,
    AnalysisType anaType = SNP_ONLY_ANALYSIS )
```

Construct an EC algorithm object.

Parameters

in	<i>ds</i>	pointer to a Dataset object
in	<i>vm</i>	reference to a Boost map of command line options
in	<i>anaType</i>	analysis type

set the EC steps to perform

set the main effects algorithm

set the interaction algorithm

Definition at line 49 of file EvaporativeCooling.cpp.

6.38.2.2 ~EvaporativeCooling()

```
EvaporativeCooling::~EvaporativeCooling ( ) [virtual]
```

Definition at line 173 of file EvaporativeCooling.cpp.

6.38.3 Member Function Documentation**6.38.3.1 ComputeClassificationErrorRandomForest()**

```
double EvaporativeCooling::ComputeClassificationErrorRandomForest ( ) [private]
```

use Random Jungle to compute the classification error of the current set of attributes with numToRemovePerIteration attributes removed

get the best attribute names based on free energy score

return the classification error on this data

Definition at line 826 of file EvaporativeCooling.cpp.

6.38.3.2 ComputeECScores()

```
bool EvaporativeCooling::ComputeECScores ( )
```

Compute the EC scores based on the current set of attributes.

Definition at line 182 of file EvaporativeCooling.cpp.

6.38.3.3 ComputeFreeEnergy()

```
bool EvaporativeCooling::ComputeFreeEnergy (
    double temperature ) [private]
```

Compute the attribute free energy using the couple temperature.

Parameters

in	<i>temperature</i>	coupling temperature T
----	--------------------	------------------------

Returns

distance

Definition at line 700 of file EvaporativeCooling.cpp.

6.38.3.4 GetAlgorithmType()

```
EcAlgorithmType EvaporativeCooling::GetAlgorithmType ()
```

Return the algorithm type.

Definition at line 366 of file EvaporativeCooling.cpp.

6.38.3.5 GetECScores()

```
AttributeScores & EvaporativeCooling::GetECScores ()
```

Get the last computed EC scores.

Definition at line 362 of file EvaporativeCooling.cpp.

6.38.3.6 GetInteractionScores()

```
AttributeScores & EvaporativeCooling::GetInteractionScores ()
```

Get the last computed [ReliefF](#) scores.

Definition at line 358 of file EvaporativeCooling.cpp.

6.38.3.7 GetMaineffectScores()

```
AttributeScores & EvaporativeCooling::GetMaineffectScores ( )
```

Get the last computed RandomJungle scores.

Definition at line 354 of file EvaporativeCooling.cpp.

6.38.3.8 OptimizeTemperature()

```
double EvaporativeCooling::OptimizeTemperature (
    std::vector< double > deltas ) [private]
```

optimize the temperature coupling constant

for each delta, run a classifier on the best attributes according to the free energy and update best temperature

if classification error is lower at this delta, update best temperature and best classification error

Definition at line 786 of file EvaporativeCooling.cpp.

6.38.3.9 PrintAllAttributeScores()

```
void EvaporativeCooling::PrintAllAttributeScores (
    std::ofstream & outStream )
```

Write the EC scores and attribute names to stream, includes evaporated attributes with score of zero.

Parameters

in	<i>outStream</i>	stream to write score-attribute name pairs
----	------------------	--

Definition at line 378 of file EvaporativeCooling.cpp.

6.38.3.10 PrintAllScoresTabular()

```
bool EvaporativeCooling::PrintAllScoresTabular ( )
```

Print the current attributes scores to stdout in tab-delimited format.

Definition at line 529 of file EvaporativeCooling.cpp.

6.38.3.11 PrintAttributeScores()

```
void EvaporativeCooling::PrintAttributeScores (
    std::ofstream & outStream )
```

Write the EC scores and attribute names to stream.

Parameters

in	<i>outStream</i>	stream to write score-attribute name pairs
----	------------------	--

Definition at line 370 of file EvaporativeCooling.cpp.

6.38.3.12 PrintInteractionAttributeScores()

```
void EvaporativeCooling::PrintInteractionAttributeScores (
    std::ofstream & outStream )
```

Write the interaction scores and attribute names to stream.

Parameters

in	<i>outStream</i>	stream to write score-attribute name pairs
----	------------------	--

Definition at line 400 of file EvaporativeCooling.cpp.

6.38.3.13 PrintKendallTaus()

```
bool EvaporativeCooling::PrintKendallTaus ( )
```

Print the Kendall taus between the main effects and interactions scores.

Definition at line 563 of file EvaporativeCooling.cpp.

6.38.3.14 PrintMainEffectAttributeScores()

```
void EvaporativeCooling::PrintMainEffectAttributeScores (
    std::ofstream & outStream )
```

Write the main effect scores and attribute names to stream.

Parameters

in	<i>outStream</i>	stream to write score-attribute name pairs
----	------------------	--

Definition at line 391 of file EvaporativeCooling.cpp.

6.38.3.15 RemoveWorstAttributes()

```
bool EvaporativeCooling::RemoveWorstAttributes (
    unsigned int numToRemove = 1 ) [private]
```

Remove the worst attribute based on free energy scores.

Parameters

in	<i>numToRemove</i>	number of attributes to remove/evaporate
----	--------------------	--

Returns

distance

Definition at line 744 of file EvaporativeCooling.cpp.

6.38.3.16 RunRandomForest()

```
bool EvaporativeCooling::RunRandomForest ( ) [private]
```

Run the random forest algorithm.

Definition at line 606 of file EvaporativeCooling.cpp.

6.38.3.17 RunReliefF()

```
bool EvaporativeCooling::RunReliefF ( ) [private]
```

Run the [ReliefF](#) algorithm.

postcondition: interactionScores contains the newly-computed scores

Definition at line 648 of file EvaporativeCooling.cpp.

6.38.3.18 WriteAttributeScores()

```
void EvaporativeCooling::WriteAttributeScores (
    std::string basefilename )
```

Write the scores and attribute names to file.

Parameters

in	<i>baseFilename</i>	filename to write score-attribute name pairs
----	---------------------	--

Definition at line 409 of file EvaporativeCooling.cpp.

6.38.3.19 WriteClassificationErrors()

```
void EvaporativeCooling::WriteClassificationErrors (
    std::string filename )
```

Write the classification errors to file.

Parameters

in	<i>filename</i>	filename to write classification errors
----	-----------------	---

Definition at line 495 of file EvaporativeCooling.cpp.

6.38.3.20 WriteTemperatures()

```
void EvaporativeCooling::WriteTemperatures (
    std::string filename )
```

Write the temperatures to file.

Parameters

in	<i>filename</i>	filename to write temperatures
----	-----------------	--------------------------------

Definition at line 512 of file EvaporativeCooling.cpp.

6.38.4 Member Data Documentation**6.38.4.1 algorithmType**

[EcAlgorithmType](#) EvaporativeCooling::algorithmType [private]

algorithm steps to perform

Definition at line 130 of file EvaporativeCooling.h.

6.38.4.2 analysisType

```
AnalysisType EvaporativeCooling::analysisType [private]
```

type of analysis to perform

See also

[ReliefF](#)
[RandomForest](#)

Definition at line 128 of file EvaporativeCooling.h.

6.38.4.3 bestClassificationError

```
double EvaporativeCooling::bestClassificationError [private]
```

Definition at line 144 of file EvaporativeCooling.h.

6.38.4.4 classificationErrors

```
std::vector<double> EvaporativeCooling::classificationErrors [private]
```

classification errors used by the T optimization algorithm

Definition at line 146 of file EvaporativeCooling.h.

6.38.4.5 dataset

```
Dataset* EvaporativeCooling::dataset [private]
```

pointer to a [Dataset](#) object

Definition at line 121 of file EvaporativeCooling.h.

6.38.4.6 ecScores

`AttributeScores` EvaporativeCooling::ecScores [private]

current set of ec scores

Definition at line 167 of file EvaporativeCooling.h.

6.38.4.7 evaporatedAttributes

`AttributeScores` EvaporativeCooling::evaporatedAttributes [private]

attributes that have been evaporated so far

Definition at line 165 of file EvaporativeCooling.h.

6.38.4.8 freeEnergyScores

`AttributeScores` EvaporativeCooling::freeEnergyScores [private]

current free energy scores

Definition at line 153 of file EvaporativeCooling.h.

6.38.4.9 interactionAlgorithm

`AttributeRanker*` EvaporativeCooling::interactionAlgorithm [private]

pointer to an interaction ranker algorithm object

Definition at line 139 of file EvaporativeCooling.h.

6.38.4.10 interactionScores

`AttributeScores` EvaporativeCooling::interactionScores [private]

current interaction scores

Definition at line 151 of file EvaporativeCooling.h.

6.38.4.11 itAlgorithmType

`EcItAlgorithmType` `EvaporativeCooling::itAlgorithmType` [private]

interactions algorithm

Definition at line 134 of file `EvaporativeCooling.h`.

6.38.4.12 maineffectAlgorithm

`AttributeRanker*` `EvaporativeCooling::maineffectAlgorithm` [private]

pointer to a main effects algorithm object

Definition at line 137 of file `EvaporativeCooling.h`.

6.38.4.13 maineffectScores

`AttributeScores` `EvaporativeCooling::maineffectScores` [private]

current random jungle scores

Definition at line 149 of file `EvaporativeCooling.h`.

6.38.4.14 meAlgorithmType

`EcMeAlgorithmType` `EvaporativeCooling::meAlgorithmType` [private]

main effects algorithm

Definition at line 132 of file `EvaporativeCooling.h`.

6.38.4.15 numRFTThreads

`unsigned int` `EvaporativeCooling::numRFTThreads` [private]

Definition at line 156 of file `EvaporativeCooling.h`.

6.38.4.16 numTargetAttributes

```
unsigned int EvaporativeCooling::numTargetAttributes [private]
```

number of target attributes

Definition at line 163 of file EvaporativeCooling.h.

6.38.4.17 numToRemoveNextIteration

```
unsigned int EvaporativeCooling::numToRemoveNextIteration [private]
```

number of attributes to remove next iteration

Definition at line 160 of file EvaporativeCooling.h.

6.38.4.18 numToRemovePerIteration

```
unsigned int EvaporativeCooling::numToRemovePerIteration [private]
```

number of attributes to remove per iteration

Definition at line 158 of file EvaporativeCooling.h.

6.38.4.19 optimalTemperature

```
double EvaporativeCooling::optimalTemperature [private]
```

Definition at line 142 of file EvaporativeCooling.h.

6.38.4.20 optimizeTemperature

```
bool EvaporativeCooling::optimizeTemperature [private]
```

Definition at line 141 of file EvaporativeCooling.h.

6.38.4.21 PP

`Plink*` `EvaporativeCooling::PP` [private]

command line parameters map

Definition at line 123 of file `EvaporativeCooling.h`.

6.38.4.22 temperatures

`std::vector<double>` `EvaporativeCooling::temperatures` [private]

Definition at line 143 of file `EvaporativeCooling.h`.

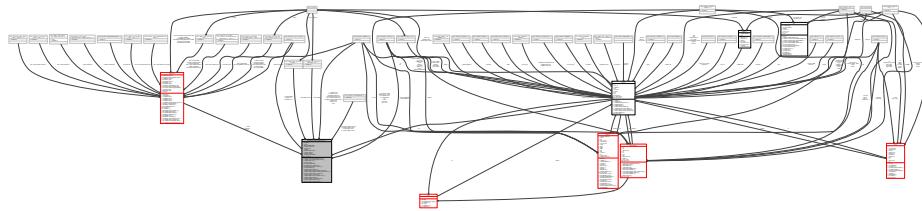
The documentation for this class was generated from the following files:

- [src/EvaporativeCooling.h](#)
- [src/EvaporativeCooling.cpp](#)

6.39 EvaporativeCoolingPrivacy Class Reference

```
#include <EvaporativeCoolingPrivacy.h>
```

Collaboration diagram for `EvaporativeCoolingPrivacy`:



Public Member Functions

- [EvaporativeCoolingPrivacy](#) (`Dataset *trainset, Dataset *holdoset, Dataset *testset, Plink *plinkPtr, bool datasetsAreSims`)
- virtual [~EvaporativeCoolingPrivacy](#) ()
- `bool ComputeScores ()`
- `void PrintState ()`
- [ResultsLists GetKeptRemoved \(\)](#)
- `bool WriteBestAttributes (std::string fileSuffix)`
- `std::pair< uint, double > CheckDetectedAttributes ()`
- `bool UsingSimData ()`

Private Member Functions

- bool `ComputeImportance ()`
- bool `ComputeAttributeProbabilities ()`
- bool `GenerateRandomUniformProbabilities ()`
- uint `EvaporateWorstAttributes (uint numToRemove)`
- double `ClassifyAttributeSet (std::vector< std::string > attrs, DATASET_TYPE)`
- bool `ComputeBestAttributesErrors ()`
- bool `UpdateTemperature ()`
- bool `ComputeInverseImportance ()`
- uint `CurrentNumberCorrect (std::vector< std::string > testSet)`
- bool `RemoveImportanceScore (std::string varToRemove)`

Private Attributes

- std::mt19937_64 `engine`
- `Plink * PP`
- double `Q_EPS`
- uint `MAX_ITERATIONS`
- uint `updateInterval`
- uint `iteration`
- uint `update`
- bool `dataIsSimulated`
- Dataset * `train`
- Dataset * `holdout`
- Dataset * `test`
- uint `numInstances`
- uint `numVariables`
- uint `numSignalsInData`
- vector< string > `signalNames`
- std::vector< std::string > `curVarNames`
- std::map< std::string, unsigned int > `curVarMap`
- AttributeScores `trainImportance`
- AttributeScores `holdoutImportance`
- AttributeScores `trainInvImportance`
- AttributeScores `holdoutInvImportance`
- std::map< std::string, double > `diffImportance`
- std::vector< double > `diffScores`
- std::vector< double > `diff`
- double `deltaQ`
- double `threshold`
- double `tolerance`
- double `startTemp`
- double `currentTemp`
- double `finalTemp`
- double `tau`
- std::vector< double > `attributeProbability`
- double `summedProbabilities`
- std::vector< double > `scaledProbabilities`
- std::vector< double > `cumulativeProbabilities`

- std::vector< double > `randUniformProbs`
- double `randUniformValue`
- vector< string > `possiblyRemove`
- std::vector< std::string > `removeAttrs`
- std::vector< std::string > `keepAttrs`
- double `randomForestPredictError`
- double `trainError`
- double `holdError`
- double `testError`
- std::vector< double > `trainErrors`
- std::vector< double > `holdoutErrors`
- std::vector< double > `testErrors`

6.39.1 Detailed Description

Definition at line 30 of file EvaporativeCoolingPrivacy.h.

6.39.2 Constructor & Destructor Documentation

6.39.2.1 EvaporativeCoolingPrivacy()

```
EvaporativeCoolingPrivacy::EvaporativeCoolingPrivacy (
    Dataset * trainset,
    Dataset * holdset,
    Dataset * testset,
    Plink * plinkPtr,
    bool datasetsAreSims )
```

Definition at line 42 of file EvaporativeCoolingPrivacy.cpp.

6.39.2.2 ~EvaporativeCoolingPrivacy()

```
EvaporativeCoolingPrivacy::~EvaporativeCoolingPrivacy ( ) [virtual]
```

Definition at line 112 of file EvaporativeCoolingPrivacy.cpp.

6.39.3 Member Function Documentation

6.39.3.1 CheckDetectedAttributes()

```
pair< uint, double > EvaporativeCoolingPrivacy::CheckDetectedAttributes ( )
```

Definition at line 324 of file EvaporativeCoolingPrivacy.cpp.

6.39.3.2 ClassifyAttributeSet()

```
double EvaporativeCoolingPrivacy::ClassifyAttributeSet (  
    std::vector< std::string > attrs,  
    DATASET_TYPE dataType ) [private]
```

Definition at line 616 of file EvaporativeCoolingPrivacy.cpp.

6.39.3.3 ComputeAttributeProbabilities()

```
bool EvaporativeCoolingPrivacy::ComputeAttributeProbabilities ( ) [private]
```

Definition at line 395 of file EvaporativeCoolingPrivacy.cpp.

6.39.3.4 ComputeBestAttributesErrors()

```
bool EvaporativeCoolingPrivacy::ComputeBestAttributesErrors ( ) [private]
```

Definition at line 580 of file EvaporativeCoolingPrivacy.cpp.

6.39.3.5 ComputeImportance()

```
bool EvaporativeCoolingPrivacy::ComputeImportance ( ) [private]
```

Definition at line 342 of file EvaporativeCoolingPrivacy.cpp.

6.39.3.6 ComputeInverseImportance()

```
bool EvaporativeCoolingPrivacy::ComputeInverseImportance ( ) [private]
```

Definition at line 671 of file EvaporativeCoolingPrivacy.cpp.

6.39.3.7 ComputeScores()

```
bool EvaporativeCoolingPrivacy::ComputeScores ( )
```

Definition at line 115 of file EvaporativeCoolingPrivacy.cpp.

6.39.3.8 CurrentNumberCorrect()

```
uint EvaporativeCoolingPrivacy::CurrentNumberCorrect ( std::vector< std::string > testSet ) [private]
```

Definition at line 252 of file EvaporativeCoolingPrivacy.cpp.

6.39.3.9 EvaporateWorstAttributes()

```
uint EvaporativeCoolingPrivacy::EvaporateWorstAttributes ( uint numToRemove ) [private]
```

Definition at line 478 of file EvaporativeCoolingPrivacy.cpp.

6.39.3.10 GenerateRandomUniformProbabilities()

```
bool EvaporativeCoolingPrivacy::GenerateRandomUniformProbabilities ( ) [private]
```

Definition at line 458 of file EvaporativeCoolingPrivacy.cpp.

6.39.3.11 GetKeptRemoved()

```
ResultsLists EvaporativeCoolingPrivacy::GetKeptRemoved ( )
```

Definition at line 295 of file EvaporativeCoolingPrivacy.cpp.

6.39.3.12 PrintState()

```
void EvaporativeCoolingPrivacy::PrintState ( )
```

Definition at line 267 of file EvaporativeCoolingPrivacy.cpp.

6.39.3.13 RemoveImportanceScore()

```
bool EvaporativeCoolingPrivacy::RemoveImportanceScore (
    std::string varToRemove ) [private]
```

Definition at line 532 of file EvaporativeCoolingPrivacy.cpp.

6.39.3.14 UpdateTemperature()

```
bool EvaporativeCoolingPrivacy::UpdateTemperature ( ) [private]
```

Definition at line 662 of file EvaporativeCoolingPrivacy.cpp.

6.39.3.15 UsingSimData()

```
bool EvaporativeCoolingPrivacy::UsingSimData ( ) [inline]
```

Definition at line 41 of file EvaporativeCoolingPrivacy.h.

6.39.3.16 WriteBestAttributes()

```
bool EvaporativeCoolingPrivacy::WriteBestAttributes (
    std::string fileSuffix )
```

Definition at line 303 of file EvaporativeCoolingPrivacy.cpp.

6.39.4 Member Data Documentation

6.39.4.1 attributeProbability

```
std::vector<double> EvaporativeCoolingPrivacy::attributeProbability [private]
```

Definition at line 103 of file EvaporativeCoolingPrivacy.h.

6.39.4.2 cummulativeProbabilities

```
std::vector<double> EvaporativeCoolingPrivacy::cummulativeProbabilities [private]
```

Definition at line 106 of file EvaporativeCoolingPrivacy.h.

6.39.4.3 currentTemp

```
double EvaporativeCoolingPrivacy::currentTemp [private]
```

Definition at line 98 of file EvaporativeCoolingPrivacy.h.

6.39.4.4 curVarMap

```
std::map<std::string, unsigned int> EvaporativeCoolingPrivacy::curVarMap [private]
```

Definition at line 80 of file EvaporativeCoolingPrivacy.h.

6.39.4.5 curVarNames

```
std::vector<std::string> EvaporativeCoolingPrivacy::curVarNames [private]
```

Definition at line 79 of file EvaporativeCoolingPrivacy.h.

6.39.4.6 **dataIsSimulated**

```
bool EvaporativeCoolingPrivacy::dataIsSimulated [private]
```

Definition at line 69 of file EvaporativeCoolingPrivacy.h.

6.39.4.7 **deltaQ**

```
double EvaporativeCoolingPrivacy::deltaQ [private]
```

Definition at line 92 of file EvaporativeCoolingPrivacy.h.

6.39.4.8 **diff**

```
std::vector<double> EvaporativeCoolingPrivacy::diff [private]
```

Definition at line 91 of file EvaporativeCoolingPrivacy.h.

6.39.4.9 **diffImportance**

```
std::map<std::string, double> EvaporativeCoolingPrivacy::diffImportance [private]
```

Definition at line 89 of file EvaporativeCoolingPrivacy.h.

6.39.4.10 **diffScores**

```
std::vector<double> EvaporativeCoolingPrivacy::diffScores [private]
```

Definition at line 90 of file EvaporativeCoolingPrivacy.h.

6.39.4.11 **engine**

```
std::mt19937_64 EvaporativeCoolingPrivacy::engine [private]
```

Definition at line 56 of file EvaporativeCoolingPrivacy.h.

6.39.4.12 finalTemp

```
double EvaporativeCoolingPrivacy::finalTemp [private]
```

Definition at line 99 of file EvaporativeCoolingPrivacy.h.

6.39.4.13 holdError

```
double EvaporativeCoolingPrivacy::holdError [private]
```

Definition at line 116 of file EvaporativeCoolingPrivacy.h.

6.39.4.14 holdout

```
Dataset* EvaporativeCoolingPrivacy::holdout [private]
```

Definition at line 71 of file EvaporativeCoolingPrivacy.h.

6.39.4.15 holdoutErrors

```
std::vector<double> EvaporativeCoolingPrivacy::holdoutErrors [private]
```

Definition at line 119 of file EvaporativeCoolingPrivacy.h.

6.39.4.16 holdoutImportance

```
AttributeScores EvaporativeCoolingPrivacy::holdoutImportance [private]
```

Definition at line 84 of file EvaporativeCoolingPrivacy.h.

6.39.4.17 holdoutInvImportance

```
AttributeScores EvaporativeCoolingPrivacy::holdoutInvImportance [private]
```

Definition at line 86 of file EvaporativeCoolingPrivacy.h.

6.39.4.18 iteration

```
uint EvaporativeCoolingPrivacy::iteration [private]
```

Definition at line 65 of file EvaporativeCoolingPrivacy.h.

6.39.4.19 keepAttrs

```
std::vector<std::string> EvaporativeCoolingPrivacy::keepAttrs [private]
```

Definition at line 111 of file EvaporativeCoolingPrivacy.h.

6.39.4.20 MAX_ITERATIONS

```
uint EvaporativeCoolingPrivacy::MAX_ITERATIONS [private]
```

Definition at line 61 of file EvaporativeCoolingPrivacy.h.

6.39.4.21 numInstances

```
uint EvaporativeCoolingPrivacy::numInstances [private]
```

Definition at line 73 of file EvaporativeCoolingPrivacy.h.

6.39.4.22 numSignalsInData

```
uint EvaporativeCoolingPrivacy::numSignalsInData [private]
```

Definition at line 75 of file EvaporativeCoolingPrivacy.h.

6.39.4.23 numVariables

```
uint EvaporativeCoolingPrivacy::numVariables [private]
```

Definition at line 74 of file EvaporativeCoolingPrivacy.h.

6.39.4.24 possiblyRemove

```
vector<string> EvaporativeCoolingPrivacy::possiblyRemove [private]
```

Definition at line 109 of file EvaporativeCoolingPrivacy.h.

6.39.4.25 PP

```
Plink* EvaporativeCoolingPrivacy::PP [private]
```

Definition at line 58 of file EvaporativeCoolingPrivacy.h.

6.39.4.26 Q_EPS

```
double EvaporativeCoolingPrivacy::Q_EPS [private]
```

Definition at line 60 of file EvaporativeCoolingPrivacy.h.

6.39.4.27 randomForestPredictError

```
double EvaporativeCoolingPrivacy::randomForestPredictError [private]
```

Definition at line 114 of file EvaporativeCoolingPrivacy.h.

6.39.4.28 randUniformProbs

```
std::vector<double> EvaporativeCoolingPrivacy::randUniformProbs [private]
```

Definition at line 107 of file EvaporativeCoolingPrivacy.h.

6.39.4.29 randUniformValue

```
double EvaporativeCoolingPrivacy::randUniformValue [private]
```

Definition at line 108 of file EvaporativeCoolingPrivacy.h.

6.39.4.30 removeAttrs

```
std::vector<std::string> EvaporativeCoolingPrivacy::removeAttrs [private]
```

Definition at line 110 of file EvaporativeCoolingPrivacy.h.

6.39.4.31 scaledProbabilities

```
std::vector<double> EvaporativeCoolingPrivacy::scaledProbabilities [private]
```

Definition at line 105 of file EvaporativeCoolingPrivacy.h.

6.39.4.32 signalNames

```
vector<string> EvaporativeCoolingPrivacy::signalNames [private]
```

Definition at line 76 of file EvaporativeCoolingPrivacy.h.

6.39.4.33 startTemp

```
double EvaporativeCoolingPrivacy::startTemp [private]
```

Definition at line 97 of file EvaporativeCoolingPrivacy.h.

6.39.4.34 summedProbabilities

```
double EvaporativeCoolingPrivacy::summedProbabilities [private]
```

Definition at line 104 of file EvaporativeCoolingPrivacy.h.

6.39.4.35 tau

```
double EvaporativeCoolingPrivacy::tau [private]
```

Definition at line 100 of file EvaporativeCoolingPrivacy.h.

6.39.4.36 test

```
Dataset* EvaporativeCoolingPrivacy::test [private]
```

Definition at line 72 of file EvaporativeCoolingPrivacy.h.

6.39.4.37 testError

```
double EvaporativeCoolingPrivacy::testError [private]
```

Definition at line 117 of file EvaporativeCoolingPrivacy.h.

6.39.4.38 testErrors

```
std::vector<double> EvaporativeCoolingPrivacy::testErrors [private]
```

Definition at line 120 of file EvaporativeCoolingPrivacy.h.

6.39.4.39 threshold

```
double EvaporativeCoolingPrivacy::threshold [private]
```

Definition at line 93 of file EvaporativeCoolingPrivacy.h.

6.39.4.40 tolerance

```
double EvaporativeCoolingPrivacy::tolerance [private]
```

Definition at line 94 of file EvaporativeCoolingPrivacy.h.

6.39.4.41 train

```
Dataset* EvaporativeCoolingPrivacy::train [private]
```

Definition at line 70 of file EvaporativeCoolingPrivacy.h.

6.39.4.42 trainError

```
double EvaporativeCoolingPrivacy::trainError [private]
```

Definition at line 115 of file EvaporativeCoolingPrivacy.h.

6.39.4.43 trainErrors

```
std::vector<double> EvaporativeCoolingPrivacy::trainErrors [private]
```

Definition at line 118 of file EvaporativeCoolingPrivacy.h.

6.39.4.44 trainImportance

```
AttributeScores EvaporativeCoolingPrivacy::trainImportance [private]
```

Definition at line 83 of file EvaporativeCoolingPrivacy.h.

6.39.4.45 trainInvImportance

```
AttributeScores EvaporativeCoolingPrivacy::trainInvImportance [private]
```

Definition at line 85 of file EvaporativeCoolingPrivacy.h.

6.39.4.46 update

```
uint EvaporativeCoolingPrivacy::update [private]
```

Definition at line 66 of file EvaporativeCoolingPrivacy.h.

6.39.4.47 updateInterval

```
uint EvaporativeCoolingPrivacy::updateInterval [private]
```

Definition at line 64 of file `EvaporativeCoolingPrivacy.h`.

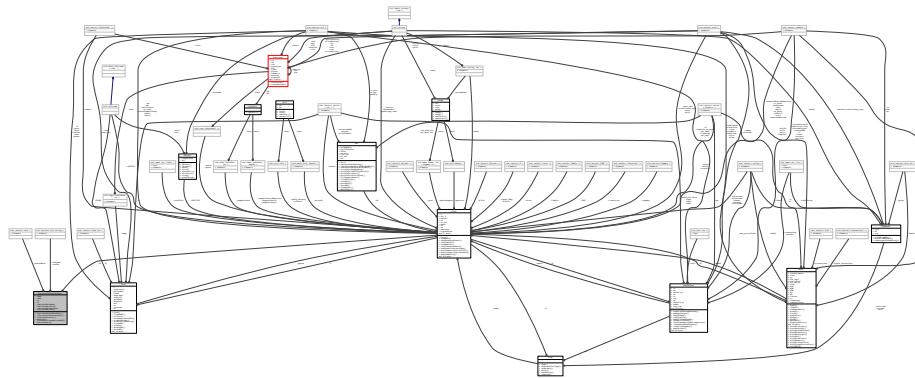
The documentation for this class was generated from the following files:

- [src/EvaporativeCoolingPrivacy.h](#)
- [src/EvaporativeCoolingPrivacy.cpp](#)

6.40 ExpressionDataSimulator Class Reference

```
#include <ExpressionDataSimulator.h>
```

Collaboration diagram for ExpressionDataSimulator:



Public Member Functions

- [ExpressionDataSimulator \(Plink *plinkPtr\)](#)
- virtual [~ExpressionDataSimulator \(\)](#)
- bool [CreateSimulation \(uint n, uint d, double pb, double bias, std::string type\)](#)
- const arma::mat & [GetData \(SplitType splitType\)](#)

Private Member Functions

- bool [CreateDiffCoexpMatrixNoME \(uint M, uint N, double meanExpression, arma::mat A, double randSdNoise, double sdNoise, std::vector< uint > sampleIndicesInteraction\)](#)
- bool [SimulateData \(uint n_e, uint n_db, uint n_ns, std::vector< std::string > sv_db, std::vector< std::string > sv_ns, double sd_b, double sd_gam, double sd_u, bool conf, std::string distr_db, double p_b, double p_gam, double p_ov\)](#)

Private Attributes

- `Plink * PP`
- `utin dimM`
- `uint dimN`
- `uint n1`
- `uint n2`
- `arma::mat simulatedDataBase`
- `std::vector< std::string > sublds`
- `std::vector< std::string > colNames`
- `std::vector< uint > phenosBase`
- `arma::mat simulatedDataTrain`
- `arma::mat simulatedDataHoldout`
- `arma::mat simulatedDataTest`

6.40.1 Detailed Description

Definition at line 28 of file ExpressionDataSimulator.h.

6.40.2 Constructor & Destructor Documentation

6.40.2.1 ExpressionDataSimulator()

```
ExpressionDataSimulator::ExpressionDataSimulator (
    Plink * plinkPtr )
```

6.40.2.2 ~ExpressionDataSimulator()

```
virtual ExpressionDataSimulator::~ExpressionDataSimulator ( ) [virtual]
```

6.40.3 Member Function Documentation

6.40.3.1 CreateDiffCoexpMatrixNoME()

```
bool ExpressionDataSimulator::CreateDiffCoexpMatrixNoME (
    uint M,
    uint N,
    double meanExpression,
    arma::mat A,
    double randSdNoise,
    double sdNoise,
    std::vector< uint > sampleIndicesInteraction ) [private]
```

6.40.3.2 CreateSimulation()

```
bool ExpressionDataSimulator::CreateSimulation (
    uint n,
    uint d,
    double pb,
    double bias,
    std::string type )
```

6.40.3.3 GetData()

```
const arma::mat& ExpressionDataSimulator::GetData (
    SplitType splitType )
```

6.40.3.4 SimulateData()

```
bool ExpressionDataSimulator::SimulateData (
    uint n_e,
    uint n_db,
    uint n_ns,
    std::vector< std::string > sv_db,
    std::vector< std::string > sv_ns,
    double sd_b,
    double sd_gam,
    double sd_u,
    bool conf,
    std::string distr_db,
    double p_b,
    double p_gam,
    double p_ov ) [private]
```

6.40.4 Member Data Documentation

6.40.4.1 colNames

```
std::vector<std::string> ExpressionDataSimulator::colNames [private]
```

Definition at line 49 of file ExpressionDataSimulator.h.

6.40.4.2 dimM

```
utin ExpressionDataSimulator::dimM [private]
```

Definition at line 43 of file ExpressionDataSimulator.h.

6.40.4.3 dimN

```
uint ExpressionDataSimulator::dimN [private]
```

Definition at line 44 of file ExpressionDataSimulator.h.

6.40.4.4 n1

```
uint ExpressionDataSimulator::n1 [private]
```

Definition at line 45 of file ExpressionDataSimulator.h.

6.40.4.5 n2

```
uint ExpressionDataSimulator::n2 [private]
```

Definition at line 46 of file ExpressionDataSimulator.h.

6.40.4.6 phenosBase

```
std::vector<uint> ExpressionDataSimulator::phenosBase [private]
```

Definition at line 50 of file ExpressionDataSimulator.h.

6.40.4.7 PP

```
Plink* ExpressionDataSimulator::PP [private]
```

Definition at line 42 of file ExpressionDataSimulator.h.

6.40.4.8 simulatedDataBase

```
arma::mat ExpressionDataSimulator::simulatedDataBase [private]
```

Definition at line 47 of file ExpressionDataSimulator.h.

6.40.4.9 simulatedDataHoldout

```
arma::mat ExpressionDataSimulator::simulatedDataHoldout [private]
```

Definition at line 52 of file ExpressionDataSimulator.h.

6.40.4.10 simulatedDataTest

```
arma::mat ExpressionDataSimulator::simulatedDataTest [private]
```

Definition at line 53 of file ExpressionDataSimulator.h.

6.40.4.11 simulatedDataTrain

```
arma::mat ExpressionDataSimulator::simulatedDataTrain [private]
```

Definition at line 51 of file ExpressionDataSimulator.h.

6.40.4.12 sublds

```
std::vector<std::string> ExpressionDataSimulator::subIds [private]
```

Definition at line 48 of file ExpressionDataSimulator.h.

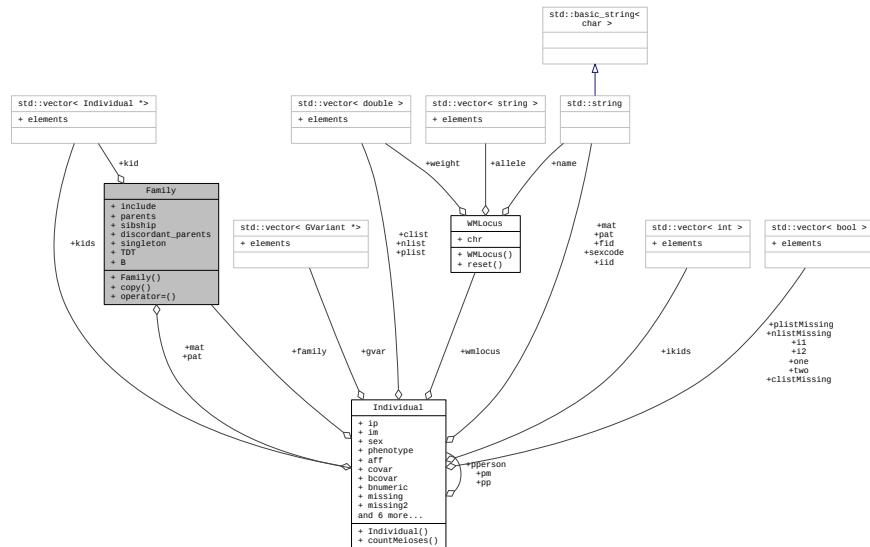
The documentation for this class was generated from the following file:

- `src/ExpressionDataSimulator.h`

6.41 Family Class Reference

```
#include <plink.h>
```

Collaboration diagram for Family:



Public Member Functions

- `Family ()`
 - `void copy (const Family &rhs)`
 - `Family & operator= (const Family &rhs)`

Public Attributes

- bool `include`
- bool `parents`
- bool `sibship`
- bool `discordant_parents`
- bool `singleton`
- bool `TDT`
- `Individual *` `pat`
- `Individual *` `mat`
- `vector< Individual * >` `kid`
- double `B`

6.41.1 Detailed Description

Definition at line 336 of file plink.h.

6.41.2 Constructor & Destructor Documentation

6.41.2.1 Family()

```
Family::Family ( ) [inline]
```

Definition at line 339 of file plink.h.

6.41.3 Member Function Documentation

6.41.3.1 copy()

```
void Family::copy (
    const Family & rhs ) [inline]
```

Definition at line 350 of file plink.h.

6.41.3.2 operator=()

```
Family& Family::operator= (
    const Family & rhs ) [inline]
```

Definition at line 359 of file plink.h.

6.41.4 Member Data Documentation

6.41.4.1 B

```
double Family::B
```

Definition at line 376 of file plink.h.

6.41.4.2 discordant_parents

```
bool Family::discordant_parents
```

Definition at line 367 of file plink.h.

6.41.4.3 include

```
bool Family::include
```

Definition at line 364 of file plink.h.

6.41.4.4 kid

```
vector<Individual *> Family::kid
```

Definition at line 373 of file plink.h.

6.41.4.5 mat

`Individual* Family::mat`

Definition at line 372 of file plink.h.

6.41.4.6 parents

`bool Family::parents`

Definition at line 365 of file plink.h.

6.41.4.7 pat

`Individual* Family::pat`

Definition at line 371 of file plink.h.

6.41.4.8 sibship

`bool Family::sibship`

Definition at line 366 of file plink.h.

6.41.4.9 singleton

`bool Family::singleton`

Definition at line 368 of file plink.h.

6.41.4.10 TDT

`bool Family::TDT`

Definition at line 369 of file plink.h.

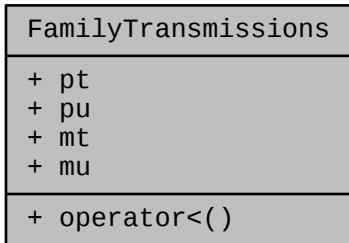
The documentation for this class was generated from the following file:

- [src/plink.h](#)

6.42 FamilyTransmissions Class Reference

```
#include <phase.h>
```

Collaboration diagram for FamilyTransmissions:



Public Member Functions

- bool `operator<` (const `FamilyTransmissions` &`b`) const

Public Attributes

- int `pt`
- int `pu`
- int `mt`
- int `mu`

6.42.1 Detailed Description

Definition at line 20 of file phase.h.

6.42.2 Member Function Documentation

6.42.2.1 `operator<()`

```
bool FamilyTransmissions::operator< (
    const FamilyTransmissions & b ) const [inline]
```

Definition at line 25 of file phase.h.

6.42.3 Member Data Documentation

6.42.3.1 mt

```
int FamilyTransmissions::mt
```

Definition at line 23 of file phase.h.

6.42.3.2 mu

```
int FamilyTransmissions::mu
```

Definition at line 23 of file phase.h.

6.42.3.3 pt

```
int FamilyTransmissions::pt
```

Definition at line 23 of file phase.h.

6.42.3.4 pu

```
int FamilyTransmissions::pu
```

Definition at line 23 of file phase.h.

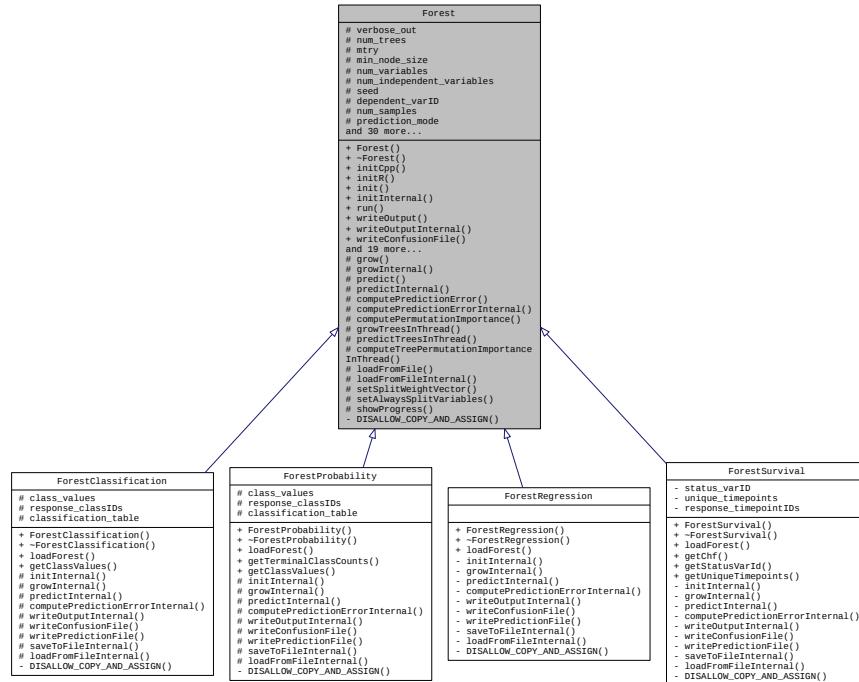
The documentation for this class was generated from the following file:

- [src/phase.h](#)
-

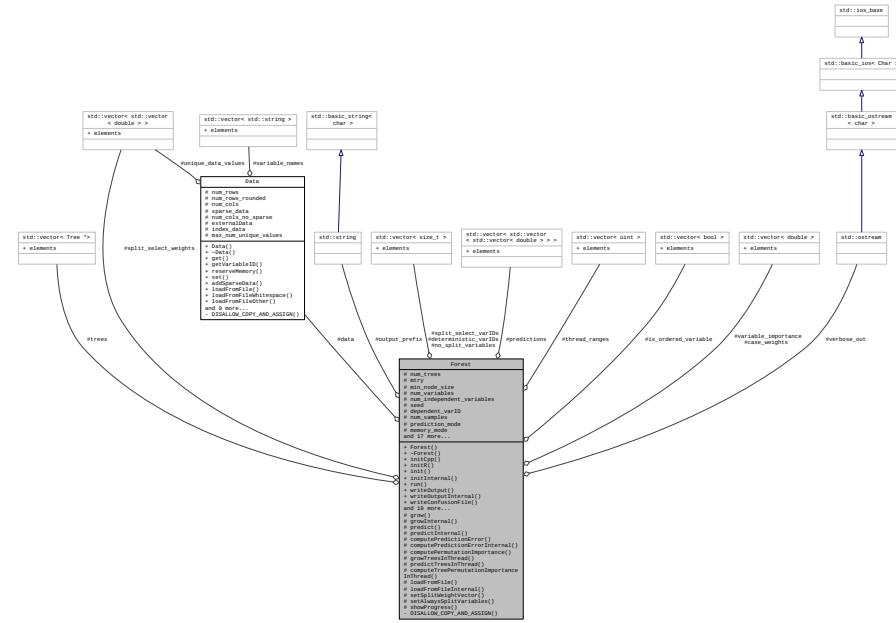
6.43 Forest Class Reference

```
#include <Forest.h>
```

Inheritance diagram for Forest:



Collaboration diagram for Forest:



Public Member Functions

- `Forest ()`
 - `virtual ~Forest ()`
 - `void initCpp (std::string dependent_variable_name, MemoryMode memory_mode, std::string input_file, uint mtry, std::string output_prefix, uint num_trees, std::ostream *verbose_out, uint seed, uint num_threads, std::string load_forest_filename, ImportanceMode importance_mode, uint min_node_size, std::string split_select_weights_file, std::vector< std::string > &always_split_variable_names, std::string status_variable_name, bool sample_with_replacement, std::vector< std::string > &unordered_variable_names, bool memory_saving_splitting, SplitRule splitrule, std::string case_weights_file, bool predict_all, double sample_fraction, double alpha, double minprop, bool holdout, PredictionType prediction_type, bool useMask, Dataset *dataset)`
 - `void initR (std::string dependent_variable_name, Data *input_data, uint mtry, uint num_trees, std::ostream *verbose_out, uint seed, uint num_threads, ImportanceMode importance_mode, uint min_node_size, std::vector< std::vector< double >> &split_select_weights, std::vector< std::string > &always_split_variable_names, std::string status_variable_name, bool prediction_mode, bool sample_with_replacement, std::vector< std::string > &unordered_variable_names, bool memory_saving_splitting, SplitRule splitrule, std::vector< double > &case_weights, bool predict_all, bool keep_inbag, double sample_fraction, double alpha, double minprop, bool holdout, PredictionType prediction_type)`
 - `void init (std::string dependent_variable_name, MemoryMode memory_mode, Data *input_data, uint mtry, std::string output_prefix, uint num_trees, uint seed, uint num_threads, ImportanceMode importance_mode, uint min_node_size, std::string status_variable_name, bool prediction_mode, bool sample_with_replacement, std::vector< std::string > &unordered_variable_names, bool memory_saving_splitting, SplitRule splitrule, bool predict_all, double sample_fraction, double alpha, double minprop, bool holdout, PredictionType prediction_type)`
 - `virtual void initInternal (std::string status_variable_name)=0`
 - `void run (bool verbose)`
 - `void writeOutput ()`
 - `virtual void writeOutputInternal ()=0`

- virtual void `writeConfusionFile ()=0`
- virtual void `writePredictionFile ()=0`
- void `writelImportanceFile ()`
- void `saveToFile ()`
- virtual void `saveToFileInternal (std::ofstream &outfile)=0`
- std::vector< std::vector< std::vector< size_t > >> `getChildNodeIDs ()`
- std::vector< std::vector< size_t > > `getSplitVarIDs ()`
- std::vector< std::vector< double > > `getSplitValues ()`
- const std::vector< double > & `getVariableImportance () const`
- double `getOverallPredictionError () const`
- const std::vector< std::vector< std::vector< double > > > & `getPredictions () const`
- virtual std::vector< double > `getPredictionValues ()`
- size_t `getDependentVarId () const`
- size_t `getNumTrees () const`
- uint `getMtry () const`
- uint `getMinNodeSize () const`
- size_t `getNumIndependentVariables () const`
- const std::vector< bool > & `getIsOrderedVariable () const`
- std::vector< std::vector< size_t > > `getInbagCounts () const`
- bool `setVerboseOutput (std::ostream *new_verbose_out)`

Protected Member Functions

- void `grow ()`
- virtual void `growInternal ()=0`
- void `predict ()`
- virtual void `predictInternal ()=0`
- void `computePredictionError ()`
- virtual void `computePredictionErrorInternal ()=0`
- void `computePermutationImportance ()`
- void `growTreesInThread (uint thread_idx, std::vector< double > *variable_importance)`
- void `predictTreesInThread (uint thread_idx, const Data *prediction_data, bool oob_prediction)`
- void `computeTreePermutationImportanceInThread (uint thread_idx, std::vector< double > *importance, std::vector< double > *variance)`
- void `loadFromFile (std::string filename)`
- virtual void `loadFromFileInternal (std::ifstream &infile)=0`
- void `setSplitWeightVector (std::vector< std::vector< double > >> &split_select_weights)`
- void `setAlwaysSplitVariables (std::vector< std::string > &always_split_variable_names)`
- void `showProgress (std::string operation)`

Protected Attributes

- std::ostream * `verbose_out`
- size_t `num_trees`
- uint `mtry`
- uint `min_node_size`
- size_t `num_variables`
- size_t `num_independent_variables`
- uint `seed`

- `size_t dependent_varID`
- `size_t num_samples`
- `bool prediction_mode`
- `MemoryMode memory_mode`
- `bool sample_with_replacement`
- `bool memory_saving_splitting`
- `SplitRule splitrule`
- `bool predict_all`
- `bool keep_inbag`
- `double sample_fraction`
- `bool holdout`
- `PredictionType prediction_type`
- `double alpha`
- `double minprop`
- `std::vector< bool > is_ordered_variable`
- `std::vector< size_t > no_split_variables`
- `uint num_threads`
- `std::vector< uint > thread_ranges`
- `std::mutex mutex`
- `std::condition_variable condition_variable`
- `std::vector< Tree * > trees`
- `Data * data`
- `std::vector< std::vector< std::vector< double > > > predictions`
- `double overall_prediction_error`
- `std::vector< size_t > deterministic_varIDs`
- `std::vector< size_t > split_select_varIDs`
- `std::vector< std::vector< double > > split_select_weights`
- `std::vector< double > case_weights`
- `std::mt19937_64 random_number_generator`
- `std::string output_prefix`
- `ImportanceMode importance_mode`
- `std::vector< double > variable_importance`
- `size_t progress`

Private Member Functions

- `DISALLOW_COPY_AND_ASSIGN (Forest)`

6.43.1 Detailed Description

Definition at line 49 of file Forest.h.

6.43.2 Constructor & Destructor Documentation

6.43.2.1 Forest()

```
Forest::Forest ( )
```

Definition at line 51 of file Forest.cpp.

6.43.2.2 ~Forest()

```
Forest::~Forest ( ) [virtual]
```

Definition at line 59 of file Forest.cpp.

6.43.3 Member Function Documentation

6.43.3.1 computePermutationImportance()

```
void Forest::computePermutationImportance ( ) [protected]
```

Definition at line 600 of file Forest.cpp.

6.43.3.2 computePredictionError()

```
void Forest::computePredictionError ( ) [protected]
```

Definition at line 566 of file Forest.cpp.

6.43.3.3 computePredictionErrorInternal()

```
virtual void Forest::computePredictionErrorInternal ( ) [protected], [pure virtual]
```

Implemented in [ForestSurvival](#), [ForestProbability](#), [ForestClassification](#), and [ForestRegression](#).

6.43.3.4 computeTreePermutationImportanceInThread()

```
void Forest::computeTreePermutationImportanceInThread (
    uint thread_idx,
    std::vector< double > * importance,
    std::vector< double > * variance ) [protected]
```

Definition at line 737 of file Forest.cpp.

6.43.3.5 DISALLOW_COPY_AND_ASSIGN()

```
Forest::DISALLOW_COPY_AND_ASSIGN (
    Forest ) [private]
```

6.43.3.6 getChildNodeIDs()

```
std::vector<std::vector<std::vector<size_t> > > Forest::getChildNodeIDs () [inline]
```

Definition at line 92 of file Forest.h.

6.43.3.7 getDependentVarId()

```
size_t Forest::getDependentVarId () const [inline]
```

Definition at line 133 of file Forest.h.

6.43.3.8 getInbagCounts()

```
std::vector<std::vector<size_t> > Forest::getInbagCounts () const [inline]
```

Definition at line 156 of file Forest.h.

6.43.3.9 getIsOrderedVariable()

```
const std::vector<bool>& Forest::getIsOrderedVariable ( ) const [inline]
```

Definition at line 151 of file Forest.h.

6.43.3.10 getMinNodeSize()

```
uint Forest::getMinNodeSize ( ) const [inline]
```

Definition at line 143 of file Forest.h.

6.43.3.11 getMtry()

```
uint Forest::getMtry ( ) const [inline]
```

Definition at line 139 of file Forest.h.

6.43.3.12 getNumIndependentVariables()

```
size_t Forest::getNumIndependentVariables ( ) const [inline]
```

Definition at line 147 of file Forest.h.

6.43.3.13 getNumTrees()

```
size_t Forest::getNumTrees ( ) const [inline]
```

Definition at line 136 of file Forest.h.

6.43.3.14 getOverallPredictionError()

```
double Forest::getOverallPredictionError ( ) const [inline]
```

Definition at line 116 of file Forest.h.

6.43.3.15 getPredictions()

```
const std::vector<std::vector<std::vector<double> > >& Forest::getPredictions ( ) const [inline]
```

Definition at line 119 of file Forest.h.

6.43.3.16 getPredictionValues()

```
virtual std::vector<double> Forest::getPredictionValues ( ) [inline], [virtual]
```

Definition at line 122 of file Forest.h.

6.43.3.17 getSplitValues()

```
std::vector<std::vector<double> > Forest::getSplitValues ( ) [inline]
```

Definition at line 106 of file Forest.h.

6.43.3.18 getSplitVarIDs()

```
std::vector<std::vector<size_t> > Forest::getSplitVarIDs ( ) [inline]
```

Definition at line 99 of file Forest.h.

6.43.3.19 getVariableImportance()

```
const std::vector<double>& Forest::getVariableImportance ( ) const [inline]
```

Definition at line 113 of file Forest.h.

6.43.3.20 grow()

```
void Forest::grow ( ) [protected]
```

Definition at line 430 of file Forest.cpp.

6.43.3.21 growInternal()

```
virtual void Forest::growInternal ( ) [protected], [pure virtual]
```

Implemented in [ForestSurvival](#), [ForestProbability](#), [ForestClassification](#), and [ForestRegression](#).

6.43.3.22 growTreesInThread()

```
void Forest::growTreesInThread (
    uint thread_idx,
    std::vector< double > * variable_importance ) [protected]
```

Definition at line 691 of file [Forest.cpp](#).

6.43.3.23 init()

```
void Forest::init (
    std::string dependent_variable_name,
    MemoryMode memory_mode,
    Data * input_data,
    uint mtry,
    std::string output_prefix,
    uint num_trees,
    uint seed,
    uint num_threads,
    ImportanceMode importance_mode,
    uint min_node_size,
    std::string status_variable_name,
    bool prediction_mode,
    bool sample_with_replacement,
    std::vector< std::string > & unordered_variable_names,
    bool memory_saving_splitting,
    SplitRule splitrule,
    bool predict_all,
    double sample_fraction,
    double alpha,
    double minprop,
    bool holdout,
    PredictionType prediction_type )
```

Definition at line 219 of file [Forest.cpp](#).

6.43.3.24 initCpp()

```
void Forest::initCpp (
    std::string dependent_variable_name,
    MemoryMode memory_mode,
    std::string input_file,
    uint mtry,
    std::string output_prefix,
    uint num_trees,
    std::ostream * verbose_out,
    uint seed,
    uint num_threads,
    std::string load_forest_filename,
    ImportanceMode importance_mode,
    uint min_node_size,
    std::string split_select_weights_file,
    std::vector< std::string > & always_split_variable_names,
    std::string status_variable_name,
    bool sample_with_replacement,
    std::vector< std::string > & unordered_variable_names,
    bool memory_saving_splitting,
    SplitRule splitrule,
    std::string case_weights_file,
    bool predict_all,
    double sample_fraction,
    double alpha,
    double minprop,
    bool holdout,
    PredictionType prediction_type,
    bool useMask,
    Dataset * dataset )
```

Definition at line 70 of file Forest.cpp.

6.43.3.25 initInternal()

```
virtual void Forest::initInternal (
    std::string status_variable_name ) [pure virtual]
```

Implemented in [ForestSurvival](#), [ForestProbability](#), [ForestClassification](#), and [ForestRegression](#).

6.43.3.26 initR()

```
void Forest::initR (
    std::string dependent_variable_name,
    Data * input_data,
    uint mtry,
    uint num_trees,
    std::ostream * verbose_out,
    uint seed,
    uint num_threads,
    ImportanceMode importance_mode,
    uint min_node_size,
    std::vector< std::vector< double >> & split_select_weights,
    std::vector< std::string > & always_split_variable_names,
    std::string status_variable_name,
    bool prediction_mode,
    bool sample_with_replacement,
    std::vector< std::string > & unordered_variable_names,
    bool memory_saving_splitting,
    SplitRule splitrule,
    std::vector< double > & case_weights,
    bool predict_all,
    bool keep_inbag,
    double sample_fraction,
    double alpha,
    double minprop,
    bool holdout,
    PredictionType prediction_type )
```

Definition at line 181 of file Forest.cpp.

6.43.3.27 loadFromFile()

```
void Forest::loadFromFile (
    std::string filename ) [protected]
```

Definition at line 762 of file Forest.cpp.

6.43.3.28 loadFromFileInternal()

```
virtual void Forest::loadFromFileInternal (
    std::ifstream & infile ) [protected], [pure virtual]
```

Implemented in [ForestSurvival](#), [ForestProbability](#), [ForestClassification](#), and [ForestRegression](#).

6.43.3.29 predict()

```
void Forest::predict ( ) [protected]
```

Definition at line 526 of file Forest.cpp.

6.43.3.30 predictInternal()

```
virtual void Forest::predictInternal ( ) [protected], [pure virtual]
```

Implemented in [ForestSurvival](#), [ForestProbability](#), [ForestClassification](#), and [ForestRegression](#).

6.43.3.31 predictTreesInThread()

```
void Forest::predictTreesInThread (
    uint thread_idx,
    const Data * prediction_data,
    bool oob_prediction ) [protected]
```

Definition at line 714 of file Forest.cpp.

6.43.3.32 run()

```
void Forest::run (
    bool verbose )
```

Definition at line 308 of file Forest.cpp.

6.43.3.33 saveToFile()

```
void Forest::saveToFile ( )
```

Definition at line 399 of file Forest.cpp.

6.43.3.34 saveToFileInternal()

```
virtual void Forest::saveToFileInternal (
    std::ofstream & outfile ) [pure virtual]
```

Implemented in [ForestSurvival](#), [ForestProbability](#), [ForestClassification](#), and [ForestRegression](#).

6.43.3.35 setAlwaysSplitVariables()

```
void Forest::setAlwaysSplitVariables (
    std::vector< std::string > & always_split_variable_names ) [protected]
```

Definition at line 851 of file Forest.cpp.

6.43.3.36 setSplitWeightVector()

```
void Forest::setSplitWeightVector (
    std::vector< std::vector< double >> & split_select_weights ) [protected]
```

Definition at line 788 of file Forest.cpp.

6.43.3.37 setVerboseOutput()

```
bool Forest::setVerboseOutput (
    std::ostream * new_verbose_out )
```

Definition at line 65 of file Forest.cpp.

6.43.3.38 showProgress()

```
void Forest::showProgress (
    std::string operation ) [protected]
```

Definition at line 885 of file Forest.cpp.

6.43.3.39 writeConfusionFile()

```
virtual void Forest::writeConfusionFile ( ) [pure virtual]
```

Implemented in [ForestSurvival](#), [ForestProbability](#), [ForestClassification](#), and [ForestRegression](#).

6.43.3.40 writeImportanceFile()

```
void Forest::writeImportanceFile ( )
```

Definition at line 373 of file Forest.cpp.

6.43.3.41 writeOutput()

```
void Forest::writeOutput ( )
```

Definition at line 336 of file Forest.cpp.

6.43.3.42 writeOutputInternal()

```
virtual void Forest::writeOutputInternal ( ) [pure virtual]
```

Implemented in [ForestSurvival](#), [ForestProbability](#), [ForestClassification](#), and [ForestRegression](#).

6.43.3.43 writePredictionFile()

```
virtual void Forest::writePredictionFile ( ) [pure virtual]
```

Implemented in [ForestSurvival](#), [ForestProbability](#), [ForestClassification](#), and [ForestRegression](#).

6.43.4 Member Data Documentation

6.43.4.1 alpha

```
double Forest::alpha [protected]
```

Definition at line 220 of file Forest.h.

6.43.4.2 case_weights

```
std::vector<double> Forest::case_weights [protected]
```

Definition at line 250 of file Forest.h.

6.43.4.3 condition_variable

```
std::condition_variable Forest::condition_variable [protected]
```

Definition at line 234 of file Forest.h.

6.43.4.4 data

```
Data* Forest::data [protected]
```

Definition at line 238 of file Forest.h.

6.43.4.5 dependent_varID

```
size_t Forest::dependent_varID [protected]
```

Definition at line 206 of file Forest.h.

6.43.4.6 deterministic_varIDs

```
std::vector<size_t> Forest::deterministic_varIDs [protected]
```

Definition at line 245 of file Forest.h.

6.43.4.7 holdout

```
bool Forest::holdout [protected]
```

Definition at line 216 of file Forest.h.

6.43.4.8 importance_mode

```
ImportanceMode Forest::importance_mode [protected]
```

Definition at line 256 of file Forest.h.

6.43.4.9 is_ordered_variable

```
std::vector<bool> Forest::is_ordered_variable [protected]
```

Definition at line 224 of file Forest.h.

6.43.4.10 keep_inbag

```
bool Forest::keep_inbag [protected]
```

Definition at line 214 of file Forest.h.

6.43.4.11 memory_mode

```
MemoryMode Forest::memory_mode [protected]
```

Definition at line 209 of file Forest.h.

6.43.4.12 memory_saving_splitting

```
bool Forest::memory_saving_splitting [protected]
```

Definition at line 211 of file Forest.h.

6.43.4.13 min_node_size

```
uint Forest::min_node_size [protected]
```

Definition at line 202 of file Forest.h.

6.43.4.14 minprop

```
double Forest::minprop [protected]
```

Definition at line 221 of file Forest.h.

6.43.4.15 mtry

```
uint Forest::mtry [protected]
```

Definition at line 201 of file Forest.h.

6.43.4.16 mutex

```
std::mutex Forest::mutex [protected]
```

Definition at line 233 of file Forest.h.

6.43.4.17 no_split_variables

```
std::vector<size_t> Forest::no_split_variables [protected]
```

Definition at line 227 of file Forest.h.

6.43.4.18 num_independent_variables

```
size_t Forest::num_independent_variables [protected]
```

Definition at line 204 of file Forest.h.

6.43.4.19 num_samples

```
size_t Forest::num_samples [protected]
```

Definition at line 207 of file Forest.h.

6.43.4.20 num_threads

```
uint Forest::num_threads [protected]
```

Definition at line 230 of file Forest.h.

6.43.4.21 num_trees

```
size_t Forest::num_trees [protected]
```

Definition at line 200 of file Forest.h.

6.43.4.22 num_variables

```
size_t Forest::num_variables [protected]
```

Definition at line 203 of file Forest.h.

6.43.4.23 output_prefix

```
std::string Forest::output_prefix [protected]
```

Definition at line 255 of file Forest.h.

6.43.4.24 overall_prediction_error

```
double Forest::overall_prediction_error [protected]
```

Definition at line 241 of file Forest.h.

6.43.4.25 predict_all

```
bool Forest::predict_all [protected]
```

Definition at line 213 of file Forest.h.

6.43.4.26 prediction_mode

```
bool Forest::prediction_mode [protected]
```

Definition at line 208 of file Forest.h.

6.43.4.27 prediction_type

```
PredictionType Forest::prediction_type [protected]
```

Definition at line 217 of file Forest.h.

6.43.4.28 predictions

```
std::vector<std::vector<std::vector<double>>> Forest::predictions [protected]
```

Definition at line 240 of file Forest.h.

6.43.4.29 progress

```
size_t Forest::progress [protected]
```

Definition at line 262 of file Forest.h.

6.43.4.30 random_number_generator

```
std::mt19937_64 Forest::random_number_generator [protected]
```

Definition at line 253 of file Forest.h.

6.43.4.31 sample_fraction

```
double Forest::sample_fraction [protected]
```

Definition at line 215 of file Forest.h.

6.43.4.32 sample_with_replacement

```
bool Forest::sample_with_replacement [protected]
```

Definition at line 210 of file Forest.h.

6.43.4.33 seed

```
uint Forest::seed [protected]
```

Definition at line 205 of file Forest.h.

6.43.4.34 split_select_varIDs

```
std::vector<size_t> Forest::split_select_varIDs [protected]
```

Definition at line 246 of file Forest.h.

6.43.4.35 split_select_weights

```
std::vector<std::vector<double>> Forest::split_select_weights [protected]
```

Definition at line 247 of file Forest.h.

6.43.4.36 splitrule

```
SplitRule Forest::splitrule [protected]
```

Definition at line 212 of file Forest.h.

6.43.4.37 thread_ranges

```
std::vector<uint> Forest::thread_ranges [protected]
```

Definition at line 231 of file Forest.h.

6.43.4.38 trees

```
std::vector<Tree*> Forest::trees [protected]
```

Definition at line 237 of file Forest.h.

6.43.4.39 variable_importance

```
std::vector<double> Forest::variable_importance [protected]
```

Definition at line 259 of file Forest.h.

6.43.4.40 verbose_out

```
std::ostream* Forest::verbose_out [protected]
```

Definition at line 198 of file Forest.h.

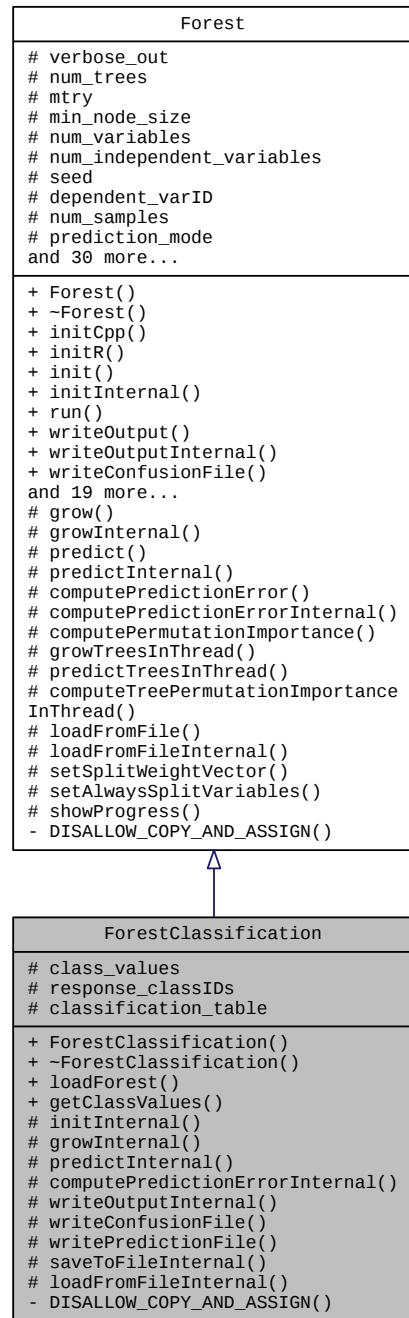
The documentation for this class was generated from the following files:

- src/[Forest.h](#)
- src/[Forest.cpp](#)

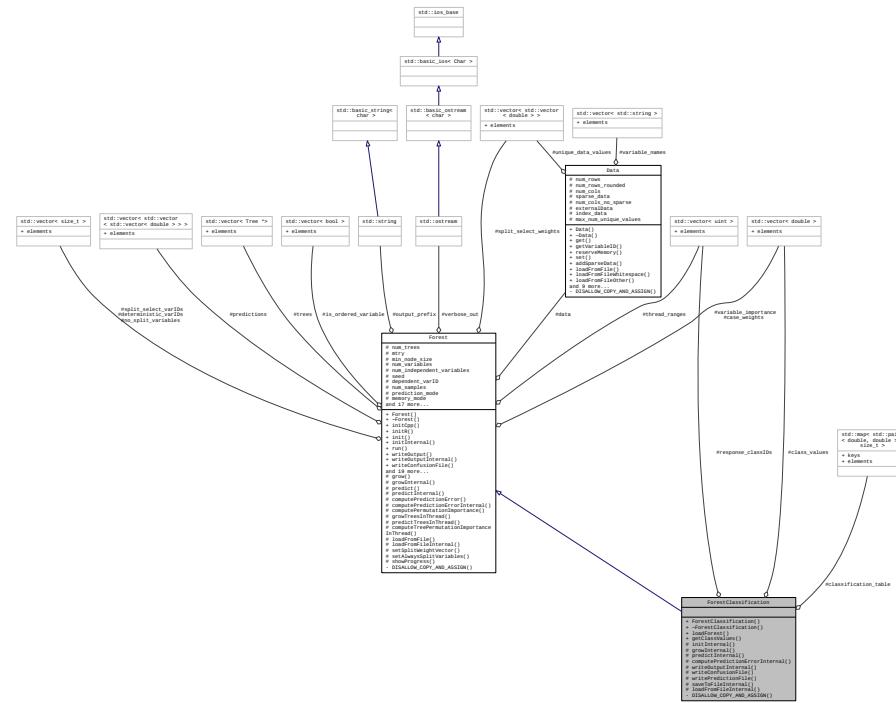
6.44 ForestClassification Class Reference

```
#include <ForestClassification.h>
```

Inheritance diagram for ForestClassification:



Collaboration diagram for ForestClassification:



Public Member Functions

- `ForestClassification ()`
 - `virtual ~ForestClassification ()`
 - `void loadForest (size_t dependent_varID, size_t num_trees, std::vector< std::vector< std::vector< std::vector< size_t >>>>`
`&forest_child_nodeIDs, std::vector< std::vector< size_t >> &forest_split_varIDs, std::vector< std::vector< double >> &forest_split_values, std::vector< double > &class_values, std::vector< bool > &is_ordered_variable)`
 - `const std::vector< double > & getClassValues () const`

Protected Member Functions

- void `initInternal` (std::string status_variable_name) override
 - void `growInternal` () override
 - void `predictInternal` () override
 - void `computePredictionErrorInternal` () override
 - void `writeOutputInternal` () override
 - void `writeConfusionFile` () override
 - void `writePredictionFile` () override
 - void `saveToFileInternal` (std::ofstream &outfile) override
 - void `loadFromFileInternal` (std::ifstream &infile) override

Protected Attributes

- std::vector< double > `class_values`
- std::vector< uint > `response_classIDs`
- std::map< std::pair< double, double >, size_t > `classification_table`

Private Member Functions

- `DISALLOW_COPY_AND_ASSIGN (ForestClassification)`

6.44.1 Detailed Description

Definition at line 40 of file `ForestClassification.h`.

6.44.2 Constructor & Destructor Documentation

6.44.2.1 `ForestClassification()`

```
ForestClassification::ForestClassification ( )
```

Definition at line 42 of file `ForestClassification.cpp`.

6.44.2.2 `~ForestClassification()`

```
ForestClassification::~ForestClassification ( ) [virtual]
```

Definition at line 45 of file `ForestClassification.cpp`.

6.44.3 Member Function Documentation

6.44.3.1 `computePredictionErrorInternal()`

```
void ForestClassification::computePredictionErrorInternal ( ) [override], [protected], [virtual]
```

Implements [Forest](#).

Definition at line 148 of file `ForestClassification.cpp`.

6.44.3.2 `DISALLOW_COPY_AND_ASSIGN()`

```
ForestClassification::DISALLOW_COPY_AND_ASSIGN (
    ForestClassification ) [private]
```

6.44.3.3 `getClassValues()`

```
const std::vector<double>& ForestClassification::getClassValues ( ) const [inline]
```

Definition at line 50 of file ForestClassification.h.

6.44.3.4 `growInternal()`

```
void ForestClassification::growInternal ( ) [override], [protected], [virtual]
```

Implements [Forest](#).

Definition at line 103 of file ForestClassification.cpp.

6.44.3.5 `initInternal()`

```
void ForestClassification::initInternal (
    std::string status_variable_name ) [override], [protected], [virtual]
```

Implements [Forest](#).

Definition at line 70 of file ForestClassification.cpp.

6.44.3.6 `loadForest()`

```
void ForestClassification::loadForest (
    size_t dependent_varID,
    size_t num_trees,
    std::vector< std::vector< std::vector< size_t >> > & forest_child_nodeIDs,
    std::vector< std::vector< size_t >> & forest_split_varIDs,
    std::vector< std::vector< double >> & forest_split_values,
    std::vector< double > & class_values,
    std::vector< bool > & is_ordered_variable )
```

Definition at line 48 of file ForestClassification.cpp.

6.44.3.7 loadFromFileInternal()

```
void ForestClassification::loadFromFileInternal (
    std::ifstream & infile) [override], [protected], [virtual]
```

Implements [Forest](#).

Definition at line 288 of file ForestClassification.cpp.

6.44.3.8 predictInternal()

```
void ForestClassification::predictInternal () [override], [protected], [virtual]
```

Implements [Forest](#).

Definition at line 110 of file ForestClassification.cpp.

6.44.3.9 saveToFileInternal()

```
void ForestClassification::saveToFileInternal (
    std::ofstream & outfile) [override], [protected], [virtual]
```

Implements [Forest](#).

Definition at line 275 of file ForestClassification.cpp.

6.44.3.10 writeConfusionFile()

```
void ForestClassification::writeConfusionFile () [override], [protected], [virtual]
```

Implements [Forest](#).

Definition at line 197 of file ForestClassification.cpp.

6.44.3.11 writeOutputInternal()

```
void ForestClassification::writeOutputInternal () [override], [protected], [virtual]
```

Implements [Forest](#).

Definition at line 193 of file ForestClassification.cpp.

6.44.3.12 writePredictionFile()

```
void ForestClassification::writePredictionFile ( ) [override], [protected], [virtual]
```

Implements [Forest](#).

Definition at line 240 of file [ForestClassification.cpp](#).

6.44.4 Member Data Documentation

6.44.4.1 class_values

```
std::vector<double> ForestClassification::class_values [protected]
```

Definition at line 66 of file [ForestClassification.h](#).

6.44.4.2 classification_table

```
std::map<std::pair<double, double>, size_t> ForestClassification::classification_table [protected]
```

Definition at line 70 of file [ForestClassification.h](#).

6.44.4.3 response_classIDs

```
std::vector<uint> ForestClassification::response_classIDs [protected]
```

Definition at line 67 of file [ForestClassification.h](#).

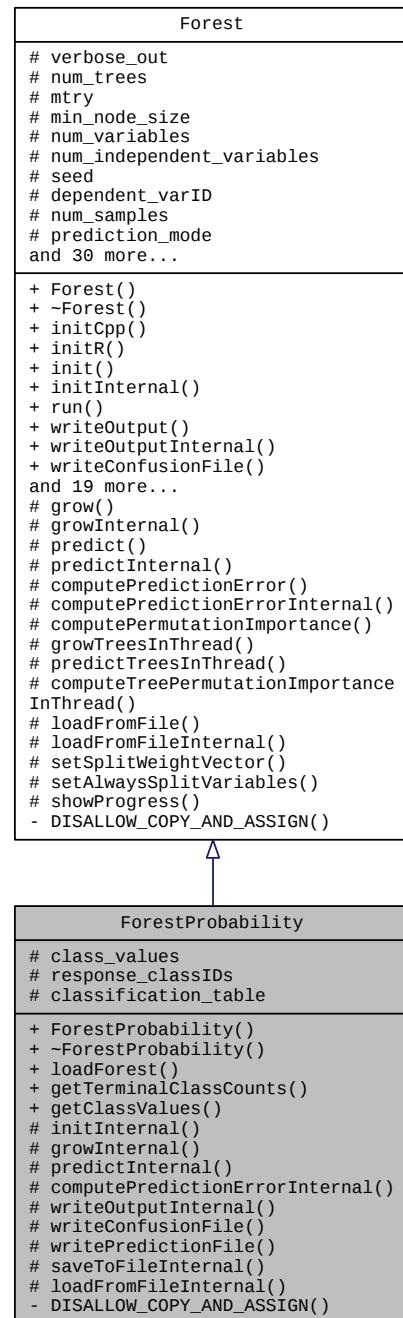
The documentation for this class was generated from the following files:

- src/[ForestClassification.h](#)
- src/[ForestClassification.cpp](#)

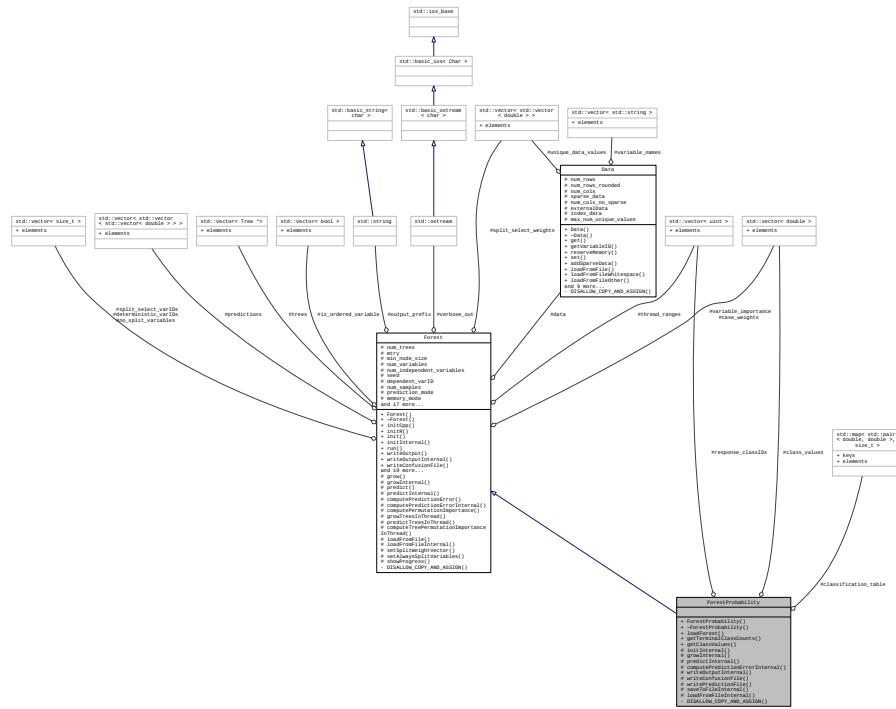
6.45 ForestProbability Class Reference

```
#include <ForestProbability.h>
```

Inheritance diagram for ForestProbability:



Collaboration diagram for ForestProbability:



Public Member Functions

- `ForestProbability ()`
- virtual `~ForestProbability ()`
- void `loadForest (size_t dependent_varID, size_t num_trees, std::vector< std::vector< std::vector< size_t >> > &forest_child_nodeIDs, std::vector< std::vector< size_t >> &forest_split_varIDs, std::vector< std::vector< double >> &forest_split_values, std::vector< double > &class_values, std::vector< std::vector< std::vector< double >>> &forest_terminal_class_counts, std::vector< bool > &is_ordered_variable)`
- `std::vector< std::vector< std::vector< double >>> getTerminalClassCounts ()`
- const `std::vector< double > & getClassValues () const`

Protected Member Functions

- void `initInternal (std::string status_variable_name)`
- void `growInternal ()`
- void `predictInternal ()`
- void `computePredictionErrorInternal ()`
- void `writeOutputInternal ()`
- void `writeConfusionFile ()`
- void `writePredictionFile ()`
- void `saveToFileInternal (std::ofstream &outfile)`
- void `loadFromFileInternal (std::ifstream &infile)`

Protected Attributes

- std::vector< double > [class_values](#)
- std::vector< uint > [response_classIDs](#)
- std::map< std::pair< double, double >, size_t > [classification_table](#)

Private Member Functions

- [DISALLOW_COPY_AND_ASSIGN \(ForestProbability\)](#)

6.45.1 Detailed Description

Definition at line 40 of file ForestProbability.h.

6.45.2 Constructor & Destructor Documentation

6.45.2.1 ForestProbability()

```
ForestProbability::ForestProbability ( )
```

Definition at line 36 of file ForestProbability.cpp.

6.45.2.2 ~ForestProbability()

```
ForestProbability::~ForestProbability ( ) [virtual]
```

Definition at line 39 of file ForestProbability.cpp.

6.45.3 Member Function Documentation

6.45.3.1 computePredictionErrorInternal()

```
void ForestProbability::computePredictionErrorInternal ( ) [protected], [virtual]
```

Implements [Forest](#).

Definition at line 142 of file ForestProbability.cpp.

6.45.3.2 `DISALLOW_COPY_AND_ASSIGN()`

```
ForestProbability::DISALLOW_COPY_AND_ASSIGN (
    ForestProbability ) [private]
```

6.45.3.3 `getClassValues()`

```
const std::vector<double>& ForestProbability::getClassValues ( ) const [inline]
```

Definition at line 60 of file `ForestProbability.h`.

6.45.3.4 `getTerminalClassCounts()`

```
std::vector<std::vector<std::vector<double> > > ForestProbability::getTerminalClassCounts ( )
[inline]
```

Definition at line 50 of file `ForestProbability.h`.

6.45.3.5 `growInternal()`

```
void ForestProbability::growInternal ( ) [protected], [virtual]
```

Implements [Forest](#).

Definition at line 97 of file `ForestProbability.cpp`.

6.45.3.6 `initInternal()`

```
void ForestProbability::initInternal (
    std::string status_variable_name ) [protected], [virtual]
```

Implements [Forest](#).

Definition at line 64 of file `ForestProbability.cpp`.

6.45.3.7 loadForest()

```
void ForestProbability::loadForest (
    size_t dependent_varID,
    size_t num_trees,
    std::vector< std::vector< std::vector< size_t >> > & forest_child_nodeIDs,
    std::vector< std::vector< size_t >> & forest_split_varIDs,
    std::vector< std::vector< double >> & forest_split_values,
    std::vector< double > & class_values,
    std::vector< std::vector< std::vector< double >>> & forest_terminal_class_counts,
    std::vector< bool > & is_ordered_variable )
```

Definition at line 42 of file ForestProbability.cpp.

6.45.3.8 loadFromFileInternal()

```
void ForestProbability::loadFromFileInternal (
    std::ifstream & infile ) [protected], [virtual]
```

Implements [Forest](#).

Definition at line 258 of file ForestProbability.cpp.

6.45.3.9 predictInternal()

```
void ForestProbability::predictInternal ( ) [protected], [virtual]
```

Implements [Forest](#).

Definition at line 104 of file ForestProbability.cpp.

6.45.3.10 saveToFileInternal()

```
void ForestProbability::saveToFileInternal (
    std::ofstream & outfile ) [protected], [virtual]
```

Implements [Forest](#).

Definition at line 245 of file ForestProbability.cpp.

6.45.3.11 writeConfusionFile()

```
void ForestProbability::writeConfusionFile ( ) [protected], [virtual]
```

Implements [Forest](#).

Definition at line 186 of file ForestProbability.cpp.

6.45.3.12 writeOutputInternal()

```
void ForestProbability::writeOutputInternal ( ) [protected], [virtual]
```

Implements [Forest](#).

Definition at line 182 of file ForestProbability.cpp.

6.45.3.13 writePredictionFile()

```
void ForestProbability::writePredictionFile ( ) [protected], [virtual]
```

Implements [Forest](#).

Definition at line 203 of file ForestProbability.cpp.

6.45.4 Member Data Documentation

6.45.4.1 class_values

```
std::vector<double> ForestProbability::class_values [protected]
```

Definition at line 76 of file ForestProbability.h.

6.45.4.2 classification_table

```
std::map<std::pair<double, double>, size_t> ForestProbability::classification_table [protected]
```

Definition at line 80 of file ForestProbability.h.

6.45.4.3 response_classIDs

```
std::vector<uint> ForestProbability::response_classIDs [protected]
```

Definition at line 77 of file ForestProbability.h.

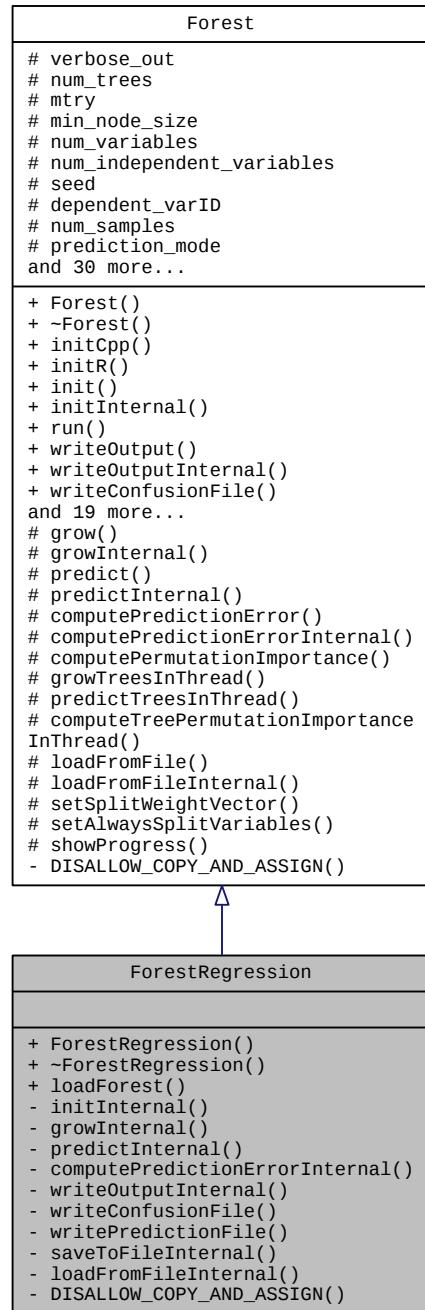
The documentation for this class was generated from the following files:

- src/[ForestProbability.h](#)
- src/[ForestProbability.cpp](#)

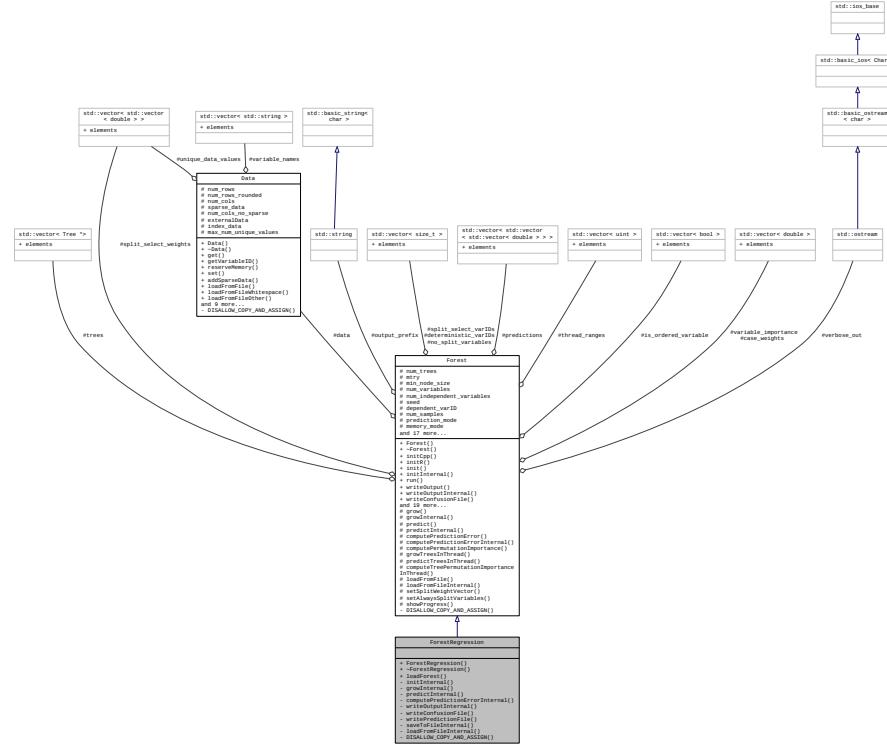
6.46 ForestRegression Class Reference

```
#include <ForestRegression.h>
```

Inheritance diagram for ForestRegression:



Collaboration diagram for ForestRegression:



Public Member Functions

- `ForestRegression ()`
 - `virtual ~ForestRegression ()`
 - `void loadForest (size_t dependent_varID, size_t num_trees, std::vector< std::vector< std::vector< size_t >> > &forest_child_nodeIDs, std::vector< std::vector< size_t >> &forest_split_varIDs, std::vector< std::vector< double >> &forest_split_values, std::vector< bool > &is_ordered_variable)`

Private Member Functions

- void `initInternal` (std::string status_variable_name)
 - void `growInternal` ()
 - void `predictInternal` ()
 - void `computePredictionErrorInternal` ()
 - void `writeOutputInternal` ()
 - void `writeConfusionFile` ()
 - void `writePredictionFile` ()
 - void `saveToFileInternal` (std::ofstream &outfile)
 - void `loadFromFileInternal` (std::ifstream &infile)
 - `DISALLOW_COPY_AND_ASSIGN` (ForestRegression)

Additional Inherited Members

6.46.1 Detailed Description

Definition at line 38 of file ForestRegression.h.

6.46.2 Constructor & Destructor Documentation

6.46.2.1 ForestRegression()

```
ForestRegression::ForestRegression ( )
```

Definition at line 38 of file ForestRegression.cpp.

6.46.2.2 ~ForestRegression()

```
ForestRegression::~ForestRegression ( ) [virtual]
```

Definition at line 41 of file ForestRegression.cpp.

6.46.3 Member Function Documentation

6.46.3.1 computePredictionErrorInternal()

```
void ForestRegression::computePredictionErrorInternal ( ) [private], [virtual]
```

Implements [Forest](#).

Definition at line 123 of file ForestRegression.cpp.

6.46.3.2 DISALLOW_COPY_AND_ASSIGN()

```
ForestRegression::DISALLOW_COPY_AND_ASSIGN ( 
    ForestRegression ) [private]
```

6.46.3.3 growInternal()

```
void ForestRegression::growInternal ( ) [private], [virtual]
```

Implements [Forest](#).

Definition at line 84 of file ForestRegression.cpp.

6.46.3.4 initInternal()

```
void ForestRegression::initInternal (
    std::string status_variable_name ) [private], [virtual]
```

Implements [Forest](#).

Definition at line 65 of file ForestRegression.cpp.

6.46.3.5 loadForest()

```
void ForestRegression::loadForest (
    size_t dependent_varID,
    size_t num_trees,
    std::vector< std::vector< std::vector< size_t >> > & forest_child_nodeIDs,
    std::vector< std::vector< size_t >> & forest_split_varIDs,
    std::vector< std::vector< double >> & forest_split_values,
    std::vector< bool > & is_ordered_variable )
```

Definition at line 44 of file ForestRegression.cpp.

6.46.3.6 loadFromFileInternal()

```
void ForestRegression::loadFromFileInternal (
    std::ifstream & infile ) [private], [virtual]
```

Implements [Forest](#).

Definition at line 222 of file ForestRegression.cpp.

6.46.3.7 predictInternal()

```
void ForestRegression::predictInternal ( ) [private], [virtual]
```

Implements [Forest](#).

Definition at line 91 of file ForestRegression.cpp.

6.46.3.8 saveToFileInternal()

```
void ForestRegression::saveToFileInternal (
    std::ofstream & outfile ) [private], [virtual]
```

Implements [Forest](#).

Definition at line 212 of file ForestRegression.cpp.

6.46.3.9 writeConfusionFile()

```
void ForestRegression::writeConfusionFile ( ) [private], [virtual]
```

Implements [Forest](#).

Definition at line 160 of file ForestRegression.cpp.

6.46.3.10 writeOutputInternal()

```
void ForestRegression::writeOutputInternal ( ) [private], [virtual]
```

Implements [Forest](#).

Definition at line 156 of file ForestRegression.cpp.

6.46.3.11 writePredictionFile()

```
void ForestRegression::writePredictionFile ( ) [private], [virtual]
```

Implements [Forest](#).

Definition at line 177 of file [ForestRegression.cpp](#).

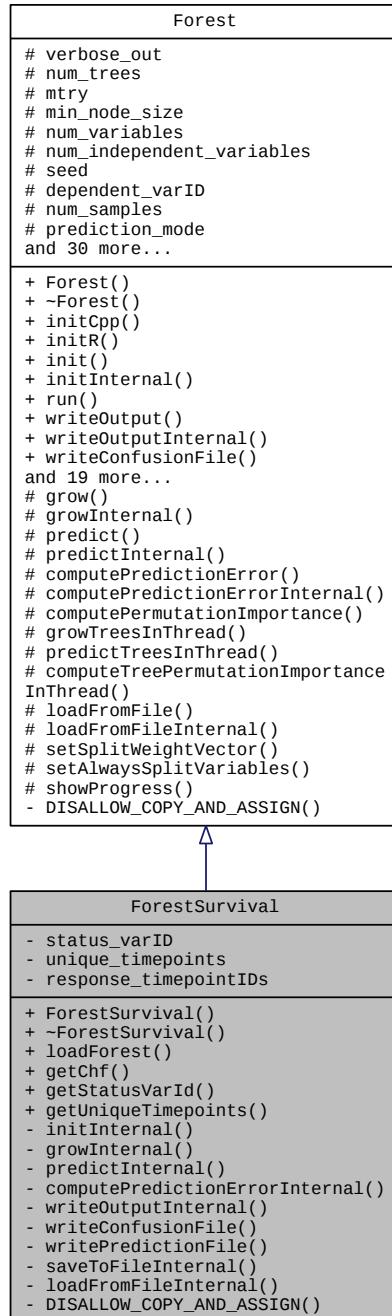
The documentation for this class was generated from the following files:

- src/[ForestRegression.h](#)
- src/[ForestRegression.cpp](#)

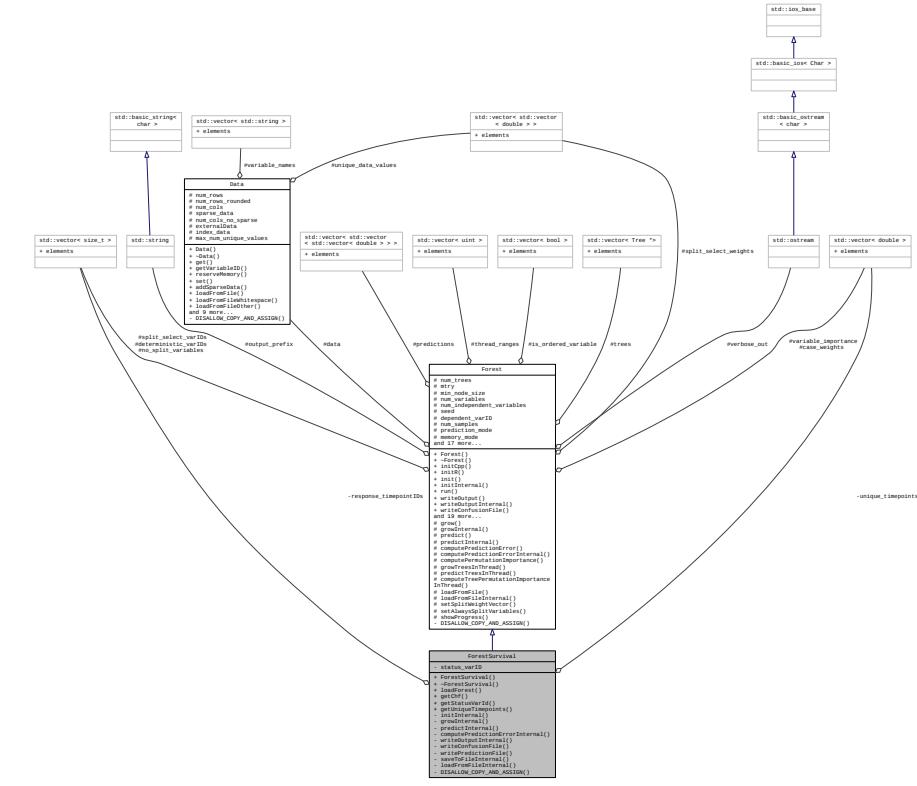
6.47 ForestSurvival Class Reference

```
#include <ForestSurvival.h>
```

Inheritance diagram for ForestSurvival:



Collaboration diagram for ForestSurvival:



Public Member Functions

- `ForestSurvival ()`
 - `virtual ~ForestSurvival ()`
 - `void loadForest (size_t dependent_varID, size_t num_trees, std::vector< std::vector< std::vector< std::vector< size_t >>> &forest_child_nodeIDs, std::vector< std::vector< size_t >> &forest_split_varIDs, std::vector< std::vector< double >> &forest_split_values, size_t status_varID, std::vector< std::vector< std::vector< double >>> &forest_chf, std::vector< double > &unique_timepoints, std::vector< bool > &is_ordered_variable)`
 - `std::vector< std::vector< std::vector< double >>> getChf ()`
 - `size_t getStatusVarID () const`
 - `const std::vector< double > & getUniqueTimepoints () const`

Private Member Functions

- void `initInternal` (std::string status_variable_name)
 - void `growInternal` ()
 - void `predictInternal` ()
 - void `computePredictionErrorInternal` ()
 - void `writeOutputInternal` ()
 - void `writeConfusionFile` ()
 - void `writePredictionFile` ()
 - void `saveToFileInternal` (std::ofstream &outfile)
 - void `loadFromFileInternal` (std::ifstream &infile)
 - **DISALLOW_COPY_AND_ASSIGN** (`ForestSurvival`)

Private Attributes

- size_t `status_varID`
- std::vector< double > `unique_timepoints`
- std::vector< size_t > `response_timepointIDs`

Additional Inherited Members

6.47.1 Detailed Description

Definition at line 39 of file ForestSurvival.h.

6.47.2 Constructor & Destructor Documentation

6.47.2.1 ForestSurvival()

```
ForestSurvival::ForestSurvival ( )
```

Definition at line 39 of file ForestSurvival.cpp.

6.47.2.2 ~ForestSurvival()

```
ForestSurvival::~ForestSurvival ( ) [virtual]
```

Definition at line 43 of file ForestSurvival.cpp.

6.47.3 Member Function Documentation

6.47.3.1 computePredictionErrorInternal()

```
void ForestSurvival::computePredictionErrorInternal ( ) [private], [virtual]
```

Implements [Forest](#).

Definition at line 156 of file ForestSurvival.cpp.

6.47.3.2 DISALLOW_COPY_AND_ASSIGN()

```
ForestSurvival::DISALLOW_COPY_AND_ASSIGN (   
    ForestSurvival ) [private]
```

6.47.3.3 getChf()

```
std::vector<std::vector<std::vector<double> > > ForestSurvival::getChf ( ) [inline]
```

Definition at line 50 of file ForestSurvival.h.

6.47.3.4 getStatusVarId()

```
size_t ForestSurvival::getStatusVarId ( ) const [inline]
```

Definition at line 59 of file ForestSurvival.h.

6.47.3.5 getUniqueTimepoints()

```
const std::vector<double>& ForestSurvival::getUniqueTimepoints ( ) const [inline]
```

Definition at line 62 of file ForestSurvival.h.

6.47.3.6 growInternal()

```
void ForestSurvival::growInternal ( ) [private], [virtual]
```

Implements [Forest](#).

Definition at line 112 of file ForestSurvival.cpp.

6.47.3.7 initInternal()

```
void ForestSurvival::initInternal (
    std::string status_variable_name ) [private], [virtual]
```

Implements [Forest](#).

Definition at line 70 of file ForestSurvival.cpp.

6.47.3.8 loadForest()

```
void ForestSurvival::loadForest (
    size_t dependent_varID,
    size_t num_trees,
    std::vector< std::vector< std::vector< size_t >> > & forest_child_nodeIDs,
    std::vector< std::vector< size_t >> & forest_split_varIDs,
    std::vector< std::vector< double >> & forest_split_values,
    size_t status_varID,
    std::vector< std::vector< std::vector< double >> > & forest_chf,
    std::vector< double > & unique_timepoints,
    std::vector< bool > & is_ordered_variable )
```

Definition at line 46 of file ForestSurvival.cpp.

6.47.3.9 loadFromFileInternal()

```
void ForestSurvival::loadFromFileInternal (
    std::ifstream & infile ) [private], [virtual]
```

Implements [Forest](#).

Definition at line 279 of file ForestSurvival.cpp.

6.47.3.10 predictInternal()

```
void ForestSurvival::predictInternal ( ) [private], [virtual]
```

Implements [Forest](#).

Definition at line 119 of file ForestSurvival.cpp.

6.47.3.11 saveToFileInternal()

```
void ForestSurvival::saveToFileInternal (
    std::ofstream & outfile ) [private], [virtual]
```

Implements [Forest](#).

Definition at line 263 of file ForestSurvival.cpp.

6.47.3.12 writeConfusionFile()

```
void ForestSurvival::writeConfusionFile () [private], [virtual]
```

Implements [Forest](#).

Definition at line 202 of file ForestSurvival.cpp.

6.47.3.13 writeOutputInternal()

```
void ForestSurvival::writeOutputInternal () [private], [virtual]
```

Implements [Forest](#).

Definition at line 196 of file ForestSurvival.cpp.

6.47.3.14 writePredictionFile()

```
void ForestSurvival::writePredictionFile () [private], [virtual]
```

Implements [Forest](#).

Definition at line 220 of file ForestSurvival.cpp.

6.47.4 Member Data Documentation

6.47.4.1 response_timepointIDs

```
std::vector<size_t> ForestSurvival::response_timepointIDs [private]
```

Definition at line 79 of file ForestSurvival.h.

6.47.4.2 status_varID

```
size_t ForestSurvival::status_varID [private]
```

Definition at line 77 of file ForestSurvival.h.

6.47.4.3 unique_timepoints

```
std::vector<double> ForestSurvival::unique_timepoints [private]
```

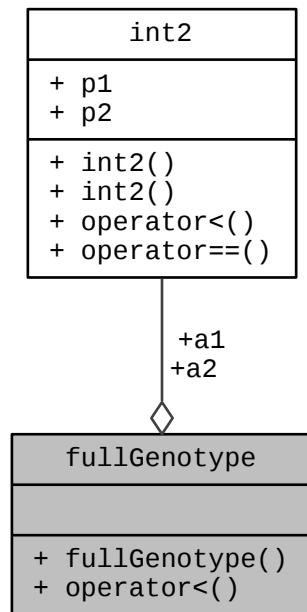
Definition at line 78 of file ForestSurvival.h.

The documentation for this class was generated from the following files:

- src/[ForestSurvival.h](#)
- src/[ForestSurvival.cpp](#)

6.48 fullGenotype Class Reference

Collaboration diagram for fullGenotype:



Public Member Functions

- `fullGenotype (int2 a, int2 b)`
- `bool operator< (const fullGenotype &b) const`

Public Attributes

- `int2 a1`
- `int2 a2`

6.48.1 Detailed Description

Definition at line 36 of file gvar.cpp.

6.48.2 Constructor & Destructor Documentation

6.48.2.1 fullGenotype()

```
fullGenotype::fullGenotype (
    int2 a,
    int2 b ) [inline]
```

Definition at line 38 of file gvar.cpp.

6.48.3 Member Function Documentation

6.48.3.1 operator<()

```
bool fullGenotype::operator< (
    const fullGenotype & b ) const [inline]
```

Definition at line 52 of file gvar.cpp.

6.48.4 Member Data Documentation

6.48.4.1 a1

```
int2 fullGenotype::a1
```

Definition at line 49 of file gvar.cpp.

6.48.4.2 a2

```
int2 fullGenotype::a2
```

Definition at line 50 of file gvar.cpp.

The documentation for this class was generated from the following file:

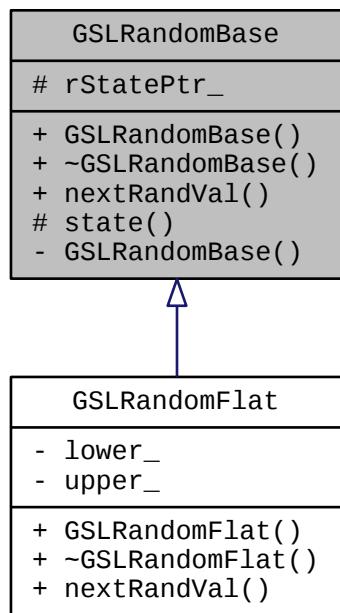
- [src/gvar.cpp](#)

6.49 GSLRandomBase Class Reference

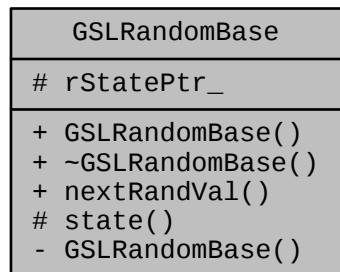
A base class for GNU Scientific Library (GSL) random number functions.

```
#include <GSLRandomBase.h>
```

Inheritance diagram for GSLRandomBase:



Collaboration diagram for GSLRandomBase:



Public Member Functions

- `GSLRandomBase (int seedVal)`
- virtual `~GSLRandomBase ()`
- virtual double `nextRandVal ()=0`

Protected Member Functions

- `gsl_rng * state ()`

Protected Attributes

- `gsl_rng * rStatePtr_`

Private Member Functions

- `GSLRandomBase (const GSLRandomBase &rhs)`

6.49.1 Detailed Description

A base class for GNU Scientific Library (GSL) random number functions.

The setup, initialization and clean-up is the same for all GSL random number functions. This class abstracts away these details, placing the setup and initialization in the class constructor and the clean-up in the class destructor. The class constructor is passed a seed value for the random number generator.

A class that provides access to one or more GSL random number functions should be derived from this class. This class must provide an implementation for the `nextRandVal()` pure virtual function. The `nextRandVal` will call the specific random number function (for example `gsl_ran_ugaussian()` for Gaussian distribution or `gsl_ran_flat()` for a flat random number distribution).

This class uses the default random number generator. At least on Windows XP using the Visual C++ 6.0 compiler the type definitions for the random functions (for example `gsl_rng_mt19937` or `gsl_rng_knuthran`) would not link properly. Perhaps they are not properly exported from the pre-built library.

I decided to use the GSL because it is supported on all major platforms (UNIX, Linux and Windows) and provides high quality pseudo-random number generation support. The standard POSIX `rand()` function is notorious for its poor quality. While the `random()` function on UNIX provides better pseudo-random number quality, but is still not as good as functions like MT19937.

Definition at line 39 of file `GSLRandomBase.h`.

6.49.2 Constructor & Destructor Documentation

6.49.2.1 `GSLRandomBase()` [1/2]

```
GSLRandomBase::GSLRandomBase (
    const GSLRandomBase & rhs) [private]
```

6.49.2.2 `GSLRandomBase()` [2/2]

```
GSLRandomBase::GSLRandomBase (
    int seedVal) [inline]
```

Definition at line 52 of file `GSLRandomBase.h`.

6.49.2.3 `~GSLRandomBase()`

```
virtual GSLRandomBase::~GSLRandomBase () [inline], [virtual]
```

Definition at line 67 of file `GSLRandomBase.h`.

6.49.3 Member Function Documentation

6.49.3.1 `nextRandVal()`

```
virtual double GSLRandomBase::nextRandVal () [pure virtual]
```

Implemented in [GSLRandomFlat](#).

6.49.3.2 `state()`

```
gsl_rng* GSLRandomBase::state () [inline], [protected]
```

Definition at line 45 of file `GSLRandomBase.h`.

6.49.4 Member Data Documentation

6.49.4.1 rStatePtr_

```
gsl_rng* GSLRandomBase::rStatePtr_ [protected]
```

Definition at line 48 of file `GSLRandomBase.h`.

The documentation for this class was generated from the following file:

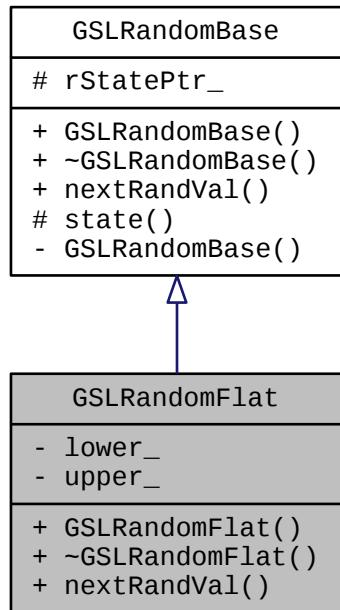
- `src/GSLRandomBase.h`

6.50 GSLRandomFlat Class Reference

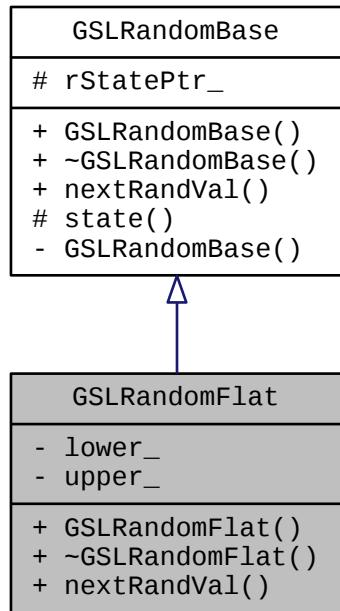
Random numbers in a flat, or uniform distribution.

```
#include <GSLRandomFlat.h>
```

Inheritance diagram for `GSLRandomFlat`:



Collaboration diagram for GSLRandomFlat:



Public Member Functions

- [GSLRandomFlat](#) (int seedVal, double lower, double upper)
- [~GSLRandomFlat \(\)](#)
- double [nextRandVal \(\)](#)

Private Attributes

- double [lower_](#)
- double [upper_](#)

Additional Inherited Members

6.50.1 Detailed Description

Random numbers in a flat, or uniform distribution.

The class constructor is given a seed and a lower and upper bound value for the uniform distribution. The random numbers that result will be a uniform distribution in the range

```
lower <= randVal < upper
```

Definition at line 22 of file [GSLRandomFlat.h](#).

6.50.2 Constructor & Destructor Documentation

6.50.2.1 `GSLRandomFlat()`

```
GSLRandomFlat::GSLRandomFlat (
    int seedVal,
    double lower,
    double upper ) [inline]
```

Definition at line 28 of file `GSLRandomFlat.h`.

6.50.2.2 `~GSLRandomFlat()`

```
GSLRandomFlat::~GSLRandomFlat () [inline]
```

Definition at line 37 of file `GSLRandomFlat.h`.

6.50.3 Member Function Documentation

6.50.3.1 `nextRandVal()`

```
double GSLRandomFlat::nextRandVal () [inline], [virtual]
```

Implements [GSLRandomBase](#).

Definition at line 41 of file `GSLRandomFlat.h`.

6.50.4 Member Data Documentation

6.50.4.1 `lower_`

```
double GSLRandomFlat::lower_ [private]
```

Definition at line 24 of file `GSLRandomFlat.h`.

6.50.4.2 upper_

```
double GSLRandomFlat::upper_ [private]
```

Definition at line 24 of file `GSLRandomFlat.h`.

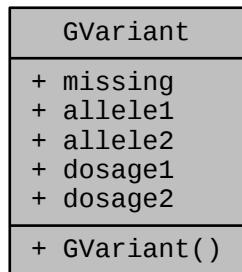
The documentation for this class was generated from the following file:

- `src/GSLRandomFlat.h`

6.51 GVariant Class Reference

```
#include <gvar.h>
```

Collaboration diagram for GVariant:



Public Member Functions

- `GVariant()`

Public Attributes

- bool `missing`
- int `allele1`
- int `allele2`
- float `dosage1`
- float `dosage2`

6.51.1 Detailed Description

Definition at line 44 of file gvar.h.

6.51.2 Constructor & Destructor Documentation

6.51.2.1 GVariant()

```
GVariant::GVariant () [inline]
```

Definition at line 47 of file gvar.h.

6.51.3 Member Data Documentation

6.51.3.1 allele1

```
int GVariant::allele1
```

Definition at line 54 of file gvar.h.

6.51.3.2 allele2

```
int GVariant::allele2
```

Definition at line 55 of file gvar.h.

6.51.3.3 dosage1

```
float GVariant::dosage1
```

Definition at line 56 of file gvar.h.

6.51.3.4 dosage2

```
float GVariant::dosage2
```

Definition at line 57 of file gvar.h.

6.51.3.5 missing

```
bool GVariant::missing
```

Definition at line 53 of file gvar.h.

The documentation for this class was generated from the following file:

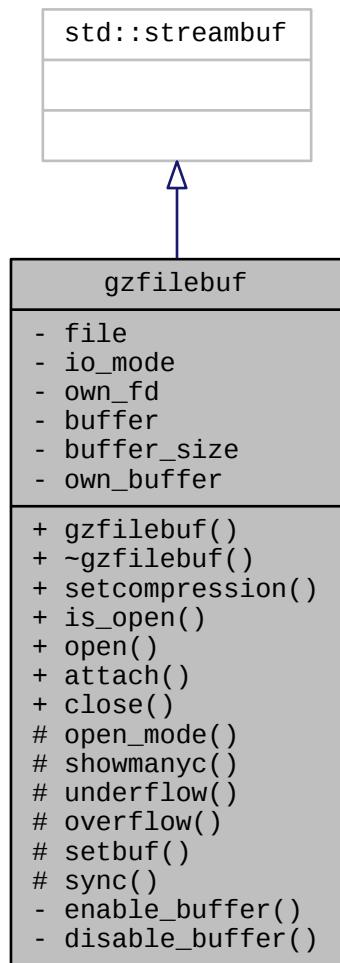
- src/[gvar.h](#)

6.52 **gzfilebuf** Class Reference

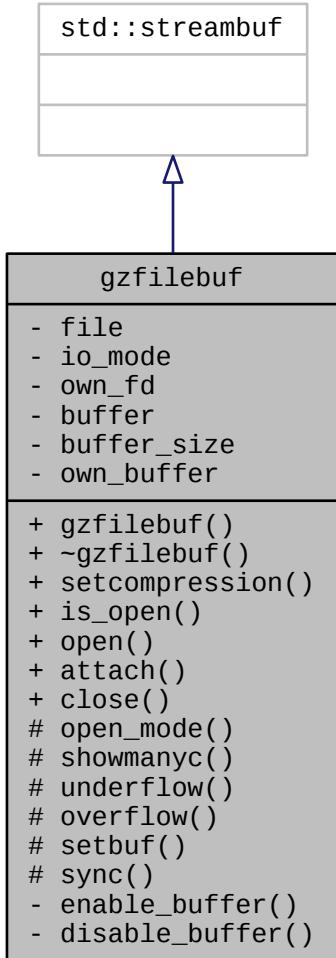
Gzipped file stream buffer class.

```
#include <zfstream.h>
```

Inheritance diagram for gzfilebuf:



Collaboration diagram for gzfilebuf:



Public Member Functions

- [`gzfilebuf \(\)`](#)
- virtual [`~gzfilebuf \(\)`](#)
- int [`setcompression`](#) (int comp_level, int comp_strategy=Z_DEFAULT_STRATEGY)
Set compression level and strategy on the fly.
- bool [`is_open \(\) const`](#)
Check if file is open.
- [`gzfilebuf * open`](#) (const char *name, std::ios_base::openmode mode)
Open gzipped file.
- [`gzfilebuf * attach`](#) (int fd, std::ios_base::openmode mode)

Attach to already open gzipped file.

- `gzfilebuf * close ()`

Close gzipped file.

Protected Member Functions

- `bool open_mode (std::ios_base::openmode mode, char *c_mode) const`
Convert ios open mode int to mode string used by zlib.
- `virtual std::streamsize showmany ()`
Number of characters available in stream buffer.
- `virtual int_type underflow ()`
Fill get area from gzipped file.
- `virtual int_type overflow (int_type c=traits_type::eof())`
Write put area to gzipped file.
- `virtual std::streambuf * setbuf (char_type *p, std::streamsize n)`
Installs external stream buffer.
- `virtual int sync ()`
Flush stream buffer to file.

Private Member Functions

- `void enable_buffer ()`
Allocate internal buffer.
- `void disable_buffer ()`
Destroy internal buffer.

Private Attributes

- `gzFile file`
Underlying file pointer.
- `std::ios_base::openmode io_mode`
Mode in which file was opened.
- `bool own_fd`
True if this object owns file descriptor.
- `char_type * buffer`
Stream buffer.
- `std::streamsize buffer_size`
Stream buffer size.
- `bool own_buffer`
True if this object owns stream buffer.

6.52.1 Detailed Description

Gzipped file stream buffer class.

This class implements basic_filebuf for gzipped files. It doesn't yet support seeking (allowed by zlib but slow/limited), putback and read/write access (tricky). Otherwise, it attempts to be a drop-in replacement for the standard file streambuf.

Definition at line 27 of file zfstream.h.

6.52.2 Constructor & Destructor Documentation

6.52.2.1 `gzfilebuf()`

```
gzfilebuf::gzfilebuf ( )
```

Definition at line 21 of file zfstream.cpp.

6.52.2.2 `~gzfilebuf()`

```
gzfilebuf::~gzfilebuf ( ) [virtual]
```

Definition at line 30 of file zfstream.cpp.

6.52.3 Member Function Documentation

6.52.3.1 `attach()`

```
gzfilebuf * gzfilebuf::attach (
    int fd,
    std::ios_base::openmode mode )
```

Attach to already open gzipped file.

Parameters

<code>fd</code>	File descriptor.
<code>mode</code>	Open mode flags.

Returns

this on success, NULL on failure.

Definition at line 79 of file zfstream.cpp.

6.52.3.2 close()

```
gzfilebuf * gzfilebuf::close ( )
```

Close gzipped file.

Returns

this on success, NULL on failure.

Definition at line 107 of file zfstream.cpp.

6.52.3.3 disable_buffer()

```
void gzfilebuf::disable_buffer ( ) [private]
```

Destroy internal buffer.

This function is safe to call multiple times. It will ensure that the internal buffer is deallocated if it exists. In any case, it will also reset the buffer pointers.

Definition at line 347 of file zfstream.cpp.

6.52.3.4 enable_buffer()

```
void gzfilebuf::enable_buffer ( ) [private]
```

Allocate internal buffer.

This function is safe to call multiple times. It will ensure that a proper internal buffer exists if it is required. If the buffer already exists or is external, the buffer pointers will be reset to their original state.

Definition at line 308 of file zfstream.cpp.

6.52.3.5 **is_open()**

```
bool gzfilebuf::is_open ( ) const [inline]
```

Check if file is open.

Returns

True if file is open.

Definition at line 57 of file `zfstream.h`.

6.52.3.6 **open()**

```
gzfilebuf * gzfilebuf::open (
    const char * name,
    std::ios_base::openmode mode )
```

Open gzipped file.

Parameters

<i>name</i>	File name.
<i>mode</i>	Open mode flags.

Returns

`this` on success, `NULL` on failure.

Definition at line 51 of file `zfstream.cpp`.

6.52.3.7 **open_mode()**

```
bool gzfilebuf::open_mode (
    std::ios_base::openmode mode,
    char * c_mode ) const [protected]
```

Convert `ios` open mode int to mode string used by zlib.

Returns

True if valid mode flag combination.

Definition at line 131 of file `zfstream.cpp`.

6.52.3.8 overflow()

```
gzfilebuf::int_type gzfilebuf::overflow (
    int_type c = traits_type::eof() ) [protected], [virtual]
```

Write put area to gzipped file.

Parameters

<i>c</i>	Extra character to add to buffer contents.
----------	--

Returns

Non-EOF on success, EOF on error.

This actually writes characters in stream buffer to gzipped file. With unbuffered output this is done one character at a time.

Definition at line 214 of file zfstream.cpp.

6.52.3.9 setbuf()

```
std::streambuf * gzfilebuf::setbuf (
    char_type * p,
    std::streamsize n ) [protected], [virtual]
```

Installs external stream buffer.

Parameters

<i>p</i>	Pointer to char buffer.
<i>n</i>	Size of external buffer.

Returns

this on success, NULL on failure.

Call setbuf(0,0) to enable unbuffered output.

Definition at line 266 of file zfstream.cpp.

6.52.3.10 setcompression()

```
int gzfilebuf::setcompression (
    int comp_level,
    int comp_strategy = Z_DEFAULT_STRATEGY )
```

Set compression level and strategy on the fly.

Parameters

<i>comp_level</i>	Compression level (see zlib.h for allowed values)
<i>comp_strategy</i>	Compression strategy (see zlib.h for allowed values)

Returns

Z_OK on success, Z_STREAM_ERROR otherwise.

Unfortunately, these parameters cannot be modified separately, as the previous `zfstream` version assumed. Since the strategy is seldom changed, it can default and `setcompression(level)` then becomes like the old `setcompression-level(level)`.

Definition at line 43 of file `zfstream.cpp`.

6.52.3.11 showmany()

```
std::streamsize gzfilebuf::showmany() [protected], [virtual]
```

Number of characters available in stream buffer.

Returns

Number of characters.

This indicates number of characters in get area of stream buffer. These characters can be read without accessing the gzipped file.

Definition at line 169 of file `zfstream.cpp`.

6.52.3.12 sync()

```
int gzfilebuf::sync ( ) [protected], [virtual]
```

Flush stream buffer to file.

Returns

0 on success, -1 on error.

This calls underflow(EOF) to do the job.

Definition at line 299 of file zfstream.cpp.

6.52.3.13 underflow()

```
gzfilebuf::int_type gzfilebuf::underflow ( ) [protected], [virtual]
```

Fill get area from gzipped file.

Returns

First character in get area on success, EOF on error.

This actually reads characters from gzipped file to stream buffer. Always buffered.

Definition at line 183 of file zfstream.cpp.

6.52.4 Member Data Documentation

6.52.4.1 buffer

```
char_type* gzfilebuf::buffer [private]
```

Stream buffer.

For simplicity this remains allocated on the free store for the entire life span of the gzfilebuf object, unless replaced by setbuf.

Definition at line 207 of file zfstream.h.

6.52.4.2 buffer_size

```
std::streamsize gzfilebuf::buffer_size [private]
```

Stream buffer size.

Defaults to system default buffer size (typically 8192 bytes). Modified by setbuf.

Definition at line 215 of file zfstream.h.

6.52.4.3 file

```
gzFile gzfilebuf::file [private]
```

Underlying file pointer.

Definition at line 186 of file zfstream.h.

6.52.4.4 io_mode

```
std::ios_base::openmode gzfilebuf::io_mode [private]
```

Mode in which file was opened.

Definition at line 191 of file zfstream.h.

6.52.4.5 own_buffer

```
bool gzfilebuf::own_buffer [private]
```

True if this object owns stream buffer.

This makes the class responsible for deleting the buffer upon destruction.

Definition at line 223 of file zfstream.h.

6.52.4.6 own_fd

```
bool gzfilebuf::own_fd [private]
```

True if this object owns file descriptor.

This makes the class responsible for closing the file upon destruction.

Definition at line 199 of file `zfstream.h`.

The documentation for this class was generated from the following files:

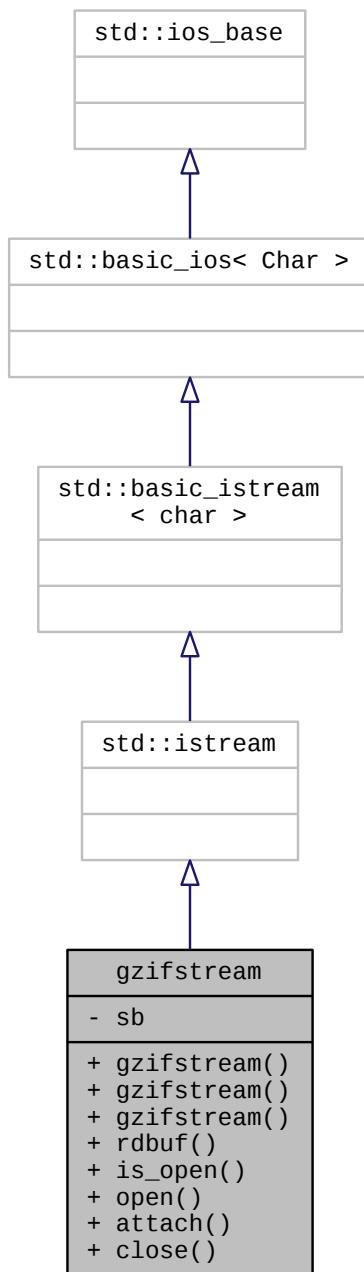
- src/[zfstream.h](#)
- src/[zfstream.cpp](#)

6.53 gzifstream Class Reference

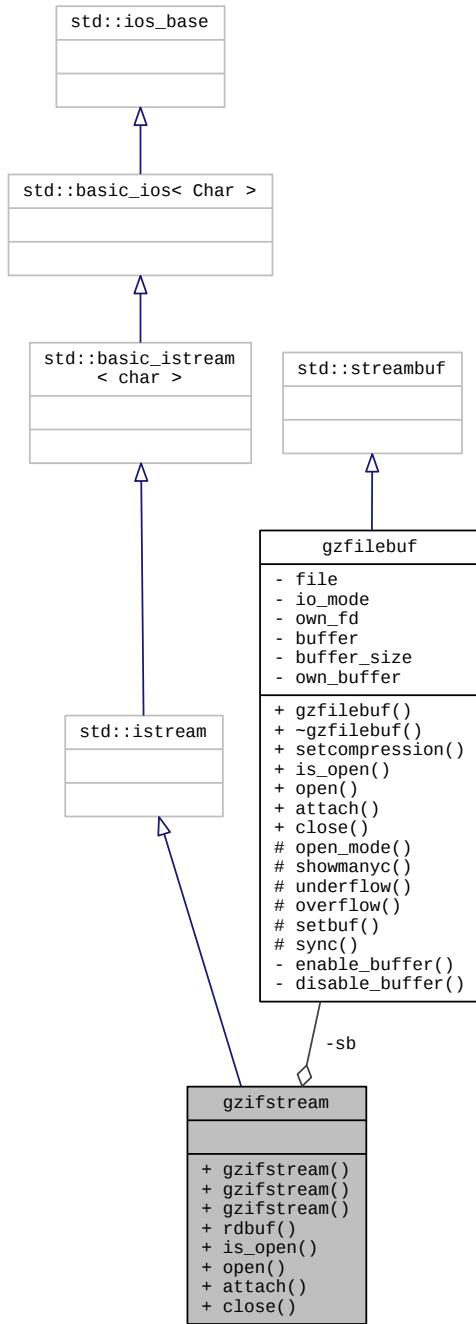
Gzipped file input stream class.

```
#include <zfstream.h>
```

Inheritance diagram for gzifstream:



Collaboration diagram for gzifstream:



Public Member Functions

- `gzifstream()`

- `gzifstream` (const char *name, std::ios_base::openmode mode=std::ios_base::in)
Construct stream on gzipped file to be opened.
- `gzifstream` (int fd, std::ios_base::openmode mode=std::ios_base::in)
Construct stream on already open gzipped file.
- `gzfilebuf * rdbuf () const`
Obtain underlying stream buffer.
- `bool is_open ()`
Check if file is open.
- `void open (const char *name, std::ios_base::openmode mode=std::ios_base::in)`
Open gzipped file.
- `void attach (int fd, std::ios_base::openmode mode=std::ios_base::in)`
Attach to already open gzipped file.
- `void close ()`
Close gzipped file.

Private Attributes

- `gzfilebuf sb`
Underlying stream buffer.

6.53.1 Detailed Description

Gzipped file input stream class.

This class implements ifstream for gzipped files. Seeking and putback is not supported yet.

Definition at line 234 of file zfstream.h.

6.53.2 Constructor & Destructor Documentation

6.53.2.1 `gzifstream()` [1/3]

```
gzifstream::gzifstream ( )
```

Definition at line 374 of file zfstream.cpp.

6.53.2.2 `gzifstream()` [2/3]

```
gzifstream::gzifstream (
    const char * name,
    std::ios_base::openmode mode = std::ios_base::in ) [explicit]
```

Construct stream on gzipped file to be opened.

Parameters

<i>name</i>	File name.
<i>mode</i>	Open mode flags (forced to contain ios::in).

Definition at line 379 of file zfstream.cpp.

6.53.2.3 gzifstream() [3/3]

```
gzifstream::gzifstream (
    int fd,
    std::ios_base::openmode mode = std::ios_base::in ) [explicit]
```

Construct stream on already open gzipped file.

Parameters

<i>fd</i>	File descriptor.
<i>mode</i>	Open mode flags (forced to contain ios::in).

Definition at line 388 of file zfstream.cpp.

6.53.3 Member Function Documentation**6.53.3.1 attach()**

```
void gzifstream::attach (
    int fd,
    std::ios_base::openmode mode = std::ios_base::in )
```

Attach to already open gzipped file.

Parameters

<i>fd</i>	File descriptor.
<i>mode</i>	Open mode flags (forced to contain ios::in).

Stream will be in state good() if attach succeeded; otherwise in state fail().

Definition at line 409 of file zfstream.cpp.

6.53.3.2 close()

```
void gzifstream::close ( )
```

Close gzipped file.

Stream will be in state fail() if close failed.

Definition at line 420 of file zfstream.cpp.

6.53.3.3 is_open()

```
bool gzifstream::is_open ( ) [inline]
```

Check if file is open.

Returns

True if file is open.

Definition at line 270 of file zfstream.h.

6.53.3.4 open()

```
void gzifstream::open (
    const char * name,
    std::ios_base::openmode mode = std::ios_base::in )
```

Open gzipped file.

Parameters

<i>name</i>	File name.
<i>mode</i>	Open mode flags (forced to contain ios::in).

Stream will be in state good() if file opens successfully; otherwise in state fail(). This differs from the behavior of ifstream, which never sets the state to good() and therefore won't allow you to reuse the stream for a second file unless you manually clear() the state. The choice is a matter of convenience.

Definition at line 398 of file zfstream.cpp.

6.53.3.5 rdbuf()

```
gzfilebuf* gzifstream::rdbuf ( ) const [inline]
```

Obtain underlying stream buffer.

Definition at line 262 of file zfstream.h.

6.53.4 Member Data Documentation

6.53.4.1 sb

```
gzfilebuf gzifstream::sb [private]
```

Underlying stream buffer.

Definition at line 312 of file zfstream.h.

The documentation for this class was generated from the following files:

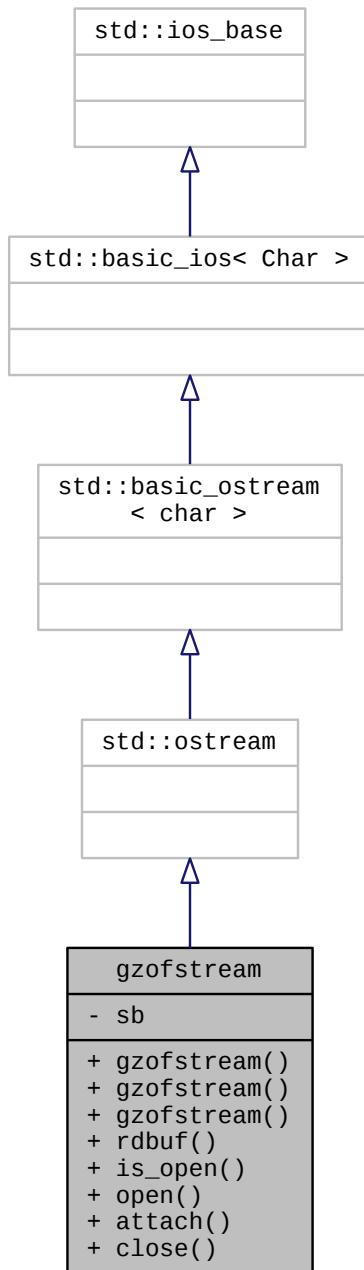
- src/zfstream.h
- src/zfstream.cpp

6.54 gzofstream Class Reference

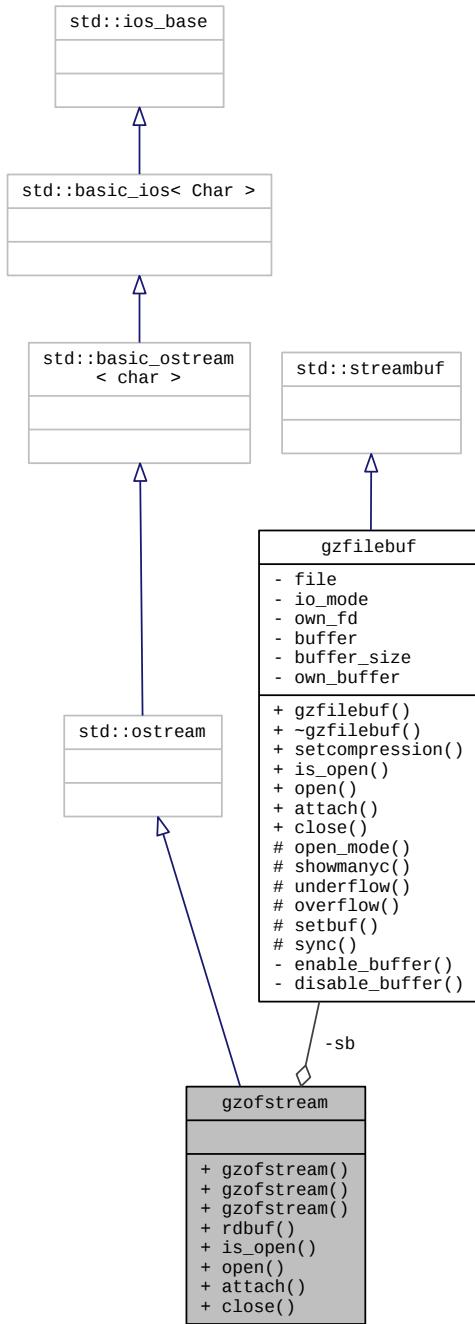
Gzipped file output stream class.

```
#include <zfstream.h>
```

Inheritance diagram for gzofstream:



Collaboration diagram for gzofstream:



Public Member Functions

- `gzofstream ()`

- **gzofstream** (const char *name, std::ios_base::openmode mode=std::ios_base::out)
Construct stream on gzipped file to be opened.
- **gzofstream** (int fd, std::ios_base::openmode mode=std::ios_base::out)
Construct stream on already open gzipped file.
- **gzfilebuf** * **rdbuf** () const
Obtain underlying stream buffer.
- **bool** **is_open** ()
Check if file is open.
- **void** **open** (const char *name, std::ios_base::openmode mode=std::ios_base::out)
Open gzipped file.
- **void** **attach** (int fd, std::ios_base::openmode mode=std::ios_base::out)
Attach to already open gzipped file.
- **void** **close** ()
Close gzipped file.

Private Attributes

- **gzfilebuf** **sb**
Underlying stream buffer.

6.54.1 Detailed Description

Gzipped file output stream class.

This class implements ofstream for gzipped files. Seeking and putback is not supported yet.

Definition at line 323 of file zfstream.h.

6.54.2 Constructor & Destructor Documentation

6.54.2.1 **gzofstream()** [1/3]

```
gzofstream::gzofstream ( )
```

Definition at line 429 of file zfstream.cpp.

6.54.2.2 **gzofstream()** [2/3]

```
gzofstream::gzofstream (
    const char * name,
    std::ios_base::openmode mode = std::ios_base::out ) [explicit]
```

Construct stream on gzipped file to be opened.

Parameters

<i>name</i>	File name.
<i>mode</i>	Open mode flags (forced to contain ios::out).

Definition at line 434 of file zfstream.cpp.

6.54.2.3 gzofstream() [3/3]

```
gzofstream::gzofstream (
    int fd,
    std::ios_base::openmode mode = std::ios_base::out ) [explicit]
```

Construct stream on already open gzipped file.

Parameters

<i>fd</i>	File descriptor.
<i>mode</i>	Open mode flags (forced to contain ios::out).

Definition at line 443 of file zfstream.cpp.

6.54.3 Member Function Documentation**6.54.3.1 attach()**

```
void gzofstream::attach (
    int fd,
    std::ios_base::openmode mode = std::ios_base::out )
```

Attach to already open gzipped file.

Parameters

<i>fd</i>	File descriptor.
<i>mode</i>	Open mode flags (forced to contain ios::out).

Stream will be in state good() if attach succeeded; otherwise in state fail().

Definition at line 464 of file zfstream.cpp.

6.54.3.2 close()

```
void gzofstream::close ( )
```

Close gzipped file.

Stream will be in state fail() if close failed.

Definition at line 475 of file zfstream.cpp.

6.54.3.3 is_open()

```
bool gzofstream::is_open ( ) [inline]
```

Check if file is open.

Returns

True if file is open.

Definition at line 359 of file zfstream.h.

6.54.3.4 open()

```
void gzofstream::open (
    const char * name,
    std::ios_base::openmode mode = std::ios_base::out )
```

Open gzipped file.

Parameters

<i>name</i>	File name.
<i>mode</i>	Open mode flags (forced to contain ios::out).

Stream will be in state good() if file opens successfully; otherwise in state fail(). This differs from the behavior of ofstream, which never sets the state to good() and therefore won't allow you to reuse the stream for a second file unless you manually clear() the state. The choice is a matter of convenience.

Definition at line 453 of file zfstream.cpp.

6.54.3.5 rdbuf()

```
gzfilebuf* gzofstream::rdbuf ( ) const [inline]
```

Obtain underlying stream buffer.

Definition at line 351 of file zfstream.h.

6.54.4 Member Data Documentation

6.54.4.1 sb

```
gzfilebuf zfstream::sb [private]
```

Underlying stream buffer.

Definition at line 401 of file zfstream.h.

The documentation for this class was generated from the following files:

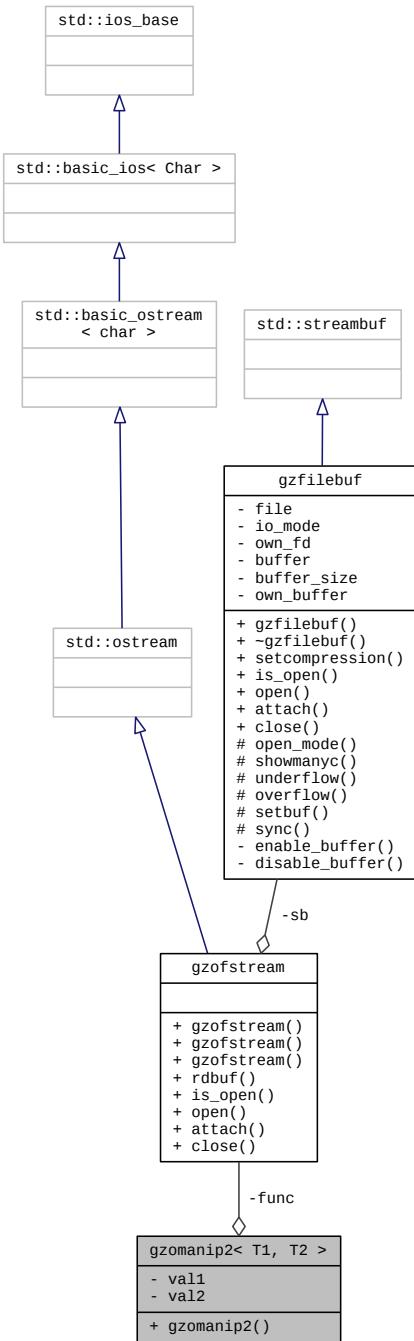
- src/zfstream.h
- src/zfstream.cpp

6.55 gzomanip2< T1, T2 > Class Template Reference

Gzipped file output stream manipulator class.

```
#include <zfstream.h>
```

Collaboration diagram for `gzomanip2< T1, T2 >`:



Public Member Functions

- `gzomanip2 (gzofstream &)(*f)(gzofstream &, T1, T2), T1 v1, T2 v2)`

Private Attributes

- `gzofstream &(* func)(gzofstream &, T1, T2)`
- `T1 val1`
- `T2 val2`

Friends

- `template<typename Ta , typename Tb >`
`gzofstream & operator<< (gzofstream &, const gzomanip2< Ta, Tb > &)`

6.55.1 Detailed Description

```
template<typename T1, typename T2>
class gzomanip2< T1, T2 >
```

Gzipped file output stream manipulator class.

This class defines a two-argument manipulator for `gzofstream`. It is used as base for the `setcompression(int,int)` manipulator.

Definition at line 413 of file `zfstream.h`.

6.55.2 Constructor & Destructor Documentation

6.55.2.1 `gzomanip2()`

```
template<typename T1 , typename T2 >
gzomanip2< T1, T2 >::gzomanip2 (
    gzofstream &(*)(gzofstream &, T1, T2) f,
    T1 v1,
    T2 v2 ) [inline]
```

Definition at line 449 of file `zfstream.h`.

6.55.3 Friends And Related Function Documentation

6.55.3.1 operator<<

```
template<typename T1, typename T2>
template<typename Ta , typename Tb >
gzofstream& operator<< (
    gzofstream & ,
    const gzomanip2< Ta, Tb > & ) [friend]
```

6.55.4 Member Data Documentation

6.55.4.1 func

```
template<typename T1, typename T2>
gzofstream& (* gzomanip2< T1, T2 >::func) (gzofstream &, T1, T2) [private]
```

Definition at line 429 of file zfstream.h.

6.55.4.2 val1

```
template<typename T1, typename T2>
T1 gzomanip2< T1, T2 >::val1 [private]
```

Definition at line 432 of file zfstream.h.

6.55.4.3 val2

```
template<typename T1, typename T2>
T2 gzomanip2< T1, T2 >::val2 [private]
```

Definition at line 433 of file zfstream.h.

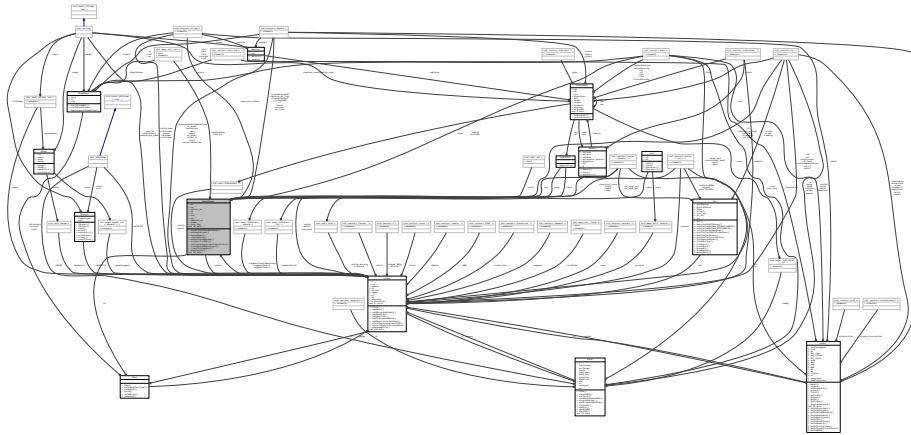
The documentation for this class was generated from the following file:

- src/zfstream.h

6.56 HaploPhase Class Reference

```
#include <phase.h>
```

Collaboration diagram for HaploPhase:



Public Member Functions

- void `enumeratePhasedWindows` (int)
- bool `makeWaplotype` (vector< int > &, vector< int > &)
- `HaploPhase (Plink &P_)`
- void `readTagFile` ()
- void `makeSlidingWindow` (string)
- void `setSpecificSNPs` (string)
- void `calculateHaplotypeFrequencies` ()
- void `trackSharedHaplotypes` ()
- void `trackThisSegment` ()
- map< int, int > `makeTestSet` (boolvec_t &, boolvec_t &)
- map< int, int > `makeSubHaplotypeSet` (boolvec_t &)
- string `getSubHaplotypeName` (boolvec_t &, boolvec_t &, int)
- `vector_t performHaplotypeTests` (bool, Perm &)
- void `imputeAllHaplotypes` ()
- void `calculatetHaplotypePhases` ()
- void `verboseDisplayWindows` (int i, bool use_ref=true)
- double `dosage` (int i, set< int > &h)
- set< int > `makeSetFromMap` (map< int, int > &h)
- void `reset` ()
- void `name` (string n)
- string `haplotypeName` (int i)
- int `nHap` ()
- double `testHaplotypeFreq` ()
- `vector_t phaseAllHaplotypes` (bool, Perm &)
- void `enumerateHaplotypes` (vector< int > &)
- void `enumerateAllPhases` ()

- void `setTestHaplotype` (string)
- vector< string > `returnHaplotypes` (vector< int > &)
- void `includeIndividuals` (int i)
- void `validateNonfounder` (int, vector< bool > &, vector< bool > &)
- void `enumerateNonfounderPhase` (int, vector< bool > &, vector< bool > &, int, int, int, int, int, vector< int > &, vector< int > &)
- bool `consistentNonfounderPhaseGivenParents` (int, int, int, int, int, int, int)
- bool `consistentNonfounderPhaseGivenGenotypes` (vector< bool > &, vector< bool > &, int, int)
- bool `consistentNonfounderMalePhaseGivenXGenotypes` (vector< bool > &, vector< bool > &s2, int)
- void `transmissionCount` (int, map< FamilyTransmissions, double > &)
- void `scoreTransmissions` (int, int, int, int, int, int, vector< int > &, vector< int > &)
- void `prunePhase` (int)
- void `queryGenotype` (int)
- void `queryThisGenotype` (int, int, int, `vector_t` &)
- void `phaseAndScoreNonfounder` (int)
- void `resolveWithKids` (int)
- void `performEM_original` ()
- void `performAlternEM` ()
- void `reportPhase` ()
- void `reportPhaseWideFormat` ()
- void `reportHaplotypeFrequencies` ()
- void `mainImputation` ()
- void `updateForImputation` ()
- void `imputeThisHaplotype` (int)
- double `imputeHaplotypes` (int, bool &, bool &)
- `vector_t` `imputeGenotype` (int, int)
- double `rsq` (int, int)
- double `dprime` (int, int)
- double `rsq_internal` (int, int)
- double `rsq_internal` (`boolvec_t` &, `boolvec_t` &, `boolvec_t` &, `boolvec_t` &)
- double `freq` (`boolvec_t` &, `boolvec_t` &)
- void `haplotypicCC` (map< int, int > &, int, bool)
- void `haplotypicWeightedCC` ()
- void `haplotypicTDT` (map< int, int > &, int, bool)
- void `haplotypicWeightedTDT` ()
- void `haplotypicQTL` (map< int, int > &, int, bool)
- void `calculateEmpiricalVariance` (int)
- void `calculateEmpiricalVariance` (set< int > &)
- set< int > `returnHaplotypeSet` (`boolvec_t` &, `boolvec_t` &)

Public Attributes

- `Plink` & `P`
- int `ns`
- int `nw`
- int `actual_nw`
- int `nh`
- int `nt`
- int `nsh`
- int `np`

- int `haploid_np`
- string `hname`
- int `test_hap`
- bool `X`
- bool `haploid`
- int `cnt_f`
- int `current`
- bool `reference_only`
- vector< vector< int > > `new_pred_locus`
- vector< string > `new_pred_allele`
- vector< map< string, double > > `new_pred_weighted_allele`
- vector< vector< bool > > `hap`
- vector< vector< int > > `hapi`
- `intvec_t S`
- `vector_t f`
- `matrix_t pp`
- vector< vector< int > > `hap1`
- vector< vector< int > > `hap2`
- map< vector< bool >, int > `hapmapb`
- map< vector< int >, int > `hapmap`
- vector< int > `ph_hap1`
- vector< int > `ph_hap2`
- vector< double > `ph_freq`
- vector< int > `haploid_ph_hap1`
- vector< double > `haploid_ph_freq`
- vector< bool > `ambig`
- vector< bool > `include`
- bool `subhaplotypes`
- map< int, int > `downcoding`
- vector< `HaploWindow` * > `windows`
- int `startWindow`
- int `finishWindow`
- vector< double > `trans`
- vector< double > `untrans`
- vector< map< `FamilyTransmissions`, double > > `phasemap`
- vector< `Locus` * > `new_map`
- vector< `Locus` * > `actual_map`
- vector< vector< bool > > `new_one`
- vector< vector< bool > > `new_two`
- int `p1`
- int `p2`
- bool `homozyg`
- ofstream `HFRQ`
- ofstream `HTEST`
- ofstream `HIMPUTE`
- ofstream `PHASE`
- ofstream `VPHASE`
- ofstream `WGT`
- double `result`
- double `pvalue`
- double `odds`

- double `case_freq`
- double `control_freq`
- `vector_t trackedIBS`
- `vector< int > trackedN`
- bool `calculateDp`
- `map< int, int > testSet`
- `set< int > sets`
- int `validN`
- bool `nonfounders`
- bool `useEmpiricalVariance`
- double `empiricalVariance`
- double `ratio`
- `vector_t transmissionX`
- `vector_t transmissionX2`
- double `transmissionTotal`

6.56.1 Detailed Description

Definition at line 53 of file phase.h.

6.56.2 Constructor & Destructor Documentation

6.56.2.1 HaploPhase()

```
HaploPhase::HaploPhase (
    Plink & P_ ) [inline]
```

Definition at line 190 of file phase.h.

6.56.3 Member Function Documentation

6.56.3.1 calculateEmpiricalVariance() [1/2]

```
void HaploPhase::calculateEmpiricalVariance (
    int h )
```

Definition at line 1345 of file haplohelper.cpp.

6.56.3.2 calculateEmpiricalVariance() [2/2]

```
void HaploPhase::calculateEmpiricalVariance (
    set< int > & hs )
```

Definition at line 1364 of file haplohelper.cpp.

6.56.3.3 calculateHaplotypeFrequencies()

```
void HaploPhase::calculateHaplotypeFrequencies ( )
```

Definition at line 146 of file haplohelper.cpp.

6.56.3.4 calculateHaplotypePhases()

```
void HaploPhase::calculateHaplotypePhases ( )
```

6.56.3.5 consistentNonfounderMalePhaseGivenXGenotypes()

```
bool HaploPhase::consistentNonfounderMalePhaseGivenXGenotypes (
    vector< bool > & s1,
    vector< bool > & s2,
    int h2 )
```

Definition at line 179 of file nonfounderphasing.cpp.

6.56.3.6 consistentNonfounderPhaseGivenGenotypes()

```
bool HaploPhase::consistentNonfounderPhaseGivenGenotypes (
    vector< bool > & s1,
    vector< bool > & s2,
    int h1,
    int h2 )
```

Definition at line 137 of file nonfounderphasing.cpp.

6.56.3.7 consistentNonfounderPhaseGivenParents()

```
bool HaploPhase::consistentNonfounderPhaseGivenParents (
    int i,
    int h1,
    int h2,
    int p1,
    int p2,
    int m1,
    int m2 )
```

Definition at line 214 of file nonfounderphasing.cpp.

6.56.3.8 dosage()

```
double HaploPhase::dosage (
    int i,
    set< int > & h )
```

Definition at line 1874 of file phase.cpp.

6.56.3.9 dprime()

```
double HaploPhase::dprime (
    int l1,
    int l2 )
```

Definition at line 1023 of file haplohelper.cpp.

6.56.3.10 enumerateAllPhases()

```
void HaploPhase::enumerateAllPhases ( )
```

Definition at line 225 of file haplohelper.cpp.

6.56.3.11 enumerateHaplotypes()

```
void HaploPhase::enumerateHaplotypes (
    vector< int > & s )
```

Definition at line 446 of file phase.cpp.

6.56.3.12 enumerateNonfounderPhase()

```
void HaploPhase::enumerateNonfounderPhase (
    int ,
    vector< bool > & ,
    vector< bool > & ,
    int ,
    int ,
    int ,
    int ,
    vector< int > & ,
    vector< int > & )
```

6.56.3.13 enumeratePhasedWindows()

```
void HaploPhase::enumeratePhasedWindows (
    int i )
```

Definition at line 66 of file metaem.cpp.

6.56.3.14 freq()

```
double HaploPhase::freq (
    boolevec_t & mask1,
    boolevec_t & alleles1 )
```

Definition at line 951 of file haplohelper.cpp.

6.56.3.15 getSubHaplotypeName()

```
string HaploPhase::getSubHaplotypeName (
    boolevec_t & mask,
    boolevec_t & allele,
    int blank )
```

Definition at line 585 of file haplohelper.cpp.

6.56.3.16 haplotypeName()

```
string HaploPhase::haplotypeName (
    int i )
```

Definition at line 35 of file haplohelper.cpp.

6.56.3.17 haplotypicCC()

```
void HaploPhase::haplotypicCC (
    map< int, int > & tests,
    int nt,
    bool display )
```

Definition at line 29 of file haploCC.cpp.

6.56.3.18 haplotypicQTL()

```
void HaploPhase::haplotypicQTL (
    map< int, int > & tests,
    int nt,
    bool display_results )
```

Definition at line 30 of file haploQTL.cpp.

6.56.3.19 haplotypicTDT()

```
void HaploPhase::haplotypicTDT (
    map< int, int > & tests,
    int nt,
    bool display )
```

Definition at line 33 of file haploTDT.cpp.

6.56.3.20 haplotypicWeightedCC()

```
void HaploPhase::haplotypicWeightedCC ( )
```

Definition at line 303 of file haploCC.cpp.

6.56.3.21 haplotypicWeightedTDT()

```
void HaploPhase::haplotypicWeightedTDT ( )
```

Definition at line 165 of file haploTDT.cpp.

6.56.3.22 imputeAllHaplotypes()

```
void HaploPhase::imputeAllHaplotypes ( )
```

Definition at line 58 of file haplohelper.cpp.

6.56.3.23 imputeGenotype()

```
vector_t HaploPhase::imputeGenotype (
    int i,
    int l )
```

Definition at line 608 of file haplohelper.cpp.

6.56.3.24 imputeHaplotypes()

```
double HaploPhase::imputeHaplotypes (
    int i,
    bool & n1,
    bool & n2 )
```

Definition at line 679 of file haplohelper.cpp.

6.56.3.25 imputeThisHaplotype()

```
void HaploPhase::imputeThisHaplotype (
    int l )
```

Definition at line 175 of file haplohelper.cpp.

6.56.3.26 includeIndividuals()

```
void HaploPhase::includeIndividuals (
    int i )
```

Definition at line 511 of file phase.cpp.

6.56.3.27 mainImputation()

```
void HaploPhase::mainImputation ( )
```

Definition at line 190 of file impute.cpp.

6.56.3.28 makeSetFromMap()

```
set< int > HaploPhase::makeSetFromMap (
    map< int, int > & h )
```

Definition at line 1854 of file phase.cpp.

6.56.3.29 makeSlidingWindow()

```
void HaploPhase::makeSlidingWindow (
    string )
```

Definition at line 514 of file prephap.cpp.

6.56.3.30 makeSubHaplotypeSet()

```
map< int, int > HaploPhase::makeSubHaplotypeSet (
    boolvec_t & mask )
```

Definition at line 531 of file haplohelper.cpp.

6.56.3.31 makeTestSet()

```
map< int, int > HaploPhase::makeTestSet (
    boolvec_t & mask,
    boolvec_t & allele )
```

Definition at line 562 of file haplohelper.cpp.

6.56.3.32 makeWaplotype()

```
bool HaploPhase::makeWaplotype (
    vector< int > & wCounter,
    vector< int > & wMax )
```

Definition at line 36 of file metaem.cpp.

6.56.3.33 name()

```
void HaploPhase::name (
    string n ) [inline]
```

Definition at line 280 of file phase.h.

6.56.3.34 nHap()

```
int HaploPhase::nHap ( ) [inline]
```

Definition at line 287 of file phase.h.

6.56.3.35 performAlternEM()

```
void HaploPhase::performAlternEM ( )
```

Definition at line 591 of file phase.cpp.

6.56.3.36 performEM_original()

```
void HaploPhase::performEM_original ( )
```

Definition at line 42 of file em.cpp.

6.56.3.37 performHaplotypeTests()

```
vector_t HaploPhase::performHaplotypeTests (
    bool display,
    Perm & perm )
```

Definition at line 1677 of file phase.cpp.

6.56.3.38 phaseAllHaplotypes()

```
vector_t HaploPhase::phaseAllHaplotypes (
    bool display,
    Perm & perm )
```

Definition at line 35 of file phase.cpp.

6.56.3.39 phaseAndScoreNonfounder()

```
void HaploPhase::phaseAndScoreNonfounder (
    int i )
```

Definition at line 273 of file nonfounderphasing.cpp.

6.56.3.40 prunePhase()

```
void HaploPhase::prunePhase (
    int i )
```

Definition at line 1814 of file phase.cpp.

6.56.3.41 queryGenotype()

```
void HaploPhase::queryGenotype (
    int s )
```

Definition at line 35 of file genoerr.cpp.

6.56.3.42 queryThisGenotype()

```
void HaploPhase::queryThisGenotype (
    int i,
    int s,
    int g,
    vector_t & geno_freq )
```

Definition at line 122 of file genoerr.cpp.

6.56.3.43 readTagFile()

```
void HaploPhase::readTagFile ( )
```

Definition at line 54 of file prephap.cpp.

6.56.3.44 reportHaplotypeFrequencies()

```
void HaploPhase::reportHaplotypeFrequencies ( )
```

Definition at line 359 of file haplohelper.cpp.

6.56.3.45 reportPhase()

```
void HaploPhase::reportPhase ( )
```

Definition at line 371 of file haplohelper.cpp.

6.56.3.46 reportPhaseWideFormat()

```
void HaploPhase::reportPhaseWideFormat ( )
```

Definition at line 458 of file haplohelper.cpp.

6.56.3.47 reset()

```
void HaploPhase::reset ( ) [inline]
```

Definition at line 248 of file phase.h.

6.56.3.48 resolveWithKids()

```
void HaploPhase::resolveWithKids (
    int i )
```

Definition at line 247 of file nonfounderphasing.cpp.

6.56.3.49 returnHaplotypes()

```
vector< string > HaploPhase::returnHaplotypes (
    vector< int > & slist )
```

Definition at line 290 of file haplohelper.cpp.

6.56.3.50 returnHaplotypeSet()

```
set< int > HaploPhase::returnHaplotypeSet (
    boolvec_t & mask,
    boolvec_t & alleles )
```

Definition at line 1324 of file haplohelper.cpp.

6.56.3.51 rsq()

```
double HaploPhase::rsq (
    int l1,
    int l2 )
```

Definition at line 994 of file haplohelper.cpp.

6.56.3.52 rsq_internal() [1/2]

```
double HaploPhase::rsq_internal (
    int s1,
    int s2 )
```

Definition at line 825 of file haplohelper.cpp.

6.56.3.53 rsq_internal() [2/2]

```
double HaploPhase::rsq_internal (
    boolevec_t & mask1,
    boolevec_t & alleles1,
    boolevec_t & mask2,
    boolevec_t & alleles2 )
```

Definition at line 843 of file haplohelper.cpp.

6.56.3.54 scoreTransmissions()

```
void HaploPhase::scoreTransmissions (
    int h1,
    int h2,
    int p1,
    int p2,
    int m1,
    int m2,
    vector< int > & t1,
    vector< int > & u1 )
```

Definition at line 783 of file nonfounderphasing.cpp.

6.56.3.55 setSpecificSNPs()

```
void HaploPhase::setSpecificSNPs (
    string )
```

Definition at line 670 of file prephap.cpp.

6.56.3.56 setTestHaplotype()

```
void HaploPhase::setTestHaplotype (
    string )
```

Definition at line 317 of file haplohelper.cpp.

6.56.3.57 testHaplotypeFreq()

```
double HaploPhase::testHaplotypeFreq ( ) [inline]
```

Definition at line 292 of file phase.h.

6.56.3.58 trackSharedHaplotypes()

```
void HaploPhase::trackSharedHaplotypes ( )
```

Definition at line 1108 of file haplohelper.cpp.

6.56.3.59 trackThisSegment()

```
void HaploPhase::trackThisSegment ( )
```

Definition at line 1201 of file haplohelper.cpp.

6.56.3.60 transmissionCount()

```
void HaploPhase::transmissionCount (
    int ,
    map< FamilyTransmissions, double > & )
```

Definition at line 691 of file nonfounderphasing.cpp.

6.56.3.61 updateForImputation()

```
void HaploPhase::updateForImputation ( )
```

Definition at line 55 of file impute.cpp.

6.56.3.62 validateNonfounder()

```
void HaploPhase::validateNonfounder (
    int i,
    vector< bool > & s1,
    vector< bool > & s2 )
```

Definition at line 62 of file nonfounderphasing.cpp.

6.56.3.63 verboseDisplayWindows()

```
void HaploPhase::verboseDisplayWindows (
    int i,
    bool use_ref = true )
```

Definition at line 1529 of file haplohelper.cpp.

6.56.4 Member Data Documentation

6.56.4.1 actual_map

```
vector<Locus*> HaploPhase::actual_map
```

Definition at line 162 of file phase.h.

6.56.4.2 actual_nw

```
int HaploPhase::actual_nw
```

Definition at line 61 of file phase.h.

6.56.4.3 ambig

```
vector<bool> HaploPhase::ambig
```

Definition at line 127 of file phase.h.

6.56.4.4 calculateDp

```
bool HaploPhase::calculateDp
```

Definition at line 388 of file phase.h.

6.56.4.5 case_freq

```
double HaploPhase::case_freq
```

Definition at line 187 of file phase.h.

6.56.4.6 cnt_f

```
int HaploPhase::cnt_f
```

Definition at line 74 of file phase.h.

6.56.4.7 control_freq

```
double HaploPhase::control_freq
```

Definition at line 188 of file phase.h.

6.56.4.8 current

```
int HaploPhase::current
```

Definition at line 75 of file phase.h.

6.56.4.9 downcoding

```
map<int,int> HaploPhase::downcoding
```

Definition at line 134 of file phase.h.

6.56.4.10 empiricalVariance

```
double HaploPhase::empiricalVariance
```

Definition at line 416 of file phase.h.

6.56.4.11 f

```
vector_t HaploPhase::f
```

Definition at line 103 of file phase.h.

6.56.4.12 finishWindow

```
int HaploPhase::finishWindow
```

Definition at line 142 of file phase.h.

6.56.4.13 hap

```
vector<vector<bool>> HaploPhase::hap
```

Definition at line 96 of file phase.h.

6.56.4.14 hap1

```
vector<vector<int> > HaploPhase::hap1
```

Definition at line 108 of file phase.h.

6.56.4.15 hap2

```
vector<vector<int> > HaploPhase::hap2
```

Definition at line 109 of file phase.h.

6.56.4.16 hapi

```
vector<vector<int> > HaploPhase::hapi
```

Definition at line 97 of file phase.h.

6.56.4.17 haploid

```
bool HaploPhase::haploid
```

Definition at line 72 of file phase.h.

6.56.4.18 haploid_np

```
int HaploPhase::haploid_np
```

Definition at line 66 of file phase.h.

6.56.4.19 haploid_ph_freq

```
vector<double> HaploPhase::haploid_ph_freq
```

Definition at line 122 of file phase.h.

6.56.4.20 haploid_ph_hap1

```
vector<int> HaploPhase::haploid_ph_hap1
```

Definition at line 121 of file phase.h.

6.56.4.21 hapmap

```
map<vector<int>, int > HaploPhase::hapmap
```

Definition at line 113 of file phase.h.

6.56.4.22 hapmapb

```
map<vector<bool>, int> HaploPhase::hapmapb
```

Definition at line 112 of file phase.h.

6.56.4.23 HFRQ

```
ofstream HaploPhase::HFRQ
```

Definition at line 175 of file phase.h.

6.56.4.24 HIMPUTE

```
ofstream HaploPhase::HIMPUTE
```

Definition at line 177 of file phase.h.

6.56.4.25 hname

```
string HaploPhase::hname
```

Definition at line 68 of file phase.h.

6.56.4.26 homozyg

```
bool HaploPhase::homzyg
```

Definition at line 172 of file phase.h.

6.56.4.27 HPHASE

```
ofstream HaploPhase::HPHASE
```

Definition at line 178 of file phase.h.

6.56.4.28 HTEST

```
ofstream HaploPhase::HTEST
```

Definition at line 176 of file phase.h.

6.56.4.29 include

```
vector<bool> HaploPhase::include
```

Definition at line 130 of file phase.h.

6.56.4.30 new_map

```
vector<Locus*> HaploPhase::new_map
```

Definition at line 161 of file phase.h.

6.56.4.31 new_one

```
vector<vector<bool>> HaploPhase::new_one
```

Definition at line 163 of file phase.h.

6.56.4.32 new_pred_allele

```
vector<string> HaploPhase::new_pred_allele
```

Definition at line 86 of file phase.h.

6.56.4.33 new_pred_locus

```
vector<vector<int> > HaploPhase::new_pred_locus
```

Definition at line 83 of file phase.h.

6.56.4.34 new_pred_weighted_allele

```
vector<map<string, double> > HaploPhase::new_pred_weighted_allele
```

Definition at line 89 of file phase.h.

6.56.4.35 new_two

```
vector<vector<bool> > HaploPhase::new_two
```

Definition at line 164 of file phase.h.

6.56.4.36 nh

```
int HaploPhase::nh
```

Definition at line 62 of file phase.h.

6.56.4.37 nonfounders

```
bool HaploPhase::nonfounders
```

Definition at line 407 of file phase.h.

6.56.4.38 np

```
int HaploPhase::np
```

Definition at line 65 of file phase.h.

6.56.4.39 ns

```
int HaploPhase::ns
```

Definition at line 59 of file phase.h.

6.56.4.40 nsh

```
int HaploPhase::nsh
```

Definition at line 64 of file phase.h.

6.56.4.41 nt

```
int HaploPhase::nt
```

Definition at line 63 of file phase.h.

6.56.4.42 nw

```
int HaploPhase::nw
```

Definition at line 60 of file phase.h.

6.56.4.43 odds

```
double HaploPhase::odds
```

Definition at line 186 of file phase.h.

6.56.4.44 P

```
Plink& HaploPhase::P
```

Definition at line 57 of file phase.h.

6.56.4.45 p1

```
int HaploPhase::p1
```

Definition at line 170 of file phase.h.

6.56.4.46 p2

```
int HaploPhase::p2
```

Definition at line 171 of file phase.h.

6.56.4.47 ph_freq

```
vector<double> HaploPhase::ph_freq
```

Definition at line 119 of file phase.h.

6.56.4.48 ph_hap1

```
vector<int> HaploPhase::ph_hap1
```

Definition at line 117 of file phase.h.

6.56.4.49 ph_hap2

```
vector<int> HaploPhase::ph_hap2
```

Definition at line 118 of file phase.h.

6.56.4.50 phasemap

```
vector<map<FamilyTransmissions,double>> HaploPhase::phasemap
```

Definition at line 156 of file phase.h.

6.56.4.51 pp

```
matrix_t HaploPhase::pp
```

Definition at line 106 of file phase.h.

6.56.4.52 pvalue

```
double HaploPhase::pvalue
```

Definition at line 185 of file phase.h.

6.56.4.53 ratio

```
double HaploPhase::ratio
```

Definition at line 417 of file phase.h.

6.56.4.54 reference_only

```
bool HaploPhase::reference_only
```

Definition at line 77 of file phase.h.

6.56.4.55 result

```
double HaploPhase::result
```

Definition at line 184 of file phase.h.

6.56.4.56 S

```
intvec_t HaploPhase::S
```

Definition at line 100 of file phase.h.

6.56.4.57 sets

```
set<int> HaploPhase::sets
```

Definition at line 402 of file phase.h.

6.56.4.58 startWindow

```
int HaploPhase::startWindow
```

Definition at line 141 of file phase.h.

6.56.4.59 subhaplotypes

```
bool HaploPhase::subhaplotypes
```

Definition at line 133 of file phase.h.

6.56.4.60 test_hap

```
int HaploPhase::test_hap
```

Definition at line 69 of file phase.h.

6.56.4.61 testSet

```
map<int,int> HaploPhase::testSet
```

Definition at line 401 of file phase.h.

6.56.4.62 trackedIBS

```
vector_t HaploPhase::trackedIBS
```

Definition at line 220 of file phase.h.

6.56.4.63 trackedN

```
vector<int> HaploPhase::trackedN
```

Definition at line 221 of file phase.h.

6.56.4.64 trans

```
vector<double> HaploPhase::trans
```

Definition at line 154 of file phase.h.

6.56.4.65 transmissionTotal

```
double HaploPhase::transmissionTotal
```

Definition at line 422 of file phase.h.

6.56.4.66 transmissionX

```
vector_t HaploPhase::transmissionX
```

Definition at line 420 of file phase.h.

6.56.4.67 transmissionX2

```
vector_t HaploPhase::transmissionX2
```

Definition at line 421 of file phase.h.

6.56.4.68 untrans

```
vector<double> HaploPhase::untrans
```

Definition at line 155 of file phase.h.

6.56.4.69 useEmpiricalVariance

```
bool HaploPhase::useEmpiricalVariance
```

Definition at line 412 of file phase.h.

6.56.4.70 validN

```
int HaploPhase::validN
```

Definition at line 404 of file phase.h.

6.56.4.71 VPHASE

```
ofstream HaploPhase::VPHASE
```

Definition at line 179 of file phase.h.

6.56.4.72 WGT

```
ofstream HaploPhase::WGT
```

Definition at line 180 of file phase.h.

6.56.4.73 windows

```
vector<HaploWindow*> HaploPhase::windows
```

Definition at line 139 of file phase.h.

6.56.4.74 X

```
bool HaploPhase::X
```

Definition at line 71 of file phase.h.

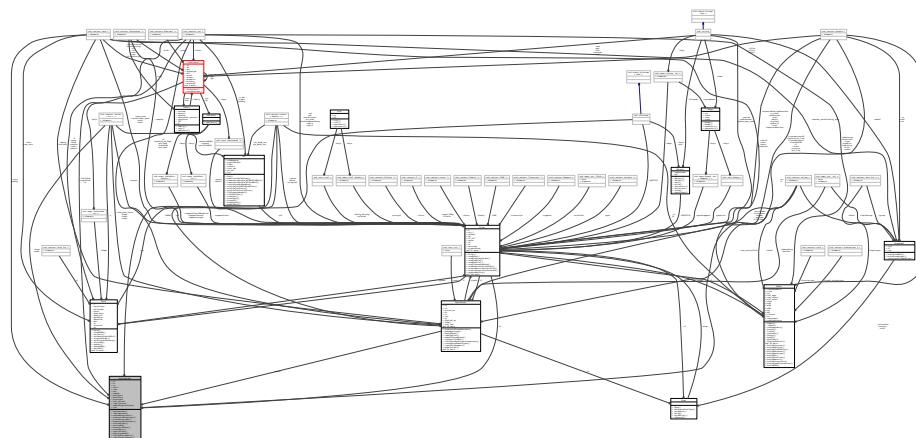
The documentation for this class was generated from the following files:

- [src/phase.h](#)
- [src/em.cpp](#)
- [src/genoerr.cpp](#)
- [src/haploCC.cpp](#)
- [src/haplohelper.cpp](#)
- [src/haploQTL.cpp](#)
- [src/haploTDT.cpp](#)
- [src/impute.cpp](#)
- [src/metaem.cpp](#)
- [src/nonfounderphasing.cpp](#)
- [src/phase.cpp](#)
- [src/prehap.cpp](#)

6.57 HaploWindow Class Reference

```
#include <haplowindow.h>
```

Collaboration diagram for HaploWindow:



Public Member Functions

- `HaploWindow (HaploPhase *, Plink *)`
- `~HaploWindow ()`
- `void expandGenogroups ()`
- `void enumerateGenogroups ()`
- `void pruneGenogroups (double t=par::haplo_plem_window_prune_phase)`
- `void enumerateHaplotypes (intvec_t &)`
- `void setStubCodes ()`
- `void performEM ()`
- `void enumeratePhase (int)`
- `void prunePhase (int, double t=par::haplo_plem_window_prune_phase)`
- `void reportPhase ()`
- `string haplotypeName (int i)`
- `vector_t leftStubFrequency ()`
- `vector_t rightStubFrequency ()`
- `void tallyUnambiguousCounts ()`

Public Attributes

- `int ns`
- `int nh`
- `int np`
- `HaploPhase * haplo`
- `Plink * P`
- `int start`
- `int stop`
- `vector_t f`
- `vector< vector< bool > > hap`
- `vector< int > leftStub`
- `vector< int > rightStub`
- `map< vector< bool >, int > hapmap`
- `intvec_t S`
- `matrix_t pp`
- `table_t hap1`
- `table_t hap2`
- `boolvec_t ambig`
- `vector_t uc`
- `set< MultiLocusGenotype * > genotypes`
- `vector< MultiLocusGenotype * > genoGroup`
- `bool converged`
- `bool left_passed`
- `bool right_passed`
- `vector< bool > zero`
- `double sampleLogLikelihood`
- `int iter`

6.57.1 Detailed Description

Definition at line 20 of file haplowindow.h.

6.57.2 Constructor & Destructor Documentation

6.57.2.1 HaploWindow()

```
HaploWindow::HaploWindow (
    HaploPhase * hp,
    PLink * plinkp )
```

Definition at line 93 of file haplowindow.cpp.

6.57.2.2 ~HaploWindow()

```
HaploWindow::~HaploWindow ( )
```

Definition at line 109 of file haplowindow.cpp.

6.57.3 Member Function Documentation

6.57.3.1 enumerateGenogroups()

```
void HaploWindow::enumerateGenogroups ( )
```

Definition at line 38 of file genogroup.cpp.

6.57.3.2 enumerateHaplotypes()

```
void HaploWindow::enumerateHaplotypes (
    intvec_t & Sall )
```

Definition at line 158 of file haplowindow.cpp.

6.57.3.3 enumeratePhase()

```
void HaploWindow::enumeratePhase ( int i )
```

Definition at line 236 of file haplowindow.cpp.

6.57.3.4 expandGenogroups()

```
void HaploWindow::expandGenogroups ( )
```

Definition at line 632 of file haplowindow.cpp.

6.57.3.5 haplotypeName()

```
string HaploWindow::haplotypeName ( int i )
```

Definition at line 71 of file haplowindow.cpp.

6.57.3.6 leftStubFrequency()

```
vector_t HaploWindow::leftStubFrequency ( )
```

Definition at line 588 of file haplowindow.cpp.

6.57.3.7 performEM()

```
void HaploWindow::performEM ( )
```

Definition at line 651 of file haplowindow.cpp.

6.57.3.8 pruneGenogroups()

```
void HaploWindow::pruneGenogroups (
    double t = par::haplo_plem_window_prune_phase )
```

Definition at line 526 of file haplowindow.cpp.

6.57.3.9 prunePhase()

```
void HaploWindow::prunePhase (
    int i,
    double t = par::haplo_plem_window_prune_phase )
```

Definition at line 538 of file haplowindow.cpp.

6.57.3.10 reportPhase()

```
void HaploWindow::reportPhase ( )
```

Definition at line 930 of file haplowindow.cpp.

6.57.3.11 rightStubFrequency()

```
vector_t HaploWindow::rightStubFrequency ( )
```

Definition at line 598 of file haplowindow.cpp.

6.57.3.12 setStubCodes()

```
void HaploWindow::setStubCodes ( )
```

Definition at line 121 of file haplowindow.cpp.

6.57.3.13 tallyUnambiguousCounts()

```
void HaploWindow::tallyUnambiguousCounts ( )
```

Definition at line 608 of file haplindow.cpp.

6.57.4 Member Data Documentation

6.57.4.1 ambig

```
boolvec_t HaploWindow::ambig
```

Definition at line 60 of file haplindow.h.

6.57.4.2 converged

```
bool HaploWindow::converged
```

Definition at line 72 of file haplindow.h.

6.57.4.3 f

```
vector_t HaploWindow::f
```

Definition at line 37 of file haplindow.h.

6.57.4.4 genoGroup

```
vector<MultiLocusGenotype*> HaploWindow::genoGroup
```

Definition at line 69 of file haplindow.h.

6.57.4.5 genotypes

```
set<MultiLocusGenotype*> HaploWindow::genotypes
```

Definition at line 66 of file haplowindow.h.

6.57.4.6 hap

```
vector<vector<bool> > HaploWindow::hap
```

Definition at line 40 of file haplowindow.h.

6.57.4.7 hap1

```
table_t HaploWindow::hap1
```

Definition at line 56 of file haplowindow.h.

6.57.4.8 hap2

```
table_t HaploWindow::hap2
```

Definition at line 57 of file haplowindow.h.

6.57.4.9 haplo

```
HaploPhase* HaploWindow::haplo
```

Definition at line 30 of file haplowindow.h.

6.57.4.10 hapmap

```
map<vector<bool>,int> HaploWindow::hapmap
```

Definition at line 47 of file haplowindow.h.

6.57.4.11 iter

```
int HaploWindow::iter
```

Definition at line 80 of file haplowindow.h.

6.57.4.12 left_passed

```
bool HaploWindow::left_passed
```

Definition at line 73 of file haplowindow.h.

6.57.4.13 leftStub

```
vector<int> HaploWindow::leftStub
```

Definition at line 43 of file haplowindow.h.

6.57.4.14 nh

```
int HaploWindow::nh
```

Definition at line 26 of file haplowindow.h.

6.57.4.15 np

```
int HaploWindow::np
```

Definition at line 27 of file haplowindow.h.

6.57.4.16 ns

```
int HaploWindow::ns
```

Definition at line 25 of file haplowindow.h.

6.57.4.17 P

```
Plink* HaploWindow::P
```

Definition at line 31 of file haplowindow.h.

6.57.4.18 pp

```
matrix_t HaploWindow::pp
```

Definition at line 53 of file haplowindow.h.

6.57.4.19 right_passed

```
bool HaploWindow::right_passed
```

Definition at line 74 of file haplowindow.h.

6.57.4.20 rightStub

```
vector<int> HaploWindow::rightStub
```

Definition at line 44 of file haplowindow.h.

6.57.4.21 S

```
intvec_t HaploWindow::S
```

Definition at line 50 of file haplowindow.h.

6.57.4.22 sampleLogLikelihood

```
double HaploWindow::sampleLogLikelihood
```

Definition at line 79 of file haplowindow.h.

6.57.4.23 start

```
int HaploWindow::start
```

Definition at line 34 of file haplowindow.h.

6.57.4.24 stop

```
int HaploWindow::stop
```

Definition at line 34 of file haplowindow.h.

6.57.4.25 uc

```
vector_t HaploWindow::uc
```

Definition at line 63 of file haplowindow.h.

6.57.4.26 zero

```
vector<bool> HaploWindow::zero
```

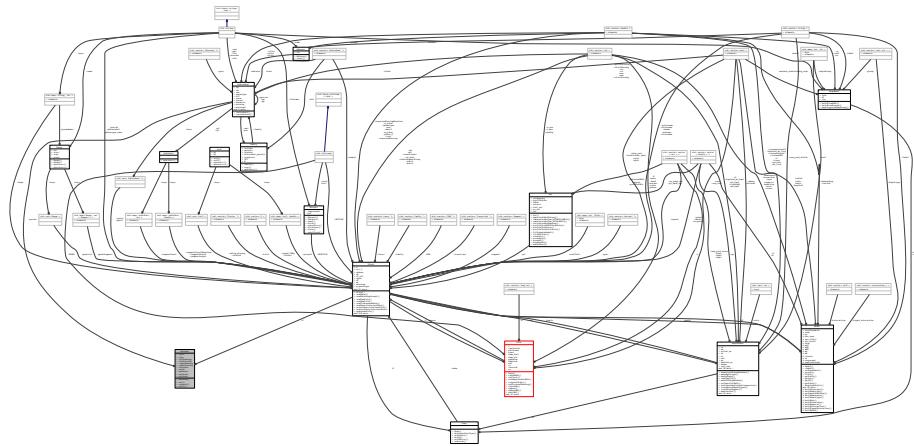
Definition at line 78 of file haplowindow.h.

The documentation for this class was generated from the following files:

- [src/haplowindow.h](#)
- [src/genogroup.cpp](#)
- [src/haplowindow.cpp](#)

6.58 HWindow Class Reference

Collaboration diagram for HWindow:



Public Member Functions

- `HWindow (Plink *, Individual *)`
- `void set (int, int)`
- `void recount ()`
- `void shift ()`

Public Attributes

- `int start`
- `int stop`
- `bool leftHomozyg`
- `bool leftMissing`
- `bool rightHomozyg`
- `bool rightMissing`
- `bool finished`
- `bool valid`
- `Individual * person`
- `Plink * P`
- `int homCount`
- `int hetCount`
- `int misCount`

6.58.1 Detailed Description

Definition at line 1938 of file homozyg.cpp.

6.58.2 Constructor & Destructor Documentation

6.58.2.1 HWindow()

```
HWindow::HWindow (
    PLink * plink,
    Individual * p )
```

Definition at line 1966 of file homozyg.cpp.

6.58.3 Member Function Documentation

6.58.3.1 recount()

```
void HWindow::recount ( )
```

Definition at line 1975 of file homozyg.cpp.

6.58.3.2 set()

```
void HWindow::set (
    int s1,
    int s2 )
```

Definition at line 2055 of file homozyg.cpp.

6.58.3.3 shift()

```
void HWindow::shift ( )
```

Definition at line 2003 of file homozyg.cpp.

6.58.4 Member Data Documentation

6.58.4.1 finished

```
bool HWindow::finished
```

Definition at line 1944 of file homozyg.cpp.

6.58.4.2 hetCount

```
int HWindow::hetCount
```

Definition at line 1950 of file homozyg.cpp.

6.58.4.3 homCount

```
int HWindow::homCount
```

Definition at line 1950 of file homozyg.cpp.

6.58.4.4 leftHomozyg

```
bool HWindow::leftHomozyg
```

Definition at line 1942 of file homozyg.cpp.

6.58.4.5 leftMissing

```
bool HWindow::leftMissing
```

Definition at line 1942 of file homozyg.cpp.

6.58.4.6 misCount

```
int HWindow::misCount
```

Definition at line 1950 of file homozyg.cpp.

6.58.4.7 P

```
Plink* HWindow::P
```

Definition at line 1948 of file homozyg.cpp.

6.58.4.8 person

```
Individual* HWindow::person
```

Definition at line 1947 of file homozyg.cpp.

6.58.4.9 rightHomozyg

```
bool HWindow::rightHomozyg
```

Definition at line 1943 of file homozyg.cpp.

6.58.4.10 rightMissing

```
bool HWindow::rightMissing
```

Definition at line 1943 of file homozyg.cpp.

6.58.4.11 start

```
int HWindow::start
```

Definition at line 1941 of file homozyg.cpp.

6.58.4.12 stop

```
int HWindow::stop
```

Definition at line 1941 of file homozyg.cpp.

6.58.4.13 valid

```
bool HWindow::valid
```

Definition at line 1945 of file homozyg.cpp.

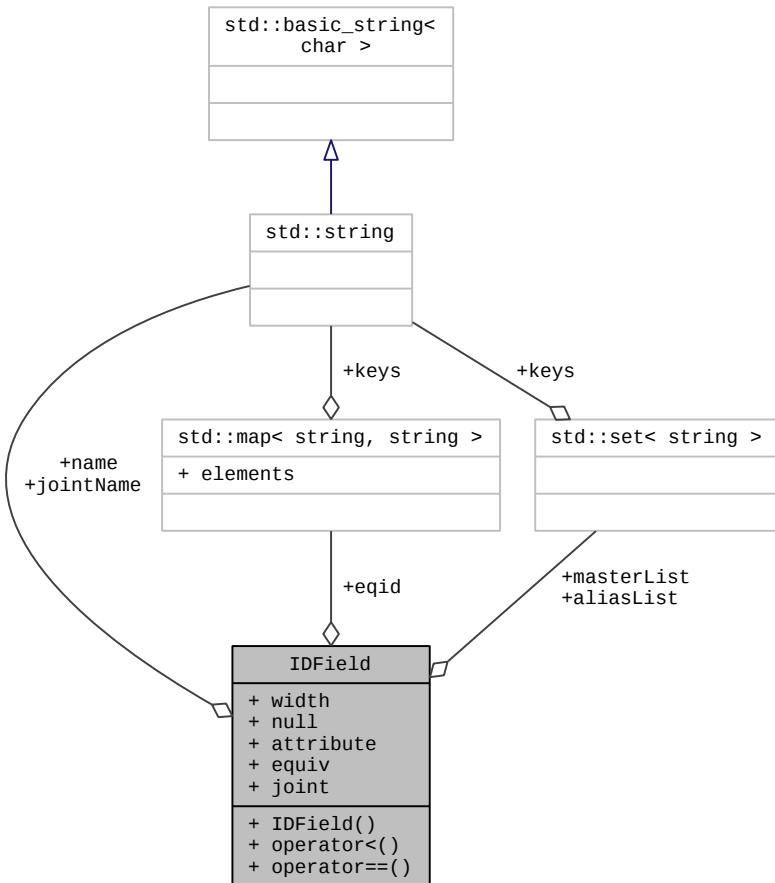
The documentation for this class was generated from the following file:

- [src/homozg.cpp](#)

6.59 IDField Class Reference

```
#include <idhelp.h>
```

Collaboration diagram for IDField:



Public Member Functions

- `IDField()`
- `bool operator< (const IDField &b) const`
- `bool operator==(const IDField &b) const`

Public Attributes

- `string name`
- `string jointName`
- `int width`
- `bool null`
- `bool attribute`
- `bool equiv`
- `bool joint`
- `map< string, string > eqid`
- `set< string > aliasList`
- `set< string > masterList`

6.59.1 Detailed Description

Definition at line 25 of file idhelp.h.

6.59.2 Constructor & Destructor Documentation

6.59.2.1 `IDField()`

```
IDField::IDField ( ) [inline]
```

Definition at line 45 of file idhelp.h.

6.59.3 Member Function Documentation

6.59.3.1 `operator<()`

```
bool IDField::operator< (
    const IDField & b ) const [inline]
```

Definition at line 54 of file idhelp.h.

6.59.3.2 operator==()

```
bool IDFA::operator== ( const IDFA & b ) const [inline]
```

Definition at line 61 of file idhelp.h.

6.59.4 Member Data Documentation

6.59.4.1 aliasList

```
set<string> IDFA::aliasList
```

Definition at line 41 of file idhelp.h.

6.59.4.2 attribute

```
bool IDFA::attribute
```

Definition at line 34 of file idhelp.h.

6.59.4.3 eqid

```
map<string, string > IDFA::eqid
```

Definition at line 40 of file idhelp.h.

6.59.4.4 equiv

```
bool IDFA::equiv
```

Definition at line 35 of file idhelp.h.

6.59.4.5 joint

```
bool IDF::joint
```

Definition at line 36 of file idhelp.h.

6.59.4.6 jointName

```
string IDF::jointName
```

Definition at line 30 of file idhelp.h.

6.59.4.7 masterList

```
set<string> IDF::masterList
```

Definition at line 42 of file idhelp.h.

6.59.4.8 name

```
string IDF::name
```

Definition at line 29 of file idhelp.h.

6.59.4.9 null

```
bool IDF::null
```

Definition at line 33 of file idhelp.h.

6.59.4.10 width

```
int IDF::width
```

Definition at line 31 of file idhelp.h.

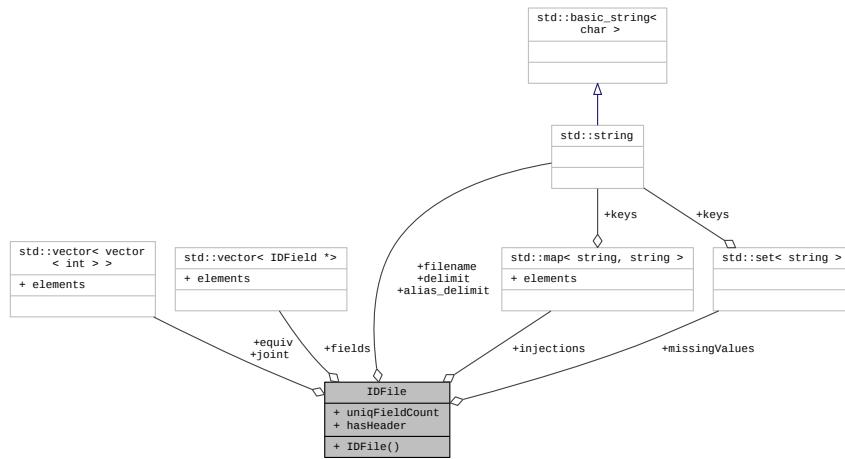
The documentation for this class was generated from the following file:

- src/[idhelp.h](#)

6.60 IDFfile Class Reference

```
#include <idhelp.h>
```

Collaboration diagram for IDFfile:



Public Member Functions

- [IDFfile \(\)](#)

Public Attributes

- string [filename](#)
- int [uniqFieldCount](#)
- bool [hasHeader](#)
- set< string > [missingValues](#)
- string [delimit](#)
- string [alias_delimit](#)
- vector< IDField * > [fields](#)
- vector< vector< int > > [joint](#)
- vector< vector< int > > [equiv](#)
- map< string, string > [injections](#)

6.60.1 Detailed Description

Definition at line 68 of file idhelp.h.

6.60.2 Constructor & Destructor Documentation

6.60.2.1 `IDFile()`

```
IDFile::IDFile ( ) [inline]
```

Definition at line 87 of file idhelp.h.

6.60.3 Member Data Documentation

6.60.3.1 `alias_delimit`

```
string IDFile::alias_delimit
```

Definition at line 80 of file idhelp.h.

6.60.3.2 `delimit`

```
string IDFile::delimit
```

Definition at line 79 of file idhelp.h.

6.60.3.3 `equiv`

```
vector< vector<int> > IDFile::equiv
```

Definition at line 84 of file idhelp.h.

6.60.3.4 `fields`

```
vector<IDField*> IDFile::fields
```

Definition at line 82 of file idhelp.h.

6.60.3.5 filename

```
string IDFfile::filename
```

Definition at line 72 of file idhelp.h.

6.60.3.6 hasHeader

```
bool IDFfile::hasHeader
```

Definition at line 76 of file idhelp.h.

6.60.3.7 injections

```
map<string, string> IDFfile::injections
```

Definition at line 85 of file idhelp.h.

6.60.3.8 joint

```
vector< vector<int> > IDFfile::joint
```

Definition at line 83 of file idhelp.h.

6.60.3.9 missingValues

```
set<string> IDFfile::missingValues
```

Definition at line 78 of file idhelp.h.

6.60.3.10 uniqFieldCount

```
int IDFfile::uniqFieldCount
```

Definition at line 74 of file idhelp.h.

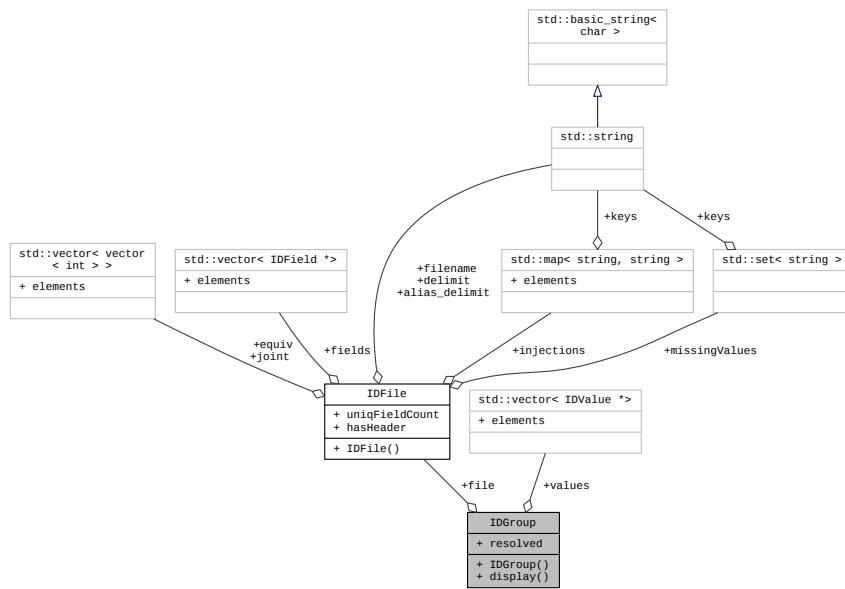
The documentation for this class was generated from the following file:

- src/[idhelp.h](#)

6.61 IDGroup Class Reference

```
#include <idhelp.h>
```

Collaboration diagram for IDGroup:



Public Member Functions

- [IDGroup \(\)](#)
- [void display \(\)](#)

Public Attributes

- `vector<IDValue*> values`
- `IDFile * file`
- `bool resolved`

6.61.1 Detailed Description

Definition at line 176 of file `idhelp.h`.

6.61.2 Constructor & Destructor Documentation

6.61.2.1 IDGroup()

```
IDGroup::IDGroup ( ) [inline]
```

Definition at line 185 of file idhelp.h.

6.61.3 Member Function Documentation

6.61.3.1 display()

```
void IDGroup::display ( ) [inline]
```

Definition at line 190 of file idhelp.h.

6.61.4 Member Data Documentation

6.61.4.1 file

```
IDFile* IDGroup::file
```

Definition at line 181 of file idhelp.h.

6.61.4.2 resolved

```
bool IDGroup::resolved
```

Definition at line 183 of file idhelp.h.

6.61.4.3 values

```
vector<IDValue*> IDGroup::values
```

Definition at line 180 of file idhelp.h.

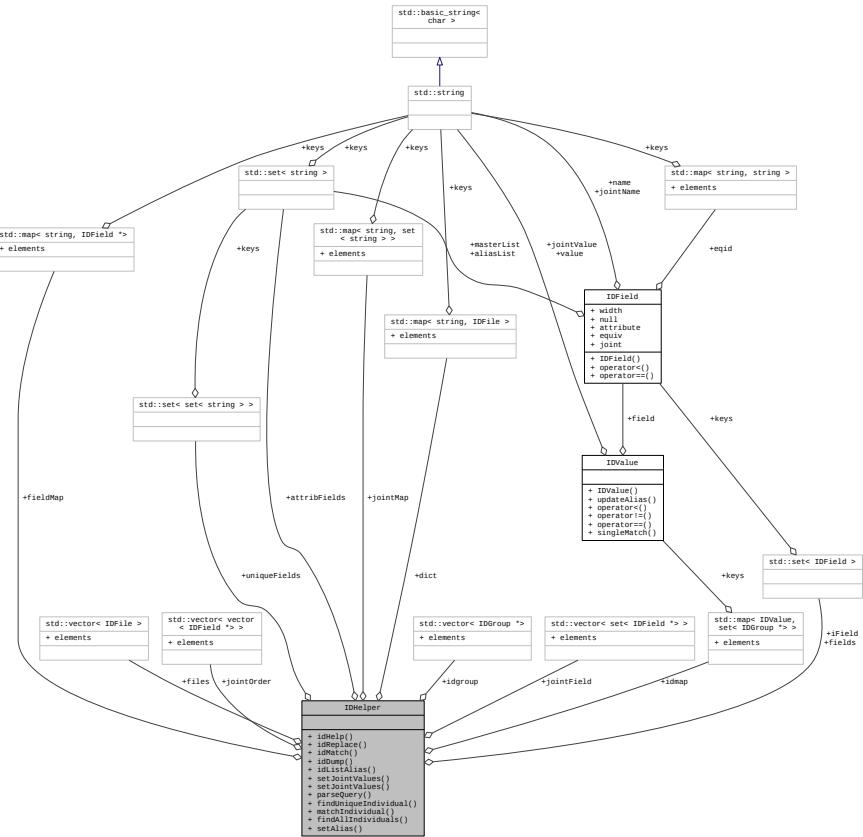
The documentation for this class was generated from the following file:

- src/[idhelp.h](#)

6.62 IDHelper Class Reference

```
#include <idhelp.h>
```

Collaboration diagram for IDHelper:



Public Member Functions

- void `idHelp ()`
- void `idReplace ()`
- void `idMatch ()`
- void `idDump ()`
- void `idListAlias ()`
- void `setJointValues (set< IDValue > &)`
- void `setJointValues (IDGroup *)`
- map<`IDField *`, `set< IDValue >`> `parseQuery (string)`
- `IDGroup * findUniqueIndividual (set< IDValue > &)`
- bool `matchIndividual (IDGroup *group, map< IDField *, set< IDValue > > &)`
- `set< IDGroup * > findAllIndividuals (map< IDField *, set< IDValue > > &)`
- bool `setAlias (IDField *, string, int, map< IDField *, string > &)`

Public Attributes

- vector< `IDFile` > files
- set< `IDField` > fields
- map< string, `IDField` * > fieldMap
- set< `IDField` >::iterator iField
- vector< `IDGroup` * > idgroup
- map< `IDValue`, set< `IDGroup` * > > idmap
- map< string, `IDFile` > dict
- set< set< string > > uniqueFields
- map< string, set< string > > jointMap
- vector< set< `IDField` * > > jointField
- vector< vector< `IDField` * > > jointOrder
- set< string > attribFields

6.62.1 Detailed Description

Definition at line 211 of file idhelp.h.

6.62.2 Member Function Documentation

6.62.2.1 findAllIndividuals()

```
set< IDGroup * > IDHelper::findAllIndividuals (
    map< IDField *, set< IDValue > > & matchTemplate )
```

Definition at line 343 of file idhelp.cpp.

6.62.2.2 findUniqueIndividual()

```
IDGroup * IDHelper::findUniqueIndividual (
    set< IDValue > & matchTemplate )
```

Definition at line 310 of file idhelp.cpp.

6.62.2.3 idDump()

```
void IDHelper::idDump ( )
```

Definition at line 2218 of file idhelp.cpp.

6.62.2.4 idHelp()

```
void IDHelper::idHelp ( )
```

Definition at line 411 of file idhelp.cpp.

6.62.2.5 idListAlias()

```
void IDHelper::idListAlias ( )
```

Definition at line 1499 of file idhelp.cpp.

6.62.2.6 idMatch()

```
void IDHelper::idMatch ( )
```

Definition at line 1818 of file idhelp.cpp.

6.62.2.7 idReplace()

```
void IDHelper::idReplace ( )
```

Definition at line 1532 of file idhelp.cpp.

6.62.2.8 matchIndividual()

```
bool IDHelper::matchIndividual (
    IDGroup * group,
    map< IDField *, set< IDValue > > & matchTemplate )
```

Definition at line 124 of file idhelp.cpp.

6.62.2.9 parseQuery()

```
map< IDField *, set< IDValue > > IDHelper::parseQuery (
    string q )
```

Definition at line 27 of file idhelp.cpp.

6.62.2.10 setAlias()

```
bool IDHelper::setAlias (
    IDField * myField,
    string val,
    int f,
    map< IDField *, string > & originalEquivalence )
```

Definition at line 356 of file idhelp.cpp.

6.62.2.11 setJointValues() [1/2]

```
void IDHelper::setJointValues (
    set< IDValue > & val )
```

Definition at line 179 of file idhelp.cpp.

6.62.2.12 setJointValues() [2/2]

```
void IDHelper::setJointValues (
    IDGroup * group )
```

Definition at line 200 of file idhelp.cpp.

6.62.3 Member Data Documentation**6.62.3.1 attribFields**

```
set<string> IDHelper::attribFields
```

Definition at line 237 of file idhelp.h.

6.62.3.2 dict

```
map<string, IDFile> IDHelper::dict
```

Definition at line 230 of file idhelp.h.

6.62.3.3 fieldMap

```
map<string, IDField*> IDHelper::fieldMap
```

Definition at line 220 of file idhelp.h.

6.62.3.4 fields

```
set<IDField> IDHelper::fields
```

Definition at line 219 of file idhelp.h.

6.62.3.5 files

```
vector<IDFile> IDHelper::files
```

Definition at line 216 of file idhelp.h.

6.62.3.6 idgroup

```
vector<IDGroup*> IDHelper::idgroup
```

Definition at line 225 of file idhelp.h.

6.62.3.7 idmap

```
map<IDValue, set<IDGroup*> > IDHelper::idmap
```

Definition at line 228 of file idhelp.h.

6.62.3.8 iField

```
set<IDField>::iterator IDHelper::iField
```

Definition at line 221 of file idhelp.h.

6.62.3.9 jointField

```
vector< set<IDField*> > IDHelper::jointField
```

Definition at line 234 of file idhelp.h.

6.62.3.10 jointMap

```
map<string, set<string> > IDHelper::jointMap
```

Definition at line 233 of file idhelp.h.

6.62.3.11 jointOrder

```
vector< vector<IDField*> > IDHelper::jointOrder
```

Definition at line 235 of file idhelp.h.

6.62.3.12 uniqueFields

```
set< set<string> > IDHelper::uniqueFields
```

Definition at line 232 of file idhelp.h.

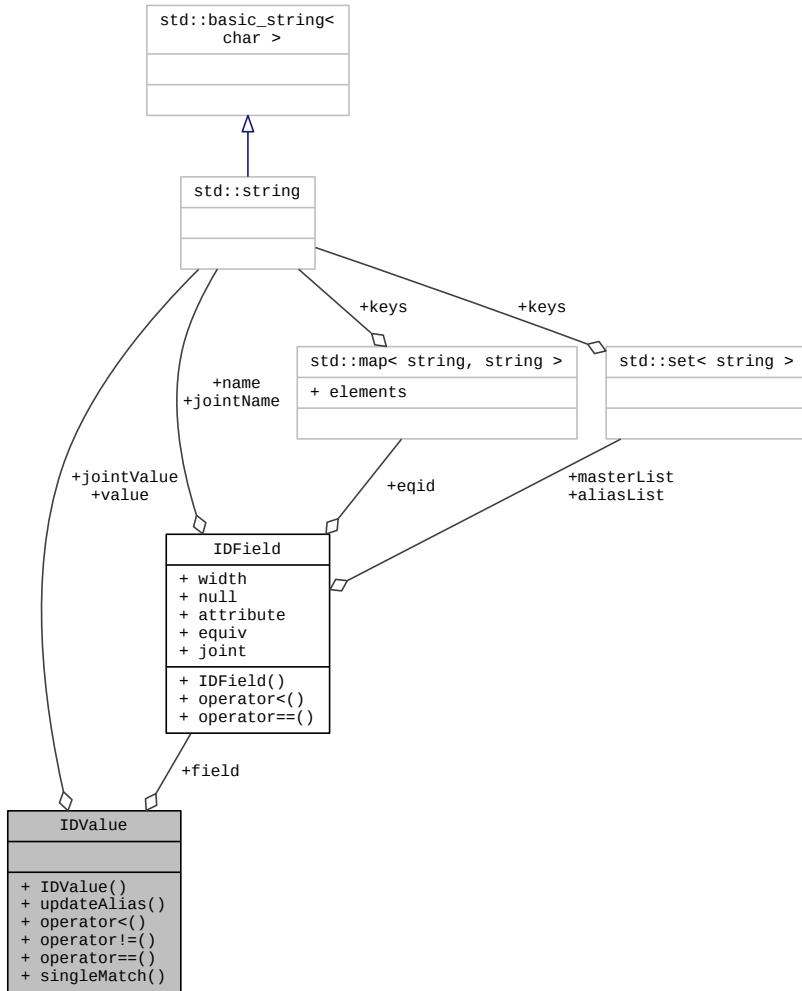
The documentation for this class was generated from the following files:

- src/idhelp.h
- src/idhelp.cpp

6.63 IDValue Class Reference

```
#include <idhelp.h>
```

Collaboration diagram for IDValue:



Public Member Functions

- `IDValue ()`
- void `updateAlias ()`
- bool `operator< (const IDValue &b) const`
- bool `operator!= (const IDValue &b) const`
- bool `operator== (const IDValue &b) const`
- bool `singleMatch (const IDValue &b) const`

Public Attributes

- `IDField * field`
- string `jointValue`
- string `value`

6.63.1 Detailed Description

Definition at line 97 of file idhelp.h.

6.63.2 Constructor & Destructor Documentation

6.63.2.1 IDValue()

```
IDValue::IDValue ( ) [inline]
```

Definition at line 101 of file idhelp.h.

6.63.3 Member Function Documentation

6.63.3.1 operator"!=()

```
bool IDValue::operator!= (
    const IDValue & b ) const [inline]
```

Definition at line 137 of file idhelp.h.

6.63.3.2 operator<()

```
bool IDValue::operator< (
    const IDValue & b ) const [inline]
```

Definition at line 123 of file idhelp.h.

6.63.3.3 operator==()

```
bool IDValue::operator== (
    const IDValue & b ) const [inline]
```

Definition at line 143 of file idhelp.h.

6.63.3.4 singleMatch()

```
bool IDValue::singleMatch (
    const IDValue & b ) const [inline]
```

Definition at line 163 of file idhelp.h.

6.63.3.5 updateAlias()

```
void IDValue::updateAlias () [inline]
```

Definition at line 108 of file idhelp.h.

6.63.4 Member Data Documentation

6.63.4.1 field

```
IDField* IDValue::field
```

Definition at line 115 of file idhelp.h.

6.63.4.2 jointValue

```
string IDValue::jointValue
```

Definition at line 117 of file idhelp.h.

6.63.4.3 value

```
string IDValue::value
```

Definition at line 120 of file idhelp.h.

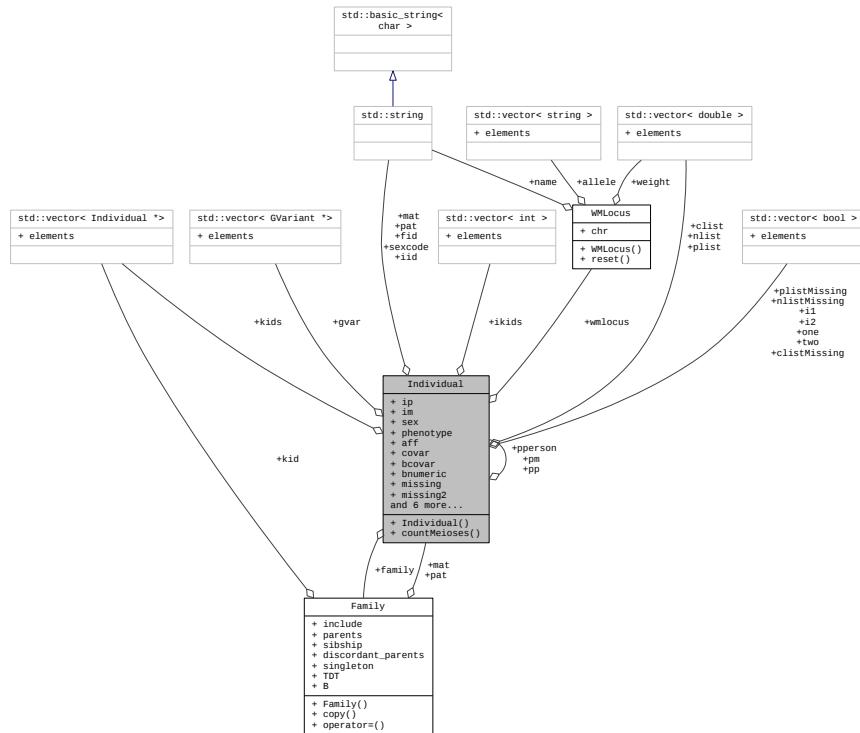
The documentation for this class was generated from the following file:

- src/idhelp.h

6.64 Individual Class Reference

```
#include <plink.h>
```

Collaboration diagram for Individual:



Public Member Functions

- Individual ()
 - int countMeioses (Individual *)

Public Attributes

- string `fid`
- string `iid`
- string `pat`
- string `mat`
- `Individual *` `pp`
- `Individual *` `pm`
- int `ip`
- int `im`
- `Individual *` `pperson`
- vector<`Individual *`> `kids`
- vector<int> `ikids`
- bool `sex`
- string `sexcode`
- double `phenotype`
- bool `aff`
- double `covar`
- bool `bcovar`
- bool `bnumeric`
- `vector_t` `nlist`
- vector<bool> `nlistMissing`
- `vector_t` `clist`
- vector<bool> `clistMissing`
- `vector_t` `plist`
- vector<bool> `plistMissing`
- bool `missing`
- bool `missing2`
- bool `flag`

WTF??? bcw 4/29/13.

- int `sol`
- bool `founder`
- `Family *` `family`
- vector<bool> `one`
- vector<bool> `two`
- vector<bool>::iterator `i1`
- vector<bool>::iterator `i2`
- vector<`GVariant *`> `gvar`
- `WMLocus` `wmlocus`
- double `T`
- double `B`
- double `W`

6.64.1 Detailed Description

Definition at line 219 of file plink.h.

6.64.2 Constructor & Destructor Documentation

6.64.2.1 Individual()

```
Individual::Individual ( ) [inline]
```

Definition at line 222 of file plink.h.

6.64.3 Member Function Documentation

6.64.3.1 countMeioses()

```
int Individual::countMeioses (
    Individual * f )
```

Definition at line 172 of file cfamily.cpp.

6.64.4 Member Data Documentation

6.64.4.1 aff

```
bool Individual::aff
```

Definition at line 281 of file plink.h.

6.64.4.2 B

```
double Individual::B
```

Definition at line 319 of file plink.h.

6.64.4.3 `bcovar`

```
bool Individual::bcovar
```

Definition at line 283 of file plink.h.

6.64.4.4 `bnumeric`

```
bool Individual::bnumeric
```

Definition at line 286 of file plink.h.

6.64.4.5 `clist`

```
vector_t Individual::clist
```

Definition at line 290 of file plink.h.

6.64.4.6 `clistMissing`

```
vector<bool> Individual::clistMissing
```

Definition at line 291 of file plink.h.

6.64.4.7 `covar`

```
double Individual::covar
```

Definition at line 282 of file plink.h.

6.64.4.8 `family`

```
Family* Individual::family
```

Definition at line 302 of file plink.h.

6.64.4.9 fid

```
string Individual::fid
```

Definition at line 253 of file plink.h.

6.64.4.10 flag

```
bool Individual::flag
```

WTF??? bkw 4/29/13.

Definition at line 298 of file plink.h.

6.64.4.11 founder

```
bool Individual::founder
```

Definition at line 301 of file plink.h.

6.64.4.12 gvar

```
vector<GVariant*> Individual::gvar
```

Definition at line 312 of file plink.h.

6.64.4.13 i1

```
vector<bool>::iterator Individual::i1
```

Definition at line 308 of file plink.h.

6.64.4.14 i2

```
vector<bool>::iterator Individual::i2
```

Definition at line 309 of file plink.h.

6.64.4.15 iid

```
string Individual::iid
```

Definition at line 254 of file plink.h.

6.64.4.16 ikids

```
vector<int> Individual::ikids
```

Definition at line 276 of file plink.h.

6.64.4.17 im

```
int Individual::im
```

Definition at line 266 of file plink.h.

6.64.4.18 ip

```
int Individual::ip
```

Definition at line 265 of file plink.h.

6.64.4.19 kids

```
vector<Individual*> Individual::kids
```

Definition at line 275 of file plink.h.

6.64.4.20 mat

```
string Individual::mat
```

Definition at line 258 of file plink.h.

6.64.4.21 missing

```
bool Individual::missing
```

Definition at line 296 of file plink.h.

6.64.4.22 missing2

```
bool Individual::missing2
```

Definition at line 297 of file plink.h.

6.64.4.23 nlist

```
vector_t Individual::nlist
```

Definition at line 287 of file plink.h.

6.64.4.24 nlistMissing

```
vector<bool> Individual::nlistMissing
```

Definition at line 288 of file plink.h.

6.64.4.25 one

```
vector<bool> Individual::one
```

Definition at line 305 of file plink.h.

6.64.4.26 pat

```
string Individual::pat
```

Definition at line 257 of file plink.h.

6.64.4.27 phenotype

```
double Individual::phenotype
```

Definition at line 280 of file plink.h.

6.64.4.28 plist

```
vector_t Individual::plist
```

Definition at line 293 of file plink.h.

6.64.4.29 plistMissing

```
vector<bool> Individual::plistMissing
```

Definition at line 294 of file plink.h.

6.64.4.30 pm

```
Individual* Individual::pm
```

Definition at line 262 of file plink.h.

6.64.4.31 pp

```
Individual* Individual::pp
```

Definition at line 261 of file plink.h.

6.64.4.32 pperson

```
Individual* Individual::pperson
```

Definition at line 272 of file plink.h.

6.64.4.33 sex

```
bool Individual::sex
```

Definition at line 278 of file plink.h.

6.64.4.34 sexcode

```
string Individual::sexcode
```

Definition at line 279 of file plink.h.

6.64.4.35 sol

```
int Individual::sol
```

Definition at line 300 of file plink.h.

6.64.4.36 T

```
double Individual::T
```

Definition at line 318 of file plink.h.

6.64.4.37 two

```
vector<bool> Individual::two
```

Definition at line 306 of file plink.h.

6.64.4.38 W

```
double Individual::w
```

Definition at line 320 of file plink.h.

6.64.4.39 wmlocus

WMlocus Individual::wmlocus

Definition at line 315 of file plink.h.

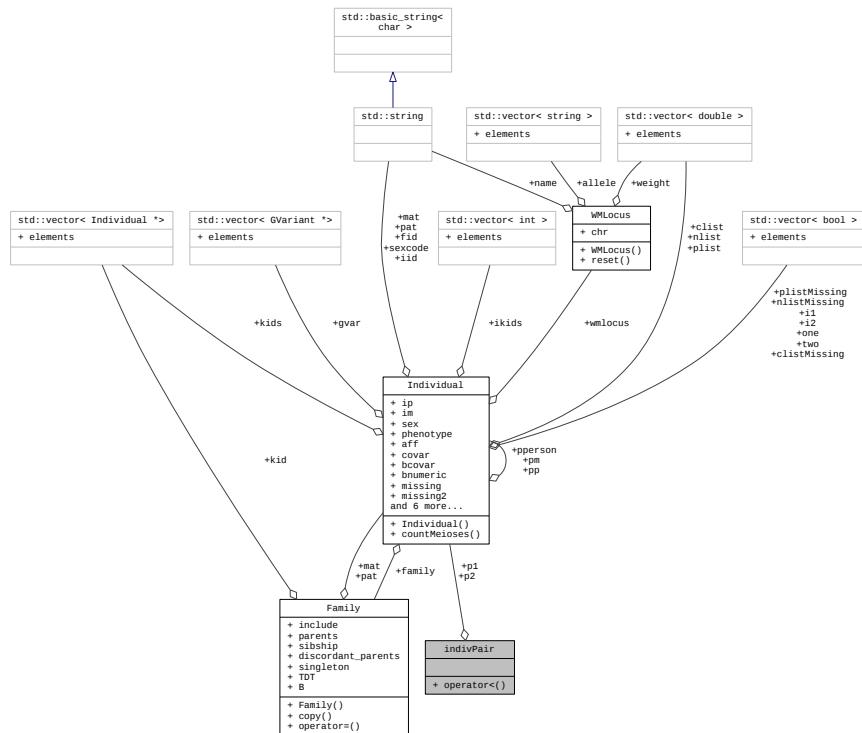
The documentation for this class was generated from the following files:

- src/plink.h
 - src/cfamily.cpp

6.65 indivPair Class Reference

```
#include <plink.h>
```

Collaboration diagram for indivPair:



Public Member Functions

- bool `operator< (const indivPair &b) const`

Public Attributes

- `Individual * p1`
- `Individual * p2`

6.65.1 Detailed Description

Definition at line 151 of file plink.h.

6.65.2 Member Function Documentation

6.65.2.1 `operator<()`

```
bool indivPair::operator< (
    const indivPair & b ) const [inline]
```

Definition at line 157 of file plink.h.

6.65.3 Member Data Documentation

6.65.3.1 `p1`

```
Individual* indivPair::p1
```

Definition at line 154 of file plink.h.

6.65.3.2 `p2`

```
Individual* indivPair::p2
```

Definition at line 155 of file plink.h.

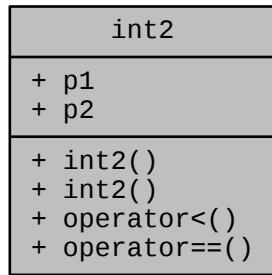
The documentation for this class was generated from the following file:

- `src/plink.h`

6.66 int2 Class Reference

```
#include <plink.h>
```

Collaboration diagram for int2:



Public Member Functions

- [int2 \(\)](#)
- [int2 \(int a, int b\)](#)
- [bool operator< \(const int2 &b\) const](#)
- [bool operator== \(const int2 &b\) const](#)

Public Attributes

- [int p1](#)
- [int p2](#)

6.66.1 Detailed Description

Definition at line 92 of file plink.h.

6.66.2 Constructor & Destructor Documentation

6.66.2.1 int2() [1/2]

```
int2::int2 ( ) [inline]
```

Definition at line 97 of file plink.h.

6.66.2.2 int2() [2/2]

```
int2::int2 (
    int a,
    int b ) [inline]
```

Definition at line 101 of file plink.h.

6.66.3 Member Function Documentation**6.66.3.1 operator<()**

```
bool int2::operator< (
    const int2 & b ) const [inline]
```

Definition at line 106 of file plink.h.

6.66.3.2 operator==()

```
bool int2::operator== (
    const int2 & b ) const [inline]
```

Definition at line 110 of file plink.h.

6.66.4 Member Data Documentation

6.66.4.1 p1

```
int int2::p1
```

Definition at line 94 of file plink.h.

6.66.4.2 p2

```
int int2::p2
```

Definition at line 95 of file plink.h.

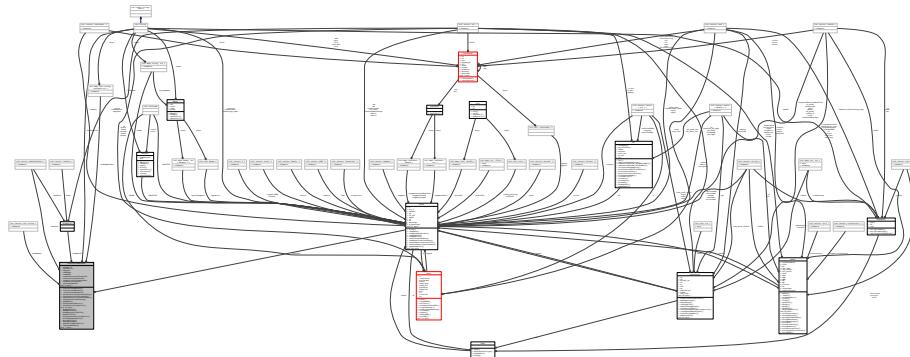
The documentation for this class was generated from the following file:

- src/[plink.h](#)

6.67 InteractionNetwork Class Reference

```
#include <InteractionNetwork.h>
```

Collaboration diagram for InteractionNetwork:



Public Member Functions

- `InteractionNetwork (std::string matrixFileParam, MatrixFileType fileType, bool isUpperTriangular, Plink *pp)`
- `InteractionNetwork (double **variablesMatrix, unsigned int dim, std::vector< std::string > &variableNames, Plink *pp)`
- `virtual ~InteractionNetwork ()`
- `bool SetConnectivityThresholding (bool connFlag)`
- `bool SetConnectivityThreshold (double threshold)`
- `bool SetConnectivityThresholdAbs (bool absFlag)`
- `bool SetBinaryThresholding (bool binaryFlag)`
- `unsigned int NumNodes ()`
- `arma::mat GetAdjacencyMatrix ()`
- `arma::mat GetConnectivityMatrix ()`
- `std::vector< std::string > GetNodeNames ()`
- `void PrintAdjacencyMatrix ()`
- `void PrintConnectivityMatrix ()`
- `void PrintSummary ()`
- `void PrintModulesSummary ()`
- `bool WriteToFile (std::string outfile, MatrixFileType fileType=CSV_FILE, NetworkMatrixType matrixType=NET_MATRIX_BOTH)`
- `ModularityResult ModularityLeadingEigenvector ()`
- `bool ripM (unsigned int pStartMergeOrder, unsigned int pMaxMergeOrder, unsigned int pMinModuleSize, unsigned int pMaxModuleSize)`
- `ModuleList GetModules ()`
- `bool Homophily (HomophilyResult &results)`
- `double ComputeQ ()`
- `bool SetModulesFromFile (std::string modulesFilename)`
- `void ShowModules ()`
- `void ShowModuleSizes ()`
- `void ShowModuleIndices ()`
- `void SaveModules (std::string saveFilename)`
- `void ShowHomophily ()`
- `bool Merge (InteractionNetwork &toMerge, double priorProbEdges, double alpha, double omega, double threshold)`
- `bool ApplyPowerTransform (double transformExponent=1.0)`
- `bool ApplyFisherTransform ()`
- `bool Deconvolve (arma::mat &nd, double alpha=1, double beta=0.9, int control=0)`
- `void SetDebugMode (bool debugFlag=true)`

Private Member Functions

- `bool ReadCsvFile (std::string matrixFilename)`
- `bool ReadGainFile (std::string gainFilename, bool isUpperTriangular=false)`
- `bool ReadBrainCorr1DFile (std::string corr1dFilename)`
- `bool ReadSifFile (std::string sifFilename)`
- `bool PrepareConnectivityMatrix ()`
- `bool WriteDelimitedFile (std::string outFilename, std::string fileType, NetworkMatrixType matrixType=NET_MATRIX_BOTH)`
- `bool WriteSifFile (std::string outFilename, NetworkMatrixType matrixType=NET_MATRIX_BOTH)`
- `void DebugMessage (std::string msg)`

- std::pair< double, arma::vec > [ModularityBestSplit](#) (arma::mat &B, double m)
- vector< unsigned int > [FlattenModules](#) ()
- [ModuleList RecursiveIndirectPathsModularity](#) ([ModuleIndices](#) thisModuleIdx)
- bool [GetNewmanModules](#) ([ModuleIndices](#) thisModuleIdx, [ModularityResult](#) &results)
- bool [MergeSmallModules](#) ([ModuleList](#) smallModules, [ModuleList](#) &results)
- bool [SumMatrixPowerSeries](#) (arma::mat &A, unsigned int maxPower, arma::mat &B)
- bool [CheckMergeResults](#) ([ModularityResult](#) results)
- bool [CheckIndices](#) ([ModuleIndices](#) toCheck)
- bool [AddModule](#) ([ModuleIndices](#) newModule)

Private Attributes

- std::string [networkFile](#)
- std::vector< std::string > [nodeNames](#)
- std::map< std::string, unsigned int > [nodeNameIndex](#)
- arma::mat [adjMatrix](#)
- arma::mat [connMatrix](#)
- arma::rowvec [degrees](#)
- double [numEdges](#)
- double [numNodes](#)
- bool [useConnectivityThreshold](#)
- double [connectivityThreshold](#)
- bool [connectivityThresholdAbs](#)
- bool [useBinaryThreshold](#)
- unsigned int [startMergeOrder](#)
- unsigned int [maxMergeOrder](#)
- unsigned int [maxModuleSize](#)
- unsigned int [minModuleSize](#)
- arma::mat [ripmMatrix](#)
- [RipmResult](#) [ripmResult](#)
- double [Q](#)
- [ModuleList](#) [modules](#)
- [Plink](#) * [inbixEnv](#)
- bool [debugMode](#)

6.67.1 Detailed Description

Definition at line 51 of file InteractionNetwork.h.

6.67.2 Constructor & Destructor Documentation

6.67.2.1 InteractionNetwork() [1/2]

```
InteractionNetwork::InteractionNetwork (
    std::string matrixFileParam,
    MatrixFileType fileType,
    bool isUpperTriangular,
    PLink * pp )
```

Definition at line 36 of file InteractionNetwork.cpp.

6.67.2.2 InteractionNetwork() [2/2]

```
InteractionNetwork::InteractionNetwork (
    double ** variablesMatrix,
    unsigned int dim,
    std::vector< std::string > & variableNames,
    PLink * pp )
```

Definition at line 83 of file InteractionNetwork.cpp.

6.67.2.3 ~InteractionNetwork()

```
InteractionNetwork::~InteractionNetwork ( ) [virtual]
```

Definition at line 116 of file InteractionNetwork.cpp.

6.67.3 Member Function Documentation

6.67.3.1 AddModule()

```
bool InteractionNetwork::AddModule (
    ModuleIndices newModule ) [private]
```

Definition at line 1582 of file InteractionNetwork.cpp.

6.67.3.2 ApplyFisherTransform()

```
bool InteractionNetwork::ApplyFisherTransform ( )
```

Definition at line 414 of file InteractionNetwork.cpp.

6.67.3.3 ApplyPowerTransform()

```
bool InteractionNetwork::ApplyPowerTransform (
    double transformExponent = 1.0 )
```

Definition at line 402 of file InteractionNetwork.cpp.

6.67.3.4 CheckIndices()

```
bool InteractionNetwork::CheckIndices (
    ModuleIndices toCheck ) [private]
```

Definition at line 1571 of file InteractionNetwork.cpp.

6.67.3.5 CheckMergeResults()

```
bool InteractionNetwork::CheckMergeResults (
    ModularityResult results ) [private]
```

Definition at line 1093 of file InteractionNetwork.cpp.

6.67.3.6 ComputeQ()

```
double InteractionNetwork::ComputeQ ( )
```

Definition at line 1276 of file InteractionNetwork.cpp.

6.67.3.7 DebugMessage()

```
void InteractionNetwork::DebugMessage (
    std::string msg) [private]
```

Definition at line 349 of file InteractionNetwork.cpp.

6.67.3.8 Deconvolve()

```
bool InteractionNetwork::Deconvolve (
    arma::mat & nd,
    double alpha = 1,
    double beta = 0.9,
    int control = 0 )
```

Definition at line 1451 of file InteractionNetwork.cpp.

6.67.3.9 FlattenModules()

```
vector< unsigned int > InteractionNetwork::FlattenModules () [private]
```

Definition at line 1433 of file InteractionNetwork.cpp.

6.67.3.10 GetAdjacencyMatrix()

```
arma::mat InteractionNetwork::GetAdjacencyMatrix ()
```

Definition at line 187 of file InteractionNetwork.cpp.

6.67.3.11 GetConnectivityMatrix()

```
arma::mat InteractionNetwork::GetConnectivityMatrix ()
```

Definition at line 191 of file InteractionNetwork.cpp.

6.67.3.12 GetModules()

```
ModuleList InteractionNetwork::GetModules ( )
```

Definition at line 1594 of file InteractionNetwork.cpp.

6.67.3.13 GetNewmanModules()

```
bool InteractionNetwork::GetNewmanModules (
    ModuleIndices thisModuleIdx,
    ModularityResult & results ) [private]
```

Definition at line 878 of file InteractionNetwork.cpp.

6.67.3.14 GetNodeNames()

```
vector< string > InteractionNetwork::GetNodeNames ( )
```

Definition at line 195 of file InteractionNetwork.cpp.

6.67.3.15 Homophily()

```
bool InteractionNetwork::Homophily (
    HomophilyResult & results )
```

Definition at line 1190 of file InteractionNetwork.cpp.

6.67.3.16 Merge()

```
bool InteractionNetwork::Merge (
    InteractionNetwork & toMerge,
    double priorProbEdges,
    double alpha,
    double omega,
    double threshold )
```

Definition at line 355 of file InteractionNetwork.cpp.

6.67.3.17 MergeSmallModules()

```
bool InteractionNetwork::MergeSmallModules (
    ModuleList smallModules,
    ModuleList & results ) [private]
```

Definition at line 1009 of file InteractionNetwork.cpp.

6.67.3.18 ModularityBestSplit()

```
pair< double, vec > InteractionNetwork::ModularityBestSplit (
    arma::mat & B,
    double m ) [private]
```

Definition at line 1391 of file InteractionNetwork.cpp.

6.67.3.19 ModularityLeadingEigenvector()

```
ModularityResult InteractionNetwork::ModularityLeadingEigenvector ( )
```

Definition at line 1104 of file InteractionNetwork.cpp.

6.67.3.20 NumNodes()

```
unsigned int InteractionNetwork::NumNodes ( )
```

Definition at line 183 of file InteractionNetwork.cpp.

6.67.3.21 PrepareConnectivityMatrix()

```
bool InteractionNetwork::PrepareConnectivityMatrix ( ) [private]
```

Definition at line 119 of file InteractionNetwork.cpp.

6.67.3.22 PrintAdjacencyMatrix()

```
void InteractionNetwork::PrintAdjacencyMatrix ( )
```

Definition at line 199 of file InteractionNetwork.cpp.

6.67.3.23 PrintConnectivityMatrix()

```
void InteractionNetwork::PrintConnectivityMatrix ( )
```

Definition at line 214 of file InteractionNetwork.cpp.

6.67.3.24 PrintModulesSummary()

```
void InteractionNetwork::PrintModulesSummary ( )
```

6.67.3.25 PrintSummary()

```
void InteractionNetwork::PrintSummary ( )
```

Definition at line 229 of file InteractionNetwork.cpp.

6.67.3.26 ReadBrainCorr1DFile()

```
bool InteractionNetwork::ReadBrainCorr1DFile (
    std::string corrlDfilename ) [private]
```

Definition at line 644 of file InteractionNetwork.cpp.

6.67.3.27 ReadCsvFile()

```
bool InteractionNetwork::ReadCsvFile (
    std::string matrixfilename ) [private]
```

Definition at line 434 of file InteractionNetwork.cpp.

6.67.3.28 ReadGainFile()

```
bool InteractionNetwork::ReadGainFile (
    std::string gainFilename,
    bool isUpperTriangular = false ) [private]
```

Definition at line 498 of file InteractionNetwork.cpp.

6.67.3.29 ReadSifFile()

```
bool InteractionNetwork::ReadSifFile (
    std::string sifFilename ) [private]
```

Definition at line 575 of file InteractionNetwork.cpp.

6.67.3.30 RecursiveIndirectPathsModularity()

```
ModuleList InteractionNetwork::RecursiveIndirectPathsModularity (
    ModuleIndices thisModuleIdx ) [private]
```

Definition at line 799 of file InteractionNetwork.cpp.

6.67.3.31 ripM()

```
bool InteractionNetwork::ripM (
    unsigned int pStartMergeOrder,
    unsigned int pMaxMergeOrder,
    unsigned int pMinModuleSize,
    unsigned int pMaxModuleSize )
```

Definition at line 714 of file InteractionNetwork.cpp.

6.67.3.32 SaveModules()

```
void InteractionNetwork::SaveModules (
    std::string saveFilename )
```

Definition at line 1375 of file InteractionNetwork.cpp.

6.67.3.33 SetBinaryThresholding()

```
bool InteractionNetwork::SetBinaryThresholding (
    bool binaryFlag )
```

Definition at line 177 of file InteractionNetwork.cpp.

6.67.3.34 SetConnectivityThreshold()

```
bool InteractionNetwork::SetConnectivityThreshold (
    double threshold )
```

Definition at line 164 of file InteractionNetwork.cpp.

6.67.3.35 SetConnectivityThresholdAbs()

```
bool InteractionNetwork::SetConnectivityThresholdAbs (
    bool absFlag )
```

Definition at line 171 of file InteractionNetwork.cpp.

6.67.3.36 SetConnectivityThresholding()

```
bool InteractionNetwork::SetConnectivityThresholding (
    bool connFlag )
```

Definition at line 158 of file InteractionNetwork.cpp.

6.67.3.37 SetDebugMode()

```
void InteractionNetwork::SetDebugMode (
    bool debugFlag = true )
```

Definition at line 1567 of file InteractionNetwork.cpp.

6.67.3.38 SetModulesFromFile()

```
bool InteractionNetwork::SetModulesFromFile ( std::string modulesFilename )
```

Definition at line 1304 of file InteractionNetwork.cpp.

6.67.3.39 ShowHomophily()

```
void InteractionNetwork::ShowHomophily ( )
```

Definition at line 1263 of file InteractionNetwork.cpp.

6.67.3.40 ShowModuleIndices()

```
void InteractionNetwork::ShowModuleIndices ( )
```

Definition at line 1355 of file InteractionNetwork.cpp.

6.67.3.41 ShowModules()

```
void InteractionNetwork::ShowModules ( )
```

Definition at line 1343 of file InteractionNetwork.cpp.

6.67.3.42 ShowModuleSizes()

```
void InteractionNetwork::ShowModuleSizes ( )
```

Definition at line 1367 of file InteractionNetwork.cpp.

6.67.3.43 SumMatrixPowerSeries()

```
bool InteractionNetwork::SumMatrixPowerSeries (
    arma::mat & A,
    unsigned int maxPower,
    arma::mat & B ) [private]
```

Definition at line 1078 of file InteractionNetwork.cpp.

6.67.3.44 WriteDelimitedFile()

```
bool InteractionNetwork::WriteDelimitedFile (
    std::string outfilename,
    std::string fileType,
    NetworkMatrixType matrixType = NET_MATRIX_BOTH ) [private]
```

Definition at line 270 of file InteractionNetwork.cpp.

6.67.3.45 WriteSifFile()

```
bool InteractionNetwork::WriteSifFile (
    std::string outfilename,
    NetworkMatrixType matrixType = NET_MATRIX_BOTH ) [private]
```

Definition at line 317 of file InteractionNetwork.cpp.

6.67.3.46 WriteToFile()

```
bool InteractionNetwork::WriteToFile (
    std::string outfile,
    MatrixFileType fileType = CSV_FILE,
    NetworkMatrixType matrixType = NET_MATRIX_BOTH )
```

Definition at line 247 of file InteractionNetwork.cpp.

6.67.4 Member Data Documentation

6.67.4.1 adjMatrix

```
arma::mat InteractionNetwork::adjMatrix [private]
```

Definition at line 154 of file InteractionNetwork.h.

6.67.4.2 connectivityThreshold

```
double InteractionNetwork::connectivityThreshold [private]
```

Definition at line 167 of file InteractionNetwork.h.

6.67.4.3 connectivityThresholdAbs

```
bool InteractionNetwork::connectivityThresholdAbs [private]
```

Definition at line 168 of file InteractionNetwork.h.

6.67.4.4 connMatrix

```
arma::mat InteractionNetwork::connMatrix [private]
```

Definition at line 156 of file InteractionNetwork.h.

6.67.4.5 debugMode

```
bool InteractionNetwork::debugMode [private]
```

Definition at line 185 of file InteractionNetwork.h.

6.67.4.6 degrees

```
arma::rowvec InteractionNetwork::degrees [private]
```

Definition at line 159 of file InteractionNetwork.h.

6.67.4.7 `inbixEnv`

```
Plink* InteractionNetwork::inbixEnv [private]
```

Definition at line 184 of file InteractionNetwork.h.

6.67.4.8 `maxMergeOrder`

```
unsigned int InteractionNetwork::maxMergeOrder [private]
```

Definition at line 173 of file InteractionNetwork.h.

6.67.4.9 `maxModuleSize`

```
unsigned int InteractionNetwork::maxModuleSize [private]
```

Definition at line 174 of file InteractionNetwork.h.

6.67.4.10 `minModuleSize`

```
unsigned int InteractionNetwork::minModuleSize [private]
```

Definition at line 175 of file InteractionNetwork.h.

6.67.4.11 `modules`

```
ModuleList InteractionNetwork::modules [private]
```

Definition at line 182 of file InteractionNetwork.h.

6.67.4.12 `networkFile`

```
std::string InteractionNetwork::networkFile [private]
```

Definition at line 148 of file InteractionNetwork.h.

6.67.4.13 nodeNameIndex

```
std::map<std::string, unsigned int> InteractionNetwork::nodeNameIndex [private]
```

Definition at line 151 of file InteractionNetwork.h.

6.67.4.14 nodeNames

```
std::vector<std::string> InteractionNetwork::nodeNames [private]
```

Definition at line 150 of file InteractionNetwork.h.

6.67.4.15 numEdges

```
double InteractionNetwork::numEdges [private]
```

Definition at line 161 of file InteractionNetwork.h.

6.67.4.16 numNodes

```
double InteractionNetwork::numNodes [private]
```

Definition at line 163 of file InteractionNetwork.h.

6.67.4.17 Q

```
double InteractionNetwork::Q [private]
```

Definition at line 181 of file InteractionNetwork.h.

6.67.4.18 ripmMatrix

```
arma::mat InteractionNetwork::ripmMatrix [private]
```

Definition at line 177 of file InteractionNetwork.h.

6.67.4.19 ripmResult

```
RipmResult InteractionNetwork::ripmResult [private]
```

Definition at line 178 of file InteractionNetwork.h.

6.67.4.20 startMergeOrder

```
unsigned int InteractionNetwork::startMergeOrder [private]
```

Definition at line 172 of file InteractionNetwork.h.

6.67.4.21 useBinaryThreshold

```
bool InteractionNetwork::useBinaryThreshold [private]
```

Definition at line 169 of file InteractionNetwork.h.

6.67.4.22 useConnectivityThreshold

```
bool InteractionNetwork::useConnectivityThreshold [private]
```

Definition at line 166 of file InteractionNetwork.h.

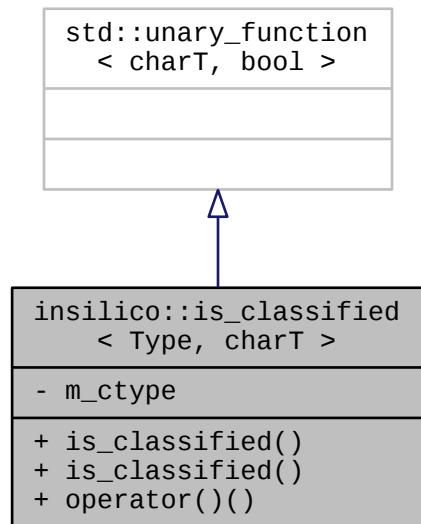
The documentation for this class was generated from the following files:

- src/[InteractionNetwork.h](#)
- src/[InteractionNetwork.cpp](#)

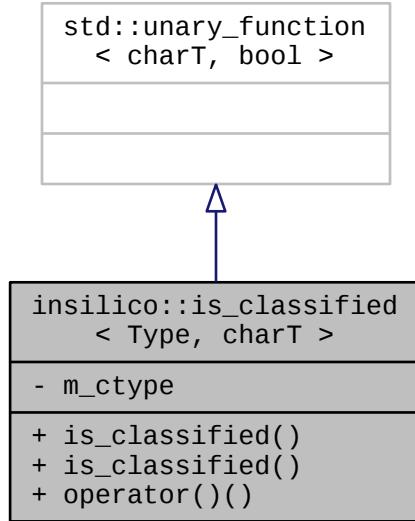
6.68 insilico::is_classified< Type, charT > Class Template Reference

```
#include <StringUtils.h>
```

Inheritance diagram for insilico::is_classified< Type, charT >:



Collaboration diagram for insilico::is_classified< Type, charT >:



Public Member Functions

- [is_classified \(std::ctype< charT > &ct\)](#)
- [is_classified \(const std::locale &loc=std::locale\(\)\)](#)
- [bool operator\(\) \(charT c\) const](#)

Private Attributes

- [std::ctype< charT > const & m_ctype](#)

6.68.1 Detailed Description

```
template<std::ctype_base::mask Type, class charT = char>
class insilico::is_classified< Type, charT >
```

Definition at line 40 of file StringUtil.h.

6.68.2 Constructor & Destructor Documentation

6.68.2.1 **is_classified()** [1/2]

```
template<std::ctype_base::mask Type, class charT = char>
insilico::is_classified< Type, charT >::is_classified (
    std::ctype< charT > & ct )  [inline]
```

Definition at line 44 of file StringUtil.h.

6.68.2.2 **is_classified()** [2/2]

```
template<std::ctype_base::mask Type, class charT = char>
insilico::is_classified< Type, charT >::is_classified (
    const std::locale & loc = std::locale() )  [inline]
```

Definition at line 47 of file StringUtil.h.

6.68.3 Member Function Documentation

6.68.3.1 **operator()()**

```
template<std::ctype_base::mask Type, class charT = char>
bool insilico::is_classified< Type, charT >::operator() (
    charT c ) const  [inline]
```

Definition at line 50 of file StringUtil.h.

6.68.4 Member Data Documentation

6.68.4.1 **m_ctype**

```
template<std::ctype_base::mask Type, class charT = char>
std::ctype<charT> const& insilico::is_classified< Type, charT >::m_ctype  [private]
```

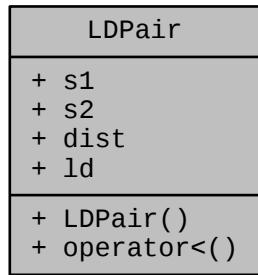
Definition at line 54 of file StringUtil.h.

The documentation for this class was generated from the following file:

- src/StringUtil.h

6.69 LDPair Class Reference

Collaboration diagram for LDPair:



Public Member Functions

- `LDPair (int s1_, int s2_, int dist_)`
- bool `operator< (const LDPair &b) const`

Public Attributes

- int `s1`
- int `s2`
- int `dist`
- double `ld`

Friends

- `ostream & operator<< (ostream &out, LDPair &v)`

6.69.1 Detailed Description

Definition at line 39 of file blox.cpp.

6.69.2 Constructor & Destructor Documentation

6.69.2.1 LDPair()

```
LDPair::LDPair (
    int s1_,
    int s2_,
    int dist_ ) [inline]
```

Definition at line 48 of file blox.cpp.

6.69.3 Member Function Documentation

6.69.3.1 operator<()

```
bool LDPair::operator< (
    const LDPair & b ) const [inline]
```

Definition at line 59 of file proxy.cpp.

6.69.4 Friends And Related Function Documentation

6.69.4.1 operator<<

```
ostream& operator<< (
    ostream & out,
    LDPair & v ) [friend]
```

Definition at line 53 of file blox.cpp.

6.69.5 Member Data Documentation

6.69.5.1 dist

```
int LDPair::dist
```

Definition at line 46 of file blox.cpp.

6.69.5.2 ld

```
double LDPair::ld
```

Definition at line 58 of file proxy.cpp.

6.69.5.3 s1

```
int LDPair::s1
```

Definition at line 44 of file blox.cpp.

6.69.5.4 s2

```
int LDPair::s2
```

Definition at line 45 of file blox.cpp.

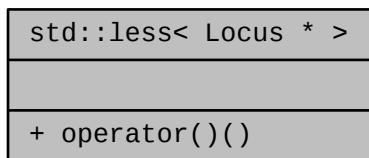
The documentation for this class was generated from the following files:

- [src/blox.cpp](#)
- [src/proxy.cpp](#)

6.70 std::less< Locus * > Class Template Reference

```
#include <plink.h>
```

Collaboration diagram for std::less< Locus * >:



Public Member Functions

- bool [operator\(\)](#) ([Locus](#) const *p1, [Locus](#) const *p2)

6.70.1 Detailed Description

```
template<>
class std::less< Locus * >
```

Definition at line 454 of file plink.h.

6.70.2 Member Function Documentation

6.70.2.1 operator()()

```
bool std::less< Locus * >::operator() (
    Locus const * p1,
    Locus const * p2 ) [inline]
```

Definition at line 457 of file plink.h.

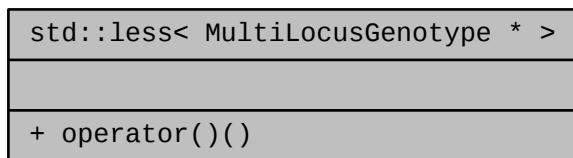
The documentation for this class was generated from the following file:

- [src/plink.h](#)

6.71 std::less< MultiLocusGenotype * > Class Template Reference

```
#include <genogroup.h>
```

Collaboration diagram for std::less< MultiLocusGenotype * >:



Public Member Functions

- bool `operator()` (`MultiLocusGenotype const *p1, MultiLocusGenotype const *p2`)

6.71.1 Detailed Description

```
template<>
class std::less< MultiLocusGenotype * >
```

Definition at line 42 of file genogroup.h.

6.71.2 Member Function Documentation

6.71.2.1 operator()()

```
bool std::less< MultiLocusGenotype * >::operator() ( 
    MultiLocusGenotype const * p1,
    MultiLocusGenotype const * p2 ) [inline]
```

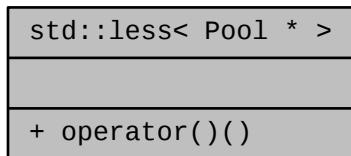
Definition at line 44 of file genogroup.h.

The documentation for this class was generated from the following file:

- [src/genogroup.h](#)

6.72 std::less< Pool * > Class Template Reference

Collaboration diagram for std::less< Pool * >:



Public Member Functions

- bool [operator\(\)](#) ([Pool](#) const *p1, [Pool](#) const *p2)

6.72.1 Detailed Description

```
template<>
class std::less< Pool * >
```

Definition at line 71 of file homozyg.cpp.

6.72.2 Member Function Documentation

6.72.2.1 operator()

```
bool std::less< Pool * >::operator() (
    Pool const * p1,
    Pool const * p2 ) [inline]
```

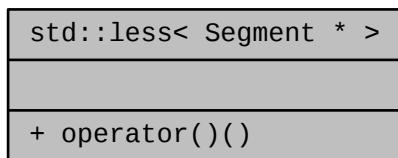
Definition at line 73 of file homozyg.cpp.

The documentation for this class was generated from the following file:

- [src/homozyg.cpp](#)

6.73 std::less< Segment * > Class Template Reference

Collaboration diagram for std::less< Segment * >:



Public Member Functions

- bool `operator()` (`Segment` const **s1*, `Segment` const **s2*)

6.73.1 Detailed Description

```
template<>
class std::less< Segment * >
```

Definition at line 34 of file homozyg.cpp.

6.73.2 Member Function Documentation

6.73.2.1 operator()

```
bool std::less< Segment * >::operator() (
    Segment const * s1,
    Segment const * s2 ) [inline]
```

Definition at line 36 of file homozyg.cpp.

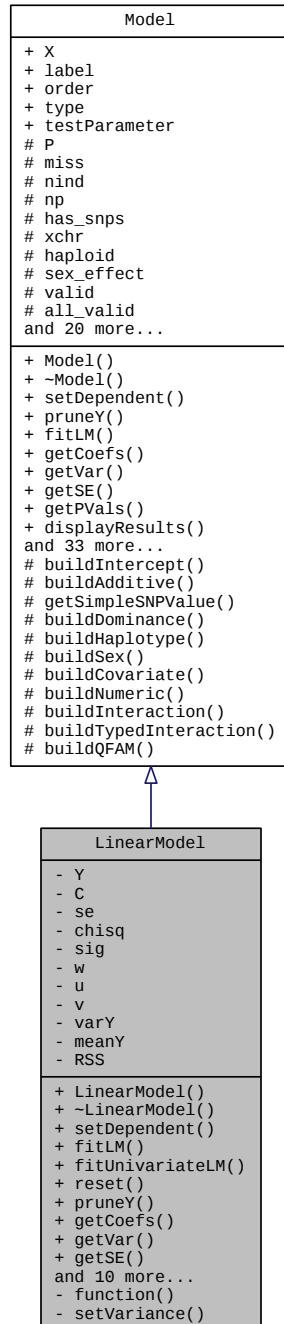
The documentation for this class was generated from the following file:

- `src/homozyg.cpp`

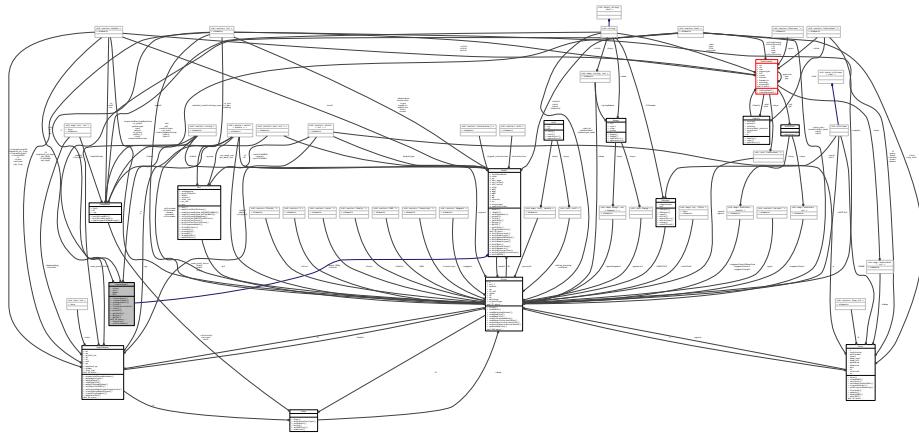
6.74 LinearModel Class Reference

```
#include <linear.h>
```

Inheritance diagram for LinearModel:



Collaboration diagram for LinearModel:



Public Member Functions

- `LinearModel (Plink *)`
- `~LinearModel ()`
- `void setDependent ()`
- `void fitLM ()`
- `void fitUnivariateLM ()`
- `void reset ()`
- `void pruneY ()`
- `vector_t getCoefs ()`
- `vector_t getVar ()`
- `vector_t getSE ()`
- `void displayResults (ofstream &, Locus *)`
- `vector_t getPVals ()`
- `double getPValue ()`
- `void HuberWhite ()`
- `void standardise ()`
- `double calculateRSS ()`
- `double calculateRSquared ()`
- `double calculateAdjustedRSquared ()`
- `double calculateMallowC (LinearModel *)`
- `double calculateFTest (LinearModel *)`

Private Member Functions

- `void function (const int i, vector< double > &p)`
- `void setVariance ()`

Private Attributes

- `vector_t Y`
- `vector< int > C`
- `vector< double > se`
- `double chisq`
- `vector< double > sig`
- `vector< double > w`
- `vector< vector< double > > u`
- `vector< vector< double > > v`
- `double varY`
- `double meanY`
- `double RSS`

Additional Inherited Members

6.74.1 Detailed Description

Definition at line 21 of file linear.h.

6.74.2 Constructor & Destructor Documentation

6.74.2.1 LinearModel()

```
LinearModel::LinearModel (   
    PLink * p_ )
```

Definition at line 20 of file linear.cpp.

6.74.2.2 ~LinearModel()

```
LinearModel::~LinearModel ( ) [inline]
```

Definition at line 24 of file linear.h.

6.74.3 Member Function Documentation

6.74.3.1 calculateAdjustedRSquared()

```
double LinearModel::calculateAdjustedRSquared ( )
```

Definition at line 668 of file linear.cpp.

6.74.3.2 calculateFTest()

```
double LinearModel::calculateFTest (  
    LinearModel * submodel )
```

Definition at line 683 of file linear.cpp.

6.74.3.3 calculateMallowC()

```
double LinearModel::calculateMallowC (   
    LinearModel * submodel )
```

Definition at line 675 of file linear.cpp.

6.74.3.4 calculateRSquared()

```
double LinearModel::calculateRSquared ( )
```

Definition at line 655 of file linear.cpp.

6.74.3.5 calculateRSS()

```
double LinearModel::calculateRSS ( )
```

Definition at line 635 of file linear.cpp.

6.74.3.6 displayResults()

```
void LinearModel::displayResults (
    ofstream & OUT,
    Locus * loc ) [virtual]
```

Implements [Model](#).

Definition at line 565 of file linear.cpp.

6.74.3.7 fitLM()

```
void LinearModel::fitLM ( ) [virtual]
```

Implements [Model](#).

Definition at line 241 of file linear.cpp.

6.74.3.8 fitUnivariateLM()

```
void LinearModel::fitUnivariateLM ( ) [virtual]
```

Implements [Model](#).

Definition at line 469 of file linear.cpp.

6.74.3.9 function()

```
void LinearModel::function (
    const int i,
    vector< double > & p ) [private]
```

6.74.3.10 getCoefs()

```
vector_t LinearModel::getCoefs ( ) [virtual]
```

Implements [Model](#).

Definition at line 518 of file linear.cpp.

6.74.3.11 getPVals()

```
vector_t LinearModel::getPVals ( ) [virtual]
```

Implements [Model](#).

Definition at line 705 of file linear.cpp.

6.74.3.12 getPValue()

```
double LinearModel::getPValue ( )
```

Definition at line 691 of file linear.cpp.

6.74.3.13 getSE()

```
vector_t LinearModel::getSE ( ) [virtual]
```

Implements [Model](#).

Definition at line 539 of file linear.cpp.

6.74.3.14 getVar()

```
vector_t LinearModel::getVar ( ) [virtual]
```

Implements [Model](#).

Definition at line 522 of file linear.cpp.

6.74.3.15 HuberWhite()

```
void LinearModel::HuberWhite ( ) [virtual]
```

Implements [Model](#).

Definition at line 716 of file linear.cpp.

6.74.3.16 pruneY()

```
void LinearModel::pruneY ( ) [virtual]
```

Implements [Model](#).

Definition at line 38 of file linear.cpp.

6.74.3.17 reset()

```
void LinearModel::reset ( )
```

Definition at line 554 of file linear.cpp.

6.74.3.18 setDependent()

```
void LinearModel::setDependent ( ) [virtual]
```

Implements [Model](#).

Definition at line 27 of file linear.cpp.

6.74.3.19 setVariance()

```
void LinearModel::setVariance ( ) [private]
```

Definition at line 173 of file linear.cpp.

6.74.3.20 standardise()

```
void LinearModel::standardise ( )
```

Definition at line 198 of file linear.cpp.

6.74.4 Member Data Documentation

6.74.4.1 C

```
vector<int> LinearModel::C [private]
```

Definition at line 49 of file linear.h.

6.74.4.2 chisq

```
double LinearModel::chisq [private]
```

Definition at line 51 of file linear.h.

6.74.4.3 meanY

```
double LinearModel::meanY [private]
```

Definition at line 59 of file linear.h.

6.74.4.4 RSS

```
double LinearModel::RSS [private]
```

Definition at line 61 of file linear.h.

6.74.4.5 se

```
vector<double> LinearModel::se [private]
```

Definition at line 50 of file linear.h.

6.74.4.6 sig

```
vector<double> LinearModel::sig [private]
```

Definition at line 53 of file linear.h.

6.74.4.7 u

```
vector<vector<double> > LinearModel::u [private]
```

Definition at line 55 of file linear.h.

6.74.4.8 v

```
vector<vector<double> > LinearModel::v [private]
```

Definition at line 56 of file linear.h.

6.74.4.9 varY

```
double LinearModel::varY [private]
```

Definition at line 58 of file linear.h.

6.74.4.10 w

```
vector<double> LinearModel::w [private]
```

Definition at line 54 of file linear.h.

6.74.4.11 Y

```
vector_t LinearModel::Y [private]
```

Definition at line 48 of file linear.h.

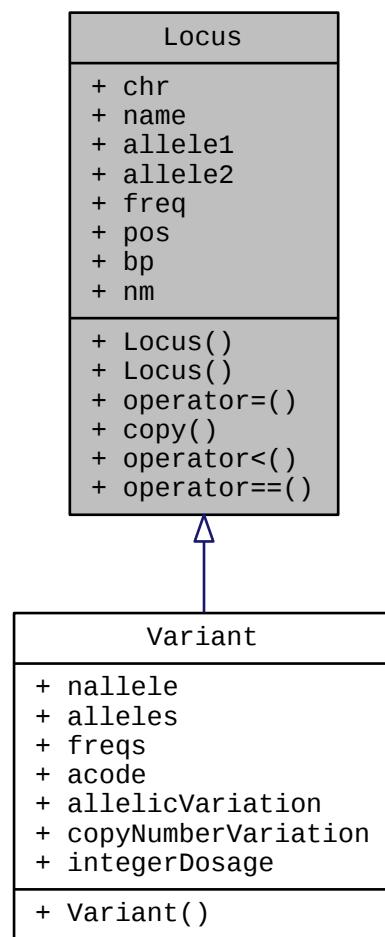
The documentation for this class was generated from the following files:

- src/[linear.h](#)
- src/[linear.cpp](#)

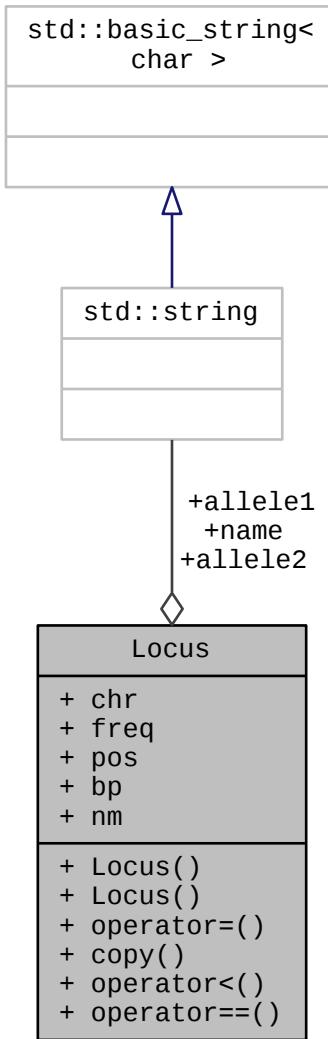
6.75 Locus Class Reference

```
#include <plink.h>
```

Inheritance diagram for Locus:



Collaboration diagram for Locus:



Public Member Functions

- `Locus ()`
- `Locus (const Locus &h1)`
- `Locus & operator= (const Locus &h1)`
- `void copy (const Locus &h1)`
- `bool operator< (const Locus &p2) const`
- `bool operator== (const Locus &p2) const`

Public Attributes

- int `chr`
- string `name`
- string `allele1`
- string `allele2`
- double `freq`
- double `pos`
- int `bp`
- int `nm`

6.75.1 Detailed Description

Definition at line 379 of file plink.h.

6.75.2 Constructor & Destructor Documentation

6.75.2.1 Locus() [1/2]

```
Locus::Locus ( ) [inline]
```

Definition at line 382 of file plink.h.

6.75.2.2 Locus() [2/2]

```
Locus::Locus ( const Locus & h1 ) [inline]
```

Definition at line 405 of file plink.h.

6.75.3 Member Function Documentation

6.75.3.1 copy()

```
void Locus::copy ( const Locus & h1 ) [inline]
```

Definition at line 414 of file plink.h.

6.75.3.2 operator<()

```
bool Locus::operator< (
    const Locus & p2 ) const [inline]
```

Definition at line 425 of file plink.h.

6.75.3.3 operator=()

```
Locus& Locus::operator= (
    const Locus & h1 ) [inline]
```

Definition at line 409 of file plink.h.

6.75.3.4 operator==()

```
bool Locus::operator== (
    const Locus & p2 ) const [inline]
```

Definition at line 429 of file plink.h.

6.75.4 Member Data Documentation

6.75.4.1 allele1

```
string Locus::allele1
```

Definition at line 395 of file plink.h.

6.75.4.2 allele2

```
string Locus::allele2
```

Definition at line 396 of file plink.h.

6.75.4.3 bp

```
int Locus::bp
```

Definition at line 400 of file plink.h.

6.75.4.4 chr

```
int Locus::chr
```

Definition at line 393 of file plink.h.

6.75.4.5 freq

```
double Locus::freq
```

Definition at line 398 of file plink.h.

6.75.4.6 name

```
string Locus::name
```

Definition at line 394 of file plink.h.

6.75.4.7 nm

```
int Locus::nm
```

Definition at line 401 of file plink.h.

6.75.4.8 pos

```
double Locus::pos
```

Definition at line 399 of file plink.h.

The documentation for this class was generated from the following file:

- src/[plink.h](#)

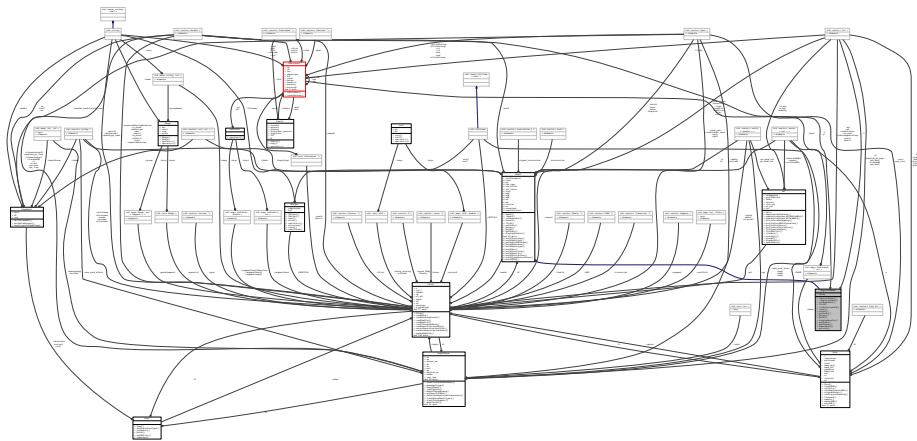
6.76 LogisticModel Class Reference

```
#include <logistic.h>
```

Inheritance diagram for LogisticModel:



Collaboration diagram for LogisticModel:



Public Member Functions

- `LogisticModel (Plink *)`
- `~LogisticModel ()`
- `void setDependent ()`
- `void fitLM ()`
- `void fitUnivariateLM ()`
- `void reset ()`
- `void pruneY ()`
- `vector_t getCoefs ()`
- `vector_t getVar ()`
- `vector_t getSE ()`
- `void displayResults (ofstream &, Locus *)`
- `vector_t getPVals ()`
- `double getPValue ()`
- `void HuberWhite ()`
- `double getLnLk ()`

Private Attributes

- `vector_t p`
- `vector< int > Y`
- `vector_t V`
- `double chisq`

Additional Inherited Members

6.76.1 Detailed Description

Definition at line 21 of file logistic.h.

6.76.2 Constructor & Destructor Documentation

6.76.2.1 LogisticModel()

```
LogisticModel::LogisticModel (
    Plink * p_ )
```

Definition at line 21 of file logistic.cpp.

6.76.2.2 ~LogisticModel()

```
LogisticModel::~LogisticModel ( ) [inline]
```

Definition at line 24 of file logistic.h.

6.76.3 Member Function Documentation

6.76.3.1 displayResults()

```
void LogisticModel::displayResults (
    ofstream & OUT,
    Locus * loc ) [virtual]
```

Implements [Model](#).

Definition at line 229 of file logistic.cpp.

6.76.3.2 fitLM()

```
void LogisticModel::fitLM ( ) [virtual]
```

Implements [Model](#).

Definition at line 60 of file logistic.cpp.

6.76.3.3 fitUnivariateLM()

```
void LogisticModel::fitUnivariateLM ( ) [inline], [virtual]
```

Implements [Model](#).

Definition at line 29 of file logistic.h.

6.76.3.4 getCoefs()

```
vector_t LogisticModel::getCoefs ( ) [virtual]
```

Implements [Model](#).

Definition at line 199 of file logistic.cpp.

6.76.3.5 getLnLk()

```
double LogisticModel::getLnLk ( )
```

Definition at line 329 of file logistic.cpp.

6.76.3.6 getPVals()

```
vector_t LogisticModel::getPVals ( ) [virtual]
```

Implements [Model](#).

Definition at line 319 of file logistic.cpp.

6.76.3.7 getPValue()

```
double LogisticModel::getPValue ( )
```

Definition at line 303 of file logistic.cpp.

6.76.3.8 getSE()

```
vector_t LogisticModel::getSE ( ) [virtual]
```

Implements [Model](#).

Definition at line 211 of file logistic.cpp.

6.76.3.9 getVar()

```
vector_t LogisticModel::getVar ( ) [virtual]
```

Implements [Model](#).

Definition at line 203 of file logistic.cpp.

6.76.3.10 HuberWhite()

```
void LogisticModel::HuberWhite ( ) [virtual]
```

Implements [Model](#).

Definition at line 343 of file logistic.cpp.

6.76.3.11 pruneY()

```
void LogisticModel::pruneY ( ) [virtual]
```

Implements [Model](#).

Definition at line 47 of file logistic.cpp.

6.76.3.12 reset()

```
void LogisticModel::reset ( )
```

Definition at line 219 of file logistic.cpp.

6.76.3.13 setDependent()

```
void LogisticModel::setDependent ( ) [virtual]
```

Implements [Model](#).

Definition at line 27 of file logistic.cpp.

6.76.4 Member Data Documentation

6.76.4.1 chisq

```
double LogisticModel::chisq [private]
```

Definition at line 46 of file logistic.h.

6.76.4.2 p

```
vector_t LogisticModel::p [private]
```

Definition at line 43 of file logistic.h.

6.76.4.3 V

```
vector_t LogisticModel::V [private]
```

Definition at line 45 of file logistic.h.

6.76.4.4 Y

```
vector<int> LogisticModel::Y [private]
```

Definition at line 44 of file logistic.h.

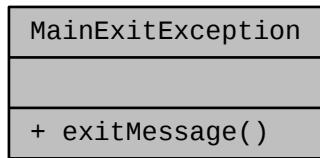
The documentation for this class was generated from the following files:

- [src/logistic.h](#)
- [src/logistic.cpp](#)

6.77 MainExitException Class Reference

```
#include <plink.h>
```

Collaboration diagram for MainExitException:



Public Member Functions

- void [exitMessage \(\)](#)

6.77.1 Detailed Description

Definition at line 484 of file plink.h.

6.77.2 Member Function Documentation

6.77.2.1 [exitMessage\(\)](#)

```
void MainExitException::exitMessage ( ) [inline]
```

Definition at line 486 of file plink.h.

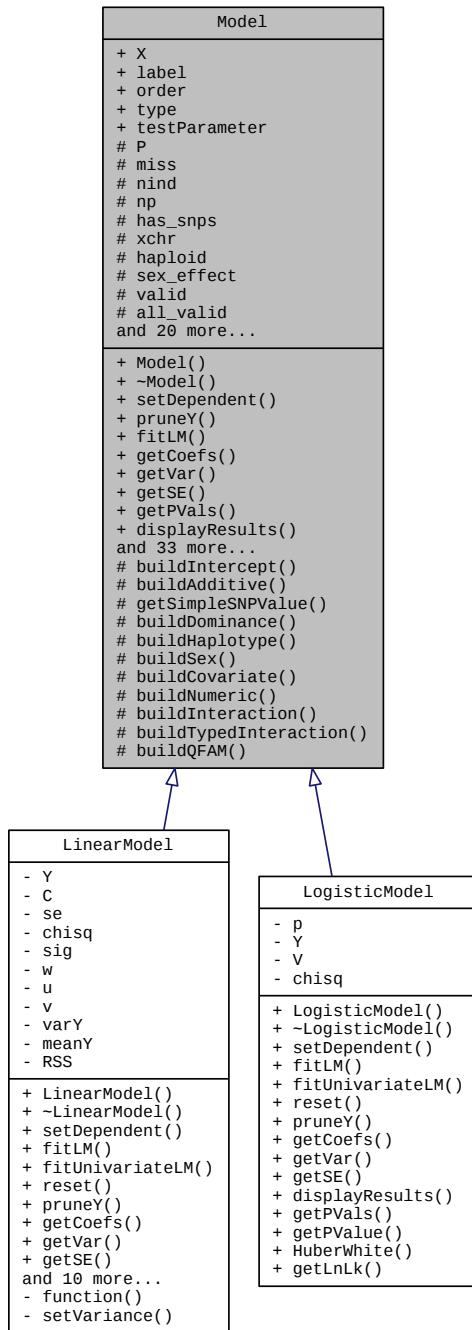
The documentation for this class was generated from the following file:

- [src/plink.h](#)

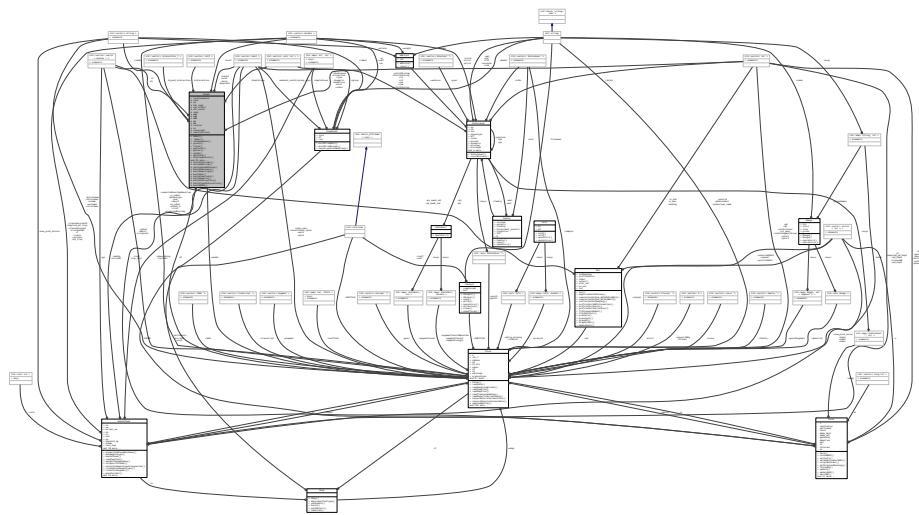
6.78 Model Class Reference

```
#include <model.h>
```

Inheritance diagram for Model:



Collaboration diagram for Model:



Public Member Functions

- `Model ()`
- `virtual ~Model ()`
- `virtual void setDependent ()=0`
- `virtual void pruneY ()=0`
- `virtual void fitLM ()=0`
- `virtual vector_t getCoefs ()=0`
- `virtual vector_t getVar ()=0`
- `virtual vector_t getSE ()=0`
- `virtual vector_t getPVals ()=0`
- `virtual void displayResults (ofstream &, Locus *)=0`
- `virtual void fitUnivariateLM ()=0`
- `void setMissing ()`
- `vector< bool > getMissing ()`
- `void setMissing (vector< bool > &)`
- `void yokeMissing (Model *)`
- `void setHaploid ()`
- `void setX ()`
- `void setDominant ()`
- `void setRecessive ()`
- `void hasSNPs (bool)`
- `void addAdditiveSNP (int)`
- `void addDominanceSNP (int)`
- `void addHaplotypeDosage (set< int > &)`
- `void addSexEffect ()`
- `bool isSexInModel ()`
- `void addCovariate (int)`
- `void addNumeric (int)`
- `void addInteraction (int, int)`

- void `addTypedInteraction` (int, ModelTermType, int, ModelTermType)
- void `buildDesignMatrix` ()
- bool `checkVIF` ()
- vector< bool > `validParameters` ()
- bool `isValid` ()
- double `getStatistic` ()
- double `linearHypothesis` (matrix_t &, vector_t &)
- int `Ysize` ()
- int `getNP` ()
- void `setValid` ()
- void `noCluster` ()
- void `setCluster` ()
- virtual void `HuberWhite` ()=0
- bool `fitConverged` ()
- int `fitNumIterations` ()

Public Attributes

- vector< vector< double > > `X`
- vector< string > `label`
- vector< int > `order`
- vector< int > `type`
- int `testParameter`

Protected Member Functions

- double `buildIntercept` ()
- double `buildAdditive` (Individual *, int)
- double `getSimpleSNPValue` (Individual *person, int snp)
- double `buildDominance` (Individual *, int)
- double `buildHaplotype` (int, int)
- double `buildSex` (Individual *)
- double `buildCovariate` (Individual *, int)
- double `buildNumeric` (Individual *, int)
- double `buildInteraction` (Individual *, int, vector_t &)
- double `buildTypedInteraction` (Individual *, int)
- double `buildQFAM` (Individual *)

Protected Attributes

- Plink * `P`
- vector< bool > `miss`
- int `nind`
- int `np`
- bool `has_snps`
- vector< bool > `xchr`
- vector< bool > `haploid`
- bool `sex_effect`

- `vector< bool > valid`
- `bool all_valid`
- `vector_t coef`
- `matrix_t S`
- `bool skip`
- `vector< int > additive`
- `int mAA`
- `int mAB`
- `int mBB`
- `double mA`
- `double mB`
- `vector< int > dominance`
- `vector< int > covariate`
- `vector< int > numeric`
- `vector< int2 > interaction`
- `vector< interaction_t > typed_interaction`
- `vector< set< int > > haplotype`
- `bool cluster`
- `vector< int > clst`
- `int nc`
- `bool converged`
- `int numIterations`

6.78.1 Detailed Description

Definition at line 20 of file model.h.

6.78.2 Constructor & Destructor Documentation

6.78.2.1 Model()

```
Model::Model ( )
```

Definition at line 19 of file model.cpp.

6.78.2.2 ~Model()

```
virtual Model::~Model ( ) [inline], [virtual]
```

Definition at line 24 of file model.h.

6.78.3 Member Function Documentation

6.78.3.1 addAdditiveSNP()

```
void Model::addAdditiveSNP (
    int a )
```

Definition at line 124 of file model.cpp.

6.78.3.2 addCovariate()

```
void Model::addCovariate (
    int c )
```

Definition at line 155 of file model.cpp.

6.78.3.3 addDominanceSNP()

```
void Model::addDominanceSNP (
    int d )
```

Definition at line 146 of file model.cpp.

6.78.3.4 addHaplotypeDosage()

```
void Model::addHaplotypeDosage (
    set< int > & h )
```

Definition at line 168 of file model.cpp.

6.78.3.5 addInteraction()

```
void Model::addInteraction (
    int a,
    int b )
```

Definition at line 174 of file model.cpp.

6.78.3.6 addNumeric()

```
void Model::addNumeric (
    int n )
```

Definition at line 161 of file model.cpp.

6.78.3.7 addSexEffect()

```
void Model::addSexEffect ( )
```

Definition at line 75 of file model.cpp.

6.78.3.8 addTypedInteraction()

```
void Model::addTypedInteraction (
    int a,
    ModelTermType typeA,
    int b,
    ModelTermType typeB )
```

Definition at line 183 of file model.cpp.

6.78.3.9 buildAdditive()

```
double Model::buildAdditive (
    Individual * person,
    int snp ) [protected]
```

Definition at line 568 of file model.cpp.

6.78.3.10 buildCovariate()

```
double Model::buildCovariate (
    Individual * person,
    int j ) [protected]
```

Definition at line 717 of file model.cpp.

6.78.3.11 buildDesignMatrix()

```
void Model::buildDesignMatrix ( )
```

Definition at line 192 of file model.cpp.

6.78.3.12 buildDominance()

```
double Model::buildDominance (
    Individual * person,
    int snp ) [protected]
```

Definition at line 671 of file model.cpp.

6.78.3.13 buildHaplotype()

```
double Model::buildHaplotype (
    int i,
    int h ) [protected]
```

Definition at line 693 of file model.cpp.

6.78.3.14 buildInteraction()

```
double Model::buildInteraction (
    Individual * person,
    int j,
    vector_t & trow ) [protected]
```

Definition at line 731 of file model.cpp.

6.78.3.15 buildIntercept()

```
double Model::buildIntercept ( ) [protected]
```

Definition at line 564 of file model.cpp.

6.78.3.16 buildNumeric()

```
double Model::buildNumeric (
    Individual * person,
    int j ) [protected]
```

Definition at line 723 of file model.cpp.

6.78.3.17 buildQFAM()

```
double Model::buildQFAM (
    Individual * person ) [protected]
```

Definition at line 780 of file model.cpp.

6.78.3.18 buildSex()

```
double Model::buildSex (
    Individual * person ) [protected]
```

Definition at line 705 of file model.cpp.

6.78.3.19 buildTypedInteraction()

```
double Model::buildTypedInteraction (
    Individual * person,
    int j ) [protected]
```

Definition at line 737 of file model.cpp.

6.78.3.20 checkVIF()

```
bool Model::checkVIF ( )
```

Definition at line 488 of file model.cpp.

6.78.3.21 `displayResults()`

```
virtual void Model::displayResults (
    ofstream & ,
    Locus * ) [pure virtual]
```

Implemented in [LinearModel](#), and [LogisticModel](#).

6.78.3.22 `fitConverged()`

```
bool Model::fitConverged ( )
```

Definition at line 824 of file model.cpp.

6.78.3.23 `fitLM()`

```
virtual void Model::fitLM ( ) [pure virtual]
```

Implemented in [LinearModel](#), and [LogisticModel](#).

6.78.3.24 `fitNumIterations()`

```
int Model::fitNumIterations ( )
```

Definition at line 828 of file model.cpp.

6.78.3.25 `fitUnivariateLM()`

```
virtual void Model::fitUnivariateLM ( ) [pure virtual]
```

Implemented in [LinearModel](#), and [LogisticModel](#).

6.78.3.26 getCoefs()

```
virtual vector_t Model::getCoefs ( ) [pure virtual]
```

Implemented in [LinearModel](#), and [LogisticModel](#).

6.78.3.27 getMissing()

```
vector< bool > Model::getMissing ( )
```

Definition at line 116 of file model.cpp.

6.78.3.28 getNP()

```
int Model::getNP ( ) [inline]
```

Definition at line 69 of file model.h.

6.78.3.29 getPVals()

```
virtual vector_t Model::getPVals ( ) [pure virtual]
```

Implemented in [LinearModel](#), and [LogisticModel](#).

6.78.3.30 getSE()

```
virtual vector_t Model::getSE ( ) [pure virtual]
```

Implemented in [LinearModel](#), and [LogisticModel](#).

6.78.3.31 getSimpleSNPValue()

```
double Model::getSimpleSNPValue (
    Individual * person,
    int snp ) [protected]
```

Definition at line 649 of file model.cpp.

6.78.3.32 getStatistic()

```
double Model::getStatistic ( )
```

Definition at line 414 of file model.cpp.

6.78.3.33 getVar()

```
virtual vector_t Model::getVar ( ) [pure virtual]
```

Implemented in [LinearModel](#), and [LogisticModel](#).

6.78.3.34 hasSNPs()

```
void Model::hasSNPs (   
    bool b )
```

Definition at line 85 of file model.cpp.

6.78.3.35 HuberWhite()

```
virtual void Model::HuberWhite ( ) [pure virtual]
```

Implemented in [LinearModel](#), and [LogisticModel](#).

6.78.3.36 isSexInModel()

```
bool Model::isSexInModel ( )
```

Definition at line 81 of file model.cpp.

6.78.3.37 isValid()

```
bool Model::isValid ( ) [inline]
```

Definition at line 58 of file model.h.

6.78.3.38 linearHypothesis()

```
double Model::linearHypothesis (
    matrix_t & H,
    vector_t & h )
```

Definition at line 429 of file model.cpp.

6.78.3.39 noCluster()

```
void Model::noCluster ( )
```

Definition at line 818 of file model.cpp.

6.78.3.40 pruneY()

```
virtual void Model::pruneY ( ) [pure virtual]
```

Implemented in [LinearModel](#), and [LogisticModel](#).

6.78.3.41 setCluster()

```
void Model::setCluster ( )
```

Definition at line 793 of file model.cpp.

6.78.3.42 setDependent()

```
virtual void Model::setDependent ( ) [pure virtual]
```

Implemented in [LinearModel](#), and [LogisticModel](#).

6.78.3.43 setDominant()

```
void Model::setDominant ( )
```

Definition at line 58 of file model.cpp.

6.78.3.44 setHaploid()

```
void Model::setHaploid ( )
```

6.78.3.45 setMissing() [1/2]

```
void Model::setMissing ( )
```

Definition at line 89 of file model.cpp.

6.78.3.46 setMissing() [2/2]

```
void Model::setMissing (
    vector< bool > & include )
```

Definition at line 101 of file model.cpp.

6.78.3.47 setRecessive()

```
void Model::setRecessive ( )
```

Definition at line 66 of file model.cpp.

6.78.3.48 setValid()

```
void Model::setValid ( ) [inline]
```

Definition at line 73 of file model.h.

6.78.3.49 setX()

```
void Model::setX ( )
```

6.78.3.50 validParameters()

```
vector< bool > Model::validParameters ( )
```

Definition at line 371 of file model.cpp.

6.78.3.51 yokeMissing()

```
void Model::yokeMissing (   
    Model * m )
```

Definition at line 120 of file model.cpp.

6.78.3.52 Ysize()

```
int Model::Ysize ( ) [inline]
```

Definition at line 65 of file model.h.

6.78.4 Member Data Documentation**6.78.4.1 additive**

```
vector<int> Model::additive [protected]
```

Definition at line 141 of file model.h.

6.78.4.2 all_valid

```
bool Model::all_valid [protected]
```

Definition at line 118 of file model.h.

6.78.4.3 clst

```
vector<int> Model::clst [protected]
```

Definition at line 170 of file model.h.

6.78.4.4 cluster

```
bool Model::cluster [protected]
```

Definition at line 169 of file model.h.

6.78.4.5 coef

```
vector_t Model::coef [protected]
```

Definition at line 121 of file model.h.

6.78.4.6 converged

```
bool Model::converged [protected]
```

Definition at line 174 of file model.h.

6.78.4.7 covariate

```
vector<int> Model::covariate [protected]
```

Definition at line 153 of file model.h.

6.78.4.8 dominance

```
vector<int> Model::dominance [protected]
```

Definition at line 150 of file model.h.

6.78.4.9 haploid

```
vector<bool> Model::haploid [protected]
```

Definition at line 113 of file model.h.

6.78.4.10 haplotype

```
vector<set<int>> Model::haplotype [protected]
```

Definition at line 166 of file model.h.

6.78.4.11 has_snps

```
bool Model::has_snps [protected]
```

Definition at line 110 of file model.h.

6.78.4.12 interaction

```
vector<int2> Model::interaction [protected]
```

Definition at line 160 of file model.h.

6.78.4.13 label

```
vector<string> Model::label
```

Definition at line 93 of file model.h.

6.78.4.14 mA

```
double Model::mA [protected]
```

Definition at line 147 of file model.h.

6.78.4.15 mAA

```
int Model::mAA [protected]
```

Definition at line 143 of file model.h.

6.78.4.16 mAB

```
int Model::mAB [protected]
```

Definition at line 144 of file model.h.

6.78.4.17 mB

```
double Model::mB [protected]
```

Definition at line 147 of file model.h.

6.78.4.18 mBB

```
int Model::mBB [protected]
```

Definition at line 145 of file model.h.

6.78.4.19 miss

```
vector<bool> Model::miss [protected]
```

Definition at line 103 of file model.h.

6.78.4.20 nc

```
int Model::nc [protected]
```

Definition at line 171 of file model.h.

6.78.4.21 nind

```
int Model::nind [protected]
```

Definition at line 106 of file model.h.

6.78.4.22 np

```
int Model::np [protected]
```

Definition at line 108 of file model.h.

6.78.4.23 numeric

```
vector<int> Model::numeric [protected]
```

Definition at line 156 of file model.h.

6.78.4.24 numIterations

```
int Model::numIterations [protected]
```

Definition at line 175 of file model.h.

6.78.4.25 order

```
vector<int> Model::order
```

Definition at line 94 of file model.h.

6.78.4.26 P

`Plink*` `Model::P` [protected]

Definition at line 100 of file model.h.

6.78.4.27 S

`matrix_t` `Model::S` [protected]

Definition at line 123 of file model.h.

6.78.4.28 sex_effect

`bool Model::sex_effect` [protected]

Definition at line 115 of file model.h.

6.78.4.29 skip

`bool Model::skip` [protected]

Definition at line 137 of file model.h.

6.78.4.30 testParameter

`int Model::testParameter`

Definition at line 98 of file model.h.

6.78.4.31 type

`vector<int> Model::type`

Definition at line 95 of file model.h.

6.78.4.32 typed_interaction

```
vector<interaction\_t> Model::typed_interaction [protected]
```

Definition at line 163 of file model.h.

6.78.4.33 valid

```
vector<bool> Model::valid [protected]
```

Definition at line 117 of file model.h.

6.78.4.34 X

```
vector<vector<double>> Model::X
```

Definition at line 90 of file model.h.

6.78.4.35 xchr

```
vector<bool> Model::xchr [protected]
```

Definition at line 112 of file model.h.

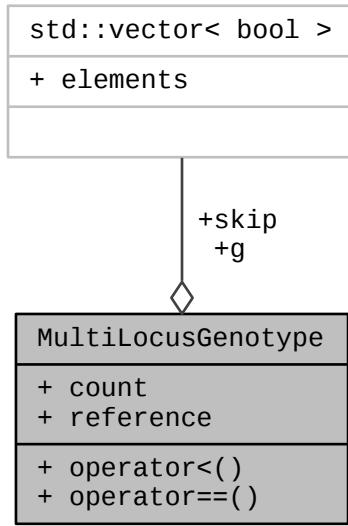
The documentation for this class was generated from the following files:

- [src/model.h](#)
- [src/model.cpp](#)

6.79 MultiLocusGenotype Class Reference

```
#include <genogroup.h>
```

Collaboration diagram for MultiLocusGenotype:



Public Member Functions

- bool `operator<` (const `MultiLocusGenotype` &`b`) const
- bool `operator==` (const `MultiLocusGenotype` &`b`) const

Public Attributes

- `vector< bool > g`
- int `count`
- int `reference`
- `vector< bool > skip`

6.79.1 Detailed Description

Definition at line 18 of file genogroup.h.

6.79.2 Member Function Documentation

6.79.2.1 operator<()

```
bool MultiLocusGenotype::operator< (
    const MultiLocusGenotype & b ) const [inline]
```

Definition at line 26 of file genogroup.h.

6.79.2.2 operator==()

```
bool MultiLocusGenotype::operator== (
    const MultiLocusGenotype & b ) const [inline]
```

Definition at line 33 of file genogroup.h.

6.79.3 Member Data Documentation

6.79.3.1 count

```
int MultiLocusGenotype::count
```

Definition at line 22 of file genogroup.h.

6.79.3.2 g

```
vector<bool> MultiLocusGenotype::g
```

Definition at line 21 of file genogroup.h.

6.79.3.3 reference

```
int MultiLocusGenotype::reference
```

Definition at line 23 of file genogroup.h.

6.79.3.4 skip

```
vector<bool> MultiLocusGenotype::skip
```

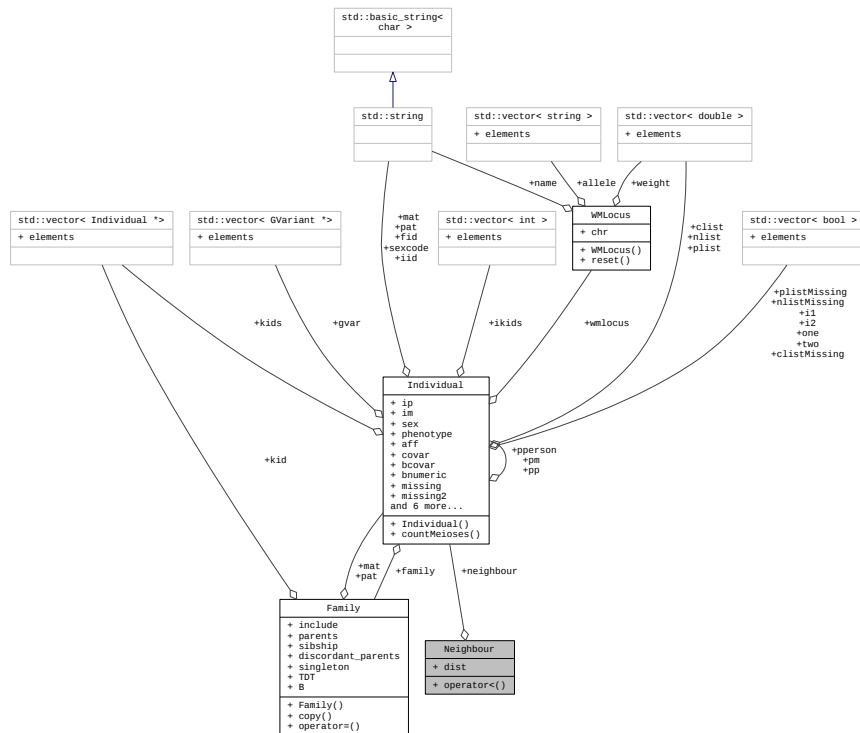
Definition at line 24 of file genogroup.h.

The documentation for this class was generated from the following file:

- [src/genogroup.h](#)

6.80 Neighbour Class Reference

Collaboration diagram for Neighbour:



Public Member Functions

- bool `operator<` (const `Neighbour` &`s2`) const

Public Attributes

- double `dist`
- `Individual` * `neighbour`

6.80.1 Detailed Description

Definition at line 51 of file cluster.cpp.

6.80.2 Member Function Documentation

6.80.2.1 `operator<()`

```
bool Neighbour::operator< (
    const Neighbour & s2 ) const [inline]
```

Definition at line 56 of file cluster.cpp.

6.80.3 Member Data Documentation

6.80.3.1 `dist`

```
double Neighbour::dist
```

Definition at line 53 of file cluster.cpp.

6.80.3.2 `neighbour`

```
Individual* Neighbour::neighbour
```

Definition at line 54 of file cluster.cpp.

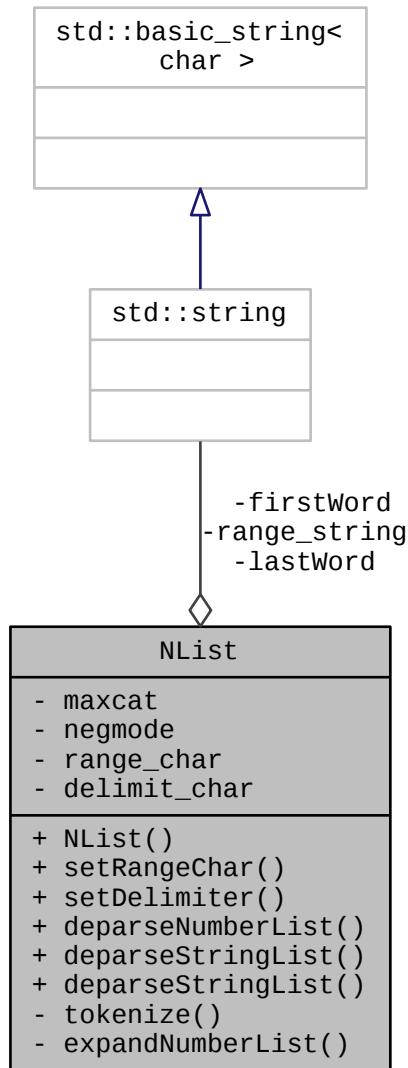
The documentation for this class was generated from the following file:

- `src/cluster.cpp`

6.81 NList Class Reference

```
#include <nlist.h>
```

Collaboration diagram for NList:



Public Member Functions

- [NList](#) (int n, bool nmode=true)
- void [setRangeChar](#) (string s)

- void `setDelimiter` (string s)
- vector< int > `deparseNumberList` (string)
- vector< int > `deparseStringList` (string, map< string, int > *)
- vector< string > `deparseStringList` (string)

Private Member Functions

- vector< string > `tokenize` (string)
- vector< int > `expandNumberList` (vector< int > &)

Private Attributes

- int `maxcat`
- bool `negmode`
- string `firstWord`
- string `lastWord`
- char `range_char`
- string `range_string`
- char `delimit_char`

6.81.1 Detailed Description

Definition at line 23 of file nlist.h.

6.81.2 Constructor & Destructor Documentation

6.81.2.1 NList()

```
NList::NList (
    int n,
    bool nmode = true ) [inline]
```

Definition at line 39 of file nlist.h.

6.81.3 Member Function Documentation

6.81.3.1 deparseNumberList()

```
vector< int > NList::deparseNumberList (
    string input )
```

Definition at line 21 of file nlist.cpp.

6.81.3.2 deparseStringList() [1/2]

```
vector< int > NList::deparseStringList (
    string input,
    map< string, int > * mapping )
```

Definition at line 47 of file nlist.cpp.

6.81.3.3 deparseStringList() [2/2]

```
vector< string > NList::deparseStringList (
    string input )
```

Definition at line 17 of file nlist.cpp.

6.81.3.4 expandNumberList()

```
vector< int > NList::expandNumberList (
    vector< int > & nlist ) [private]
```

Definition at line 85 of file nlist.cpp.

6.81.3.5 setDelimiter()

```
void NList::setDelimiter (
    string s ) [inline]
```

Definition at line 52 of file nlist.h.

6.81.3.6 setRangeChar()

```
void NList::setRangeChar (
    string s ) [inline]
```

Definition at line 47 of file nlist.h.

6.81.3.7 tokenize()

```
vector< string > NList::tokenize (
    string s ) [private]
```

Definition at line 151 of file nlist.cpp.

6.81.4 Member Data Documentation

6.81.4.1 delimit_char

```
char NList::delimit_char [private]
```

Definition at line 35 of file nlist.h.

6.81.4.2 firstWord

```
string NList::firstWord [private]
```

Definition at line 29 of file nlist.h.

6.81.4.3 lastWord

```
string NList::lastWord [private]
```

Definition at line 30 of file nlist.h.

6.81.4.4 maxcat

```
int NList::maxcat [private]
```

Definition at line 26 of file nlist.h.

6.81.4.5 negmode

```
bool NList::negmode [private]
```

Definition at line 27 of file nlist.h.

6.81.4.6 range_char

```
char NList::range_char [private]
```

Definition at line 32 of file nlist.h.

6.81.4.7 range_string

```
string NList::range_string [private]
```

Definition at line 33 of file nlist.h.

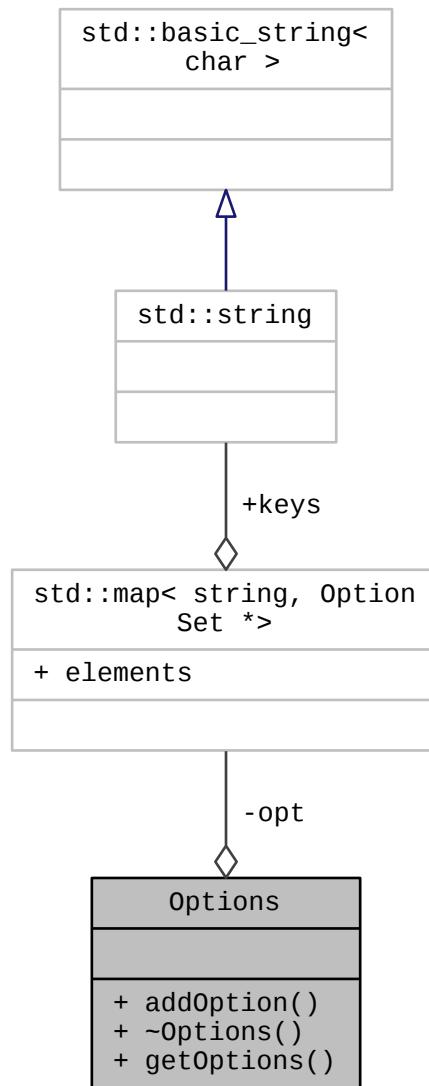
The documentation for this class was generated from the following files:

- src/nlist.h
- src/nlist.cpp

6.82 Options Class Reference

```
#include <options.h>
```

Collaboration diagram for Options:



Public Member Functions

- `OptionSet * addOption (string s)`
- `~Options ()`
- `OptionSet * getOptions (string s)`

Private Attributes

- map< string, OptionSet * > opt

6.82.1 Detailed Description

Definition at line 73 of file options.h.

6.82.2 Constructor & Destructor Documentation

6.82.2.1 ~Options()

```
Options::~Options ( ) [inline]
```

Definition at line 87 of file options.h.

6.82.3 Member Function Documentation

6.82.3.1 addOption()

```
OptionSet* Options::addOption (
    string s ) [inline]
```

Definition at line 78 of file options.h.

6.82.3.2 getOptions()

```
OptionSet* Options::getOptions (
    string s ) [inline]
```

Definition at line 95 of file options.h.

6.82.4 Member Data Documentation

6.82.4.1 opt

```
map<string, OptionSet*> Options::opt [private]
```

Definition at line 74 of file options.h.

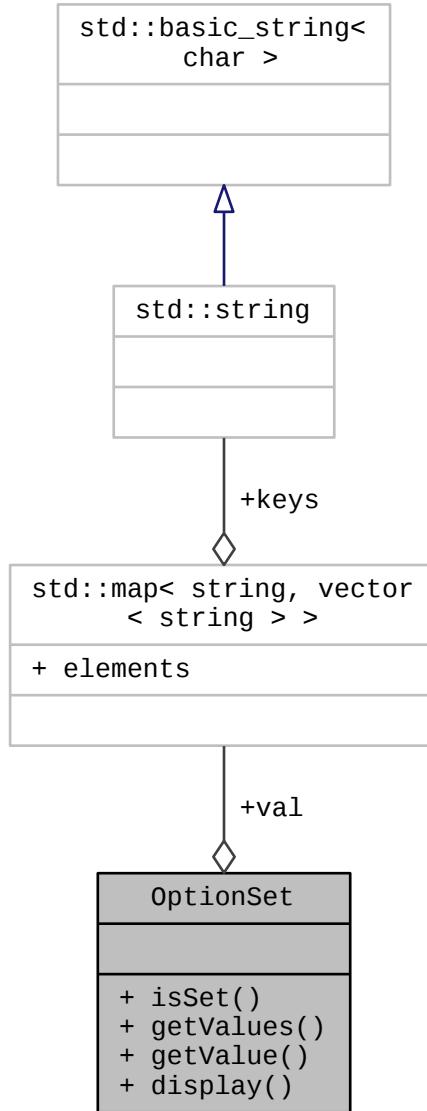
The documentation for this class was generated from the following file:

- [src/options.h](#)

6.83 OptionSet Class Reference

```
#include <options.h>
```

Collaboration diagram for OptionSet:



Public Member Functions

- bool `isSet` (string s)
- vector< string > `getValues` (string s)
- string `getValue` (string s)
- void `display` ()

Public Attributes

- map< string, vector< string > > **val**

6.83.1 Detailed Description

Definition at line 29 of file options.h.

6.83.2 Member Function Documentation

6.83.2.1 display()

```
void OptionSet::display ( ) [inline]
```

Definition at line 57 of file options.h.

6.83.2.2 getValue()

```
string OptionSet::getValue (
    string s ) [inline]
```

Definition at line 46 of file options.h.

6.83.2.3 getValues()

```
vector<string> OptionSet::getValues (
    string s ) [inline]
```

Definition at line 38 of file options.h.

6.83.2.4 isSet()

```
bool OptionSet::isSet (
    string s ) [inline]
```

Definition at line 34 of file options.h.

6.83.3 Member Data Documentation

6.83.3.1 val

```
map<string, vector<string> > OptionSet::val
```

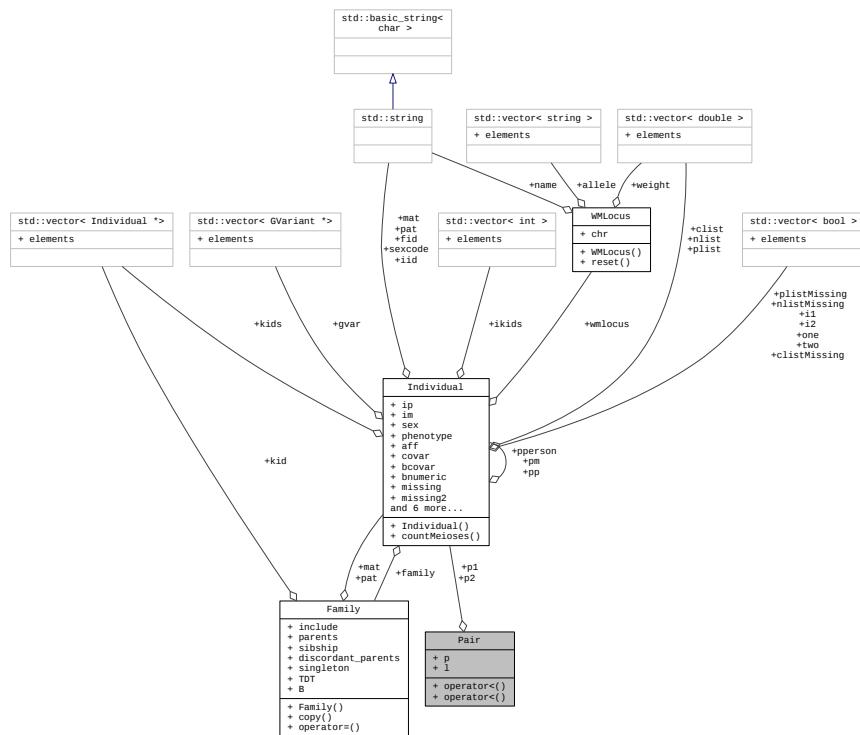
Definition at line 32 of file options.h.

The documentation for this class was generated from the following file:

- [src/options.h](#)

6.84 Pair Class Reference

Collaboration diagram for Pair:



Public Member Functions

- `bool operator< (const Pair &b) const`
- `bool operator< (const Pair &p2) const`

Public Attributes

- `Individual * p1`
- `Individual * p2`
- `double p`
- `int l`

6.84.1 Detailed Description

Definition at line 21 of file informative.cpp.

6.84.2 Member Function Documentation

6.84.2.1 `operator<()` [1/2]

```
bool Pair::operator< (
    const Pair & b ) const [inline]
```

Definition at line 26 of file informative.cpp.

6.84.2.2 `operator<()` [2/2]

```
bool Pair::operator< (
    const Pair & p2 ) const [inline]
```

Definition at line 44 of file multiple.cpp.

6.84.3 Member Data Documentation

6.84.3.1 `l`

```
int Pair::l
```

Definition at line 42 of file multiple.cpp.

6.84.3.2 p

```
double Pair::p
```

Definition at line 41 of file multiple.cpp.

6.84.3.3 p1

```
Individual* Pair::p1
```

Definition at line 23 of file informative.cpp.

6.84.3.4 p2

```
Individual* Pair::p2
```

Definition at line 24 of file informative.cpp.

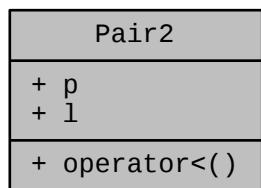
The documentation for this class was generated from the following files:

- src/informative.cpp
- src/multiple.cpp

6.85 Pair2 Class Reference

```
#include <plink.h>
```

Collaboration diagram for Pair2:



Public Member Functions

- bool `operator<` (const `Pair2` &`p2`) const

Public Attributes

- double `p`
- int `l`

6.85.1 Detailed Description

Definition at line 140 of file plink.h.

6.85.2 Member Function Documentation

6.85.2.1 `operator<()`

```
bool Pair2::operator< (
    const Pair2 & p2 ) const [inline]
```

Definition at line 146 of file plink.h.

6.85.3 Member Data Documentation

6.85.3.1 `l`

```
int Pair2::l
```

Definition at line 144 of file plink.h.

6.85.3.2 `p`

```
double Pair2::p
```

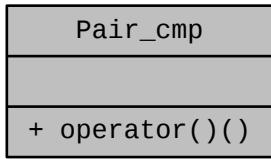
Definition at line 143 of file plink.h.

The documentation for this class was generated from the following file:

- `src/plink.h`

6.86 Pair_cmp Struct Reference

Collaboration diagram for Pair_cmp:



Public Member Functions

- bool [operator\(\)](#) (const LDPair &a, const LDPair &b) const

6.86.1 Detailed Description

Definition at line 61 of file blox.cpp.

6.86.2 Member Function Documentation

6.86.2.1 operator()()

```
bool Pair_cmp::operator() (
    const LDPair & a,
    const LDPair & b ) const [inline]
```

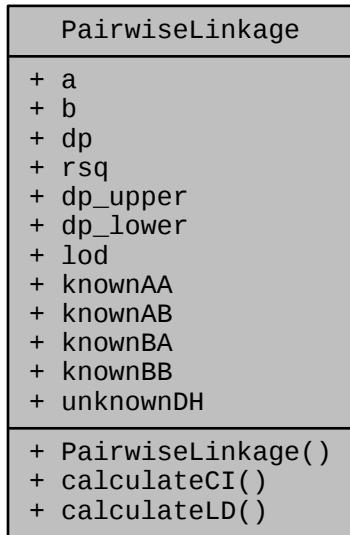
Definition at line 63 of file blox.cpp.

The documentation for this struct was generated from the following file:

- [src/blox.cpp](#)

6.87 PairwiseLinkage Class Reference

Collaboration diagram for PairwiseLinkage:



Public Member Functions

- [PairwiseLinkage \(int a_, int b_\)](#)
- void [calculateCI \(\)](#)
- void [calculateLD \(\)](#)

Public Attributes

- int [a](#)
- int [b](#)
- double [dp](#)
- double [rsq](#)
- double [dp_upper](#)
- double [dp_lower](#)
- double [lod](#)
- int [knownAA](#)
- int [knownAB](#)
- int [knownBA](#)
- int [knownBB](#)
- int [unknownDH](#)

6.87.1 Detailed Description

Definition at line 83 of file blox.cpp.

6.87.2 Constructor & Destructor Documentation

6.87.2.1 PairwiseLinkage()

```
PairwiseLinkage::PairwiseLinkage (
    int a__,
    int b__ ) [inline]
```

Definition at line 86 of file blox.cpp.

6.87.3 Member Function Documentation

6.87.3.1 calculateCI()

```
void PairwiseLinkage::calculateCI ( )
```

Definition at line 454 of file blox.cpp.

6.87.3.2 calculateLD()

```
void PairwiseLinkage::calculateLD ( )
```

Definition at line 631 of file blox.cpp.

6.87.4 Member Data Documentation

6.87.4.1 a

```
int PairwiseLinkage::a
```

Definition at line 93 of file blox.cpp.

6.87.4.2 b

```
int PairwiseLinkage::b
```

Definition at line 94 of file blox.cpp.

6.87.4.3 dp

```
double PairwiseLinkage::dp
```

Definition at line 95 of file blox.cpp.

6.87.4.4 dp_lower

```
double PairwiseLinkage::dp_lower
```

Definition at line 96 of file blox.cpp.

6.87.4.5 dp_upper

```
double PairwiseLinkage::dp_upper
```

Definition at line 96 of file blox.cpp.

6.87.4.6 knownAA

```
int PairwiseLinkage::knownAA
```

Definition at line 100 of file blox.cpp.

6.87.4.7 knownAB

```
int PairwiseLinkage::knownAB
```

Definition at line 100 of file blox.cpp.

6.87.4.8 knownBA

```
int PairwiseLinkage::knownBA
```

Definition at line 100 of file blox.cpp.

6.87.4.9 knownBB

```
int PairwiseLinkage::knownBB
```

Definition at line 100 of file blox.cpp.

6.87.4.10 lod

```
double PairwiseLinkage::lod
```

Definition at line 97 of file blox.cpp.

6.87.4.11 rsq

```
double PairwiseLinkage::rsq
```

Definition at line 95 of file blox.cpp.

6.87.4.12 unknownDH

```
int PairwiseLinkage::unknownDH
```

Definition at line 100 of file blox.cpp.

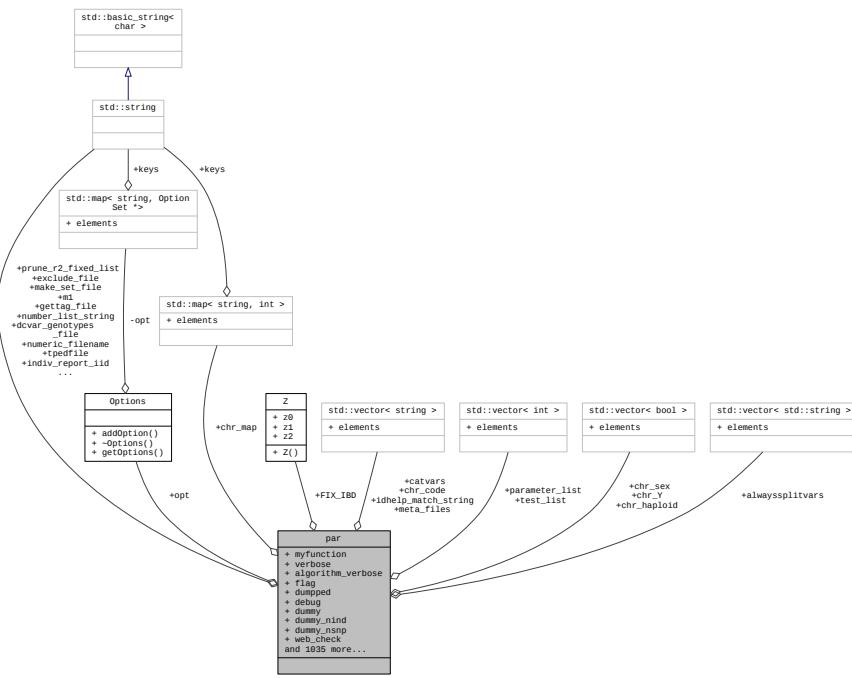
The documentation for this class was generated from the following file:

- [src/blox.cpp](#)

6.88 par Class Reference

```
#include <options.h>
```

Collaboration diagram for par:



Static Public Attributes

- static bool `myfunction` = false
- static Options `opt`
- static bool `verbose` = false
- static bool `algorithm_verbose` = false
- static bool `flag` = false
- static bool `dumpped` = false
- static bool `debug` = false
- static bool `dummy` = false
- static int `dummy_nind` = 0
- static int `dummy_nsnp` = 0
- static bool `web_check` = true
- static bool `tucc` = false
- static bool `do_not_load_snps` = false
- static const int `logistic_model_iterations` = 25
- static const double `logistic_tolerance_epsilon` = 1e-8
- static const double `epsilon` = 1e-12
- static long unsigned int `random_seed` = 0
- static int `simul_ncases` = 1000

- static int `simul_ncontrols` = 1000
- static string `simul_label` = ""
- static double `simul_prevalence` = 0.01
- static bool `simul` = false
- static string `simul_file` = ""
- static bool `simul_tags` = false
- static bool `simul_haps` = false
- static bool `simul_qt` = false
- static double `simul_qt_var` = 0.05
- static bool `lookup` = false
- static bool `lookup_single_snp` = false
- static bool `lookup_to_file` = false
- static string `lookup.snp` = "rs1234"
- static string `lookup_gene_name` = "GENE1"
- static bool `lookup_gene` = false
- static bool `lookup_multiple_genes` = false
- static int `lookup_gene_kb_window` = 20
- static int `lookup.snp_kb_window` = 100
- static bool `lookup2` = false
- static string `lookup2_cmd` = ""
- static bool `idhelp` = false
- static string `idhelp_output_delimit` = " "
- static string `idhelp_dictionary` = ""
- static bool `idhelp_dump_from_dict` = false
- static string `idhelp_dump_from_dict_cmd` = ""
- static bool `idhelp_auto_alias` = false
- static bool `idhelp_lookup` = false
- static string `idhelp_lookup_string` = ""
- static bool `idhelp_subset` = false
- static string `idhelp_subset_string` = ""
- static bool `idhelp_replace` = false
- static string `idhelp_replace_string` = ""
- static bool `idhelp_match` = false
- static vector< string > `idhelp_match_string`
- static bool `idhelp_no_dict` = false
- static bool `idhelp_list_aliases` = false
- static bool `idhelp_alias_update` = true
- static string `idhelp_command`
- static string `idhelp_input`
- static bool `run_R_script` = false
- static bool `run_R_write_script` = false
- static string `R_script` = "script.R"
- static bool `run_R_chisq` = false
- static bool `run_R_z` = false
- static int `run_R_nsnp` = 100
- static int `R_port` = 6311
- static bool `recode` = false
- static bool `recode_transpose` = false
- static bool `recode_long` = false
- static bool `recode_long_ref` = false
- static bool `recode_mutlist` = false

- static bool `recode_12` = false
- static bool `recode_AD` = false
- static bool `recode_AD_Aonly` = false
- static bool `recode_AD_fixed` = false
- static bool `recode_allele_coding` = false
- static string `recode_allele_coding_file` = "file.lst"
- static bool `recode_1234` = false
- static bool `recode_ACGT` = false
- static bool `set_reference_allele` = false
- static string `set_reference_allele_file` = "dummy.file"
- static bool `lfile_allele_count` = false
- static string `recode_delimit` = " "
- static string `recode_indelimit` = " "
- static bool `recode_HV` = false
- static bool `recode_whap` = false
- static bool `recode_fastphase` = false
- static bool `recode_structure` = false
- static bool `recode_bim bam` = false
- static bool `preserve_all_genotypes` = false
- static bool `preserve_mendel_errors` = false
- static bool `zero_cluster` = false
- static string `zero_cluster_filename` = "inbix.zero"
- static bool `oblig_missing` = false
- static string `oblig_missing_filename` = "inbix.zero"
- static string `oblig_clusters_filename` = "inbix.clst"
- static bool `loop_over` = false
- static string `loop_over_label` = ""
- static int `loop_counter` = 0
- static string `loop_over_filename` = "inbix.clst"
- static bool `list_by_allele` = false
- static bool `list_twolocus` = false
- static string `twolocus_snp1` = ""
- static string `twolocus_snp2` = ""
- static bool `indiv_report` = false
- static string `indiv_report_fid` = "fid1"
- static string `indiv_report_iid` = "iid1"
- static bool `plist` = false
- static string `plist_fid1` = ""
- static string `plist_iid1` = ""
- static string `plist_fid2` = ""
- static string `plist_iid2` = ""
- static bool `merge_data` = false
- static bool `merge_force_strand` = false
- static int `merge_mode` = 1
- static bool `merge_binary` = false
- static bool `merge_list` = false
- static string `merge_list_filename` = "merge.list"
- static string `merge_pedfile` = "merge.ped"
- static string `merge_mapfile` = "merge.map"
- static string `merge_bedfile` = "merge.bed"
- static string `merge_bimfile` = "merge.bim"

- static string `merge_famfile` = "merge.fam"
- static bool `write_snplist` = false
- static bool `update_map` = false
- static bool `update_cm` = false
- static bool `update_chr` = false
- static bool `update_name` = false
- static bool `update_ids` = false
- static string `update_ids_file` = ""
- static bool `update_sex` = false
- static string `update_sex_file` = ""
- static bool `update_parents` = false
- static string `update_parents_file` = ""
- static bool `update_pheno` = false
- static string `update_pheno_file` = ""
- static string `update_mapfile` = "new.map"
- static string `range_delimiter` = "-"
- static bool `update_alleles` = false
- static string `update_allele_file` = "dummy"
- static bool `compound_genotype_code` = false
- static string `tpedfile` = "inbix.tped"
- static string `tfamfile` = "inbix.tfam"
- static bool `tfile_input` = false
- static string `lpedfile` = "inbix.lgen"
- static string `lfamfile`
- static bool `lfile_input` = false
- static bool `ref_file` = false
- static string `ref_file_name` = ""
- static bool `gvar` = false
- static bool `gvar_write` = false
- static bool `gvar_to_standard` = false
- static bool `load_gvar` = false
- static bool `gvar_verbose_association` = false
- static string `gmapfile` = "inbix.map"
- static string `gfamfile` = "inbix.fam"
- static string `gvarfile` = "inbix.gvar"
- static bool `gvar_include_all_variants` = false
- static bool `gvar_full_report` = false
- static bool `flip_strand` = false
- static string `flip_file` = "inbix.flip"
- static bool `flip_subset` = false
- static string `flip_subset_file` = "inbix.file"
- static bool `read_bitfile` = false
- static bool `write_bitfile` = false
- static bool `fast_binary` = false
- static string `bitfilename` = "inbix.bed"
- static string `famfile` = "inbix.fam"
- static string `bitfilename_map` = "inbix.bim"
- static bool `SNP_major` = true
- static bool `out_SNP_major` = true
- static bool `compress_file` = false
- static bool `uncompress_file` = false

- static string `compress_filename` = ""
- static bool `read_ped` = false
- static string `pedfile` = "inbix.ped"
- static string `mapfile` = "inbix.map"
- static bool `ped_from_stdin` = false
- static string `fileroot` = "inbix"
- static bool `map3` = false
- static bool `liability` = false
- static bool `ped_skip_sex` = false
- static bool `ped_skip_parents` = false
- static bool `ped_skip_fid` = false
- static bool `ped_skip_pheno` = false
- static string `output_file_name` = "inbix"
- static bool `silent` = false
- static bool `gplink` = false
- static bool `cli` = false
- static string `missing_genotype` = "0"
- static string `out_missing_genotype` = "0"
- static string `missing_phenotype` = "-9"
- static string `out_missing_phenotype` = "-9"
- static bool `missing_genotype_explicit` = false
- static bool `missing_phenotype_explicit` = false
- static bool `ignore_missing_sex` = false
- static bool `pheno_file` = false
- static bool `covar_file` = false
- static bool `clist` = false
- static bool `have_snps` = false
- static bool `have_numerics` = false
- static bool `numeric_file` = false
- static string `numeric_filename` = "inbix.num"
- static bool `do_covariance_matrix` = false
- static bool `do_coexpression_all` = false
- static bool `do_coexpression_casecontrol` = false
- static bool `do_numeric_summary` = false
- static bool `do_numeric_extract` = false
- static string `numeric_extract_file` = ""
- static bool `do_differential_coexpression` = false
- static bool `do_dcgain_abs` = false
- static bool `do_dcvar` = false
- static bool `do_dcvar_pfilter` = false
- static double `dcvar_pfilter_value` = 0.05
- static string `dcvar_pfilter_type` = "bon"
- static string `dcvar_var_model` = "dom"
- static string `dcvar_genotypes_file` = ""
- static string `dcvar_snp_locations_file` = ""
- static string `dcvar_gene_expression_file` = ""
- static string `dcvar_chip_seq_file` = ""
- static bool `do_dcvar_chipseq` = false
- static bool `dcvar_resume.snp` = false
- static uint `dcvar_radius` = 1000
- static bool `do_differential_modularity` = false

- static bool `do_dmgain_abs` = false
- static bool `do_numeric_standardize` = false
- static bool `do_numeric_lowval_filter` = false
- static bool `do_numeric_lowvar_filter` = false
- static double `numeric_lowval_percentile` = 0.1
- static double `numeric_lowvar_percentile` = 0.1
- static bool `do_regain` = false
- static bool `do_regain_post` = false
- static bool `do_regain_pvalue_threshold` = false
- static string `regainFile` = "no file"
- static const double `regainLargeCoefPValue` = 0.99
- static const double `regainLargeCoefTvalue` = 10
- static const double `regainMaxBetaValue` = 100e3
- static bool `regainCompress` = false
- static bool `regainComponents` = false
- static double `regainFdr` = 0.5
- static bool `regainFdrPrune` = false
- static bool `regainSifFilter` = false
- static double `regainSifThreshold` = -9999999
- static bool `regainUseBetaValues` = false
- static bool `regainPureInteractions` = false
- static double `regainFailValue` = 0
- static double `regainPvalueThreshold` = 0.1
- static string `regainMatrixTransform` = "none"
- static string `regainMatrixFormat` = "full"
- static bool `regainMatrixThreshold` = false
- static double `regainMatrixThresholdValue` = 0.0
- static bool `regainMatrixToSif` = false
- static bool `do_deconvolution` = false
- static double `deconvolutionAlpha` = 1.0
- static double `deconvolutionBeta` = 0.9
- static int `deconvolutionControl` = 0
- static bool `do_modularity` = false
- static double `modConnectivityThreshold` = 0.0
- static bool `modEnableConnectivityThreshold` = false
- static bool `modUseBinaryThreshold` = false
- static bool `modComputeHomophily` = false
- static bool `modPowerTransform` = false
- static double `modPowerTransformExponent` = 1
- static bool `modFisherTransform` = false
- static double `modFisherTransformCutoff` = 0.999999
- static string `sifFile` = ""
- static bool `sifNetwork` = false
- static bool `sifToGain` = false
- static string `afni1dFile` = ""
- static bool `afniNetwork` = false
- static bool `do_relief` = false
- static bool `do_write_best_k` = false
- static bool `do_write_each_k_scores` = false
- static bool `do_normalize_scores` = false
- static bool `do_iterative_removal` = false

- static bool `do_kopt` = false
- static string `distanceMatrixFilename` = ""
- static string `gainMatrixFilename` = ""
- static string `titvFilename` = ""
- static string `algorithmMode` = "relieff"
- static string `algorithmSeqMode` = "snr"
- static string `algorithmSnrMode` = "snr"
- static string `algorithmTstatMode` = "pval"
- static double `algorithmSeqS0` = 0.05
- static unsigned int `k` = 10
- static unsigned int `koptBegin` = 1
- static unsigned int `koptEnd` = 2
- static unsigned int `koptStep` = 1
- static unsigned int `m` = 0
- static string `snpDiffMetricName` = "gm"
- static string `snpNearestNeighborMetricName` = "gm"
- static string `numDiffMetricName` = "manhattan"
- static string `weightByDistanceMethod` = "equal"
- static double `weightByDistanceSigma` = 2.0
- static string `reliefMode` = "relieff"
- static string `reliefSeqAlgorithmMode` = "snr"
- static double `reliefSeqAlgorithmS0` = 0.05
- static string `reliefSeqSnrMode` = "snr"
- static string `reliefSeqTstatMode` = "pval"
- static unsigned int `relieffNumTarget` = 0
- static unsigned int `relieffIterNumToRemove` = 0
- static unsigned int `relieffIterPercentToRemove` = 0
- static bool `do_randomforest` = false
- static vector< std::string > `alwaysplitvars`
- static string `caseweights` = ""
- static string `depvarname` = "Class"
- static double `fraction` = 1.0
- static bool `holdout` = false
- static `MemoryMode` `memmode` = `MEM_DOUBLE`
- static bool `savemem` = false
- static string `predict` = ""
- static string `splitweights` = ""
- static unsigned int `nrthreads` = `DEFAULT_NUM_THREADS`
- static bool `predall` = false
- static double `alpha` = `DEFAULT_ALPHA`
- static double `minprop` = `DEFAULT_MINPROP`
- static vector< string > `catvars`
- static `ImportanceMode` `impmeasure` = `IMP_NONE`
- static unsigned int `targetpartitionsize` = 0
- static unsigned int `mtry` = 0
- static bool `do_rfprobability` = false
- static `SplitRule` `splitrule` = `DEFAULT_SPLITRULE`
- static string `statusvarname` = ""
- static unsigned int `ntree` = `DEFAULT_NUM_TREE`
- static bool `rfreplace` = true
- static `TreeType` `treetype` = `TREE_CLASSIFICATION`

- static bool `writeforest` = false
- static bool `do_ec` = false
- static bool `do_iterative_removal_ec` = false
- static bool `do_iterative_write_scores` = false
- static string `ecAlgorithmSteps` = "all"
- static string `ecMeAlgorithm` = "randomforest"
- static string `ecItAlgorithm` = "relieff"
- static bool `ecOptimizeTemp` = false
- static unsigned int `ecNumTarget` = 0
- static unsigned int `eclterNumToRemove` = 0
- static unsigned int `eclterPercentToRemove` = 0
- static bool `do_ec_privacy` = false
- static string `ecPrivacyTrainFile` = ""
- static string `ecPrivacyHoldoutFile` = ""
- static string `ecPrivacyTestFile` = ""
- static double `ecPrivacyStartTemp` = 0.1
- static double `ecPrivacyFinalTemp` = 0.00001
- static double `ecPrivacyTau` = 100
- static uint `ecPrivacyUpdateFrequency` = 50
- static double `ecPrivacyPercentSignal` = 0.1
- static bool `do_ripm` = false
- static string `thresholdType` = "hard"
- static double `thresholdValue` = 0.8
- static int `startMergeOrder` = 2
- static int `maxMergeOrder` = 4
- static int `maxModuleSize` = 200
- static int `minModuleSize` = 30
- static bool `useAbs` = true
- static bool `useWeighted` = false
- static bool `do_ranking` = false
- static string `ranker_method` = ""
- static int `ranker_top_n` = -1
- static string `ranker_save_data_file` = ""
- static string `ranker_input_file` = ""
- static double `ranker_centrality_gamma` = 0.85
- static bool `do_ranker_permutation` = false
- static string `rankerPermMethod` = "regain"
- static int `rankerPermNum` = 100
- static double `rankerPermThreshold` = 0.05
- static bool `exportArff` = false
- static bool `exportDelimited` = false
- static string `exportDelimiter` = "\t"
- static bool `do_iqtl` = false
- static string `iqtl_expression_file` = ""
- static string `iqtl_coord_file` = ""
- static bool `iqtl_local_cis` = false
- static int `iqtl_radius` = 1000
- static bool `iqtl_interaction_full` = false
- static bool `do_iqtl_tf` = false
- static int `iqtl_tf_radius` = 0
- static string `iqtl_tf_coord_file` = ""

- static double `iqt1_pvalue` = 0.5
- static bool `no_show_covar` = false
- static bool `dump_covar` = false
- static bool `dump_covar_with_phenotype` = false
- static bool `dump_covar_dummy_coding` = false
- static bool `filter_on_covar` = false
- static int `clist_number` = 0
- static int `plist_number` = 0
- static bool `snp_attrib_filter` = false
- static string `snp_attrib_value` = ""
- static string `snp_attrib_file` = ""
- static bool `ind_attrib_filter` = false
- static string `ind_attrib_value` = ""
- static string `ind_attrib_file` = ""
- static bool `multiple_phenotypes` = false
- static string `multiple_phenotype_file` = ""
- static string `make_pheno_filename` = ""
- static string `make_pheno_value` = ""
- static bool `make_pheno` = false
- static bool `make_pheno_present` = false
- static bool `dump_clst` = false
- static bool `clist_selection` = false
- static bool `clist_selection_name` = false
- static bool `clist_selection_number` = false
- static string `clist_selection_string` = ""
- static bool `plist_selection` = false
- static bool `plist_selection_name` = false
- static bool `plist_selection_number` = false
- static string `plist_selection_string` = ""
- static int `mult_pheno` = 1
- static string `name_pheno` = ""
- static bool `all_pheno` = false
- static int `mult_covar` = 1
- static int `mult_clst` = 1
- static int `mult_filter` = 1
- static string `filter_value` = "1"
- static string `number_list_string` = ""
- static bool `number_list_positive` = true
- static string `pheno_filename` = "inbix.phe"
- static string `covar_filename` = "inbix.cov"
- static string `clist_filename` = "inbix.cov"
- static string `filter_filename` = "inbix.cov"
- static bool `cm_map` = false
- static double `grid` = 0.005
- static double `fringe` = .01
- static bool `singlepoint` = false
- static int `inter_grid` = 2
- static bool `done_global_pihat` = false
- static bool `sol_family` = false
- static bool `summ_nonfounders` = false
- static bool `make_founders` = false

- static bool `has_nofounders` = false
- static bool `make_missing_parents` = false
- static bool `score_risk` = false
- static string `score_risk_file` = "inbix.risk"
- static bool `score_risk_ranges` = false
- static string `score_risk_ranges_file` = "inbix.ranges"
- static int `score_risk_ranges_min` = 0
- static bool `score_impute_expected` = true
- static bool `score_risk_on_qrange` = false
- static string `score_qrange_file` = ""
- static string `score_qfile` = ""
- static bool `score_test` = false
- static bool `profile_sets` = false
- static bool `report_missing` = false
- static bool `test_missing` = false
- static bool `mishap_test` = false
- static int `mishap_window` = 1
- static bool `calcFst` = false
- static bool `proxy_assoc` = false
- static bool `proxy_glm` = false
- static bool `proxy_all` = false
- static bool `proxy_full_report` = false
- static bool `proxy_error` = false
- static bool `proxy_impute` = false
- static bool `proxy_impute_replace` = false
- static bool `proxy_impute_preserve_genotyped` = false
- static bool `proxy_record_dosage` = false
- static bool `proxy_impute_genotypic_concordance` = false
- static double `proxy_impute_threshold` = 0.9
- static double `proxy_info_threshold` = 0.5
- static bool `impute_verbose` = false
- static bool `proxy_exclude` = false
- static string `proxy_exclude_list` = "pexclude.list"
- static bool `proxy_exclude_from_file` = false
- static bool `proxy_reference_only` = false
- static bool `proxy_leave_out` = false
- static bool `proxy_include_reference` = false
- static bool `proxy_CC` = false
- static bool `proxy_TDT` = false
- static string `proxy_assoc_snp` = "rs1234"
- static int `proxy_window` = 15
- static bool `proxy_list` = false
- static string `proxy_list_file` = "proxy.hap"
- static bool `proxy_all_list` = false
- static string `proxy_all_list_file` = "proxy.list"
- static double `proxy_kb` = 250
- static double `proxy_r2` = 0.5
- static double `proxy_maf` = 0.005
- static double `proxy_mhf` = 0.01
- static double `proxy_genotype` = 0.2
- static bool `proxy_list_proxies` = false

- static int `proxy_maxhap` = 3
- static bool `proxy_r2_filter` = true
- static double `proxy_r2_filter_A` = 0.00
- static double `proxy_r2_filter_B` = 0.05
- static double `proxy_r2_filter_C` = 0.50
- static int `proxy.snp_filter` = 5
- static double `proxy_kb_planA` = 250
- static int `proxy_window_planA` = 15
- static int `proxy.snp_filter_planA` = 5
- static double `proxy_r2_filter_A_planA` = 0.00
- static double `proxy_r2_filter_B_planA` = 0.25
- static double `proxy_r2_filter_C_planA` = 0.50
- static double `proxy_planB_threshold` = 0.1
- static double `proxy_kb_planB` = 500
- static int `proxy_window_planB` = 30
- static int `proxy.snp_filter_planB` = 10
- static double `proxy_r2_filter_A_planB` = 0.00
- static double `proxy_r2_filter_B_planB` = 0.01
- static double `proxy_r2_filter_C_planB` = 0.50
- static bool `greport` = false
- static string `greport_results` = "file1"
- static string `greport_gene_list` = "file2"
- static bool `greport_subset` = false
- static string `greport_subset_file` = "file3"
- static bool `greport_display_empty` = false
- static bool `annot_file` = false
- static string `annot_filename` = ""
- static bool `meta_analysis` = false
- static vector< string > `meta_files`
- static bool `set_screen` = false
- static string `set_screen_resultfile` = ""
- static bool `gettag_mode` = false
- static bool `gettag_mode1` = true
- static bool `gettag_mode2` = false
- static string `gettag_file` = ""
- static double `gettag_r2` = 0.8
- static int `gettag_kb` = 250000
- static bool `gettag_listall` = false
- static bool `clumpld` = false
- static bool `clumpld_best` = false
- static string `clumpld_results` = "inbx.assoc"
- static string `clumpld_column` = "P"
- static bool `clumpld_verbose` = false
- static bool `clumpld_indep` = true
- static int `clumpld_kb` = 250000
- static double `clumpld_r2` = 0.5
- static double `clumpld_p1` = 1e-4
- static double `clumpld_p2` = 1e-2
- static bool `clumpld_index1` = false
- static bool `clumpld_only_show_replications` = false
- static bool `clumpld_only_show_replications_list` = false

- static bool `clumpld_annot` = false
- static string `clumpld_annot_fields` = ""
- static string `clumpld_range_file` = "range.list"
- static bool `clumpld_range_annotation` = false
- static int `clumpld_min` = 0
- static double `min_af` = 0.01
- static double `max_af` = 1
- static bool `make_minor_allele` = true
- static double `min_hf` = 0.01
- static double `max_hf` = 1
- static int `min_genotype_cell` = 5
- static double `rarer_maf_threshold` = 0.1
- static double `rarer_dist_threshold` = 100000
- static int `rarer_interval` = 100
- static bool `rare_test` = false
- static bool `rare_test_weight1` = false
- static bool `rare_test_print_details` = false
- static string `rare_test_print_details_snp` = ""
- static bool `elf_pcmode` = false
- static bool `elf_pcmode_2sided` = false
- static bool `elf_baseline` = false
- static bool `rare_test_score_range` = false
- static double `rare_test_score_range_threshold` = 0.01
- static string `rare_test_score_results_file` = ""
- static string `rare_test_score_range_file` = ""
- static bool `rare_test_summary_controls` = false
- static vector< bool > `chr_haploid`
- static vector< bool > `chr_sex`
- static vector< bool > `chr_Y`
- static vector< string > `chr_code`
- static map< string, int > `chr_map`
- static bool `species_dog` = false
- static bool `species_cow` = false
- static bool `species_sheep` = false
- static bool `species_horse` = false
- static bool `species_rice` = false
- static bool `species_mouse` = false
- static int `run_start` = 0
- static int `run_end` = 0
- static int `run_chr` = 0
- static string `m1` = ""
- static string `m2` = ""
- static double `window` = 0
- static bool `position_window` = false
- static int `from_window` = 0
- static int `to_window` = 0
- static bool `qt` = false
- static bool `bt` = true
- static bool `coding01` = false
- static bool `ignore_phenotypes` = true
- static bool `filter_cases` = false

- static bool `filter_controls` = false
- static bool `filter_males` = false
- static bool `filter_females` = false
- static bool `filter_founders` = false
- static bool `filter_nofounders` = false
- static bool `SD` = true
- static bool `CP` = false
- static bool `affpair` = false
- static bool `remove_unaffected_pairs` = false
- static bool `fix_prev` = false
- static double `fixed_prev` = 0
- static string `tagfile` = "inbix.tag"
- static string `mapfile_impute` = "inbix.impute.map"
- static bool `make_tags`
- static bool `impute_tags` = false
- static bool `sliding_window` = false
- static string `sliding_window_size` = "2"
- static bool `make_blocks` = false
- static bool `meta_large_phase` = false
- static bool `phase_snps` = false
- static bool `phase_hap_all` = false
- static double `hap_post_prob` = 0.8
- static double `hap_missing_geno` = 0.5
- static double `hap_min_phase_prob` = 1e-2
- static int `hap_max_nf_phases` = 1024
- static bool `display_hap_freqs` = false
- static int `haplo_plem_window` = 6
- static int `haplo_plem_overlap` = 2
- static int `haplo_plem_original_overlap` = 2
- static int `haplo_plem_iter` = 20
- static bool `haplo_plem_verbose` = false
- static bool `haplo_plem_follow` = false
- static int `haplo_plem_follow_ind` = -1
- static string `haplo_plem_follow_fid` = "FID1"
- static string `haplo_plem_follow_iid` = "IID1"
- static int `haplo_plem_likelihood_iter` = 5
- static double `haplo_plem_window_prune_phase` = 1e-10
- static double `haplo_plem_window_tol` = 1e-4
- static double `haplo_plem_zero_threshold` = -1
- static bool `haplo_plem_nonzero_threshold` = true
- static int `haplo_plem_meta_window` = 2
- static double `haplo_plem_meta_prune_haplotype` = 1e-6
- static double `haplo_plem_meta_prune_phase` = 0.01
- static int `haplo_plem_meta_iter` = 200
- static int `haplo_plem_meta_likelihood_iter` = 5
- static double `haplo_plem_meta_tol` = 1e-4
- static bool `test_hap_CC` = false
- static bool `test_hap_TDT` = false
- static bool `test_hap_QTL` = false
- static bool `test_hap_only` = false
- static bool `test_hap_GLM` = false

- static bool `test_hap_GLM_omnibus` = false
- static bool `display_phase_probs` = false
- static bool `display_phase_probs_wide` = false
- static bool `weighted_mm` = false
- static bool `chap_test` = false
- static bool `chap_sole_variant` = false
- static bool `chap_sole_variant_specific_alleles` = false
- static string `chap_sole_variant_specific_allele_list` = ""
- static bool `chap_independent_effect` = false
- static bool `chap_haplotype_specific` = false
- static string `chap_entity` = ""
- static bool `chap_specified_groups` = false
- static bool `chap_specified_snps` = false
- static string `chap_model1` = ""
- static string `chap_model0` = ""
- static bool `chap_drop_snps` = false
- static string `chap_drop_snps_list` = ""
- static bool `chap_add_grp_specifics` = false
- static bool `assoc_test` = false
- static bool `assoc_counts` = false
- static bool `assoc_glm` = false
- static bool `standard_beta` = false
- static bool `assoc_glm_without_main_snp` = false
- static bool `assoc_test_alt_perm` = false
- static bool `full_model_assoc` = false
- static bool `trend_only` = false
- static bool `fisher_test` = false
- static bool `return_beta` = false
- static bool `hap_specific_snps` = false
- static string `hap_specific_snps_list` = ""
- static bool `output_pheno_perm` = false
- static bool `qt_means` = false
- static bool `conditioning_snp_single` = false
- static string `conditioning.snp_name` = "rs1234"
- static bool `conditioning_snps` = false
- static string `conditioning_snps_file` = "inbix.list"
- static int `xchr_model` = 1
- static bool `glm_sex_effect` = false
- static bool `glm_no_auto_sex_effect` = false
- static bool `glm_dominant` = false
- static bool `glm_recessive` = false
- static double `vif_threshold` = 50
- static bool `twoDFmodel` = false
- static bool `twoDFmodel_hethom` = false
- static bool `test_full_model` = false
- static bool `simple_interaction` = false
- static vector< int > `parameter_list`
- static vector< int > `test_list`
- static bool `glm_user_test` = false
- static bool `glm_user_parameters` = false
- static bool `qt_with_covariates` = false

- static bool `model_perm_best` = false
- static bool `model_perm_gen` = false
- static bool `model_perm_dom` = false
- static bool `model_perm_rec` = false
- static bool `model_perm_trend` = false
- static bool `assoc_gxe` = false
- static bool `QTDT_test` = false
- static bool `QFAM_total` = false
- static bool `QFAM_between` = false
- static bool `QFAM_within1` = false
- static bool `QFAM_within2` = false
- static bool `QFAM_adaptive` = false
- static bool `TDT_test` = false
- static bool `sibTDT_test` = false
- static bool `mating_tests` = false
- static bool `dfam_tdt` = true
- static bool `dfam_sibs` = true
- static bool `dfam_unrelateds` = true
- static bool `perm_TDT_basic` = true
- static bool `perm_TDT_parent` = false
- static bool `discordant_parents` = false
- static bool `parent_of_origin` = false
- static bool `perm_POO_poo` = true
- static bool `perm_POO_pat` = false
- static bool `perm_POO_mat` = false
- static bool `perm_POO_best` = false
- static bool `built_families` = false
- static bool `MENDEL_test` = false
- static bool `MENDEL_report` = false
- static double `MENDEL.snp` = 0.1
- static double `MENDEL.ind` = 0.1
- static bool `HWD_test` = false
- static bool `HWD_report` = false
- static double `HWD_limit` = 0.001
- static bool `HWD_standard` = false
- static bool `HWD_filter_on_all` = false
- static bool `CMH_test_1` = false
- static bool `CMH_test_2` = false
- static bool `CMH_test_ORD` = false
- static bool `breslowday` = false
- static bool `OR_homog_test` = false
- static double `ci_level` = 0.95
- static double `ci_zt` = 0
- static bool `display_ci` = false
- static bool `pfilter` = false
- static double `pfvalue` = 1e-5
- static bool `multtest` = false
- static bool `use_GC` = false
- static bool `fix_lambda` = false
- static double `lambda` = 1
- static bool `qq_plot` = false

- static bool `logscale` = false
- static bool `ibs_sharing_test` = false
- static bool `extract_set` = false
- static bool `exclude_set` = false
- static bool `snp_range_list` = false
- static bool `thin_snps` = false
- static double `thin_param` = 0
- static bool `make_set` = false
- static string `make_set_file` = "inbix.set"
- static int `make_set_border` = 0
- static bool `make_set_collapse` = false
- static bool `make_set_ignore_group` = false
- static string `make_set_collapse_label` = "SET"
- static bool `make_set_complement` = false
- static bool `write_set` = false
- static bool `read_set` = false
- static string `exclude_file` = "inbix.list"
- static string `extract_file` = "inbix.list"
- static string `keep_file` = "inbix.list"
- static string `remove_file` = "inbix.list"
- static bool `read_snp_qual` = false
- static string `snp_qual_file` = "dummy"
- static double `snp_qual_min` = 0
- static double `snp_qual_max` = 1
- static bool `read_genotype_qual` = false
- static string `geno_qual_file` = "dummy"
- static double `geno_qual_min` = 0
- static double `geno_qual_max` = 1
- static bool `snp_include_from_cl` = false
- static string `snp_include_range` = ""
- static bool `dump_gene` = false
- static string `dump_genename` = ""
- static bool `hotel` = false
- static bool `set_test` = false
- static bool `set_p2` = false
- static int `set_min` = -1
- static int `set_max` = 5
- static bool `set_r2` = false
- static double `set_r2_val` = 0.5
- static bool `set_r2_phase` = false
- static double `set_chisq_threshold` = 3.84146
- static bool `set_r2_write` = false
- static bool `set_r2_read` = false
- static string `set_r2_read_file` = "inbix.Idset"
- static string `subsetfile` = "dummy.file"
- static bool `use_subset` = false
- static string `setfile` = "inbix.set"
- static bool `set_score` = false
- static double `set_score_p` = 1
- static double `set_step_in` = 0.05
- static bool `set_step` = false

- static bool `set_table` = false
- static bool `permute_within_sol` = false
- static bool `boot` = false
- static bool `disp_r1` = false
- static bool `disp_r2` = false
- static bool `disp_r_window` = false
- static int `disp_r_window.snp` = 10
- static int `disp_r_window.kb` = 1000000
- static double `disp_r_window.r2` = 0.2
- static bool `ld_anchor` = false
- static bool `ld_anchor_list` = false
- static bool `flip_scan` = false
- static double `flip_scan_threshold` = 0.5
- static bool `flip_scan_verbose` = false
- static bool `prune_ld` = false
- static bool `prune_ld_pairwise` = false
- static bool `prune_ld_pairwise_maf` = true
- static double `prune_ld_vif` = 2
- static double `prune_ld_r2` = 1 - 1e-6
- static int `prune_ld_win` = 100
- static int `prune_ld_step` = 50
- static bool `prune_r2_prefer` = false
- static string `prune_r2_prefer_list` = "dummy"
- static bool `prune_r2_fixed` = false
- static string `prune_r2_fixed_list` = "dummy"
- static bool `calc_SNPSNP_LD` = false
- static string `ld_SNP1` = ""
- static string `ld_SNP1_file` = ""
- static string `ld_SNP2` = ""
- static bool `epistasis` = false
- static bool `fast_epistasis` = false
- static bool `epi_caseonly` = false
- static double `epi_caseonly_kb_gap` = 1000
- static bool `epi_filter` = true
- static double `epi_alpha1` = 0.0001
- static double `epi_alpha2` = 0.01
- static bool `set_by_set` = true
- static bool `epi_genebased` = false
- static bool `epi_quickscan` = false
- static bool `drop_sets` = true
- static bool `inbreeding` = false
- static bool `check_sex` = false
- static bool `impute_sex` = false
- static double `sex_threshold_male` = 0.8
- static double `sex_threshold_female` = 0.2
- static bool `homo_run` = false
- static bool `homo_run_consensus_match` = false
- static bool `homo_run_kb` = false
- static bool `homo_run_snps` = false
- static double `homo_run_density` = 50
- static int `homo_run_gap` = 1000

- static bool `homo_miss_as_hom` = false
- static int `homoWindowSize` = 50
- static int `homo_windowKB` = 5000
- static int `homo_windowAllowedHet` = 1
- static int `homo_windowAllowedMissing` = 5
- static double `homo_threshold` = 0.05
- static int `homo_run_length_kb` = 1000
- static int `homo_run_length_snps` = 100
- static int `homo_run_het` = 1
- static bool `homo_summary_allelic_match` = false
- static double `fuzzy_homo` = 0.99
- static bool `homozyg_verbose` = false
- static int `pool_size_min` = 2
- static bool `ibs_run` = false
- static int `ibs_run_length_snps` = 100
- static int `ibs_run_length_kb` = 100
- static double `ibs_run_density` = 0.01
- static int `ibs_inner_run_length_kb` = 100
- static int `ibs_inner_run_length_snp` = 20
- static int `ibs_join_kb` = 100
- static int `ibs_join.snp` = 1
- static int `ibs_run_missing` = 2
- static int `ibs_run_0` = 1
- static int `ibs_inter.snp_distance` = 1000000
- static bool `ibs_2only` = false
- static bool `miss_run` = false
- static int `miss_run_length` = 100
- static bool `miss_run_length_kb` = false
- static double `miss_run_level` = 0.80
- static bool `segment_haplotrack` = false
- static string `segment_haplotrack_fid1` = "1"
- static string `segment_haplotrack_iid1` = "1"
- static string `segment_haplotrack_fid2` = "2"
- static string `segment_haplotrack_iid2` = "2"
- static bool `mk_datfile` = false
- static bool `segment_output` = false
- static bool `segment_minimal` = false
- static bool `segment_silently_return_groups` = false
- static int `segment_current_focal.snp` = -1
- static bool `segment_overlap` = false
- static bool `segment_verbose` = false
- static bool `segment_validate` = false
- static bool `segment_test_individual` = false
- static bool `segment_test_specific_segs` = false
- static bool `segment_test_fisher` = false
- static bool `segment_test_1sided` = true
- static bool `segment_test_force_1sided` = false
- static bool `segment_test_ignore_discordant` = false
- static int `segment.snp1` = -1
- static int `segment.snp2` = -1
- static string `segment.m1` = ""

- static string `segment_m2` = ""
- static bool `force_span` = false
- static int `segment_length` = 1000000
- static int `segment_snp` = 100
- static bool `segment_output_started` = false
- static bool `read_segment_file` = false
- static string `read_segment_filename` = ""
- static int `segment_inter.snp_distance` = 1000
- static bool `multi_output` = false
- static bool `gmulti_output` = false
- static bool `pihat_filter` = true
- static bool `genome_output` = false
- static bool `compress_genome` = false
- static bool `genome_only_check_rels` = false
- static bool `genome_output_minimal` = false
- static bool `genome_output_full` = false
- static bool `genome_2sets` = false
- static string `genome_setlist1` = "inbix.set1"
- static string `genome_setlist2` = "inbix.set2"
- static bool `genome_test` = false
- static double `genome_test_threshold` = 0.01
- static int `genome_test_min.snp` = 20
- static bool `ibs_test` = false
- static int `ibs_test_min.snp` = 20
- static bool `ibs_test_method2` = false
- static bool `summary_ibd_output` = false
- static double `IBD_threshold` = 0.2
- static double `segment_threshold_start` = 0.25
- static double `segment_threshold_finish` = 0.25
- static bool `nudge` = false
- static bool `bound` = true
- static bool `show_impossible_IBD` = true
- static bool `IBD_within` = false
- static bool `permute` = false
- static int `replicates` = 1000
- static bool `perm_count` = false
- static bool `mperm_save_best` = false
- static bool `mperm_save_all` = false
- static bool `mperm_rank` = false
- static bool `adaptive_perm` = true
- static int `adaptive_min` = 5
- static int `adaptive_max` = 1000000
- static int `adaptive_interval` = 1
- static double `adaptive_interval2` = 0.001
- static double `adaptive_alpha` = 0.00
- static double `adaptive_ci` = 0.0001
- static bool `perm_genedrop` = false
- static bool `perm_genedrop_and_swap` = false
- static bool `perm_genedrop_unrel` = false
- static bool `perm_genedrop_parents` = false
- static bool `perm_genedrop_sibships` = false

- static bool **FIXED** = false
- static bool **FIXED_p** = false
- static Z **FIX_IBD**
- static double **FIX_p** = 0.5
- static bool **matrix** = false
- static bool **distance_matrix** = false
- static bool **cluster** = false
- static bool **cluster_euclidean** = false
- static bool **cluster_group_avg** = false
- static bool **cluster_plot** = false
- static bool **force_initial_cluster** = false
- static int **cluster_mds_dim** = 2
- static bool **mds_by_individual** = true
- static bool **genome_groups** = false
- static bool **cluster_ibm_constraint** = false
- static double **cluster_ibm_constraint_value** = 0
- static bool **cluster_missing** = false
- static bool **cluster_selcon** = false
- static string **cluster_selcon_file** = "inbix.clst"
- static int **max_cluster_N** = -1
- static double **merge_p** = 0
- static int **ibstest_gap** = 500000
- static int **max_cluster_size** = 0
- static int **max_cluster_case** = 0
- static int **max_cluster_control** = 0
- static bool **include_cluster** = false
- static bool **include_cluster_from_file** = false
- static string **include_cluster_filename** = "inbix.clst"
- static int **analyse_cluster** = 0
- static bool **cluster_on_phenotype** = false
- static bool **cluster_on_mcc** = false
- static int **min_neighbour** = 1
- static int **max_neighbour** = 10
- static bool **outlier_detection** = false
- static bool **bmatch** = false
- static bool **bmatch_usertype** = false
- static bool **qmatch** = false
- static string **bmatch_filename** = "inbix.bmatch"
- static string **bmatch_direction_filename** = "inbix.bm"
- static string **qmatch_filename** = "inbix.qmatch"
- static string **qmatch_threshold_filename** = "inbix.qt"
- static bool **include_all_pairs** = false
- static double **include_all_z1** = 0.001
- static double **MIN_PIHAT** = 0.0025
- static double **MAX_PIHAT** = 1.0000
- static double **MAX_CORR_PIHAT_PIHAT_G** = 0.9
- static double **MAX_GENO_MISSING** = 0.1
- static double **MAX_IND_MISSING** = 0.1
- static int **MAX_LINE_LENGTH** = 2000000
- static bool **remove_indiv** = false
- static string **remove_indiv_list** = "inbix.list"

- static string `keep_indiv_list` = "inbix.list"
- static bool `keep_indiv` = false
- static bool `extract_before_exclude` = true
- static bool `remove_before_keep` = true
- static bool `locked` = false
- static bool `af_read` = false
- static bool `af_write` = false
- static bool `ibd_read` = false
- static string `ibd_file` = "inbix.genome"
- static bool `ibd_read_minimal` = false
- static bool `ibd_read_list` = false
- static string `ibd_file_list` = "inbix.genome.list"
- static string `af_file` = "inbix.frq"
- static bool `af_count` = false
- static bool `inc_write` = false
- static bool `inc_read` = false
- static string `inc_file` = "inbix.inc"
- static int `pp_maxsnp` = 6
- static int `pp_maxfid` = 6
- static int `pp_maxiid` = 6
- static int `BATCH_SIZE` = 500000
- static bool `plink` = false
- static bool `display_segment_long` = false
- static bool `display_cnv_track` = false
- static int `cnv_col` = 0
- static bool `cnv_makemap` = false
- static bool `cnv_writelist` = false
- static bool `cnv_list` = false
- static string `cnv_listname` = "inbix.cnv"
- static int `cnv_min_kb` = -1
- static double `cnv_min_score` = -1
- static int `cnv_min_sites` = -1
- static int `cnv_max_kb` = -1
- static double `cnv_max_score` = -1
- static int `cnv_max_sites` = -1
- static bool `cnv_del_only` = false
- static bool `cnv_dup_only` = false
- static int `cnv_type` = -1
- static bool `cnv_intersect` = false
- static bool `cnv_exclude` = false
- static string `cnv_intersect_file` = "inbix.file"
- static bool `cnv_intersect_subset` = false
- static string `cnv_intersect_subset_file` = "inbix.file"
- static bool `cnv_count` = false
- static double `cnv_overlap` = -1
- static bool `cnv_defined_overlap` = false
- static bool `cnv_indiv_perm` = false
- static bool `cnv_pos_perm` = false
- static bool `cnv_drop_no_segment` = false
- static bool `cnv_freq_method2` = false
- static double `cnv_freq_method2_threshold` = 0.8

- static bool `cnv_write_freq` = false
- static bool `cnv_freq_include` = false
- static bool `cnv_freq_include_below` = true
- static bool `cnv_freq_include_exact` = false
- static bool `cnv_freq_include_exact_exclude` = false
- static int `cnv_freq_include_cnt` = -1
- static bool `cnv_unique` = false
- static bool `cnv_intersect_writeback` = false
- static bool `cnv_intersect_writeback_verbose` = false
- static bool `cnv_disrupt` = false
- static int `cnv_region_border` = 0
- static bool `cnv_union_overlap` = false
- static bool `cnv_region_overlap` = false
- static bool `cnv_check_overlap` = false
- static bool `cnv_count_baseline` = false
- static string `cnv_count_baseline_file` = ""
- static bool `cnv_weighted_gene_test` = false
- static bool `cnv_enrichment_test` = false
- static int `cnv_en_model` = 4
- static bool `cnv_glm` = false
- static bool `seg_test_window` = false
- static double `seg_test_window_bp` = 100000
- static bool `seg_test_region` = false
- static bool `dosage_assoc` = false
- static string `dosage_file` = ""
- static bool `dosage_hard_call` = false
- static double `dosage_hard_call_thresh` = 0.99
- static int `dosage_hard_call_thresh2` = 0
- static bool `dosage_hasMap` = false
- static bool `write_dosage` = false

6.88.1 Detailed Description

Definition at line 106 of file options.h.

6.88.2 Member Data Documentation

6.88.2.1 adaptive_alpha

```
double par::adaptive_alpha = 0.00 [static]
```

Definition at line 1206 of file options.h.

6.88.2.2 adaptive_ci

```
double par::adaptive_ci = 0.0001 [static]
```

Definition at line 1207 of file options.h.

6.88.2.3 adaptive_interval

```
int par::adaptive_interval = 1 [static]
```

Definition at line 1204 of file options.h.

6.88.2.4 adaptive_interval2

```
double par::adaptive_interval2 = 0.001 [static]
```

Definition at line 1205 of file options.h.

6.88.2.5 adaptive_max

```
int par::adaptive_max = 1000000 [static]
```

Definition at line 1203 of file options.h.

6.88.2.6 adaptive_min

```
int par::adaptive_min = 5 [static]
```

Definition at line 1202 of file options.h.

6.88.2.7 adaptive_perm

```
bool par::adaptive_perm = true [static]
```

Definition at line 1201 of file options.h.

6.88.2.8 af_count

```
bool par::af_count = false [static]
```

Definition at line 1287 of file options.h.

6.88.2.9 af_file

```
string par::af_file = "inbix.frq" [static]
```

Definition at line 1286 of file options.h.

6.88.2.10 af_read

```
bool par::af_read = false [static]
```

Definition at line 1277 of file options.h.

6.88.2.11 af_write

```
bool par::af_write = false [static]
```

Definition at line 1278 of file options.h.

6.88.2.12 affpair

```
bool par::affpair = false [static]
```

Definition at line 804 of file options.h.

6.88.2.13 afni1dFile

```
string par::afnildFile = "" [static]
```

Definition at line 427 of file options.h.

6.88.2.14 afniNetwork

```
bool par::afniNetwork = false [static]
```

Definition at line 428 of file options.h.

6.88.2.15 algorithm_verbose

```
bool par::algorithm_verbose = false [static]
```

Definition at line 113 of file options.h.

6.88.2.16 algorithmMode

```
string par::algorithmMode = "relieff" [static]
```

Definition at line 440 of file options.h.

6.88.2.17 algorithmSeqMode

```
string par::algorithmSeqMode = "snr" [static]
```

Definition at line 441 of file options.h.

6.88.2.18 algorithmSeqS0

```
double par::algorithmSeqS0 = 0.05 [static]
```

Definition at line 444 of file options.h.

6.88.2.19 algorithmSnrMode

```
string par::algorithmSnrMode = "snr" [static]
```

Definition at line 442 of file options.h.

6.88.2.20 algorithmTstatMode

```
string par::algorithmTstatMode = "pval" [static]
```

Definition at line 443 of file options.h.

6.88.2.21 all_pheno

```
bool par::all_pheno = false [static]
```

Definition at line 594 of file options.h.

6.88.2.22 alpha

```
double par::alpha = DEFAULT_ALPHA [static]
```

Definition at line 478 of file options.h.

6.88.2.23 alwaysplitvars

```
vector< std::string > par::alwaysplitvars [static]
```

Definition at line 467 of file options.h.

6.88.2.24 analyse_cluster

```
int par::analyse_cluster = 0 [static]
```

Definition at line 1244 of file options.h.

6.88.2.25 annot_file

```
bool par::annot_file = false [static]
```

Definition at line 705 of file options.h.

6.88.2.26 annot_filename

```
string par::annot_filename = "" [static]
```

Definition at line 706 of file options.h.

6.88.2.27 assoc_counts

```
bool par::assoc_counts = false [static]
```

Definition at line 879 of file options.h.

6.88.2.28 assoc_glm

```
bool par::assoc_glm = false [static]
```

Definition at line 880 of file options.h.

6.88.2.29 assoc_glm_without_main_snp

```
bool par::assoc_glm_without_main_snp = false [static]
```

Definition at line 882 of file options.h.

6.88.2.30 assoc_gxe

```
bool par::assoc_gxe = false [static]
```

Definition at line 924 of file options.h.

6.88.2.31 assoc_test

```
bool par::assoc_test = false [static]
```

Definition at line 878 of file options.h.

6.88.2.32 assoc_test_alt_perm

```
bool par::assoc_test_alt_perm = false [static]
```

Definition at line 883 of file options.h.

6.88.2.33 BATCH_SIZE

```
int par::BATCH_SIZE = 500000 [static]
```

Definition at line 1297 of file options.h.

6.88.2.34 bitfilename

```
string par::bitfilename = "inbix.bed" [static]
```

Definition at line 297 of file options.h.

6.88.2.35 bitfilename_map

```
string par::bitfilename_map = "inbix.bim" [static]
```

Definition at line 299 of file options.h.

6.88.2.36 bmatch

```
bool par::bmatch = false [static]
```

Definition at line 1250 of file options.h.

6.88.2.37 bmatch_direction_filename

```
string par::bmatch_direction_filename = "inbix.bm" [static]
```

Definition at line 1254 of file options.h.

6.88.2.38 bmatch_filename

```
string par::bmatch_filename = "inbix.bmatch" [static]
```

Definition at line 1253 of file options.h.

6.88.2.39 bmatch_usertype

```
bool par::bmatch_usertype = false [static]
```

Definition at line 1251 of file options.h.

6.88.2.40 boot

```
bool par::boot = false [static]
```

Definition at line 1046 of file options.h.

6.88.2.41 bound

```
bool par::bound = true [static]
```

Definition at line 1191 of file options.h.

6.88.2.42 breslowday

```
bool par::breslowday = false [static]
```

Definition at line 964 of file options.h.

6.88.2.43 bt

```
bool par::bt = true [static]
```

Definition at line 791 of file options.h.

6.88.2.44 built_families

```
bool par::built_families = false [static]
```

Definition at line 948 of file options.h.

6.88.2.45 calc_SNPSNP_LD

```
bool par::calc_SNPSNP_LD = false [static]
```

Definition at line 1072 of file options.h.

6.88.2.46 calcFst

```
bool par::calcFst = false [static]
```

Definition at line 640 of file options.h.

6.88.2.47 caseweights

```
string par::caseweights = "" [static]
```

Definition at line 468 of file options.h.

6.88.2.48 catvars

```
vector< string > par::catvars [static]
```

Definition at line 480 of file options.h.

6.88.2.49 chap_add_grp_specifics

```
bool par::chap_add_grp_specifics = false [static]
```

Definition at line 876 of file options.h.

6.88.2.50 chap_drop_snps

```
bool par::chap_drop_snps = false [static]
```

Definition at line 874 of file options.h.

6.88.2.51 chap_drop_snps_list

```
string par::chap_drop_snps_list = "" [static]
```

Definition at line 875 of file options.h.

6.88.2.52 chap_entity

```
string par::chap_entity = "" [static]
```

Definition at line 869 of file options.h.

6.88.2.53 chap_haplotype_specific

```
bool par::chap_haplotype_specific = false [static]
```

Definition at line 868 of file options.h.

6.88.2.54 chap_independent_effect

```
bool par::chap_independent_effect = false [static]
```

Definition at line 867 of file options.h.

6.88.2.55 chap_model0

```
string par::chap_model0 = "" [static]
```

Definition at line 873 of file options.h.

6.88.2.56 chap_model1

```
string par::chap_model1 = "" [static]
```

Definition at line 872 of file options.h.

6.88.2.57 chap_sole_variant

```
bool par::chap_sole_variant = false [static]
```

Definition at line 864 of file options.h.

6.88.2.58 chap_sole_variant_specific_allele_list

```
string par::chap_sole_variant_specific_allele_list = "" [static]
```

Definition at line 866 of file options.h.

6.88.2.59 chap_sole_variant_specific_alleles

```
bool par::chap_sole_variant_specific_alleles = false [static]
```

Definition at line 865 of file options.h.

6.88.2.60 chap_specified_groups

```
bool par::chap_specified_groups = false [static]
```

Definition at line 870 of file options.h.

6.88.2.61 chap_specified_snps

```
bool par::chap_specified_snps = false [static]
```

Definition at line 871 of file options.h.

6.88.2.62 chap_test

```
bool par::chap_test = false [static]
```

Definition at line 863 of file options.h.

6.88.2.63 check_sex

```
bool par::check_sex = false [static]
```

Definition at line 1092 of file options.h.

6.88.2.64 chr_code

```
vector< string > par::chr_code [static]
```

Definition at line 770 of file options.h.

6.88.2.65 chr_haploid

```
vector< bool > par::chr_haploid [static]
```

Definition at line 767 of file options.h.

6.88.2.66 chr_map

```
map< string, int > par::chr_map [static]
```

Definition at line 771 of file options.h.

6.88.2.67 chr_sex

```
vector< bool > par::chr_sex [static]
```

Definition at line 768 of file options.h.

6.88.2.68 chr_Y

```
vector< bool > par::chr_Y [static]
```

Definition at line 769 of file options.h.

6.88.2.69 ci_level

```
double par::ci_level = 0.95 [static]
```

Definition at line 968 of file options.h.

6.88.2.70 ci_zt

```
double par::ci_zt = 0 [static]
```

Definition at line 969 of file options.h.

6.88.2.71 cli

```
bool par::cli = false [static]
```

Definition at line 324 of file options.h.

6.88.2.72 clist

```
bool par::clist = false [static]
```

Definition at line 337 of file options.h.

6.88.2.73 clist_filename

```
string par::clist_filename = "inbix cov" [static]
```

Definition at line 605 of file options.h.

6.88.2.74 `clist_number`

```
int par::clist_number = 0 [static]
```

Definition at line 562 of file options.h.

6.88.2.75 `clist_selection`

```
bool par::clist_selection = false [static]
```

Definition at line 582 of file options.h.

6.88.2.76 `clist_selection_name`

```
bool par::clist_selection_name = false [static]
```

Definition at line 583 of file options.h.

6.88.2.77 `clist_selection_number`

```
bool par::clist_selection_number = false [static]
```

Definition at line 584 of file options.h.

6.88.2.78 `clist_selection_string`

```
string par::clist_selection_string = "" [static]
```

Definition at line 585 of file options.h.

6.88.2.79 `clumpld`

```
bool par::clumpld = false [static]
```

Definition at line 722 of file options.h.

6.88.2.80 clumpld_annot

```
bool par::clumpld_annot = false [static]
```

Definition at line 735 of file options.h.

6.88.2.81 clumpld_annot_fields

```
string par::clumpld_annot_fields = "" [static]
```

Definition at line 736 of file options.h.

6.88.2.82 clumpld_best

```
bool par::clumpld_best = false [static]
```

Definition at line 723 of file options.h.

6.88.2.83 clumpld_column

```
string par::clumpld_column = "P" [static]
```

Definition at line 725 of file options.h.

6.88.2.84 clumpld_indep

```
bool par::clumpld_indep = true [static]
```

Definition at line 727 of file options.h.

6.88.2.85 clumpld_index1

```
bool par::clumpld_index1 = false [static]
```

Definition at line 732 of file options.h.

6.88.2.86 clumpld_kb

```
int par::clumpld_kb = 250000 [static]
```

Definition at line 728 of file options.h.

6.88.2.87 clumpld_min

```
int par::clumpld_min = 0 [static]
```

Definition at line 739 of file options.h.

6.88.2.88 clumpld_only_show_replications

```
bool par::clumpld_only_show_replications = false [static]
```

Definition at line 733 of file options.h.

6.88.2.89 clumpld_only_show_replications_list

```
bool par::clumpld_only_show_replications_list = false [static]
```

Definition at line 734 of file options.h.

6.88.2.90 clumpld_p1

```
double par::clumpld_p1 = 1e-4 [static]
```

Definition at line 730 of file options.h.

6.88.2.91 clumpld_p2

```
double par::clumpld_p2 = 1e-2 [static]
```

Definition at line 731 of file options.h.

6.88.2.92 clumpld_r2

```
double par::clumpld_r2 = 0.5 [static]
```

Definition at line 729 of file options.h.

6.88.2.93 clumpld_range_annotation

```
bool par::clumpld_range_annotation = false [static]
```

Definition at line 738 of file options.h.

6.88.2.94 clumpld_range_file

```
string par::clumpld_range_file = "range.list" [static]
```

Definition at line 737 of file options.h.

6.88.2.95 clumpld_results

```
string par::clumpld_results = "inbix.assoc" [static]
```

Definition at line 724 of file options.h.

6.88.2.96 clumpld_verbose

```
bool par::clumpld_verbose = false [static]
```

Definition at line 726 of file options.h.

6.88.2.97 cluster

```
bool par::cluster = false [static]
```

Definition at line 1222 of file options.h.

6.88.2.98 cluster_euclidean

```
bool par::cluster_euclidean = false [static]
```

Definition at line 1223 of file options.h.

6.88.2.99 cluster_group_avg

```
bool par::cluster_group_avg = false [static]
```

Definition at line 1224 of file options.h.

6.88.2.100 cluster_ibm_constraint

```
bool par::cluster_ibm_constraint = false [static]
```

Definition at line 1230 of file options.h.

6.88.2.101 cluster_ibm_constraint_value

```
double par::cluster_ibm_constraint_value = 0 [static]
```

Definition at line 1231 of file options.h.

6.88.2.102 cluster_mds_dim

```
int par::cluster_mds_dim = 2 [static]
```

Definition at line 1227 of file options.h.

6.88.2.103 cluster_missing

```
bool par::cluster_missing = false [static]
```

Definition at line 1232 of file options.h.

6.88.2.104 cluster_on_mcc

```
bool par::cluster_on_mcc = false [static]
```

Definition at line 1246 of file options.h.

6.88.2.105 cluster_on_phenotype

```
bool par::cluster_on_phenotype = false [static]
```

Definition at line 1245 of file options.h.

6.88.2.106 cluster_plot

```
bool par::cluster_plot = false [static]
```

Definition at line 1225 of file options.h.

6.88.2.107 cluster_selcon

```
bool par::cluster_selcon = false [static]
```

Definition at line 1233 of file options.h.

6.88.2.108 cluster_selcon_file

```
string par::cluster_selcon_file = "inbix.clst" [static]
```

Definition at line 1234 of file options.h.

6.88.2.109 cm_map

```
bool par::cm_map = false [static]
```

Definition at line 608 of file options.h.

6.88.2.110 CMH_test_1

```
bool par::CMH_test_1 = false [static]
```

Definition at line 961 of file options.h.

6.88.2.111 CMH_test_2

```
bool par::CMH_test_2 = false [static]
```

Definition at line 962 of file options.h.

6.88.2.112 CMH_test_ORD

```
bool par::CMH_test_ORD = false [static]
```

Definition at line 963 of file options.h.

6.88.2.113 cnv_check_overlap

```
bool par::cnv_check_overlap = false [static]
```

Definition at line 1343 of file options.h.

6.88.2.114 cnv_col

```
int par::cnv_col = 0 [static]
```

Definition at line 1303 of file options.h.

6.88.2.115 cnv_count

```
bool par::cnv_count = false [static]
```

Definition at line 1322 of file options.h.

6.88.2.116 cnv_count_baseline

```
bool par::cnv_count_baseline = false [static]
```

Definition at line 1344 of file options.h.

6.88.2.117 cnv_count_baseline_file

```
string par::cnv_count_baseline_file = "" [static]
```

Definition at line 1345 of file options.h.

6.88.2.118 cnv_defined_overlap

```
bool par::cnv_defined_overlap = false [static]
```

Definition at line 1324 of file options.h.

6.88.2.119 cnv_del_only

```
bool par::cnv_del_only = false [static]
```

Definition at line 1314 of file options.h.

6.88.2.120 cnv_disrupt

```
bool par::cnv_disrupt = false [static]
```

Definition at line 1339 of file options.h.

6.88.2.121 cnv_drop_no_segment

```
bool par::cnv_drop_no_segment = false [static]
```

Definition at line 1327 of file options.h.

6.88.2.122 cnv_dup_only

```
bool par::cnv_dup_only = false [static]
```

Definition at line 1315 of file options.h.

6.88.2.123 cnv_en_model

```
int par::cnv_en_model = 4 [static]
```

Definition at line 1348 of file options.h.

6.88.2.124 cnv_enrichment_test

```
bool par::cnv_enrichment_test = false [static]
```

Definition at line 1347 of file options.h.

6.88.2.125 cnv_exclude

```
bool par::cnv_exclude = false [static]
```

Definition at line 1318 of file options.h.

6.88.2.126 cnv_freq_include

```
bool par::cnv_freq_include = false [static]
```

Definition at line 1331 of file options.h.

6.88.2.127 cnv_freq_include_below

```
bool par::cnv_freq_include_below = true [static]
```

Definition at line 1332 of file options.h.

6.88.2.128 cnv_freq_include_cnt

```
int par::cnv_freq_include_cnt = -1 [static]
```

Definition at line 1335 of file options.h.

6.88.2.129 cnv_freq_include_exact

```
bool par::cnv_freq_include_exact = false [static]
```

Definition at line 1333 of file options.h.

6.88.2.130 cnv_freq_include_exact_exclude

```
bool par::cnv_freq_include_exact_exclude = false [static]
```

Definition at line 1334 of file options.h.

6.88.2.131 cnv_freq_method2

```
bool par::cnv_freq_method2 = false [static]
```

Definition at line 1328 of file options.h.

6.88.2.132 cnv_freq_method2_threshold

```
double par::cnv_freq_method2_threshold = 0.8 [static]
```

Definition at line 1329 of file options.h.

6.88.2.133 cnv_glm

```
bool par::cnv_glm = false [static]
```

Definition at line 1349 of file options.h.

6.88.2.134 cnv_indiv_perm

```
bool par::cnv_indiv_perm = false [static]
```

Definition at line 1325 of file options.h.

6.88.2.135 cnv_intersect

```
bool par::cnv_intersect = false [static]
```

Definition at line 1317 of file options.h.

6.88.2.136 cnv_intersect_file

```
string par::cnv_intersect_file = "inbix.file" [static]
```

Definition at line 1319 of file options.h.

6.88.2.137 cnv_intersect_subset

```
bool par::cnv_intersect_subset = false [static]
```

Definition at line 1320 of file options.h.

6.88.2.138 cnv_intersect_subset_file

```
string par::cnv_intersect_subset_file = "inbix.file" [static]
```

Definition at line 1321 of file options.h.

6.88.2.139 cnv_intersect_writeback

```
bool par::cnv_intersect_writeback = false [static]
```

Definition at line 1337 of file options.h.

6.88.2.140 cnv_intersect_writeback_verbose

```
bool par::cnv_intersect_writeback_verbose = false [static]
```

Definition at line 1338 of file options.h.

6.88.2.141 cnv_list

```
bool par::cnv_list = false [static]
```

Definition at line 1306 of file options.h.

6.88.2.142 cnv_listname

```
string par::cnv_listname = "inbix.cnv" [static]
```

Definition at line 1307 of file options.h.

6.88.2.143 cnv_makemap

```
bool par::cnv_makemap = false [static]
```

Definition at line 1304 of file options.h.

6.88.2.144 cnv_max_kb

```
int par::cnv_max_kb = -1 [static]
```

Definition at line 1311 of file options.h.

6.88.2.145 cnv_max_score

```
double par::cnv_max_score = -1 [static]
```

Definition at line 1312 of file options.h.

6.88.2.146 cnv_max_sites

```
int par::cnv_max_sites = -1 [static]
```

Definition at line 1313 of file options.h.

6.88.2.147 cnv_min_kb

```
int par::cnv_min_kb = -1 [static]
```

Definition at line 1308 of file options.h.

6.88.2.148 cnv_min_score

```
double par::cnv_min_score = -1 [static]
```

Definition at line 1309 of file options.h.

6.88.2.149 cnv_min_sites

```
int par::cnv_min_sites = -1 [static]
```

Definition at line 1310 of file options.h.

6.88.2.150 cnv_overlap

```
double par::cnv_overlap = -1 [static]
```

Definition at line 1323 of file options.h.

6.88.2.151 cnv_pos_perm

```
bool par::cnv_pos_perm = false [static]
```

Definition at line 1326 of file options.h.

6.88.2.152 cnv_region_border

```
int par::cnv_region_border = 0 [static]
```

Definition at line 1340 of file options.h.

6.88.2.153 cnv_region_overlap

```
bool par::cnv_region_overlap = false [static]
```

Definition at line 1342 of file options.h.

6.88.2.154 cnv_type

```
int par::cnv_type = -1 [static]
```

Definition at line 1316 of file options.h.

6.88.2.155 cnv_union_overlap

```
bool par::cnv_union_overlap = false [static]
```

Definition at line 1341 of file options.h.

6.88.2.156 cnv_unique

```
bool par::cnv_unique = false [static]
```

Definition at line 1336 of file options.h.

6.88.2.157 cnv_weighted_gene_test

```
bool par::cnv_weighted_gene_test = false [static]
```

Definition at line 1346 of file options.h.

6.88.2.158 cnv_write_freq

```
bool par::cnv_write_freq = false [static]
```

Definition at line 1330 of file options.h.

6.88.2.159 cnv_writelist

```
bool par::cnv_writelist = false [static]
```

Definition at line 1305 of file options.h.

6.88.2.160 coding01

```
bool par::coding01 = false [static]
```

Definition at line 792 of file options.h.

6.88.2.161 compound_genotype_code

```
bool par::compound_genotype_code = false [static]
```

Definition at line 265 of file options.h.

6.88.2.162 compress_file

```
bool par::compress_file = false [static]
```

Definition at line 304 of file options.h.

6.88.2.163 compress_filename

```
string par::compress_filename = "" [static]
```

Definition at line 306 of file options.h.

6.88.2.164 compress_genome

```
bool par::compress_genome = false [static]
```

Definition at line 1173 of file options.h.

6.88.2.165 conditioning_snp_name

```
string par::conditioning_snp_name = "rs1234" [static]
```

Definition at line 895 of file options.h.

6.88.2.166 conditioning_snp_single

```
bool par::conditioning_snp_single = false [static]
```

Definition at line 894 of file options.h.

6.88.2.167 conditioning_snps

```
bool par::conditioning_snps = false [static]
```

Definition at line 896 of file options.h.

6.88.2.168 conditioning_snps_file

```
string par::conditioning_snps_file = "inbix.list" [static]
```

Definition at line 897 of file options.h.

6.88.2.169 covar_file

```
bool par::covar_file = false [static]
```

Definition at line 336 of file options.h.

6.88.2.170 covar_filename

```
string par::covar_filename = "inbix.cov" [static]
```

Definition at line 604 of file options.h.

6.88.2.171 CP

```
bool par::CP = false [static]
```

Definition at line 803 of file options.h.

6.88.2.172 dcvar_chip_seq_file

```
string par::dcvar_chip_seq_file = "" [static]
```

Definition at line 364 of file options.h.

6.88.2.173 dcvar_gene_expression_file

```
string par::dcvar_gene_expression_file = "" [static]
```

Definition at line 363 of file options.h.

6.88.2.174 dcvar_genotypes_file

```
string par::dcvar_genotypes_file = "" [static]
```

Definition at line 361 of file options.h.

6.88.2.175 dcvar_pfilter_type

```
string par::dcvar_pfilter_type = "bon" [static]
```

Definition at line 358 of file options.h.

6.88.2.176 dcvar_pfilter_value

```
double par::dcvar_pfilter_value = 0.05 [static]
```

Definition at line 357 of file options.h.

6.88.2.177 dcvar_radius

```
uint par::dcvar_radius = 1000 [static]
```

Definition at line 368 of file options.h.

6.88.2.178 dcvar_resume.snp

```
bool par::dcvar_resume.snp = false [static]
```

Definition at line 366 of file options.h.

6.88.2.179 dcvar.snp.locations_file

```
string par::dcvar.snp.locations_file = "" [static]
```

Definition at line 362 of file options.h.

6.88.2.180 dcvar_var_model

```
string par::dcvar_var_model = "dom" [static]
```

Definition at line 359 of file options.h.

6.88.2.181 debug

```
bool par::debug = false [static]
```

Definition at line 116 of file options.h.

6.88.2.182 deconvolutionAlpha

```
double par::deconvolutionAlpha = 1.0 [static]
```

Definition at line 410 of file options.h.

6.88.2.183 deconvolutionBeta

```
double par::deconvolutionBeta = 0.9 [static]
```

Definition at line 411 of file options.h.

6.88.2.184 deconvolutionControl

```
int par::deconvolutionControl = 0 [static]
```

Definition at line 412 of file options.h.

6.88.2.185 depvarname

```
string par::depvarname = "Class" [static]
```

Definition at line 469 of file options.h.

6.88.2.186 dfam_sibs

```
bool par::dfam_sibs = true [static]
```

Definition at line 937 of file options.h.

6.88.2.187 dfam_tdt

```
bool par::dfam_tdt = true [static]
```

Definition at line 936 of file options.h.

6.88.2.188 dfam_unrelateds

```
bool par::dfam_unrelateds = true [static]
```

Definition at line 938 of file options.h.

6.88.2.189 discordant_parents

```
bool par::discordant_parents = false [static]
```

Definition at line 942 of file options.h.

6.88.2.190 disp_r1

```
bool par::disp_r1 = false [static]
```

Definition at line 1047 of file options.h.

6.88.2.191 disp_r2

```
bool par::disp_r2 = false [static]
```

Definition at line 1048 of file options.h.

6.88.2.192 disp_r_window

```
bool par::disp_r_window = false [static]
```

Definition at line 1049 of file options.h.

6.88.2.193 disp_r_window_kb

```
int par::disp_r_window_kb = 1000000 [static]
```

Definition at line 1051 of file options.h.

6.88.2.194 disp_r_window_r2

```
double par::disp_r_window_r2 = 0.2 [static]
```

Definition at line 1052 of file options.h.

6.88.2.195 disp_r_window.snp

```
int par::disp_r_window.snp = 10 [static]
```

Definition at line 1050 of file options.h.

6.88.2.196 display_ci

```
bool par::display_ci = false [static]
```

Definition at line 970 of file options.h.

6.88.2.197 display_cnv_track

```
bool par::display_cnv_track = false [static]
```

Definition at line 1302 of file options.h.

6.88.2.198 display_hap_freqs

```
bool par::display_hap_freqs = false [static]
```

Definition at line 826 of file options.h.

6.88.2.199 display_phase_probs

```
bool par::display_phase_probs = false [static]
```

Definition at line 859 of file options.h.

6.88.2.200 display_phase_probs_wide

```
bool par::display_phase_probs_wide = false [static]
```

Definition at line 860 of file options.h.

6.88.2.201 display_segment_long

```
bool par::display_segment_long = false [static]
```

Definition at line 1301 of file options.h.

6.88.2.202 distance_matrix

```
bool par::distance_matrix = false [static]
```

Definition at line 1221 of file options.h.

6.88.2.203 distanceMatrixFilename

```
string par::distanceMatrixFilename = "" [static]
```

Definition at line 437 of file options.h.

6.88.2.204 do_coexpression_all

```
bool par::do_coexpression_all = false [static]
```

Definition at line 345 of file options.h.

6.88.2.205 do_coexpression_casecontrol

```
bool par::do_coexpression_casecontrol = false [static]
```

Definition at line 346 of file options.h.

6.88.2.206 do_covariance_matrix

```
bool par::do_covariance_matrix = false [static]
```

Definition at line 344 of file options.h.

6.88.2.207 do_dcgain_abs

```
bool par::do_dcgain_abs = false [static]
```

Definition at line 353 of file options.h.

6.88.2.208 do_dcvar

```
bool par::do_dcvar = false [static]
```

Definition at line 355 of file options.h.

6.88.2.209 do_dcvar_chipseq

```
bool par::do_dcvar_chipseq = false [static]
```

Definition at line 365 of file options.h.

6.88.2.210 do_dcvar_pfilter

```
bool par::do_dcvar_pfilter = false [static]
```

Definition at line 356 of file options.h.

6.88.2.211 do_deconvolution

```
bool par::do_deconvolution = false [static]
```

Definition at line 409 of file options.h.

6.88.2.212 do_differential_coexpression

```
bool par::do_differential_coexpression = false [static]
```

Definition at line 352 of file options.h.

6.88.2.213 do_differential_modularity

```
bool par::do_differential_modularity = false [static]
```

Definition at line 371 of file options.h.

6.88.2.214 do_dmgain_abs

```
bool par::do_dmgain_abs = false [static]
```

Definition at line 372 of file options.h.

6.88.2.215 do_ec

```
bool par::do_ec = false [static]
```

Definition at line 493 of file options.h.

6.88.2.216 do_ec_privacy

```
bool par::do_ec_privacy = false [static]
```

Definition at line 505 of file options.h.

6.88.2.217 do_iqtl

```
bool par::do_iqtl = false [static]
```

Definition at line 545 of file options.h.

6.88.2.218 do_iqtl_tf

```
bool par::do_iqtl_tf = false [static]
```

Definition at line 552 of file options.h.

6.88.2.219 do_iterative_removal

```
bool par::do_iterative_removal = false [static]
```

Definition at line 435 of file options.h.

6.88.2.220 do_iterative_removal_ec

```
bool par::do_iterative_removal_ec = false [static]
```

Definition at line 494 of file options.h.

6.88.2.221 do_iterative_write_scores

```
bool par::do_iterative_write_scores = false [static]
```

Definition at line 495 of file options.h.

6.88.2.222 do_kopt

```
bool par::do_kopt = false [static]
```

Definition at line 436 of file options.h.

6.88.2.223 do_modularity

```
bool par::do_modularity = false [static]
```

Definition at line 415 of file options.h.

6.88.2.224 do_normalize_scores

```
bool par::do_normalize_scores = false [static]
```

Definition at line 434 of file options.h.

6.88.2.225 do_not_load_snps

```
bool par::do_not_load_snps = false [static]
```

Definition at line 122 of file options.h.

6.88.2.226 do_numeric_extract

```
bool par::do_numeric_extract = false [static]
```

Definition at line 348 of file options.h.

6.88.2.227 do_numeric_lowval_filter

```
bool par::do_numeric_lowval_filter = false [static]
```

Definition at line 379 of file options.h.

6.88.2.228 do_numeric_lowvar_filter

```
bool par::do_numeric_lowvar_filter = false [static]
```

Definition at line 380 of file options.h.

6.88.2.229 do_numeric_standardize

```
bool par::do_numeric_standardize = false [static]
```

Definition at line 375 of file options.h.

6.88.2.230 do_numeric_summary

```
bool par::do_numeric_summary = false [static]
```

Definition at line 347 of file options.h.

6.88.2.231 do_randomforest

```
bool par::do_randomforest = false [static]
```

Definition at line 466 of file options.h.

6.88.2.232 do_ranker_permutation

```
bool par::do_ranker_permutation = false [static]
```

Definition at line 534 of file options.h.

6.88.2.233 do_ranking

```
bool par::do_ranking = false [static]
```

Definition at line 527 of file options.h.

6.88.2.234 do_regain

```
bool par::do_regain = false [static]
```

Definition at line 385 of file options.h.

6.88.2.235 do_regain_post

```
bool par::do_regain_post = false [static]
```

Definition at line 386 of file options.h.

6.88.2.236 do_regain_pvalue_threshold

```
bool par::do_regain_pvalue_threshold = false [static]
```

Definition at line 387 of file options.h.

6.88.2.237 do_relieff

```
bool par::do_relieff = false [static]
```

Definition at line 431 of file options.h.

6.88.2.238 do_rfprobability

```
bool par::do_rfprobability = false [static]
```

Definition at line 484 of file options.h.

6.88.2.239 do_ripm

```
bool par::do_ripm = false [static]
```

Definition at line 516 of file options.h.

6.88.2.240 do_write_best_k

```
bool par::do_write_best_k = false [static]
```

Definition at line 432 of file options.h.

6.88.2.241 do_write_each_k_scores

```
bool par::do_write_each_k_scores = false [static]
```

Definition at line 433 of file options.h.

6.88.2.242 done_global_pihat

```
bool par::done_global_pihat = false [static]
```

Definition at line 614 of file options.h.

6.88.2.243 dosage_assoc

```
bool par::dosage_assoc = false [static]
```

Definition at line 1355 of file options.h.

6.88.2.244 dosage_file

```
string par::dosage_file = "" [static]
```

Definition at line 1356 of file options.h.

6.88.2.245 dosage_hard_call

```
bool par::dosage_hard_call = false [static]
```

Definition at line 1358 of file options.h.

6.88.2.246 dosage_hard_call_thresh

```
double par::dosage_hard_call_thresh = 0.99 [static]
```

Definition at line 1359 of file options.h.

6.88.2.247 dosage_hard_call_thresh2

```
int par::dosage_hard_call_thresh2 = 0 [static]
```

Definition at line 1360 of file options.h.

6.88.2.248 dosage_hasMap

```
bool par::dosage_hasMap = false [static]
```

Definition at line 1361 of file options.h.

6.88.2.249 drop_sets

```
bool par::drop_sets = true [static]
```

Definition at line 1089 of file options.h.

6.88.2.250 dummy

```
bool par::dummy = false [static]
```

Definition at line 117 of file options.h.

6.88.2.251 dummy_nind

```
int par::dummy_nind = 0 [static]
```

Definition at line 118 of file options.h.

6.88.2.252 dummy_nsnp

```
int par::dummy_nsnp = 0 [static]
```

Definition at line 119 of file options.h.

6.88.2.253 dump_clst

```
bool par::dump_clst = false [static]
```

Definition at line 580 of file options.h.

6.88.2.254 dump_covar

```
bool par::dump_covar = false [static]
```

Definition at line 558 of file options.h.

6.88.2.255 dump_covar_dummy_coding

```
bool par::dump_covar_dummy_coding = false [static]
```

Definition at line 560 of file options.h.

6.88.2.256 dump_covar_with_phenotype

```
bool par::dump_covar_with_phenotype = false [static]
```

Definition at line 559 of file options.h.

6.88.2.257 dump_gene

```
bool par::dump_gene = false [static]
```

Definition at line 1018 of file options.h.

6.88.2.258 dump_genename

```
string par::dump_genename = "" [static]
```

Definition at line 1019 of file options.h.

6.88.2.259 dumpped

```
bool par::dumpped = false [static]
```

Definition at line 115 of file options.h.

6.88.2.260 ecAlgorithmSteps

```
string par::ecAlgorithmSteps = "all" [static]
```

Definition at line 496 of file options.h.

6.88.2.261 ecltAlgorithm

```
string par::ecltAlgorithm = "relieff" [static]
```

Definition at line 498 of file options.h.

6.88.2.262 eclterNumToRemove

```
unsigned int par::eclterNumToRemove = 0 [static]
```

Definition at line 501 of file options.h.

6.88.2.263 eclterPercentToRemove

```
unsigned int par::eclterPercentToRemove = 0 [static]
```

Definition at line 502 of file options.h.

6.88.2.264 ecMeAlgorithm

```
string par::ecMeAlgorithm = "randomforest" [static]
```

Definition at line 497 of file options.h.

6.88.2.265 ecNumTarget

```
unsigned int par::ecNumTarget = 0 [static]
```

Definition at line 500 of file options.h.

6.88.2.266 ecOptimizeTemp

```
bool par::ecOptimizeTemp = false [static]
```

Definition at line 499 of file options.h.

6.88.2.267 ecPrivacyFinalTemp

```
double par::ecPrivacyFinalTemp = 0.00001 [static]
```

Definition at line 510 of file options.h.

6.88.2.268 ecPrivacyHoldoutFile

```
string par::ecPrivacyHoldoutFile = "" [static]
```

Definition at line 507 of file options.h.

6.88.2.269 ecPrivacyPercentSignal

```
double par::ecPrivacyPercentSignal = 0.1 [static]
```

Definition at line 513 of file options.h.

6.88.2.270 ecPrivacyStartTemp

```
double par::ecPrivacyStartTemp = 0.1 [static]
```

Definition at line 509 of file options.h.

6.88.2.271 ecPrivacyTau

```
double par::ecPrivacyTau = 100 [static]
```

Definition at line 511 of file options.h.

6.88.2.272 ecPrivacyTestFile

```
string par::ecPrivacyTestFile = "" [static]
```

Definition at line 508 of file options.h.

6.88.2.273 ecPrivacyTrainFile

```
string par::ecPrivacyTrainFile = "" [static]
```

Definition at line 506 of file options.h.

6.88.2.274 ecPrivacyUpdateFrequency

```
uint par::ecPrivacyUpdateFrequency = 50 [static]
```

Definition at line 512 of file options.h.

6.88.2.275 elf_baseline

```
bool par::elf_baseline = false [static]
```

Definition at line 759 of file options.h.

6.88.2.276 elf_pcmode

```
bool par::elf_pcmode = false [static]
```

Definition at line 757 of file options.h.

6.88.2.277 elf_pcmode_2sided

```
bool par::elf_pcmode_2sided = false [static]
```

Definition at line 758 of file options.h.

6.88.2.278 epi_alpha1

```
double par::epi_alpha1 = 0.0001 [static]
```

Definition at line 1083 of file options.h.

6.88.2.279 epi_alpha2

```
double par::epi_alpha2 = 0.01 [static]
```

Definition at line 1084 of file options.h.

6.88.2.280 epi_caseonly

```
bool par::epi_caseonly = false [static]
```

Definition at line 1080 of file options.h.

6.88.2.281 epi_caseonly_kb_gap

```
double par::epi_caseonly_kb_gap = 1000 [static]
```

Definition at line 1081 of file options.h.

6.88.2.282 epi_filter

```
bool par::epi_filter = true [static]
```

Definition at line 1082 of file options.h.

6.88.2.283 epi_genebased

```
bool par::epi_genebased = false [static]
```

Definition at line 1086 of file options.h.

6.88.2.284 epi_quickscan

```
bool par::epi_quickscan = false [static]
```

Definition at line 1087 of file options.h.

6.88.2.285 epistasis

```
bool par::epistasis = false [static]
```

Definition at line 1077 of file options.h.

6.88.2.286 epsilon

```
double const par::epsilon = 1e-12 [static]
```

Definition at line 128 of file options.h.

6.88.2.287 exclude_file

```
string par::exclude_file = "inbix.list" [static]
```

Definition at line 1001 of file options.h.

6.88.2.288 exclude_set

```
bool par::exclude_set = false [static]
```

Definition at line 985 of file options.h.

6.88.2.289 exportArff

```
bool par::exportArff = false [static]
```

Definition at line 540 of file options.h.

6.88.2.290 exportDelimited

```
bool par::exportDelimited = false [static]
```

Definition at line 541 of file options.h.

6.88.2.291 exportDelimiter

```
string par::exportDelimiter = "\t" [static]
```

Definition at line 542 of file options.h.

6.88.2.292 extract_before_exclude

```
bool par::extract_before_exclude = true [static]
```

Definition at line 1272 of file options.h.

6.88.2.293 extract_file

```
string par::extract_file = "inbix.list" [static]
```

Definition at line 1002 of file options.h.

6.88.2.294 extract_set

```
bool par::extract_set = false [static]
```

Definition at line 984 of file options.h.

6.88.2.295 famfile

```
string par::famfile = "inbix.fam" [static]
```

Definition at line 298 of file options.h.

6.88.2.296 fast_binary

```
bool par::fast_binary = false [static]
```

Definition at line 296 of file options.h.

6.88.2.297 fast_epistasis

```
bool par::fast_epistasis = false [static]
```

Definition at line 1078 of file options.h.

6.88.2.298 fileroot

```
string par::fileroot = "inbix" [static]
```

Definition at line 312 of file options.h.

6.88.2.299 filter_cases

```
bool par::filter_cases = false [static]
```

Definition at line 795 of file options.h.

6.88.2.300 filter_controls

```
bool par::filter_controls = false [static]
```

Definition at line 796 of file options.h.

6.88.2.301 filter_females

```
bool par::filter_females = false [static]
```

Definition at line 798 of file options.h.

6.88.2.302 filter_filename

```
string par::filter_filename = "inbix.cov" [static]
```

Definition at line 606 of file options.h.

6.88.2.303 filter_founders

```
bool par::filter_founders = false [static]
```

Definition at line 799 of file options.h.

6.88.2.304 filter_males

```
bool par::filter_males = false [static]
```

Definition at line 797 of file options.h.

6.88.2.305 filter_nonfounders

```
bool par::filter_nonfounders = false [static]
```

Definition at line 800 of file options.h.

6.88.2.306 filter_on_covar

```
bool par::filter_on_covar = false [static]
```

Definition at line 561 of file options.h.

6.88.2.307 filter_value

```
string par::filter_value = "1" [static]
```

Definition at line 598 of file options.h.

6.88.2.308 fisher_test

```
bool par::fisher_test = false [static]
```

Definition at line 886 of file options.h.

6.88.2.309 FIX_IBD

```
Z par::FIX_IBD [static]
```

Definition at line 1217 of file options.h.

6.88.2.310 fix_lambda

```
bool par::fix_lambda = false [static]
```

Definition at line 977 of file options.h.

6.88.2.311 FIX_p

```
double par::FIX_p = 0.5 [static]
```

Definition at line 1218 of file options.h.

6.88.2.312 fix_prev

```
bool par::fix_prev = false [static]
```

Definition at line 806 of file options.h.

6.88.2.313 FIXED

```
bool par::FIXED = false [static]
```

Definition at line 1215 of file options.h.

6.88.2.314 FIXED_p

```
bool par::FIXED_p = false [static]
```

Definition at line 1216 of file options.h.

6.88.2.315 fixed_prev

```
double par::fixed_prev = 0 [static]
```

Definition at line 807 of file options.h.

6.88.2.316 flag

```
bool par::flag = false [static]
```

Definition at line 114 of file options.h.

6.88.2.317 flip_file

```
string par::flip_file = "inbix.flip" [static]
```

Definition at line 290 of file options.h.

6.88.2.318 flip_scan

```
bool par::flip_scan = false [static]
```

Definition at line 1056 of file options.h.

6.88.2.319 flip_scan_threshold

```
double par::flip_scan_threshold = 0.5 [static]
```

Definition at line 1057 of file options.h.

6.88.2.320 flip_scan_verbose

```
bool par::flip_scan_verbose = false [static]
```

Definition at line 1058 of file options.h.

6.88.2.321 flip_strand

```
bool par::flip_strand = false [static]
```

Definition at line 289 of file options.h.

6.88.2.322 flip_subset

```
bool par::flip_subset = false [static]
```

Definition at line 291 of file options.h.

6.88.2.323 flip_subset_file

```
string par::flip_subset_file = "inbix.file" [static]
```

Definition at line 292 of file options.h.

6.88.2.324 force_initial_cluster

```
bool par::force_initial_cluster = false [static]
```

Definition at line 1226 of file options.h.

6.88.2.325 force_span

```
bool par::force_span = false [static]
```

Definition at line 1161 of file options.h.

6.88.2.326 fraction

```
double par::fraction = 1.0 [static]
```

Definition at line 470 of file options.h.

6.88.2.327 fringe

```
double par::fringe = .01 [static]
```

Definition at line 610 of file options.h.

6.88.2.328 from_window

```
int par::from_window = 0 [static]
```

Definition at line 787 of file options.h.

6.88.2.329 full_model_assoc

```
bool par::full_model_assoc = false [static]
```

Definition at line 884 of file options.h.

6.88.2.330 fuzzy_homo

```
double par::fuzzy_homo = 0.99 [static]
```

Definition at line 1115 of file options.h.

6.88.2.331 gainMatrixFilename

```
string par::gainMatrixFilename = "" [static]
```

Definition at line 438 of file options.h.

6.88.2.332 geno_qual_file

```
string par::geno_qual_file = "dummy" [static]
```

Definition at line 1011 of file options.h.

6.88.2.333 geno_qual_max

```
double par::geno_qual_max = 1 [static]
```

Definition at line 1013 of file options.h.

6.88.2.334 geno_qual_min

```
double par::geno_qual_min = 0 [static]
```

Definition at line 1012 of file options.h.

6.88.2.335 genome_2sets

```
bool par::genome_2sets = false [static]
```

Definition at line 1177 of file options.h.

6.88.2.336 genome_groups

```
bool par::genome_groups = false [static]
```

Definition at line 1229 of file options.h.

6.88.2.337 genome_only_check_rels

```
bool par::genome_only_check_rels = false [static]
```

Definition at line 1174 of file options.h.

6.88.2.338 genome_output

```
bool par::genome_output = false [static]
```

Definition at line 1172 of file options.h.

6.88.2.339 genome_output_full

```
bool par::genome_output_full = false [static]
```

Definition at line 1176 of file options.h.

6.88.2.340 genome_output_minimal

```
bool par::genome_output_minimal = false [static]
```

Definition at line 1175 of file options.h.

6.88.2.341 genome_setlist1

```
string par::genome_setlist1 = "inbix.set1" [static]
```

Definition at line 1178 of file options.h.

6.88.2.342 genome_setlist2

```
string par::genome_setlist2 = "inbix.set2" [static]
```

Definition at line 1179 of file options.h.

6.88.2.343 genome_test

```
bool par::genome_test = false [static]
```

Definition at line 1180 of file options.h.

6.88.2.344 genome_test_min.snp

```
int par::genome_test_min.snp = 20 [static]
```

Definition at line 1182 of file options.h.

6.88.2.345 genome_test_threshold

```
double par::genome_test_threshold = 0.01 [static]
```

Definition at line 1181 of file options.h.

6.88.2.346 gettag_file

```
string par::gettag_file = "" [static]
```

Definition at line 717 of file options.h.

6.88.2.347 gettag_kb

```
int par::gettag_kb = 250000 [static]
```

Definition at line 719 of file options.h.

6.88.2.348 gettag_listall

```
bool par::gettag_listall = false [static]
```

Definition at line 720 of file options.h.

6.88.2.349 gettag_mode

```
bool par::gettag_mode = false [static]
```

Definition at line 714 of file options.h.

6.88.2.350 gettag_mode1

```
bool par::gettag_mode1 = true [static]
```

Definition at line 715 of file options.h.

6.88.2.351 gettag_mode2

```
bool par::gettag_mode2 = false [static]
```

Definition at line 716 of file options.h.

6.88.2.352 gettag_r2

```
double par::gettag_r2 = 0.8 [static]
```

Definition at line 718 of file options.h.

6.88.2.353 gfamfile

```
string par::gfamfile = "inbix.fam" [static]
```

Definition at line 284 of file options.h.

6.88.2.354 glm_dominant

```
bool par::glm_dominant = false [static]
```

Definition at line 902 of file options.h.

6.88.2.355 glm_no_auto_sex_effect

```
bool par::glm_no_auto_sex_effect = false [static]
```

Definition at line 901 of file options.h.

6.88.2.356 glm_recessive

```
bool par::glm_recessive = false [static]
```

Definition at line 903 of file options.h.

6.88.2.357 glm_sex_effect

```
bool par::glm_sex_effect = false [static]
```

Definition at line 900 of file options.h.

6.88.2.358 glm_user_parameters

```
bool par::glm_user_parameters = false [static]
```

Definition at line 914 of file options.h.

6.88.2.359 glm_user_test

```
bool par::glm_user_test = false [static]
```

Definition at line 913 of file options.h.

6.88.2.360 gmapfile

```
string par::gmapfile = "inbix.map" [static]
```

Definition at line 283 of file options.h.

6.88.2.361 gmulti_output

```
bool par::gmulti_output = false [static]
```

Definition at line 1170 of file options.h.

6.88.2.362 gmlink

```
bool par::gmlink = false [static]
```

Definition at line 323 of file options.h.

6.88.2.363 greport

```
bool par::greport = false [static]
```

Definition at line 698 of file options.h.

6.88.2.364 greport_display_empty

```
bool par::greport_display_empty = false [static]
```

Definition at line 703 of file options.h.

6.88.2.365 greport_gene_list

```
string par::greport_gene_list = "file2" [static]
```

Definition at line 700 of file options.h.

6.88.2.366 greport_results

```
string par::greport_results = "file1" [static]
```

Definition at line 699 of file options.h.

6.88.2.367 greport_subset

```
bool par::greport_subset = false [static]
```

Definition at line 701 of file options.h.

6.88.2.368 greport_subset_file

```
string par::greport_subset_file = "file3" [static]
```

Definition at line 702 of file options.h.

6.88.2.369 grid

```
double par::grid = 0.005 [static]
```

Definition at line 609 of file options.h.

6.88.2.370 gvar

```
bool par::gvar = false [static]
```

Definition at line 278 of file options.h.

6.88.2.371 gvar_full_report

```
bool par::gvar_full_report = false [static]
```

Definition at line 287 of file options.h.

6.88.2.372 gvar_include_all_variants

```
bool par::gvar_include_all_variants = false [static]
```

Definition at line 286 of file options.h.

6.88.2.373 gvar_to_standard

```
bool par::gvar_to_standard = false [static]
```

Definition at line 280 of file options.h.

6.88.2.374 gvar_verbose_association

```
bool par::gvar_verbose_association = false [static]
```

Definition at line 282 of file options.h.

6.88.2.375 gvar_write

```
bool par::gvar_write = false [static]
```

Definition at line 279 of file options.h.

6.88.2.376 gvarfile

```
string par::gvarfile = "inbix.gvar" [static]
```

Definition at line 285 of file options.h.

6.88.2.377 hap_max_nf_phases

```
int par::hap_max_nf_phases = 1024 [static]
```

Definition at line 825 of file options.h.

6.88.2.378 hap_min_phase_prob

```
double par::hap_min_phase_prob = 1e-2 [static]
```

Definition at line 824 of file options.h.

6.88.2.379 hap_missing_genotype

```
double par::hap_missing_genotype = 0.5 [static]
```

Definition at line 823 of file options.h.

6.88.2.380 hap_post_prob

```
double par::hap_post_prob = 0.8 [static]
```

Definition at line 822 of file options.h.

6.88.2.381 hap_specific_snps

```
bool par::hap_specific_snps = false [static]
```

Definition at line 888 of file options.h.

6.88.2.382 hap_specific_snps_list

```
string par::hap_specific_snps_list = "" [static]
```

Definition at line 889 of file options.h.

6.88.2.383 haplo_plem_follow

```
bool par::haplo_plem_follow = false [static]
```

Definition at line 834 of file options.h.

6.88.2.384 haplo_plem_follow_fid

```
string par::haplo_plem_follow_fid = "FID1" [static]
```

Definition at line 836 of file options.h.

6.88.2.385 haplo_plem_follow_iid

```
string par::haplo_plem_follow_iid = "IID1" [static]
```

Definition at line 837 of file options.h.

6.88.2.386 haplo_plem_follow_ind

```
int par::haplo_plem_follow_ind = -1 [static]
```

Definition at line 835 of file options.h.

6.88.2.387 haplo_plem_iter

```
int par::haplo_plem_iter = 20 [static]
```

Definition at line 831 of file options.h.

6.88.2.388 haplo_plem_likelihood_iter

```
int par::haplo_plem_likelihood_iter = 5 [static]
```

Definition at line 839 of file options.h.

6.88.2.389 haplo_plem_meta_iter

```
int par::haplo_plem_meta_iter = 200 [static]
```

Definition at line 849 of file options.h.

6.88.2.390 haplo_plem_meta_likelihood_iter

```
int par::haplo_plem_meta_likelihood_iter = 5 [static]
```

Definition at line 850 of file options.h.

6.88.2.391 haplo_plem_meta_prune_haplotype

```
double par::haplo_plem_meta_prune_haplotype = 1e-6 [static]
```

Definition at line 847 of file options.h.

6.88.2.392 haplo_plem_meta_prune_phase

```
double par::haplo_plem_meta_prune_phase = 0.01 [static]
```

Definition at line 848 of file options.h.

6.88.2.393 haplo_plem_meta_tol

```
double par::haplo_plem_meta_tol = 1e-4 [static]
```

Definition at line 851 of file options.h.

6.88.2.394 haplo_plem_meta_window

```
int par::haplo_plem_meta_window = 2 [static]
```

Definition at line 846 of file options.h.

6.88.2.395 haplo_plem_nonzero_threshold

```
bool par::haplo_plem_nonzero_threshold = true [static]
```

Definition at line 844 of file options.h.

6.88.2.396 haplo_plem_original_overlap

```
int par::haplo_plem_original_overlap = 2 [static]
```

Definition at line 830 of file options.h.

6.88.2.397 haplo_plem_overlap

```
int par::haplo_plem_overlap = 2 [static]
```

Definition at line 829 of file options.h.

6.88.2.398 haplo_plem_verbose

```
bool par::haplo_plem_verbose = false [static]
```

Definition at line 832 of file options.h.

6.88.2.399 haplo_plem_window

```
int par::haplo_plem_window = 6 [static]
```

Definition at line 828 of file options.h.

6.88.2.400 haplo_plem_window_prune_phase

```
double par::haplo_plem_window_prune_phase = 1e-10 [static]
```

Definition at line 840 of file options.h.

6.88.2.401 haplo_plem_window_tol

```
double par::haplo_plem_window_tol = 1e-4 [static]
```

Definition at line 842 of file options.h.

6.88.2.402 haplo_plem_zero_threshold

```
double par::haplo_plem_zero_threshold = -1 [static]
```

Definition at line 843 of file options.h.

6.88.2.403 has_nonfounders

```
bool par::has_nonfounders = false [static]
```

Definition at line 620 of file options.h.

6.88.2.404 have_numerics

```
bool par::have_numerics = false [static]
```

Definition at line 341 of file options.h.

6.88.2.405 have_snps

```
bool par::have_snps = false [static]
```

Definition at line 340 of file options.h.

6.88.2.406 holdout

```
bool par::holdout = false [static]
```

Definition at line 471 of file options.h.

6.88.2.407 homo_miss_as_hom

```
bool par::homo_miss_as_hom = false [static]
```

Definition at line 1103 of file options.h.

6.88.2.408 homo_run

```
bool par::homo_run = false [static]
```

Definition at line 1097 of file options.h.

6.88.2.409 homo_run_consensus_match

```
bool par::homo_run_consensus_match = false [static]
```

Definition at line 1098 of file options.h.

6.88.2.410 homo_run_density

```
double par::homo_run_density = 50 [static]
```

Definition at line 1101 of file options.h.

6.88.2.411 homo_run_gap

```
int par::homo_run_gap = 1000 [static]
```

Definition at line 1102 of file options.h.

6.88.2.412 homo_run_het

```
int par::homo_run_het = 1 [static]
```

Definition at line 1113 of file options.h.

6.88.2.413 homo_run_kb

```
bool par::homo_run_kb = false [static]
```

Definition at line 1099 of file options.h.

6.88.2.414 homo_run_length_kb

```
int par::homo_run_length_kb = 1000 [static]
```

Definition at line 1111 of file options.h.

6.88.2.415 homo_run_length_snps

```
int par::homo_run_length_snps = 100 [static]
```

Definition at line 1112 of file options.h.

6.88.2.416 homo_run_snps

```
bool par::homo_run_snps = false [static]
```

Definition at line 1100 of file options.h.

6.88.2.417 homo_summary_allelic_match

```
bool par::homo_summary_allelic_match = false [static]
```

Definition at line 1114 of file options.h.

6.88.2.418 homo_threshold

```
double par::homo_threshold = 0.05 [static]
```

Definition at line 1109 of file options.h.

6.88.2.419 homo_windowAllowedHet

```
int par::homo_windowAllowedHet = 1 [static]
```

Definition at line 1107 of file options.h.

6.88.2.420 homo_windowAllowedMissing

```
int par::homo_windowAllowedMissing = 5 [static]
```

Definition at line 1108 of file options.h.

6.88.2.421 homo_windowKB

```
int par::homo_windowKB = 5000 [static]
```

Definition at line 1106 of file options.h.

6.88.2.422 homo_windowSize

```
int par::homoWindowSize = 50 [static]
```

Definition at line 1105 of file options.h.

6.88.2.423 homozyg_verbose

```
bool par::homozyg_verbose = false [static]
```

Definition at line 1116 of file options.h.

6.88.2.424 hotel

```
bool par::hotel = false [static]
```

Definition at line 1021 of file options.h.

6.88.2.425 HWD_filter_on_all

```
bool par::HWD_filter_on_all = false [static]
```

Definition at line 959 of file options.h.

6.88.2.426 HWD_limit

```
double par::HWD_limit = 0.001 [static]
```

Definition at line 957 of file options.h.

6.88.2.427 HWD_report

```
bool par::HWD_report = false [static]
```

Definition at line 956 of file options.h.

6.88.2.428 HWD_standard

```
bool par::HWD_standard = false [static]
```

Definition at line 958 of file options.h.

6.88.2.429 HWD_test

```
bool par::HWD_test = false [static]
```

Definition at line 955 of file options.h.

6.88.2.430 ibd_file

```
string par::ibd_file = "inbix.genome" [static]
```

Definition at line 1281 of file options.h.

6.88.2.431 ibd_file_list

```
string par::ibd_file_list = "inbix.genome.list" [static]
```

Definition at line 1284 of file options.h.

6.88.2.432 ibd_read

```
bool par::ibd_read = false [static]
```

Definition at line 1280 of file options.h.

6.88.2.433 ibd_read_list

```
bool par::ibd_read_list = false [static]
```

Definition at line 1283 of file options.h.

6.88.2.434 ibd_read_minimal

```
bool par::ibd_read_minimal = false [static]
```

Definition at line 1282 of file options.h.

6.88.2.435 IBD_threshold

```
double par::IBD_threshold = 0.2 [static]
```

Definition at line 1187 of file options.h.

6.88.2.436 IBD_within

```
bool par::IBD_within = false [static]
```

Definition at line 1193 of file options.h.

6.88.2.437 ibs_2only

```
bool par::ibs_2only = false [static]
```

Definition at line 1130 of file options.h.

6.88.2.438 ibs_inner_run_length_kb

```
int par::ibs_inner_run_length_kb = 100 [static]
```

Definition at line 1123 of file options.h.

6.88.2.439 ibs_inner_run_length_snp

```
int par::ibs_inner_run_length_snp = 20 [static]
```

Definition at line 1124 of file options.h.

6.88.2.440 ibs_inter.snp_distance

```
int par::ibs_inter.snp_distance = 1000000 [static]
```

Definition at line 1129 of file options.h.

6.88.2.441 ibs_join_kb

```
int par::ibs_join_kb = 100 [static]
```

Definition at line 1125 of file options.h.

6.88.2.442 ibs_join.snp

```
int par::ibs_join.snp = 1 [static]
```

Definition at line 1126 of file options.h.

6.88.2.443 ibs_run

```
bool par::ibs_run = false [static]
```

Definition at line 1119 of file options.h.

6.88.2.444 ibs_run_0

```
int par::ibs_run_0 = 1 [static]
```

Definition at line 1128 of file options.h.

6.88.2.445 ibs_run_density

```
double par::ibs_run_density = 0.01 [static]
```

Definition at line 1122 of file options.h.

6.88.2.446 ibs_run_length_kb

```
int par::ibs_run_length_kb = 100 [static]
```

Definition at line 1121 of file options.h.

6.88.2.447 ibs_run_length_snps

```
int par::ibs_run_length_snps = 100 [static]
```

Definition at line 1120 of file options.h.

6.88.2.448 ibs_run_missing

```
int par::ibs_run_missing = 2 [static]
```

Definition at line 1127 of file options.h.

6.88.2.449 ibs_sharing_test

```
bool par::ibs_sharing_test = false [static]
```

Definition at line 982 of file options.h.

6.88.2.450 ibs_test

```
bool par::ibs_test = false [static]
```

Definition at line 1183 of file options.h.

6.88.2.451 ibs_test_method2

```
bool par::ibs_test_method2 = false [static]
```

Definition at line 1185 of file options.h.

6.88.2.452 ibs_test_min.snp

```
int par::ibs_test_min.snp = 20 [static]
```

Definition at line 1184 of file options.h.

6.88.2.453 ibstest_gap

```
int par::ibstest_gap = 500000 [static]
```

Definition at line 1237 of file options.h.

6.88.2.454 idhelp

```
bool par::idhelp = false [static]
```

Definition at line 155 of file options.h.

6.88.2.455 idhelp_alias_update

```
bool par::idhelp_alias_update = true [static]
```

Definition at line 172 of file options.h.

6.88.2.456 idhelp_auto_alias

```
bool par::idhelp_auto_alias = false [static]
```

Definition at line 160 of file options.h.

6.88.2.457 idhelp_command

```
string par::idhelp_command [static]
```

Definition at line 174 of file options.h.

6.88.2.458 idhelp_dictionary

```
string par::idhelp_dictionary = "" [static]
```

Definition at line 157 of file options.h.

6.88.2.459 idhelp_dump_from_dict

```
bool par::idhelp_dump_from_dict = false [static]
```

Definition at line 158 of file options.h.

6.88.2.460 idhelp_dump_from_dict_cmd

```
string par::idhelp_dump_from_dict_cmd = "" [static]
```

Definition at line 159 of file options.h.

6.88.2.461 idhelp_input

```
string par::idhelp_input [static]
```

Definition at line 175 of file options.h.

6.88.2.462 idhelp_list_aliases

```
bool par::idhelp_list_aliases = false [static]
```

Definition at line 171 of file options.h.

6.88.2.463 idhelp_lookup

```
bool par::idhelp_lookup = false [static]
```

Definition at line 161 of file options.h.

6.88.2.464 idhelp_lookup_string

```
string par::idhelp_lookup_string = "" [static]
```

Definition at line 162 of file options.h.

6.88.2.465 idhelp_match

```
bool par::idhelp_match = false [static]
```

Definition at line 167 of file options.h.

6.88.2.466 idhelp_match_string

```
vector< string > par::idhelp_match_string [static]
```

Definition at line 168 of file options.h.

6.88.2.467 idhelp_no_dict

```
bool par::idhelp_no_dict = false [static]
```

Definition at line 170 of file options.h.

6.88.2.468 idhelp_output_delimit

```
string par::idhelp_output_delimit = " " [static]
```

Definition at line 156 of file options.h.

6.88.2.469 idhelp_replace

```
bool par::idhelp_replace = false [static]
```

Definition at line 165 of file options.h.

6.88.2.470 idhelp_replace_string

```
string par::idhelp_replace_string = "" [static]
```

Definition at line 166 of file options.h.

6.88.2.471 idhelp_subset

```
bool par::idhelp_subset = false [static]
```

Definition at line 163 of file options.h.

6.88.2.472 idhelp_subset_string

```
string par::idhelp_subset_string ="" [static]
```

Definition at line 164 of file options.h.

6.88.2.473 ignore_missing_sex

```
bool par::ignore_missing_sex = false [static]
```

Definition at line 333 of file options.h.

6.88.2.474 ignore_phenotypes

```
bool par::ignore_phenotypes = true [static]
```

Definition at line 794 of file options.h.

6.88.2.475 impmeasure

```
ImportanceMode par::impmeasure = IMP_NONE [static]
```

Definition at line 481 of file options.h.

6.88.2.476 impute_sex

```
bool par::impute_sex = false [static]
```

Definition at line 1093 of file options.h.

6.88.2.477 impute_tags

```
bool par::impute_tags = false [static]
```

Definition at line 812 of file options.h.

6.88.2.478 impute_verbose

```
bool par::impute_verbose = false [static]
```

Definition at line 654 of file options.h.

6.88.2.479 inbreeding

```
bool par::inbreeding = false [static]
```

Definition at line 1091 of file options.h.

6.88.2.480 inc_file

```
string par::inc_file = "inbix.inc" [static]
```

Definition at line 1291 of file options.h.

6.88.2.481 inc_read

```
bool par::inc_read = false [static]
```

Definition at line 1290 of file options.h.

6.88.2.482 inc_write

```
bool par::inc_write = false [static]
```

Definition at line 1289 of file options.h.

6.88.2.483 include_all_pairs

```
bool par::include_all_pairs = false [static]
```

Definition at line 1258 of file options.h.

6.88.2.484 include_all_z1

```
double par::include_all_z1 = 0.001 [static]
```

Definition at line 1259 of file options.h.

6.88.2.485 include_cluster

```
bool par::include_cluster = false [static]
```

Definition at line 1241 of file options.h.

6.88.2.486 include_cluster_filename

```
string par::include_cluster_filename = "inbix.clst" [static]
```

Definition at line 1243 of file options.h.

6.88.2.487 include_cluster_from_file

```
bool par::include_cluster_from_file = false [static]
```

Definition at line 1242 of file options.h.

6.88.2.488 ind_attrib_file

```
string par::ind_attrib_file = "" [static]
```

Definition at line 570 of file options.h.

6.88.2.489 ind_attrib_filter

```
bool par::ind_attrib_filter = false [static]
```

Definition at line 568 of file options.h.

6.88.2.490 ind_attrib_value

```
string par::ind_attrib_value = "" [static]
```

Definition at line 569 of file options.h.

6.88.2.491 indiv_report

```
bool par::indiv_report = false [static]
```

Definition at line 226 of file options.h.

6.88.2.492 indiv_report_fid

```
string par::indiv_report_fid = "fid1" [static]
```

Definition at line 227 of file options.h.

6.88.2.493 indiv_report_iid

```
string par::indiv_report_iid = "iid1" [static]
```

Definition at line 228 of file options.h.

6.88.2.494 inter_grid

```
int par::inter_grid = 2 [static]
```

Definition at line 612 of file options.h.

6.88.2.495 iqtl_coord_file

```
string par::iqtl_coord_file = "" [static]
```

Definition at line 547 of file options.h.

6.88.2.496 iqtl_expression_file

```
string par::iqtl_expression_file = "" [static]
```

Definition at line 546 of file options.h.

6.88.2.497 iqtl_interaction_full

```
bool par::iqtl_interaction_full = false [static]
```

Definition at line 550 of file options.h.

6.88.2.498 iqtl_local_cis

```
bool par::iqtl_local_cis = false [static]
```

Definition at line 548 of file options.h.

6.88.2.499 iqtl_pvalue

```
double par::iqtl_pvalue = 0.5 [static]
```

Definition at line 555 of file options.h.

6.88.2.500 iqtl_radius

```
int par::iqtl_radius = 1000 [static]
```

Definition at line 549 of file options.h.

6.88.2.501 iqtl_tf_coord_file

```
string par::iqtl_tf_coord_file = "" [static]
```

Definition at line 554 of file options.h.

6.88.2.502 iqtl_tf_radius

```
int par::iqtl_tf_radius = 0 [static]
```

Definition at line 553 of file options.h.

6.88.2.503 k

```
uint par::k = 10 [static]
```

Definition at line 445 of file options.h.

6.88.2.504 keep_file

```
string par::keep_file = "inbix.list" [static]
```

Definition at line 1003 of file options.h.

6.88.2.505 keep_indiv

```
bool par::keep_indiv = false [static]
```

Definition at line 1271 of file options.h.

6.88.2.506 keep_indiv_list

```
string par::keep_indiv_list = "inbix.list" [static]
```

Definition at line 1270 of file options.h.

6.88.2.507 koptBegin

```
uint par::koptBegin = 1 [static]
```

Definition at line 446 of file options.h.

6.88.2.508 koptEnd

```
uint par::koptEnd = 2 [static]
```

Definition at line 447 of file options.h.

6.88.2.509 koptStep

```
uint par::koptStep = 1 [static]
```

Definition at line 448 of file options.h.

6.88.2.510 lambda

```
double par::lambda = 1 [static]
```

Definition at line 978 of file options.h.

6.88.2.511 ld_anchor

```
bool par::ld_anchor = false [static]
```

Definition at line 1053 of file options.h.

6.88.2.512 ld_anchor_list

```
bool par::ld_anchor_list = false [static]
```

Definition at line 1054 of file options.h.

6.88.2.513 ld_SNP1

```
string par::ld_SNP1 = "" [static]
```

Definition at line 1073 of file options.h.

6.88.2.514 ld_SNP1_file

```
string par::ld_SNP1_file = "" [static]
```

Definition at line 1074 of file options.h.

6.88.2.515 ld_SNP2

```
string par::ld_SNP2 = "" [static]
```

Definition at line 1075 of file options.h.

6.88.2.516 lfamfile

```
string par::lfamfile [static]
```

Definition at line 272 of file options.h.

6.88.2.517 lfile_allele_count

```
bool par::lfile_allele_count = false [static]
```

Definition at line 202 of file options.h.

6.88.2.518 lfile_input

```
bool par::lfile_input = false [static]
```

Definition at line 273 of file options.h.

6.88.2.519 liability

```
bool par::liability = false [static]
```

Definition at line 314 of file options.h.

6.88.2.520 list_by_allele

```
bool par::list_by_allele = false [static]
```

Definition at line 222 of file options.h.

6.88.2.521 list_twolocus

```
bool par::list_twolocus = false [static]
```

Definition at line 223 of file options.h.

6.88.2.522 load_gvar

```
bool par::load_gvar = false [static]
```

Definition at line 281 of file options.h.

6.88.2.523 locked

```
bool par::locked = false [static]
```

Definition at line 1275 of file options.h.

6.88.2.524 logistic_model_iterations

```
int const par::logistic_model_iterations = 25 [static]
```

Definition at line 125 of file options.h.

6.88.2.525 logistic_tolerance_epsilon

```
double const par::logistic_tolerance_epsilon = 1e-8 [static]
```

Definition at line 126 of file options.h.

6.88.2.526 logscale

```
bool par::logscale = false [static]
```

Definition at line 980 of file options.h.

6.88.2.527 lookup

```
bool par::lookup = false [static]
```

Definition at line 142 of file options.h.

6.88.2.528 lookup2

```
bool par::lookup2 = false [static]
```

Definition at line 152 of file options.h.

6.88.2.529 lookup2_cmd

```
string par::lookup2_cmd = "" [static]
```

Definition at line 153 of file options.h.

6.88.2.530 lookup_gene

```
bool par::lookup_gene = false [static]
```

Definition at line 147 of file options.h.

6.88.2.531 lookup_gene_kb_window

```
int par::lookup_gene_kb_window = 20 [static]
```

Definition at line 149 of file options.h.

6.88.2.532 lookup_gene_name

```
string par::lookup_gene_name = "GENE1" [static]
```

Definition at line 146 of file options.h.

6.88.2.533 lookup_multiple_genes

```
bool par::lookup_multiple_genes = false [static]
```

Definition at line 148 of file options.h.

6.88.2.534 lookup_single_snp

```
bool par::lookup_single_snp = false [static]
```

Definition at line 143 of file options.h.

6.88.2.535 lookup_snp

```
string par::lookup_snp = "rs1234" [static]
```

Definition at line 145 of file options.h.

6.88.2.536 lookup_snp_kb_window

```
int par::lookup_snp_kb_window = 100 [static]
```

Definition at line 150 of file options.h.

6.88.2.537 lookup_to_file

```
bool par::lookup_to_file = false [static]
```

Definition at line 144 of file options.h.

6.88.2.538 loop_counter

```
int par::loop_counter = 0 [static]
```

Definition at line 220 of file options.h.

6.88.2.539 loop_over

```
bool par::loop_over = false [static]
```

Definition at line 218 of file options.h.

6.88.2.540 loop_over_filename

```
string par::loop_over_filename = "inbix.clst" [static]
```

Definition at line 221 of file options.h.

6.88.2.541 loop_over_label

```
string par::loop_over_label = "" [static]
```

Definition at line 219 of file options.h.

6.88.2.542 lpedfile

```
string par::lpedfile = "inbix.lgen" [static]
```

Definition at line 271 of file options.h.

6.88.2.543 m

```
uint par::m = 0 [static]
```

Definition at line 449 of file options.h.

6.88.2.544 m1

```
string par::m1 = "" [static]
```

Definition at line 783 of file options.h.

6.88.2.545 m2

```
string par::m2 = "" [static]
```

Definition at line 784 of file options.h.

6.88.2.546 make_blocks

```
bool par::make_blocks = false [static]
```

Definition at line 816 of file options.h.

6.88.2.547 make_founders

```
bool par::make_founders = false [static]
```

Definition at line 619 of file options.h.

6.88.2.548 make_minor_allele

```
bool par::make_minor_allele = true [static]
```

Definition at line 743 of file options.h.

6.88.2.549 make_missing_parents

```
bool par::make_missing_parents = false [static]
```

Definition at line 621 of file options.h.

6.88.2.550 make_pheno

```
bool par::make_pheno = false [static]
```

Definition at line 577 of file options.h.

6.88.2.551 make_pheno_filename

```
string par::make_pheno_filename = "" [static]
```

Definition at line 575 of file options.h.

6.88.2.552 make_pheno_present

```
bool par::make_pheno_present = false [static]
```

Definition at line 578 of file options.h.

6.88.2.553 make_pheno_value

```
string par::make_pheno_value = "" [static]
```

Definition at line 576 of file options.h.

6.88.2.554 make_set

```
bool par::make_set = false [static]
```

Definition at line 991 of file options.h.

6.88.2.555 make_set_border

```
int par::make_set_border = 0 [static]
```

Definition at line 993 of file options.h.

6.88.2.556 make_set_collapse

```
bool par::make_set_collapse = false [static]
```

Definition at line 994 of file options.h.

6.88.2.557 make_set_collapse_label

```
string par::make_set_collapse_label = "SET" [static]
```

Definition at line 996 of file options.h.

6.88.2.558 make_set_complement

```
bool par::make_set_complement = false [static]
```

Definition at line 997 of file options.h.

6.88.2.559 make_set_file

```
string par::make_set_file = "inbix.set" [static]
```

Definition at line 992 of file options.h.

6.88.2.560 make_set_ignore_group

```
bool par::make_set_ignore_group = false [static]
```

Definition at line 995 of file options.h.

6.88.2.561 make_tags

```
bool par::make_tags [static]
```

Definition at line 811 of file options.h.

6.88.2.562 map3

```
bool par::map3 = false [static]
```

Definition at line 313 of file options.h.

6.88.2.563 mapfile

```
string par::mapfile = "inbix.map" [static]
```

Definition at line 310 of file options.h.

6.88.2.564 mapfile_impute

```
string par::mapfile_impute = "inbix.impute.map" [static]
```

Definition at line 810 of file options.h.

6.88.2.565 mating_tests

```
bool par::mating_tests = false [static]
```

Definition at line 935 of file options.h.

6.88.2.566 matrix

```
bool par::matrix = false [static]
```

Definition at line 1220 of file options.h.

6.88.2.567 max_af

```
double par::max_af = 1 [static]
```

Definition at line 742 of file options.h.

6.88.2.568 max_cluster_case

```
int par::max_cluster_case = 0 [static]
```

Definition at line 1239 of file options.h.

6.88.2.569 max_cluster_control

```
int par::max_cluster_control = 0 [static]
```

Definition at line 1240 of file options.h.

6.88.2.570 max_cluster_N

```
int par::max_cluster_N = -1 [static]
```

Definition at line 1235 of file options.h.

6.88.2.571 max_cluster_size

```
int par::max_cluster_size = 0 [static]
```

Definition at line 1238 of file options.h.

6.88.2.572 MAX_CORR_PIHAT_PIHAT_G

```
double par::MAX_CORR_PIHAT_PIHAT_G = 0.9 [static]
```

Definition at line 1263 of file options.h.

6.88.2.573 MAX_GENO_MISSING

```
double par::MAX_GENO_MISSING = 0.1 [static]
```

Definition at line 1264 of file options.h.

6.88.2.574 max_hf

```
double par::max_hf = 1 [static]
```

Definition at line 746 of file options.h.

6.88.2.575 MAX_IND_MISSING

```
double par::MAX_IND_MISSING = 0.1 [static]
```

Definition at line 1265 of file options.h.

6.88.2.576 MAX_LINE_LENGTH

```
int par::MAX_LINE_LENGTH = 2000000 [static]
```

Definition at line 1266 of file options.h.

6.88.2.577 max_neighbour

```
int par::max_neighbour = 10 [static]
```

Definition at line 1248 of file options.h.

6.88.2.578 MAX_PIHAT

```
double par::MAX_PIHAT = 1.0000 [static]
```

Definition at line 1262 of file options.h.

6.88.2.579 maxMergeOrder

```
int par::maxMergeOrder = 4 [static]
```

Definition at line 520 of file options.h.

6.88.2.580 maxModuleSize

```
int par::maxModuleSize = 200 [static]
```

Definition at line 521 of file options.h.

6.88.2.581 mds_by_individual

```
bool par::mds_by_individual = true [static]
```

Definition at line 1228 of file options.h.

6.88.2.582 memmode

```
MemoryMode par::memmode = MEM_DOUBLE [static]
```

Definition at line 472 of file options.h.

6.88.2.583 MENDEL_ind

```
double par::MENDEL_ind = 0.1 [static]
```

Definition at line 953 of file options.h.

6.88.2.584 MENDEL_report

```
bool par::MENDEL_report = false [static]
```

Definition at line 951 of file options.h.

6.88.2.585 MENDEL.snp

```
double par::MENDEL.snp = 0.1 [static]
```

Definition at line 952 of file options.h.

6.88.2.586 MENDEL_test

```
bool par::MENDEL_test = false [static]
```

Definition at line 950 of file options.h.

6.88.2.587 merge_bedfile

```
string par::merge_bedfile = "merge.bed" [static]
```

Definition at line 242 of file options.h.

6.88.2.588 merge_bimfile

```
string par::merge_bimfile = "merge.bim" [static]
```

Definition at line 243 of file options.h.

6.88.2.589 merge_binary

```
bool par::merge_binary = false [static]
```

Definition at line 237 of file options.h.

6.88.2.590 merge_data

```
bool par::merge_data = false [static]
```

Definition at line 234 of file options.h.

6.88.2.591 merge_famfile

```
string par::merge_famfile = "merge.fam" [static]
```

Definition at line 244 of file options.h.

6.88.2.592 merge_force_strand

```
bool par::merge_force_strand = false [static]
```

Definition at line 235 of file options.h.

6.88.2.593 merge_list

```
bool par::merge_list = false [static]
```

Definition at line 238 of file options.h.

6.88.2.594 merge_list_filename

```
string par::merge_list_filename = "merge.list" [static]
```

Definition at line 239 of file options.h.

6.88.2.595 merge_mapfile

```
string par::merge_mapfile = "merge.map" [static]
```

Definition at line 241 of file options.h.

6.88.2.596 merge_mode

```
int par::merge_mode = 1 [static]
```

Definition at line 236 of file options.h.

6.88.2.597 merge_p

```
double par::merge_p = 0 [static]
```

Definition at line 1236 of file options.h.

6.88.2.598 merge_pedfile

```
string par::merge_pedfile = "merge.ped" [static]
```

Definition at line 240 of file options.h.

6.88.2.599 meta_analysis

```
bool par::meta_analysis = false [static]
```

Definition at line 708 of file options.h.

6.88.2.600 meta_files

```
vector< string > par::meta_files [static]
```

Definition at line 709 of file options.h.

6.88.2.601 meta_large_phase

```
bool par::meta_large_phase = false [static]
```

Definition at line 818 of file options.h.

6.88.2.602 min_af

```
double par::min_af = 0.01 [static]
```

Definition at line 741 of file options.h.

6.88.2.603 min_genome_cell

```
int par::min_genome_cell = 5 [static]
```

Definition at line 748 of file options.h.

6.88.2.604 min_hf

```
double par::min_hf = 0.01 [static]
```

Definition at line 745 of file options.h.

6.88.2.605 min_neighbour

```
int par::min_neighbour = 1 [static]
```

Definition at line 1247 of file options.h.

6.88.2.606 MIN_PIHAT

```
double par::MIN_PIHAT = 0.0025 [static]
```

Definition at line 1261 of file options.h.

6.88.2.607 minModuleSize

```
int par::minModuleSize = 30 [static]
```

Definition at line 522 of file options.h.

6.88.2.608 minprop

```
double par::minprop = DEFAULT_MINPROP [static]
```

Definition at line 479 of file options.h.

6.88.2.609 mishap_test

```
bool par::mishap_test = false [static]
```

Definition at line 637 of file options.h.

6.88.2.610 mishap_window

```
int par::mishap_window = 1 [static]
```

Definition at line 638 of file options.h.

6.88.2.611 miss_run

```
bool par::miss_run = false [static]
```

Definition at line 1132 of file options.h.

6.88.2.612 miss_run_length

```
int par::miss_run_length = 100 [static]
```

Definition at line 1133 of file options.h.

6.88.2.613 miss_run_length_kb

```
bool par::miss_run_length_kb = false [static]
```

Definition at line 1134 of file options.h.

6.88.2.614 miss_run_level

```
double par::miss_run_level = 0.80 [static]
```

Definition at line 1135 of file options.h.

6.88.2.615 missing_genotype

```
string par::missing_genotype = "0" [static]
```

Definition at line 326 of file options.h.

6.88.2.616 missing_genotype_explicit

```
bool par::missing_genotype_explicit = false [static]
```

Definition at line 330 of file options.h.

6.88.2.617 missing_phenotype

```
string par::missing_phenotype = "-9" [static]
```

Definition at line 328 of file options.h.

6.88.2.618 missing_phenotype_explicit

```
bool par::missing_phenotype_explicit = false [static]
```

Definition at line 331 of file options.h.

6.88.2.619 mk_datfile

```
bool par::mk_datfile = false [static]
```

Definition at line 1143 of file options.h.

6.88.2.620 modComputeHomophily

```
bool par::modComputeHomophily = false [static]
```

Definition at line 419 of file options.h.

6.88.2.621 modConnectivityThreshold

```
double par::modConnectivityThreshold = 0.0 [static]
```

Definition at line 416 of file options.h.

6.88.2.622 model_perm_best

```
bool par::model_perm_best = false [static]
```

Definition at line 918 of file options.h.

6.88.2.623 model_perm_dom

```
bool par::model_perm_dom = false [static]
```

Definition at line 920 of file options.h.

6.88.2.624 model_perm_gen

```
bool par::model_perm_gen = false [static]
```

Definition at line 919 of file options.h.

6.88.2.625 model_perm_rec

```
bool par::model_perm_rec = false [static]
```

Definition at line 921 of file options.h.

6.88.2.626 model_perm_trend

```
bool par::model_perm_trend = false [static]
```

Definition at line 922 of file options.h.

6.88.2.627 modEnableConnectivityThreshold

```
bool par::modEnableConnectivityThreshold = false [static]
```

Definition at line 417 of file options.h.

6.88.2.628 modFisherTransform

```
bool par::modFisherTransform = false [static]
```

Definition at line 422 of file options.h.

6.88.2.629 modFisherTransformCutoff

```
double par::modFisherTransformCutoff = 0.999999 [static]
```

Definition at line 423 of file options.h.

6.88.2.630 modPowerTransform

```
bool par::modPowerTransform = false [static]
```

Definition at line 420 of file options.h.

6.88.2.631 modPowerTransformExponent

```
double par::modPowerTransformExponent = 1 [static]
```

Definition at line 421 of file options.h.

6.88.2.632 modUseBinaryThreshold

```
bool par::modUseBinaryThreshold = false [static]
```

Definition at line 418 of file options.h.

6.88.2.633 mperm_rank

```
bool par::mperm_rank = false [static]
```

Definition at line 1200 of file options.h.

6.88.2.634 mperm_save_all

```
bool par::mperm_save_all = false [static]
```

Definition at line 1199 of file options.h.

6.88.2.635 mperm_save_best

```
bool par::mperm_save_best = false [static]
```

Definition at line 1198 of file options.h.

6.88.2.636 mtry

```
unsigned int par::mtry = 0 [static]
```

Definition at line 483 of file options.h.

6.88.2.637 mult_clst

```
int par::mult_clst = 1 [static]
```

Definition at line 596 of file options.h.

6.88.2.638 mult_covar

```
int par::mult_covar = 1 [static]
```

Definition at line 595 of file options.h.

6.88.2.639 mult_filter

```
int par::mult_filter = 1 [static]
```

Definition at line 597 of file options.h.

6.88.2.640 mult_pheno

```
int par::mult_pheno = 1 [static]
```

Definition at line 592 of file options.h.

6.88.2.641 multi_output

```
bool par::multi_output = false [static]
```

Definition at line 1169 of file options.h.

6.88.2.642 multiple_phenotype_file

```
string par::multiple_phenotype_file = "" [static]
```

Definition at line 573 of file options.h.

6.88.2.643 multiple_phenotypes

```
bool par::multiple_phenotypes = false [static]
```

Definition at line 572 of file options.h.

6.88.2.644 multtest

```
bool par::multtest = false [static]
```

Definition at line 975 of file options.h.

6.88.2.645 myfunction

```
bool par::myfunction = false [static]
```

Definition at line 108 of file options.h.

6.88.2.646 name_pheno

```
string par::name_pheno = "" [static]
```

Definition at line 593 of file options.h.

6.88.2.647 no_show_covar

```
bool par::no_show_covar = false [static]
```

Definition at line 557 of file options.h.

6.88.2.648 nrfthreads

```
unsigned int par::nrfthreads = DEFAULT_NUM_THREADS [static]
```

Definition at line 476 of file options.h.

6.88.2.649 ntree

```
unsigned int par::ntree = DEFAULT_NUM_TREE [static]
```

Definition at line 487 of file options.h.

6.88.2.650 nudge

```
bool par::nudge = false [static]
```

Definition at line 1190 of file options.h.

6.88.2.651 number_list_positive

```
bool par::number_list_positive = true [static]
```

Definition at line 601 of file options.h.

6.88.2.652 number_list_string

```
string par::number_list_string = "" [static]
```

Definition at line 600 of file options.h.

6.88.2.653 numDiffMetricName

```
string par::numDiffMetricName = "manhattan" [static]
```

Definition at line 452 of file options.h.

6.88.2.654 numeric_extract_file

```
string par::numeric_extract_file = "" [static]
```

Definition at line 349 of file options.h.

6.88.2.655 numeric_file

```
bool par::numeric_file = false [static]
```

Definition at line 342 of file options.h.

6.88.2.656 numeric_filename

```
string par::numeric_filename = "inbix.num" [static]
```

Definition at line 343 of file options.h.

6.88.2.657 numeric_lowval_percentile

```
double par::numeric_lowval_percentile = 0.1 [static]
```

Definition at line 381 of file options.h.

6.88.2.658 numeric_lowvar_percentile

```
double par::numeric_lowvar_percentile = 0.1 [static]
```

Definition at line 382 of file options.h.

6.88.2.659 oblig_clusters_filename

```
string par::oblig_clusters_filename = "inbix.clst" [static]
```

Definition at line 217 of file options.h.

6.88.2.660 oblig_missing

```
bool par::oblig_missing = false [static]
```

Definition at line 215 of file options.h.

6.88.2.661 oblig_missing_filename

```
string par::oblig_missing_filename = "inbix.zero" [static]
```

Definition at line 216 of file options.h.

6.88.2.662 opt

```
Options par::opt [static]
```

Definition at line 110 of file options.h.

6.88.2.663 OR_homog_test

```
bool par::OR_homog_test = false [static]
```

Definition at line 966 of file options.h.

6.88.2.664 out_missing_genotype

```
string par::out_missing_genotype = "0" [static]
```

Definition at line 327 of file options.h.

6.88.2.665 out_missing_phenotype

```
string par::out_missing_phenotype = "-9" [static]
```

Definition at line 329 of file options.h.

6.88.2.666 out_SNP_major

```
bool par::out_SNP_major = true [static]
```

Definition at line 302 of file options.h.

6.88.2.667 outlier_detection

```
bool par::outlier_detection = false [static]
```

Definition at line 1249 of file options.h.

6.88.2.668 output_file_name

```
string par::output_file_name = "inbix" [static]
```

Definition at line 321 of file options.h.

6.88.2.669 output_pheno_perm

```
bool par::output_pheno_perm = false [static]
```

Definition at line 891 of file options.h.

6.88.2.670 parameter_list

```
vector< int > par::parameter_list [static]
```

Definition at line 911 of file options.h.

6.88.2.671 parent_of_origin

```
bool par::parent_of_origin = false [static]
```

Definition at line 943 of file options.h.

6.88.2.672 ped_from_stdin

```
bool par::ped_from_stdin = false [static]
```

Definition at line 311 of file options.h.

6.88.2.673 ped_skip_fid

```
bool par::ped_skip_fid = false [static]
```

Definition at line 318 of file options.h.

6.88.2.674 ped_skip_parents

```
bool par::ped_skip_parents = false [static]
```

Definition at line 317 of file options.h.

6.88.2.675 ped_skip_pheno

```
bool par::ped_skip_pheno = false [static]
```

Definition at line 319 of file options.h.

6.88.2.676 ped_skip_sex

```
bool par::ped_skip_sex = false [static]
```

Definition at line 316 of file options.h.

6.88.2.677 pedfile

```
string par::pedfile = "inbix.ped" [static]
```

Definition at line 309 of file options.h.

6.88.2.678 perm_count

```
bool par::perm_count = false [static]
```

Definition at line 1197 of file options.h.

6.88.2.679 perm_genedrop

```
bool par::perm_genedrop = false [static]
```

Definition at line 1209 of file options.h.

6.88.2.680 perm_genedrop_and_swap

```
bool par::perm_genedrop_and_swap = false [static]
```

Definition at line 1210 of file options.h.

6.88.2.681 perm_genedrop_parents

```
bool par::perm_genedrop_parents = false [static]
```

Definition at line 1212 of file options.h.

6.88.2.682 perm_genedrop_sibships

```
bool par::perm_genedrop_sibships = false [static]
```

Definition at line 1213 of file options.h.

6.88.2.683 perm_genedrop_unrel

```
bool par::perm_genedrop_unrel = false [static]
```

Definition at line 1211 of file options.h.

6.88.2.684 perm_POO_best

```
bool par::perm_POO_best = false [static]
```

Definition at line 947 of file options.h.

6.88.2.685 perm_POO_mat

```
bool par::perm_POO_mat = false [static]
```

Definition at line 946 of file options.h.

6.88.2.686 perm_POO_pat

```
bool par::perm_POO_pat = false [static]
```

Definition at line 945 of file options.h.

6.88.2.687 perm_POO_poo

```
bool par::perm_POO_poo = true [static]
```

Definition at line 944 of file options.h.

6.88.2.688 perm_TDT_basic

```
bool par::perm_TDT_basic = true [static]
```

Definition at line 940 of file options.h.

6.88.2.689 perm_TDT_parent

```
bool par::perm_TDT_parent = false [static]
```

Definition at line 941 of file options.h.

6.88.2.690 permute

```
bool par::permute = false [static]
```

Definition at line 1195 of file options.h.

6.88.2.691 permute_within_sol

```
bool par::permute_within_sol = false [static]
```

Definition at line 1045 of file options.h.

6.88.2.692 pfilter

```
bool par::pfilter = false [static]
```

Definition at line 972 of file options.h.

6.88.2.693 pfvalue

```
double par::pfvalue = 1e-5 [static]
```

Definition at line 973 of file options.h.

6.88.2.694 phase_hap_all

```
bool par::phase_hap_all = false [static]
```

Definition at line 821 of file options.h.

6.88.2.695 phase_snps

```
bool par::phase_snps = false [static]
```

Definition at line 820 of file options.h.

6.88.2.696 pheno_file

```
bool par::pheno_file = false [static]
```

Definition at line 335 of file options.h.

6.88.2.697 pheno_filename

```
string par::pheno_filename = "inbix.phe" [static]
```

Definition at line 603 of file options.h.

6.88.2.698 pihat_filter

```
bool par::pihat_filter = true [static]
```

Definition at line 1171 of file options.h.

6.88.2.699 plink

```
bool par::plink = false [static]
```

Definition at line 1299 of file options.h.

6.88.2.700 plist

```
bool par::plist = false [static]
```

Definition at line 229 of file options.h.

6.88.2.701 plist_fid1

```
string par::plist_fid1 = "" [static]
```

Definition at line 230 of file options.h.

6.88.2.702 plist_fid2

```
string par::plist_fid2 = "" [static]
```

Definition at line 232 of file options.h.

6.88.2.703 plist_iid1

```
string par::plist_iid1 = "" [static]
```

Definition at line 231 of file options.h.

6.88.2.704 plist_iid2

```
string par::plist_iid2 = "" [static]
```

Definition at line 233 of file options.h.

6.88.2.705 plist_number

```
int par::plist_number = 0 [static]
```

Definition at line 563 of file options.h.

6.88.2.706 plist_selection

```
bool par::plist_selection = false [static]
```

Definition at line 587 of file options.h.

6.88.2.707 plist_selection_name

```
bool par::plist_selection_name = false [static]
```

Definition at line 588 of file options.h.

6.88.2.708 plist_selection_number

```
bool par::plist_selection_number = false [static]
```

Definition at line 589 of file options.h.

6.88.2.709 plist_selection_string

```
string par::plist_selection_string = "" [static]
```

Definition at line 590 of file options.h.

6.88.2.710 pool_size_min

```
int par::pool_size_min = 2 [static]
```

Definition at line 1117 of file options.h.

6.88.2.711 position_window

```
bool par::position_window = false [static]
```

Definition at line 786 of file options.h.

6.88.2.712 pp_maxfid

```
int par::pp_maxfid = 6 [static]
```

Definition at line 1294 of file options.h.

6.88.2.713 pp_maxiid

```
int par::pp_maxiid = 6 [static]
```

Definition at line 1295 of file options.h.

6.88.2.714 pp_maxsnp

```
int par::pp_maxsnp = 6 [static]
```

Definition at line 1293 of file options.h.

6.88.2.715 predall

```
bool par::predall = false [static]
```

Definition at line 477 of file options.h.

6.88.2.716 predict

```
string par::predict = "" [static]
```

Definition at line 474 of file options.h.

6.88.2.717 preserve_all_genotypes

```
bool par::preserve_all_genotypes = false [static]
```

Definition at line 211 of file options.h.

6.88.2.718 preserve_mendel_errors

```
bool par::preserve_mendel_errors = false [static]
```

Definition at line 212 of file options.h.

6.88.2.719 profile_sets

```
bool par::profile_sets = false [static]
```

Definition at line 633 of file options.h.

6.88.2.720 proxy_all

```
bool par::proxy_all = false [static]
```

Definition at line 644 of file options.h.

6.88.2.721 proxy_all_list

```
bool par::proxy_all_list = false [static]
```

Definition at line 668 of file options.h.

6.88.2.722 proxy_all_list_file

```
string par::proxy_all_list_file = "proxy.list" [static]
```

Definition at line 669 of file options.h.

6.88.2.723 proxy_assoc

```
bool par::proxy_assoc = false [static]
```

Definition at line 642 of file options.h.

6.88.2.724 proxy_assoc.snp

```
string par::proxy_assoc.snp = "rs1234" [static]
```

Definition at line 664 of file options.h.

6.88.2.725 proxy_CC

```
bool par::proxy_CC = false [static]
```

Definition at line 662 of file options.h.

6.88.2.726 proxy_error

```
bool par::proxy_error = false [static]
```

Definition at line 646 of file options.h.

6.88.2.727 proxy_exclude

```
bool par::proxy_exclude = false [static]
```

Definition at line 655 of file options.h.

6.88.2.728 proxy_exclude_from_file

```
bool par::proxy_exclude_from_file = false [static]
```

Definition at line 657 of file options.h.

6.88.2.729 proxy_exclude_list

```
string par::proxy_exclude_list = "pexclude.list" [static]
```

Definition at line 656 of file options.h.

6.88.2.730 proxy_full_report

```
bool par::proxy_full_report = false [static]
```

Definition at line 645 of file options.h.

6.88.2.731 proxy_gen0

```
double par::proxy_gen0 = 0.2 [static]
```

Definition at line 674 of file options.h.

6.88.2.732 proxy_glm

```
bool par::proxy_glm = false [static]
```

Definition at line 643 of file options.h.

6.88.2.733 proxy_impute

```
bool par::proxy_impute = false [static]
```

Definition at line 647 of file options.h.

6.88.2.734 proxy_impute_genotypic_concordance

```
bool par::proxy_impute_genotypic_concordance = false [static]
```

Definition at line 651 of file options.h.

6.88.2.735 proxy_impute_preserve_genotyped

```
bool par::proxy_impute_preserve_genotyped = false [static]
```

Definition at line 649 of file options.h.

6.88.2.736 proxy_impute_replace

```
bool par::proxy_impute_replace = false [static]
```

Definition at line 648 of file options.h.

6.88.2.737 proxy_impute_threshold

```
double par::proxy_impute_threshold = 0.9 [static]
```

Definition at line 652 of file options.h.

6.88.2.738 proxy_include_reference

```
bool par::proxy_include_reference = false [static]
```

Definition at line 661 of file options.h.

6.88.2.739 proxy_info_threshold

```
double par::proxy_info_threshold = 0.5 [static]
```

Definition at line 653 of file options.h.

6.88.2.740 proxy_kb

```
double par::proxy_kb = 250 [static]
```

Definition at line 670 of file options.h.

6.88.2.741 proxy_kb_planA

```
double par::proxy_kb_planA = 250 [static]
```

Definition at line 683 of file options.h.

6.88.2.742 proxy_kb_planB

```
double par::proxy_kb_planB = 500 [static]
```

Definition at line 691 of file options.h.

6.88.2.743 proxy_leave_out

```
bool par::proxy_leave_out = false [static]
```

Definition at line 660 of file options.h.

6.88.2.744 proxy_list

```
bool par::proxy_list = false [static]
```

Definition at line 666 of file options.h.

6.88.2.745 proxy_list_file

```
string par::proxy_list_file = "proxy.hap" [static]
```

Definition at line 667 of file options.h.

6.88.2.746 proxy_list_proxies

```
bool par::proxy_list_proxies = false [static]
```

Definition at line 675 of file options.h.

6.88.2.747 proxy_maf

```
double par::proxy_maf = 0.005 [static]
```

Definition at line 672 of file options.h.

6.88.2.748 proxy_maxhap

```
int par::proxy_maxhap = 3 [static]
```

Definition at line 676 of file options.h.

6.88.2.749 proxy_mhf

```
double par::proxy_mhf = 0.01 [static]
```

Definition at line 673 of file options.h.

6.88.2.750 proxy_planB_threshold

```
double par::proxy_planB_threshold = 0.1 [static]
```

Definition at line 690 of file options.h.

6.88.2.751 proxy_r2

```
double par::proxy_r2 = 0.5 [static]
```

Definition at line 671 of file options.h.

6.88.2.752 proxy_r2_filter

```
bool par::proxy_r2_filter = true [static]
```

Definition at line 677 of file options.h.

6.88.2.753 proxy_r2_filter_A

```
double par::proxy_r2_filter_A = 0.00 [static]
```

Definition at line 678 of file options.h.

6.88.2.754 proxy_r2_filter_A_planA

```
double par::proxy_r2_filter_A_planA = 0.00 [static]
```

Definition at line 686 of file options.h.

6.88.2.755 proxy_r2_filter_A_planB

```
double par::proxy_r2_filter_A_planB = 0.00 [static]
```

Definition at line 694 of file options.h.

6.88.2.756 proxy_r2_filter_B

```
double par::proxy_r2_filter_B = 0.05 [static]
```

Definition at line 679 of file options.h.

6.88.2.757 proxy_r2_filter_B_planA

```
double par::proxy_r2_filter_B_planA = 0.25 [static]
```

Definition at line 687 of file options.h.

6.88.2.758 proxy_r2_filter_B_planB

```
double par::proxy_r2_filter_B_planB = 0.01 [static]
```

Definition at line 695 of file options.h.

6.88.2.759 proxy_r2_filter_C

```
double par::proxy_r2_filter_C = 0.50 [static]
```

Definition at line 680 of file options.h.

6.88.2.760 proxy_r2_filter_C_planA

```
double par::proxy_r2_filter_C_planA = 0.50 [static]
```

Definition at line 688 of file options.h.

6.88.2.761 proxy_r2_filter_C_planB

```
double par::proxy_r2_filter_C_planB = 0.50 [static]
```

Definition at line 696 of file options.h.

6.88.2.762 proxy_record_dosage

```
bool par::proxy_record_dosage = false [static]
```

Definition at line 650 of file options.h.

6.88.2.763 proxy_reference_only

```
bool par::proxy_reference_only = false [static]
```

Definition at line 658 of file options.h.

6.88.2.764 proxy.snp_filter

```
int par::proxy.snp_filter = 5 [static]
```

Definition at line 681 of file options.h.

6.88.2.765 proxy.snp_filter_planA

```
int par::proxy.snp_filter_planA = 5 [static]
```

Definition at line 685 of file options.h.

6.88.2.766 proxy.snp_filter_planB

```
int par::proxy.snp_filter_planB = 10 [static]
```

Definition at line 693 of file options.h.

6.88.2.767 proxy.TDT

```
bool par::proxy.TDT = false [static]
```

Definition at line 663 of file options.h.

6.88.2.768 proxy.window

```
int par::proxy.window = 15 [static]
```

Definition at line 665 of file options.h.

6.88.2.769 proxy.window_planA

```
int par::proxy.window_planA = 15 [static]
```

Definition at line 684 of file options.h.

6.88.2.770 proxy_window_planB

```
int par::proxy_window_planB = 30 [static]
```

Definition at line 692 of file options.h.

6.88.2.771 prune_id

```
bool par::prune_id = false [static]
```

Definition at line 1060 of file options.h.

6.88.2.772 prune_id_pairwise

```
bool par::prune_id_pairwise = false [static]
```

Definition at line 1061 of file options.h.

6.88.2.773 prune_id_pairwise_maf

```
bool par::prune_id_pairwise_maf = true [static]
```

Definition at line 1062 of file options.h.

6.88.2.774 prune_id_r2

```
double par::prune_id_r2 = 1 - 1e-6 [static]
```

Definition at line 1064 of file options.h.

6.88.2.775 prune_id_step

```
int par::prune_id_step = 50 [static]
```

Definition at line 1066 of file options.h.

6.88.2.776 prune_ld_vif

```
double par::prune_ld_vif = 2 [static]
```

Definition at line 1063 of file options.h.

6.88.2.777 prune_ld_win

```
int par::prune_ld_win = 100 [static]
```

Definition at line 1065 of file options.h.

6.88.2.778 prune_r2_fixed

```
bool par::prune_r2_fixed = false [static]
```

Definition at line 1069 of file options.h.

6.88.2.779 prune_r2_fixed_list

```
string par::prune_r2_fixed_list = "dummy" [static]
```

Definition at line 1070 of file options.h.

6.88.2.780 prune_r2_prefer

```
bool par::prune_r2_prefer = false [static]
```

Definition at line 1067 of file options.h.

6.88.2.781 prune_r2_prefer_list

```
string par::prune_r2_prefer_list = "dummy" [static]
```

Definition at line 1068 of file options.h.

6.88.2.782 QFAM_adaptive

```
bool par::QFAM_adaptive = false [static]
```

Definition at line 931 of file options.h.

6.88.2.783 QFAM_between

```
bool par::QFAM_between = false [static]
```

Definition at line 928 of file options.h.

6.88.2.784 QFAM_total

```
bool par::QFAM_total = false [static]
```

Definition at line 927 of file options.h.

6.88.2.785 QFAM_within1

```
bool par::QFAM_within1 = false [static]
```

Definition at line 929 of file options.h.

6.88.2.786 QFAM_within2

```
bool par::QFAM_within2 = false [static]
```

Definition at line 930 of file options.h.

6.88.2.787 qmatch

```
bool par::qmatch = false [static]
```

Definition at line 1252 of file options.h.

6.88.2.788 qmatch_filename

```
string par::qmatch_filename = "inbix.qmatch" [static]
```

Definition at line 1255 of file options.h.

6.88.2.789 qmatch_threshold_filename

```
string par::qmatch_threshold_filename = "inbix.qt" [static]
```

Definition at line 1256 of file options.h.

6.88.2.790 qq_plot

```
bool par::qq_plot = false [static]
```

Definition at line 979 of file options.h.

6.88.2.791 qt

```
bool par::qt = false [static]
```

Definition at line 790 of file options.h.

6.88.2.792 qt_means

```
bool par::qt_means = false [static]
```

Definition at line 893 of file options.h.

6.88.2.793 qt_with_covariates

```
bool par::qt_with_covariates = false [static]
```

Definition at line 916 of file options.h.

6.88.2.794 QTDT_test

```
bool par::QTDT_test = false [static]
```

Definition at line 926 of file options.h.

6.88.2.795 R_port

```
int par::R_port = 6311 [static]
```

Definition at line 183 of file options.h.

6.88.2.796 R_script

```
string par::R_script = "script.R" [static]
```

Definition at line 179 of file options.h.

6.88.2.797 random_seed

```
long unsigned int par::random_seed = 0 [static]
```

Definition at line 129 of file options.h.

6.88.2.798 range_delimiter

```
string par::range_delimiter = "-" [static]
```

Definition at line 260 of file options.h.

6.88.2.799 ranker_centrality_gamma

```
double par::ranker_centrality_gamma = 0.85 [static]
```

Definition at line 532 of file options.h.

6.88.2.800 ranker_input_file

```
string par::ranker_input_file = "" [static]
```

Definition at line 531 of file options.h.

6.88.2.801 ranker_method

```
string par::ranker_method = "" [static]
```

Definition at line 528 of file options.h.

6.88.2.802 ranker_save_data_file

```
string par::ranker_save_data_file = "" [static]
```

Definition at line 530 of file options.h.

6.88.2.803 ranker_top_n

```
int par::ranker_top_n = -1 [static]
```

Definition at line 529 of file options.h.

6.88.2.804 rankerPermMethod

```
string par::rankerPermMethod = "regain" [static]
```

Definition at line 535 of file options.h.

6.88.2.805 rankerPermNum

```
int par::rankerPermNum = 100 [static]
```

Definition at line 536 of file options.h.

6.88.2.806 rankerPermThreshold

```
double par::rankerPermThreshold = 0.05 [static]
```

Definition at line 537 of file options.h.

6.88.2.807 rare_test

```
bool par::rare_test = false [static]
```

Definition at line 753 of file options.h.

6.88.2.808 rare_test_print_details

```
bool par::rare_test_print_details = false [static]
```

Definition at line 755 of file options.h.

6.88.2.809 rare_test_print_details_snp

```
string par::rare_test_print_details_snp = "" [static]
```

Definition at line 756 of file options.h.

6.88.2.810 rare_test_score_range

```
bool par::rare_test_score_range = false [static]
```

Definition at line 761 of file options.h.

6.88.2.811 rare_test_score_range_file

```
string par::rare_test_score_range_file = "" [static]
```

Definition at line 764 of file options.h.

6.88.2.812 rare_test_score_range_threshold

```
double par::rare_test_score_range_threshold = 0.01 [static]
```

Definition at line 762 of file options.h.

6.88.2.813 rare_test_score_results_file

```
string par::rare_test_score_results_file = "" [static]
```

Definition at line 763 of file options.h.

6.88.2.814 rare_test_summary_controls

```
bool par::rare_test_summary_controls = false [static]
```

Definition at line 765 of file options.h.

6.88.2.815 rare_test_weight1

```
bool par::rare_test_weight1 = false [static]
```

Definition at line 754 of file options.h.

6.88.2.816 rarer_dist_threshold

```
double par::rarer_dist_threshold = 100000 [static]
```

Definition at line 751 of file options.h.

6.88.2.817 rarer_interval

```
int par::rarer_interval = 100 [static]
```

Definition at line 752 of file options.h.

6.88.2.818 rarer_maf_threshold

```
double par::rarer_maf_threshold = 0.1 [static]
```

Definition at line 750 of file options.h.

6.88.2.819 read_bitfile

```
bool par::read_bitfile = false [static]
```

Definition at line 294 of file options.h.

6.88.2.820 read_geno_qual

```
bool par::read_geno_qual = false [static]
```

Definition at line 1010 of file options.h.

6.88.2.821 read_ped

```
bool par::read_ped = false [static]
```

Definition at line 308 of file options.h.

6.88.2.822 read_segment_file

```
bool par::read_segment_file = false [static]
```

Definition at line 1165 of file options.h.

6.88.2.823 read_segment_filename

```
string par::read_segment_filename = "" [static]
```

Definition at line 1166 of file options.h.

6.88.2.824 read_set

```
bool par:::read_set = false [static]
```

Definition at line 999 of file options.h.

6.88.2.825 read_snp_qual

```
bool par:::read_snp_qual = false [static]
```

Definition at line 1006 of file options.h.

6.88.2.826 recode

```
bool par:::recode = false [static]
```

Definition at line 185 of file options.h.

6.88.2.827 recode_12

```
bool par:::recode_12 = false [static]
```

Definition at line 190 of file options.h.

6.88.2.828 recode_1234

```
bool par:::recode_1234 = false [static]
```

Definition at line 197 of file options.h.

6.88.2.829 recode_ACGT

```
bool par:::recode_ACGT = false [static]
```

Definition at line 198 of file options.h.

6.88.2.830 recode_AD

```
bool par::recode_AD = false [static]
```

Definition at line 191 of file options.h.

6.88.2.831 recode_AD_Aonly

```
bool par::recode_AD_Aonly = false [static]
```

Definition at line 192 of file options.h.

6.88.2.832 recode_AD_fixed

```
bool par::recode_AD_fixed = false [static]
```

Definition at line 193 of file options.h.

6.88.2.833 recode_allele_coding

```
bool par::recode_allele_coding = false [static]
```

Definition at line 194 of file options.h.

6.88.2.834 recode_allele_coding_file

```
string par::recode_allele_coding_file = "file.lst" [static]
```

Definition at line 195 of file options.h.

6.88.2.835 recode_bim bam

```
bool par::recode_bim bam = false [static]
```

Definition at line 210 of file options.h.

6.88.2.836 recode_delimit

```
string par::recode_delimit = " " [static]
```

Definition at line 204 of file options.h.

6.88.2.837 recode_fastphase

```
bool par::recode_fastphase = false [static]
```

Definition at line 208 of file options.h.

6.88.2.838 recode_HV

```
bool par::recode_HV = false [static]
```

Definition at line 206 of file options.h.

6.88.2.839 recode_indelimit

```
string par::recode_indelimit = " " [static]
```

Definition at line 205 of file options.h.

6.88.2.840 recode_long

```
bool par::recode_long = false [static]
```

Definition at line 187 of file options.h.

6.88.2.841 recode_long_ref

```
bool par::recode_long_ref = false [static]
```

Definition at line 188 of file options.h.

6.88.2.842 recode_mutlist

```
bool par::recode_mutlist = false [static]
```

Definition at line 189 of file options.h.

6.88.2.843 recode_structure

```
bool par::recode_structure = false [static]
```

Definition at line 209 of file options.h.

6.88.2.844 recode_transpose

```
bool par::recode_transpose = false [static]
```

Definition at line 186 of file options.h.

6.88.2.845 recode_whap

```
bool par::recode_whap = false [static]
```

Definition at line 207 of file options.h.

6.88.2.846 ref_file

```
bool par::ref_file = false [static]
```

Definition at line 275 of file options.h.

6.88.2.847 ref_file_name

```
string par::ref_file_name = "" [static]
```

Definition at line 276 of file options.h.

6.88.2.848 regainComponents

```
bool par::regainComponents = false [static]
```

Definition at line 393 of file options.h.

6.88.2.849 regainCompress

```
bool par::regainCompress = false [static]
```

Definition at line 392 of file options.h.

6.88.2.850 regainFailValue

```
double par::regainFailValue = 0 [static]
```

Definition at line 400 of file options.h.

6.88.2.851 regainFdr

```
double par::regainFdr = 0.5 [static]
```

Definition at line 394 of file options.h.

6.88.2.852 regainFdrPrune

```
bool par::regainFdrPrune = false [static]
```

Definition at line 395 of file options.h.

6.88.2.853 regainFile

```
string par::regainFile = "no file" [static]
```

Definition at line 388 of file options.h.

6.88.2.854 regainLargeCoefPvalue

```
double const par::regainLargeCoefPvalue = 0.99 [static]
```

Definition at line 389 of file options.h.

6.88.2.855 regainLargeCoefTvalue

```
double const par::regainLargeCoefTvalue = 10 [static]
```

Definition at line 390 of file options.h.

6.88.2.856 regainMatrixFormat

```
string par::regainMatrixFormat = "full" [static]
```

Definition at line 404 of file options.h.

6.88.2.857 regainMatrixThreshold

```
bool par::regainMatrixThreshold = false [static]
```

Definition at line 405 of file options.h.

6.88.2.858 regainMatrixThresholdValue

```
double par::regainMatrixThresholdValue = 0.0 [static]
```

Definition at line 406 of file options.h.

6.88.2.859 regainMatrixToSif

```
bool par::regainMatrixToSif = false [static]
```

Definition at line 407 of file options.h.

6.88.2.860 regainMatrixTransform

```
string par::regainMatrixTransform = "none" [static]
```

Definition at line 403 of file options.h.

6.88.2.861 regainMaxBetaValue

```
double const par::regainMaxBetaValue = 100e3 [static]
```

Definition at line 391 of file options.h.

6.88.2.862 regainPureInteractions

```
bool par::regainPureInteractions = false [static]
```

Definition at line 399 of file options.h.

6.88.2.863 regainPvalueThreshold

```
double par::regainPvalueThreshold = 0.1 [static]
```

Definition at line 401 of file options.h.

6.88.2.864 regainSifFilter

```
bool par::regainSifFilter = false [static]
```

Definition at line 396 of file options.h.

6.88.2.865 regainSifThreshold

```
double par::regainSifThreshold = -9999999 [static]
```

Definition at line 397 of file options.h.

6.88.2.866 regainUseBetaValues

```
bool par::regainUseBetaValues = false [static]
```

Definition at line 398 of file options.h.

6.88.2.867 relieffilterNumToRemove

```
uint par::relieffilterNumToRemove = 0 [static]
```

Definition at line 461 of file options.h.

6.88.2.868 relieffilterPercentToRemove

```
uint par::relieffilterPercentToRemove = 0 [static]
```

Definition at line 462 of file options.h.

6.88.2.869 reliefNumTarget

```
uint par::reliefNumTarget = 0 [static]
```

Definition at line 460 of file options.h.

6.88.2.870 reliefMode

```
string par::reliefMode = "relieff" [static]
```

Definition at line 455 of file options.h.

6.88.2.871 reliefSeqAlgorithmMode

```
string par::reliefSeqAlgorithmMode = "snr" [static]
```

Definition at line 456 of file options.h.

6.88.2.872 reliefSeqAlgorithmS0

```
double par::reliefSeqAlgorithmS0 = 0.05 [static]
```

Definition at line 457 of file options.h.

6.88.2.873 reliefSeqSnrMode

```
string par::reliefSeqSnrMode = "snr" [static]
```

Definition at line 458 of file options.h.

6.88.2.874 reliefSeqTstatMode

```
string par::reliefSeqTstatMode = "pval" [static]
```

Definition at line 459 of file options.h.

6.88.2.875 remove_before_keep

```
bool par::remove_before_keep = true [static]
```

Definition at line 1273 of file options.h.

6.88.2.876 remove_file

```
string par::remove_file = "inbix.list" [static]
```

Definition at line 1004 of file options.h.

6.88.2.877 remove_indiv

```
bool par::remove_indiv = false [static]
```

Definition at line 1268 of file options.h.

6.88.2.878 remove_indiv_list

```
string par::remove_indiv_list = "inbix.list" [static]
```

Definition at line 1269 of file options.h.

6.88.2.879 remove_unaffected_pairs

```
bool par::remove_unaffected_pairs = false [static]
```

Definition at line 805 of file options.h.

6.88.2.880 replicates

```
int par::replicates = 1000 [static]
```

Definition at line 1196 of file options.h.

6.88.2.881 report_missing

```
bool par::report_missing = false [static]
```

Definition at line 635 of file options.h.

6.88.2.882 return_beta

```
bool par::return_beta = false [static]
```

Definition at line 887 of file options.h.

6.88.2.883 rfreplace

```
bool par::rfreplace = true [static]
```

Definition at line 488 of file options.h.

6.88.2.884 run_chr

```
int par::run_chr = 0 [static]
```

Definition at line 782 of file options.h.

6.88.2.885 run_end

```
int par::run_end = 0 [static]
```

Definition at line 781 of file options.h.

6.88.2.886 run_R_chisq

```
bool par::run_R_chisq = false [static]
```

Definition at line 180 of file options.h.

6.88.2.887 run_R_nsnps

```
int par::run_R_nsnps = 100 [static]
```

Definition at line 182 of file options.h.

6.88.2.888 run_R_script

```
bool par::run_R_script = false [static]
```

Definition at line 177 of file options.h.

6.88.2.889 run_R_write_script

```
bool par::run_R_write_script = false [static]
```

Definition at line 178 of file options.h.

6.88.2.890 run_R_z

```
bool par::run_R_z = false [static]
```

Definition at line 181 of file options.h.

6.88.2.891 run_start

```
int par::run_start = 0 [static]
```

Definition at line 780 of file options.h.

6.88.2.892 savemem

```
bool par::savemem = false [static]
```

Definition at line 473 of file options.h.

6.88.2.893 score_impute_expected

```
bool par::score_impute_expected = true [static]
```

Definition at line 628 of file options.h.

6.88.2.894 score_qfile

```
string par::score_qfile = "" [static]
```

Definition at line 631 of file options.h.

6.88.2.895 score_qrange_file

```
string par::score_qrange_file = "" [static]
```

Definition at line 630 of file options.h.

6.88.2.896 score_risk

```
bool par::score_risk = false [static]
```

Definition at line 623 of file options.h.

6.88.2.897 score_risk_file

```
string par::score_risk_file = "inbix.risk" [static]
```

Definition at line 624 of file options.h.

6.88.2.898 score_risk_on_qrange

```
bool par::score_risk_on_qrange = false [static]
```

Definition at line 629 of file options.h.

6.88.2.899 score_risk_ranges

```
bool par::score_risk_ranges = false [static]
```

Definition at line 625 of file options.h.

6.88.2.900 score_risk_ranges_file

```
string par::score_risk_ranges_file = "inbix.ranges" [static]
```

Definition at line 626 of file options.h.

6.88.2.901 score_risk_ranges_min

```
int par::score_risk_ranges_min = 0 [static]
```

Definition at line 627 of file options.h.

6.88.2.902 score_test

```
bool par::score_test = false [static]
```

Definition at line 632 of file options.h.

6.88.2.903 SD

```
bool par::SD = true [static]
```

Definition at line 802 of file options.h.

6.88.2.904 seg_test_region

```
bool par::seg_test_region = false [static]
```

Definition at line 1353 of file options.h.

6.88.2.905 seg_test_window

```
bool par::seg_test_window = false [static]
```

Definition at line 1351 of file options.h.

6.88.2.906 seg_test_window_bp

```
double par::seg_test_window_bp = 100000 [static]
```

Definition at line 1352 of file options.h.

6.88.2.907 segment_current_focal_snp

```
int par::segment_current_focal.snp = -1 [static]
```

Definition at line 1147 of file options.h.

6.88.2.908 segment_haplotrack

```
bool par::segment_haplotrack = false [static]
```

Definition at line 1137 of file options.h.

6.88.2.909 segment_haplotrack_fid1

```
string par::segment_haplotrack_fid1 = "1" [static]
```

Definition at line 1138 of file options.h.

6.88.2.910 segment_haplotrack_fid2

```
string par::segment_haplotrack_fid2 = "2" [static]
```

Definition at line 1140 of file options.h.

6.88.2.911 segment_haplotrack_iid1

```
string par::segment_haplotrack_iid1 = "1" [static]
```

Definition at line 1139 of file options.h.

6.88.2.912 segment_haplotrack_iid2

```
string par::segment_haplotrack_iid2 = "2" [static]
```

Definition at line 1141 of file options.h.

6.88.2.913 segment_inter_snp_distance

```
int par::segment_inter_snp_distance = 1000 [static]
```

Definition at line 1168 of file options.h.

6.88.2.914 segment_length

```
int par::segment_length = 1000000 [static]
```

Definition at line 1162 of file options.h.

6.88.2.915 segment_m1

```
string par::segment_m1 = "" [static]
```

Definition at line 1159 of file options.h.

6.88.2.916 segment_m2

```
string par::segment_m2 = "" [static]
```

Definition at line 1160 of file options.h.

6.88.2.917 segment_minimal

```
bool par::segment_minimal = false [static]
```

Definition at line 1145 of file options.h.

6.88.2.918 segment_output

```
bool par::segment_output = false [static]
```

Definition at line 1144 of file options.h.

6.88.2.919 segment_output_started

```
bool par::segment_output_started = false [static]
```

Definition at line 1164 of file options.h.

6.88.2.920 segment_overlap

```
bool par::segment_overlap = false [static]
```

Definition at line 1148 of file options.h.

6.88.2.921 segment_silently_return_groups

```
bool par::segment_silently_return_groups = false [static]
```

Definition at line 1146 of file options.h.

6.88.2.922 segment.snp

```
int par::segment.snp = 100 [static]
```

Definition at line 1163 of file options.h.

6.88.2.923 segment.snp1

```
int par::segment.snp1 = -1 [static]
```

Definition at line 1157 of file options.h.

6.88.2.924 segment.snp2

```
int par::segment.snp2 = -1 [static]
```

Definition at line 1158 of file options.h.

6.88.2.925 segment_test_1sided

```
bool par::segment_test_1sided = true [static]
```

Definition at line 1154 of file options.h.

6.88.2.926 segment_test_fisher

```
bool par::segment_test_fisher = false [static]
```

Definition at line 1153 of file options.h.

6.88.2.927 segment_test_force_1sided

```
bool par::segment_test_force_1sided = false [static]
```

Definition at line 1155 of file options.h.

6.88.2.928 segment_test_ignore_discordant

```
bool par::segment_test_ignore_discordant = false [static]
```

Definition at line 1156 of file options.h.

6.88.2.929 segment_test_individual

```
bool par::segment_test_individual = false [static]
```

Definition at line 1151 of file options.h.

6.88.2.930 segment_test_specific_segs

```
bool par::segment_test_specific_segs = false [static]
```

Definition at line 1152 of file options.h.

6.88.2.931 segment_threshold_finish

```
double par::segment_threshold_finish = 0.25 [static]
```

Definition at line 1189 of file options.h.

6.88.2.932 segment_threshold_start

```
double par::segment_threshold_start = 0.25 [static]
```

Definition at line 1188 of file options.h.

6.88.2.933 segment_validate

```
bool par::segment_validate = false [static]
```

Definition at line 1150 of file options.h.

6.88.2.934 segment_verbose

```
bool par::segment_verbose = false [static]
```

Definition at line 1149 of file options.h.

6.88.2.935 set_by_set

```
bool par::set_by_set = true [static]
```

Definition at line 1085 of file options.h.

6.88.2.936 set_chisq_threshold

```
double par::set_chisq_threshold = 3.84146 [static]
```

Definition at line 1030 of file options.h.

6.88.2.937 set_max

```
int par::set_max = 5 [static]
```

Definition at line 1026 of file options.h.

6.88.2.938 set_min

```
int par::set_min = -1 [static]
```

Definition at line 1025 of file options.h.

6.88.2.939 set_p2

```
bool par::set_p2 = false [static]
```

Definition at line 1024 of file options.h.

6.88.2.940 set_r2

```
bool par::set_r2 = false [static]
```

Definition at line 1027 of file options.h.

6.88.2.941 set_r2_phase

```
bool par::set_r2_phase = false [static]
```

Definition at line 1029 of file options.h.

6.88.2.942 set_r2_read

```
bool par::set_r2_read = false [static]
```

Definition at line 1032 of file options.h.

6.88.2.943 set_r2_read_file

```
string par::set_r2_read_file = "inbix.ldset" [static]
```

Definition at line 1033 of file options.h.

6.88.2.944 set_r2_val

```
double par::set_r2_val = 0.5 [static]
```

Definition at line 1028 of file options.h.

6.88.2.945 set_r2_write

```
bool par::set_r2_write = false [static]
```

Definition at line 1031 of file options.h.

6.88.2.946 set_reference_allele

```
bool par::set_reference_allele = false [static]
```

Definition at line 200 of file options.h.

6.88.2.947 set_reference_allele_file

```
string par::set_reference_allele_file = "dummy.file" [static]
```

Definition at line 201 of file options.h.

6.88.2.948 set_score

```
bool par::set_score = false [static]
```

Definition at line 1039 of file options.h.

6.88.2.949 set_score_p

```
double par::set_score_p = 1 [static]
```

Definition at line 1040 of file options.h.

6.88.2.950 set_screen

```
bool par::set_screen = false [static]
```

Definition at line 711 of file options.h.

6.88.2.951 set_screen_resultfile

```
string par::set_screen_resultfile = "" [static]
```

Definition at line 712 of file options.h.

6.88.2.952 set_step

```
bool par::set_step = false [static]
```

Definition at line 1042 of file options.h.

6.88.2.953 set_step_in

```
double par::set_step_in = 0.05 [static]
```

Definition at line 1041 of file options.h.

6.88.2.954 set_table

```
bool par::set_table = false [static]
```

Definition at line 1043 of file options.h.

6.88.2.955 set_test

```
bool par::set_test = false [static]
```

Definition at line 1023 of file options.h.

6.88.2.956 setfile

```
string par::setfile = "inbix.set" [static]
```

Definition at line 1038 of file options.h.

6.88.2.957 sex_threshold_female

```
double par::sex_threshold_female = 0.2 [static]
```

Definition at line 1095 of file options.h.

6.88.2.958 sex_threshold_male

```
double par::sex_threshold_male = 0.8 [static]
```

Definition at line 1094 of file options.h.

6.88.2.959 show_impossible_IBD

```
bool par::show_impossible_IBD = true [static]
```

Definition at line 1192 of file options.h.

6.88.2.960 sibTDT_test

```
bool par::sibTDT_test = false [static]
```

Definition at line 934 of file options.h.

6.88.2.961 sifFile

```
string par::sifFile = "" [static]
```

Definition at line 424 of file options.h.

6.88.2.962 sifNetwork

```
bool par::sifNetwork = false [static]
```

Definition at line 425 of file options.h.

6.88.2.963 sifToGain

```
bool par::sifToGain = false [static]
```

Definition at line 426 of file options.h.

6.88.2.964 silent

```
bool par::silent = false [static]
```

Definition at line 322 of file options.h.

6.88.2.965 simple_interaction

```
bool par::simple_interaction = false [static]
```

Definition at line 910 of file options.h.

6.88.2.966 simul

```
bool par::simul = false [static]
```

Definition at line 135 of file options.h.

6.88.2.967 simul_file

```
string par::simul_file = "" [static]
```

Definition at line 136 of file options.h.

6.88.2.968 simul_haps

```
bool par::simul_haps = false [static]
```

Definition at line 138 of file options.h.

6.88.2.969 simul_label

```
string par::simul_label = "" [static]
```

Definition at line 133 of file options.h.

6.88.2.970 simul_ncases

```
int par::simul_ncases = 1000 [static]
```

Definition at line 131 of file options.h.

6.88.2.971 simul_ncontrols

```
int par::simul_ncontrols = 1000 [static]
```

Definition at line 132 of file options.h.

6.88.2.972 simul_prevalence

```
double par::simul_prevalence = 0.01 [static]
```

Definition at line 134 of file options.h.

6.88.2.973 simul_qt

```
bool par::simul_qt = false [static]
```

Definition at line 139 of file options.h.

6.88.2.974 simul_qt_var

```
double par::simul_qt_var = 0.05 [static]
```

Definition at line 140 of file options.h.

6.88.2.975 simul_tags

```
bool par::simul_tags = false [static]
```

Definition at line 137 of file options.h.

6.88.2.976 singlepoint

```
bool par::singlepoint = false [static]
```

Definition at line 611 of file options.h.

6.88.2.977 sliding_window

```
bool par::sliding_window = false [static]
```

Definition at line 813 of file options.h.

6.88.2.978 sliding_window_size

```
string par::sliding_window_size = "2" [static]
```

Definition at line 814 of file options.h.

6.88.2.979 snp_attrib_file

```
string par::snp_attrib_file = "" [static]
```

Definition at line 567 of file options.h.

6.88.2.980 snp_attrib_filter

```
bool par::snp_attrib_filter = false [static]
```

Definition at line 565 of file options.h.

6.88.2.981 snp_attrib_value

```
string par::snp_attrib_value = "" [static]
```

Definition at line 566 of file options.h.

6.88.2.982 snp_include_from_cl

```
bool par::snp_include_from_cl = false [static]
```

Definition at line 1015 of file options.h.

6.88.2.983 snp_include_range

```
string par::snp_include_range = "" [static]
```

Definition at line 1016 of file options.h.

6.88.2.984 SNP_major

```
bool par::SNP_major = true [static]
```

Definition at line 301 of file options.h.

6.88.2.985 snp_qual_file

```
string par::snp_qual_file = "dummy" [static]
```

Definition at line 1007 of file options.h.

6.88.2.986 snp_qual_max

```
double par::snp_qual_max = 1 [static]
```

Definition at line 1009 of file options.h.

6.88.2.987 snp_qual_min

```
double par::snp_qual_min = 0 [static]
```

Definition at line 1008 of file options.h.

6.88.2.988 snp_range_list

```
bool par::snp_range_list = false [static]
```

Definition at line 986 of file options.h.

6.88.2.989 snpDiffMetricName

```
string par::snpDiffMetricName = "gm" [static]
```

Definition at line 450 of file options.h.

6.88.2.990 snpNearestNeighborMetricName

```
string par::snpNearestNeighborMetricName = "gm" [static]
```

Definition at line 451 of file options.h.

6.88.2.991 sol_family

```
bool par::sol_family = false [static]
```

Definition at line 616 of file options.h.

6.88.2.992 species_cow

```
bool par::species_cow = false [static]
```

Definition at line 774 of file options.h.

6.88.2.993 species_dog

```
bool par::species_dog = false [static]
```

Definition at line 773 of file options.h.

6.88.2.994 species_horse

```
bool par::species_horse = false [static]
```

Definition at line 776 of file options.h.

6.88.2.995 species_mouse

```
bool par::species_mouse = false [static]
```

Definition at line 778 of file options.h.

6.88.2.996 species_rice

```
bool par::species_rice = false [static]
```

Definition at line 777 of file options.h.

6.88.2.997 species_sheep

```
bool par::species_sheep = false [static]
```

Definition at line 775 of file options.h.

6.88.2.998 splitrule

```
SplitRule par::splitrule = DEFAULT_SPLITRULE [static]
```

Definition at line 485 of file options.h.

6.88.2.999 splitweights

```
string par::splitweights = "" [static]
```

Definition at line 475 of file options.h.

6.88.2.1000 standard_beta

```
bool par::standard_beta = false [static]
```

Definition at line 881 of file options.h.

6.88.2.1001 startMergeOrder

```
int par::startMergeOrder = 2 [static]
```

Definition at line 519 of file options.h.

6.88.2.1002 statusvarname

```
string par::statusvarname = "" [static]
```

Definition at line 486 of file options.h.

6.88.2.1003 subsetfile

```
string par::subsetfile = "dummy.file" [static]
```

Definition at line 1035 of file options.h.

6.88.2.1004 summ_nonfounders

```
bool par::summ_nonfounders = false [static]
```

Definition at line 618 of file options.h.

6.88.2.1005 summary_ibd_output

```
bool par::summary_ibd_output = false [static]
```

Definition at line 1186 of file options.h.

6.88.2.1006 tagfile

```
string par::tagfile = "inbix.tag" [static]
```

Definition at line 809 of file options.h.

6.88.2.1007 targetpartitionsize

```
unsigned int par::targetpartitionsize = 0 [static]
```

Definition at line 482 of file options.h.

6.88.2.1008 TDT_test

```
bool par::TDT_test = false [static]
```

Definition at line 933 of file options.h.

6.88.2.1009 test_full_model

```
bool par::test_full_model = false [static]
```

Definition at line 909 of file options.h.

6.88.2.1010 test_hap_CC

```
bool par::test_hap_CC = false [static]
```

Definition at line 853 of file options.h.

6.88.2.1011 test_hap_GLM

```
bool par::test_hap_GLM = false [static]
```

Definition at line 857 of file options.h.

6.88.2.1012 test_hap_GLM_omnibus

```
bool par::test_hap_GLM_omnibus = false [static]
```

Definition at line 858 of file options.h.

6.88.2.1013 test_hap_only

```
bool par::test_hap_only = false [static]
```

Definition at line 856 of file options.h.

6.88.2.1014 test_hap_QTL

```
bool par::test_hap_QTL = false [static]
```

Definition at line 855 of file options.h.

6.88.2.1015 test_hap_TDT

```
bool par::test_hap_TDT = false [static]
```

Definition at line 854 of file options.h.

6.88.2.1016 test_list

```
vector< int > par::test_list [static]
```

Definition at line 912 of file options.h.

6.88.2.1017 test_missing

```
bool par::test_missing = false [static]
```

Definition at line 636 of file options.h.

6.88.2.1018 tfamfile

```
string par::tfamfile = "inbix.tfam" [static]
```

Definition at line 268 of file options.h.

6.88.2.1019 tfile_input

```
bool par::tfile_input = false [static]
```

Definition at line 269 of file options.h.

6.88.2.1020 thin_param

```
double par::thin_param = 0 [static]
```

Definition at line 989 of file options.h.

6.88.2.1021 thin_snps

```
bool par::thin_snps = false [static]
```

Definition at line 988 of file options.h.

6.88.2.1022 thresholdType

```
string par::thresholdType = "hard" [static]
```

Definition at line 517 of file options.h.

6.88.2.1023 thresholdValue

```
double par::thresholdValue = 0.8 [static]
```

Definition at line 518 of file options.h.

6.88.2.1024 titvFilename

```
string par::titvFilename = "" [static]
```

Definition at line 439 of file options.h.

6.88.2.1025 to_window

```
int par::to_window = 0 [static]
```

Definition at line 788 of file options.h.

6.88.2.1026 tpedfile

```
string par::tpedfile = "inbix.tped" [static]
```

Definition at line 267 of file options.h.

6.88.2.1027 treetype

```
TreeType par::treetype = TREE_CLASSIFICATION [static]
```

Definition at line 489 of file options.h.

6.88.2.1028 trend_only

```
bool par::trend_only = false [static]
```

Definition at line 885 of file options.h.

6.88.2.1029 tucc

```
bool par::tucc = false [static]
```

Definition at line 121 of file options.h.

6.88.2.1030 twoDFmodel

```
bool par::twoDFmodel = false [static]
```

Definition at line 907 of file options.h.

6.88.2.1031 twoDFmodel_hethom

```
bool par::twoDFmodel_hethom = false [static]
```

Definition at line 908 of file options.h.

6.88.2.1032 twolocus_snp1

```
string par::twolocus_snp1 = "" [static]
```

Definition at line 224 of file options.h.

6.88.2.1033 twolocus_snp2

```
string par::twolocus_snp2 = "" [static]
```

Definition at line 225 of file options.h.

6.88.2.1034 uncompress_file

```
bool par::uncompress_file = false [static]
```

Definition at line 305 of file options.h.

6.88.2.1035 update_allele_file

```
string par::update_allele_file = "dummy" [static]
```

Definition at line 263 of file options.h.

6.88.2.1036 update_alleles

```
bool par::update_alleles = false [static]
```

Definition at line 262 of file options.h.

6.88.2.1037 update_chr

```
bool par::update_chr = false [static]
```

Definition at line 248 of file options.h.

6.88.2.1038 update_cm

```
bool par::update_cm = false [static]
```

Definition at line 247 of file options.h.

6.88.2.1039 update_ids

```
bool par::update_ids = false [static]
```

Definition at line 250 of file options.h.

6.88.2.1040 update_ids_file

```
string par::update_ids_file = "" [static]
```

Definition at line 251 of file options.h.

6.88.2.1041 update_map

```
bool par::update_map = false [static]
```

Definition at line 246 of file options.h.

6.88.2.1042 update_mapfile

```
string par::update_mapfile = "new.map" [static]
```

Definition at line 259 of file options.h.

6.88.2.1043 update_name

```
bool par::update_name = false [static]
```

Definition at line 249 of file options.h.

6.88.2.1044 update_parents

```
bool par::update_parents = false [static]
```

Definition at line 254 of file options.h.

6.88.2.1045 update_parents_file

```
string par::update_parents_file = "" [static]
```

Definition at line 255 of file options.h.

6.88.2.1046 update_pheno

```
bool par::update_pheno = false [static]
```

Definition at line 256 of file options.h.

6.88.2.1047 update_pheno_file

```
string par::update_pheno_file = "" [static]
```

Definition at line 257 of file options.h.

6.88.2.1048 update_sex

```
bool par::update_sex = false [static]
```

Definition at line 252 of file options.h.

6.88.2.1049 update_sex_file

```
string par::update_sex_file = "" [static]
```

Definition at line 253 of file options.h.

6.88.2.1050 use_GC

```
bool par::use_GC = false [static]
```

Definition at line 976 of file options.h.

6.88.2.1051 use_subset

```
bool par::use_subset = false [static]
```

Definition at line 1036 of file options.h.

6.88.2.1052 useAbs

```
bool par::useAbs = true [static]
```

Definition at line 523 of file options.h.

6.88.2.1053 useWeighted

```
bool par::useWeighted = false [static]
```

Definition at line 524 of file options.h.

6.88.2.1054 verbose

```
bool par::verbose = false [static]
```

Definition at line 112 of file options.h.

6.88.2.1055 vif_threshold

```
double par::vif_threshold = 50 [static]
```

Definition at line 905 of file options.h.

6.88.2.1056 web_check

```
bool par::web_check = true [static]
```

Definition at line 120 of file options.h.

6.88.2.1057 weightByDistanceMethod

```
string par::weightByDistanceMethod = "equal" [static]
```

Definition at line 453 of file options.h.

6.88.2.1058 weightByDistanceSigma

```
double par::weightByDistanceSigma = 2.0 [static]
```

Definition at line 454 of file options.h.

6.88.2.1059 weighted_mm

```
bool par::weighted_mm = false [static]
```

Definition at line 861 of file options.h.

6.88.2.1060 window

```
double par::window = 0 [static]
```

Definition at line 785 of file options.h.

6.88.2.1061 write_bitfile

```
bool par::write_bitfile = false [static]
```

Definition at line 295 of file options.h.

6.88.2.1062 write_dosage

```
bool par::write_dosage = false [static]
```

Definition at line 1362 of file options.h.

6.88.2.1063 write_set

```
bool par::write_set = false [static]
```

Definition at line 998 of file options.h.

6.88.2.1064 write_snplist

```
bool par::write_snplist = false [static]
```

Definition at line 245 of file options.h.

6.88.2.1065 writeforest

```
bool par::writeforest = false [static]
```

Definition at line 490 of file options.h.

6.88.2.1066 xchr_model

```
int par::xchr_model = 1 [static]
```

Definition at line 899 of file options.h.

6.88.2.1067 zero_cluster

```
bool par::zero_cluster = false [static]
```

Definition at line 213 of file options.h.

6.88.2.1068 zero_cluster_filename

```
string par::zero_cluster_filename = "inbix.zero" [static]
```

Definition at line 214 of file options.h.

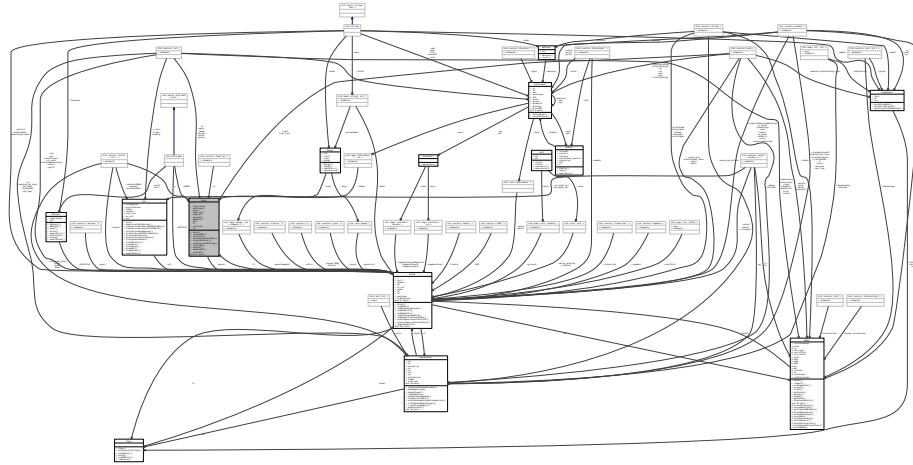
The documentation for this class was generated from the following files:

- [src/options.h](#)
- [src/options.cpp](#)

6.89 Perm Class Reference

```
#include <perm.h>
```

Collaboration diagram for Perm:



Public Member Functions

- `Perm (Plink &)`
- `void closeDUMP ()`
- `void setTests (int x)`
- `void setAdaptiveSetSNPs (int x)`
- `void originalOrder ()`
- `void setOriginalRanking (vector_t &)`
- `bool finished ()`
- `bool update (vector< double > &, vector< double > &)`
- `bool updateSNP (double, double, int)`
- `void nextSNP ()`
- `vector< double > & report ()`
- `int current_reps ()`
- `int reps_done (int)`
- `double pvalue (int)`
- `double max_pvalue (int)`
- `int rank (int)`
- `void permuteInCluster ()`
- `void setPermClusters (Plink &)`
- `void preGeneDrop ()`
- `void geneDrop ()`
- `void dropAlleles (Plink &, Individual *, int, int, vector< bool > &, vector< bool > &, vector< bool > &, map< Individual *, int > &)`

Public Attributes

- vector< int > `pheno`
- vector< int > `geno`
- vector< bool > `test`
- vector< bool > `snp_test`

Private Attributes

- int `t`
- long int `replicates`
- long int `performed`
- bool `count`
- ofstream `PDUMP`
- bool `dump_best`
- bool `dump_all`
- vector< int > `order`
- vector< int > `reorder`
- bool `genedrop`
- map< Individual *, int > `idmap`
- bool `adaptive`
- int `min`
- double `zt`
- int `interval`
- vector< int > `R`
- vector< int > `maxR`
- vector< long int > `N`
- vector< vector< int > > `s`
- int `ns`
- `Plink & P`

6.89.1 Detailed Description

Definition at line 21 of file perm.h.

6.89.2 Constructor & Destructor Documentation

6.89.2.1 Perm()

```
Perm::Perm (
    Plink & pref )
```

Definition at line 23 of file perm.cpp.

6.89.3 Member Function Documentation

6.89.3.1 closeDUMP()

```
void Perm::closeDUMP ( ) [inline]
```

Definition at line 67 of file perm.h.

6.89.3.2 current_reps()

```
int Perm::current_reps ( ) [inline]
```

Definition at line 92 of file perm.h.

6.89.3.3 dropAlleles()

```
void Perm::dropAlleles (
    PLink & P,
    Individual * person,
    int i,
    int l,
    vector< bool > & pat,
    vector< bool > & mat,
    vector< bool > & done,
    map< Individual *, int > & idmap )
```

Definition at line 239 of file genedrop.cpp.

6.89.3.4 finished()

```
bool Perm::finished ( )
```

Definition at line 113 of file perm.cpp.

6.89.3.5 geneDrop()

```
void Perm::geneDrop ( )
```

Definition at line 202 of file genedrop.cpp.

6.89.3.6 max_pvalue()

```
double Perm::max_pvalue ( int l )
```

Definition at line 541 of file perm.cpp.

6.89.3.7 nextSNP()

```
void Perm::nextSNP ( )
```

Definition at line 442 of file perm.cpp.

6.89.3.8 originalOrder()

```
void Perm::originalOrder ( )
```

Definition at line 106 of file perm.cpp.

6.89.3.9 permuteInCluster()

```
void Perm::permuteInCluster ( )
```

Definition at line 120 of file perm.cpp.

6.89.3.10 preGeneDrop()

```
void Perm::preGeneDrop ( )
```

Definition at line 24 of file genedrop.cpp.

6.89.3.11 pvalue()

```
double Perm::pvalue (
    int l )
```

Definition at line 530 of file perm.cpp.

6.89.3.12 rank()

```
int Perm::rank (
    int l )
```

Definition at line 521 of file perm.cpp.

6.89.3.13 report()

```
vector<double>& Perm::report ( )
```

6.89.3.14 reps_done()

```
int Perm::reps_done (
    int l )
```

Definition at line 549 of file perm.cpp.

6.89.3.15 setAdaptiveSetSNPs()

```
void Perm::setAdaptiveSetSNPs (
    int x )
```

Definition at line 99 of file perm.cpp.

6.89.3.16 setOriginalRanking()

```
void Perm::setOriginalRanking (
    vector_t & original )
```

Definition at line 184 of file perm.cpp.

6.89.3.17 setPermClusters()

```
void Perm::setPermClusters (
    PLink & P )
```

Definition at line 150 of file perm.cpp.

6.89.3.18 setTests()

```
void Perm::setTests (
    int x )
```

Definition at line 58 of file perm.cpp.

6.89.3.19 update()

```
bool Perm::update (
    vector< double > & result,
    vector< double > & original )
```

Definition at line 208 of file perm.cpp.

6.89.3.20 updateSNP()

```
bool Perm::updateSNP (
    double result,
    double original,
    int l )
```

Definition at line 449 of file perm.cpp.

6.89.4 Member Data Documentation

6.89.4.1 adaptive

```
bool Perm::adaptive [private]
```

Definition at line 46 of file perm.h.

6.89.4.2 count

```
bool Perm::count [private]
```

Definition at line 27 of file perm.h.

6.89.4.3 dump_all

```
bool Perm::dump_all [private]
```

Definition at line 31 of file perm.h.

6.89.4.4 dump_best

```
bool Perm::dump_best [private]
```

Definition at line 30 of file perm.h.

6.89.4.5 genedrop

```
bool Perm::genedrop [private]
```

Definition at line 39 of file perm.h.

6.89.4.6 geno

```
vector<int> Perm::geno
```

Definition at line 75 of file perm.h.

6.89.4.7 idmap

```
map<Individual*, int> Perm::idmap [private]
```

Definition at line 40 of file perm.h.

6.89.4.8 interval

```
int Perm::interval [private]
```

Definition at line 50 of file perm.h.

6.89.4.9 maxR

```
vector<int> Perm::maxR [private]
```

Definition at line 54 of file perm.h.

6.89.4.10 min

```
int Perm::min [private]
```

Definition at line 48 of file perm.h.

6.89.4.11 N

```
vector<long int> Perm::N [private]
```

Definition at line 55 of file perm.h.

6.89.4.12 ns

```
int Perm::ns [private]
```

Definition at line 59 of file perm.h.

6.89.4.13 order

```
vector<int> Perm::order [private]
```

Definition at line 33 of file perm.h.

6.89.4.14 P

```
Plink& Perm::P [private]
```

Definition at line 61 of file perm.h.

6.89.4.15 PDUMP

```
ofstream Perm::PDUMP [private]
```

Definition at line 29 of file perm.h.

6.89.4.16 performed

```
long int Perm::performed [private]
```

Definition at line 25 of file perm.h.

6.89.4.17 pheno

```
vector<int> Perm::pheno
```

Definition at line 74 of file perm.h.

6.89.4.18 R

```
vector<int> Perm::R [private]
```

Definition at line 53 of file perm.h.

6.89.4.19 reorder

```
vector<int> Perm::reorder [private]
```

Definition at line 34 of file perm.h.

6.89.4.20 replicates

```
long int Perm::replicates [private]
```

Definition at line 24 of file perm.h.

6.89.4.21 s

```
vector< vector<int> > Perm::s [private]
```

Definition at line 58 of file perm.h.

6.89.4.22 snp_test

```
vector<bool> Perm::snp_test
```

Definition at line 78 of file perm.h.

6.89.4.23 t

```
int Perm::t [private]
```

Definition at line 23 of file perm.h.

6.89.4.24 test

```
vector<bool> Perm::test
```

Definition at line 77 of file perm.h.

6.89.4.25 zt

```
double Perm::zt [private]
```

Definition at line 49 of file perm.h.

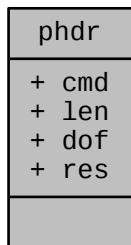
The documentation for this class was generated from the following files:

- src/perm.h
- src/genedrop.cpp
- src/perm.cpp

6.90 phdr Struct Reference

```
#include <Rsrv.h>
```

Collaboration diagram for phdr:



Public Attributes

- int cmd
- int len
- int dof
- int res

6.90.1 Detailed Description

Definition at line 80 of file Rsrv.h.

6.90.2 Member Data Documentation

6.90.2.1 cmd

```
int phdr::cmd
```

Definition at line 81 of file Rsrv.h.

6.90.2.2 dof

```
int phdr::dof
```

Definition at line 83 of file Rsrv.h.

6.90.2.3 len

```
int phdr::len
```

Definition at line 82 of file Rsrv.h.

6.90.2.4 res

```
int phdr::res
```

Definition at line 84 of file Rsrv.h.

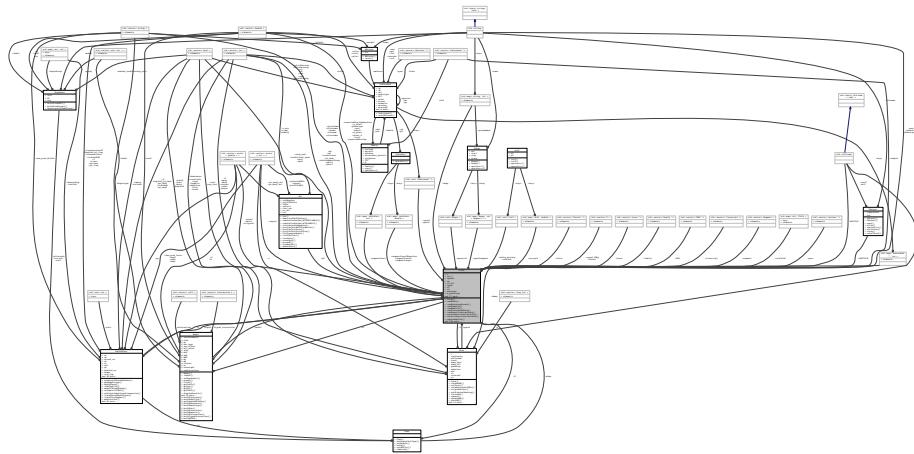
The documentation for this struct was generated from the following file:

- src/[Rsrv.h](#)

6.91 Plink Class Reference

```
#include <plink.h>
```

Collaboration diagram for Plink:



Public Member Functions

- [Plink \(\)](#)
- void [readData \(\)](#)
- void [readDataLongFormat \(\)](#)
- void [readFamFile \(string\)](#)
- void [readMapFile \(string, vector< bool > &, vector< int > &, int &\)](#)
- void [readTransposedData \(\)](#)
- void [readGenericVariantData \(\)](#)
- void [outputGenericVariantFile \(\)](#)
- void [convertGenericVariantData \(\)](#)
- void [updateMapFile \(\)](#)
- void [updateFamFile \(\)](#)
- void [updateAlleles \(\)](#)
- void [readStdIn \(\)](#)
- void [mergeData \(\)](#)
- bool [reconcileMerge \(int, int, string, string, bool, bool, ofstream &, map< string, int > &\)](#)
- void [mergeBinaryData \(\)](#)
- void [mergeList \(\)](#)
- void [dummyLoader \(\)](#)
- void [simulateSNPs \(\)](#)
- void [simulateSNPs_QT \(\)](#)
- bool [readPhenoFile \(\)](#)
- bool [setQtlPhenoFromNumericIndex \(int index\)](#)
- bool [readMultiplePhenoFile \(\)](#)
- bool [readCovariateFile \(\)](#)
- bool [readNumericFile \(\)](#)
- bool [readCovListFile \(\)](#)

- bool `readClusterFile` (bool verbose=true)
- void `readConditioningList` ()
- void `readBinData` ()
- void `readSet` ()
- void `prettyPrintLengths` ()
- void `printLOG` (string)
- void `outputSetFile` ()
- void `setAssocSummary` ()
- void `Ind2SNP` ()
- void `SNP2Ind` ()
- bool `outputArffFile` (string arffFilename)
- int `getSimpleSNPValue` (Individual *person, int snp)
- bool `outputDelimitedFile` (string delimitedFilename, string delimiter)
- bool `outputNumericExtract` (string attributeFilename)
- bool `outputSifToGain` (string sifFilename)
- bool `outputNumericFiltered` (string newFilename, boolevec_t varFlags)
- void `filterSNPs` ()
- void `processGVAR` ()
- void `calcStratifiedAlleleFreqs` ()
- void `hardyWeinbergCheck` ()
- double `calcInbreeding` (Individual *, int, int, ofstream &)
- void `sexCheck` ()
- void `calcFst` ()
- void `findAllHomozygousRuns` (Perm &)
- void `findHomoRuns` (Individual *, ofstream &)
- void `findHomoWindow` (Individual *, ofstream &)
- void `summariseHomoRuns` ()
- void `findIBSRuns` (Individual *, Individual *, ofstream &)
- void `findMissRuns` (Individual *, ofstream &)
- void `groupSegmentsSpanning` (int)
- void `displaySegmentsLong` ()
- void `displaySegmentsBED` ()
- void `setUpForCNVList` ()
- void `readCNVList` ()
- void `processCNVList` ()
- vector_t `glmCNVBurdenModel` (Perm &, bool)
- bool `missingGenotype` (int, int)
- bool `obligMissing` (int, int)
- void `outputPermedPhenotypes` (Perm &)
- void `countCNVPerRegion` (vector< int > &, vector< int > &)
- void `initialiseGeneCountAssociation` (Perm &)
- void `parseTrios` ()
- void `makeFounders` ()
- void `makeMissingParents` ()
- void `linkRelateds` (map< Individual *, int > &, map< string, Individual *> &)
- void `checkMendel` ()
- void `pseudoCaseControl` ()
- vector< double > `testTDT` (bool, bool, Perm &, vector< bool > &, vector< bool > &)
- void `perm_testTDT` (Perm &)
- vector< double > `testSibTDT` (bool, bool, Perm &, vector< bool > &, vector< bool > &)
- void `perm_testQTDT` (Perm &)

- `vector< double > calcQTDT (vector< int > &, ofstream &, bool, Perm &, vector< int > &, vector< bool > &)`
- `vector< double > testTDT_POO (bool, bool, Perm &, vector< bool > &, vector< bool > &)`
- `void perm_testTDT_POO (Perm &)`
- `vector< double > sharingIBSTest (Perm &)`
- `void perm_sharingIBSTest (Perm &)`
- `int readInformative ()`
- `int calcInformative ()`
- `void writeInformative ()`
- `void displayGenomeWideInfo ()`
- `void testGenomeIBDByCovariate (Perm &)`
- `void permutationIBSTest (Perm &)`
- `void displayGMULTI (Individual *, Individual *, int, ofstream &)`
- `void preCalcGenomeIBD ()`
- `void preCalcMultiPoint ()`
- `void preCalcSinglePoint ()`
- `void preCalcPhenotypes ()`
- `Z calcGenomeIBS (Individual *, Individual *)`
- `void calcGenomeIBM (Individual *, Individual *)`
- `Z calcGenomeIBD (Individual *, Individual *, Z)`
- `vector< Z > calcLocusIBD (Individual *, Individual *, Z)`
- `vector< double > calcMultiPoint (vector< Z > &, Z, ofstream &)`
- `vector< double > calcSinglePoint (vector< Z > &, Z)`
- `short calcPhenotypes (vector< double > &, Individual *p1, Individual *p2)`
- `void calcRegression (int)`
- `vector< double > doRegression (int, vector< double > &)`
- `void preCalcRegression_PHENO (vector< double > &)`
- `void preCalcRegression_PIHAT ()`
- `void calcAssociationWithPermutation (Perm &)`
- `void calcAssociationWithBootstrap ()`
- `void perm_testGXE2 (Perm &)`
- `vector< double > testQAssocGXE2 (bool, Perm &)`
- `void calcGXE (Perm &)`
- `void perm_testHotel (Perm &)`
- `vector< double > calcHotel (bool, Perm &, Set &, int, int)`
- `void calcMH ()`
- `void calcHomog ()`
- `vector< double > calcMantelHaenszel_2x2xK (Perm &, bool)`
- `vector< double > calcMantelHaenszel_ORD (vector< int > &, vector< int > &, vector< int > &)`
- `vector< double > calcMantelHaenszel_IxJxK (vector< int > &, vector< int > &, vector< int > &)`
- `void calcLDStatistics ()`
- `void calcPairwiseLD ()`
- `double correlation2SNP (int, int, bool, bool, bool useFlag=false)`
- `void pruneLD ()`
- `void calcFlipScan ()`
- `void setReferenceAllele ()`
- `map< Range, vector< int > > mkBlks (int, int)`
- `void setFlagToCase ()`
- `void setFlagToControl ()`
- `void calcEpistasis ()`
- `void driverSCREEPI ()`
- `vector< double > testMiss (Perm &, bool)`

- void `performMisHapTests ()`
- void `proxyWrapper ()`
- void `performProxyTests (int)`
- void `scoreIndividuals ()`
- void `calculateProfile (map< int, double > &, map< int, bool > &, vector_t &, matrix_t &, vector< int > &, vector< int > &)`
- vector< double > `testAssoc (int &, int &, vector< int > &, vector< int > &, vector< int > &, vector< double > &, vector< double > &, vector< double > &, vector< double > &, Perm &, bool)`
- vector< double > `testQAssoc (bool, Perm &)`
- vector< double > `fullModelAssoc (bool, Perm &)`
- void `displayQTMeans (ofstream &, int l)`
- vector_t `glmAssoc (bool, Perm &)`
- vector_t `conditionalHaplotypeTest (bool, Perm &)`
- vector_t `glmHaplotypeTest (bool, Perm &)`
- void `multcomp (vector< double > &, string)`
- void `buildT (double, bool, double, double)`
- void `setMarkerRange ()`
- void `buildCluster ()`
- void `generateMDS ()`
- void `groupGenome ()`
- void `summaryIBD ()`
- void `findSegments (int, int, vector_t &, ofstream &)`
- void `summaryIBDsegments (Perm &perm)`
- void `summaryIBSsegments (Perm &perm)`
- void `indivSegmentSummary ()`
- void `indivSegmentSummaryCalc (map< indivPair, int > &, map< indivPair, double > &, bool, bool)`
- void `readSegmentFile (ifstream &)`
- void `readSegmentFileMinimal (ifstream &)`
- void `readHomozygSegmentFile (ifstream &)`
- void `segmentPermutationTest (Perm &, bool, string, vector< int > &, vector< int > &, vector< int > &)`
- void `segmentIndividualTest (Perm &)`
- vector_t `perm_segmentIndividualTest (Perm &, bool, int, int, map< Individual *, int > &)`
- void `homozygousSegmentPermutationTest (Perm &, string, vector< int > &, vector< int > &)`
- void `validateSegments ()`
- void `positionPermuteSegments ()`
- void `runTestCNVwithQT (Perm &)`
- vector_t `testCNVwithQT (double, int, int, vector_t &, vector_t &, vector_t &)`
- void `runTestCNVwithGLM (Perm &)`
- vector_t `testCNVwithGLM (bool, Perm &, vector< int > &)`
- void `displayGenomePV ()`
- void `extractExcludeSet (bool)`
- void `removeIndividuals (bool)`
- void `keep2SetsForGenome ()`
- void `filterQualSNPs ()`
- void `filterQualGenotypes ()`
- void `makePhenotype ()`
- void `filterOnCovariate ()`
- void `filterOnCase ()`
- void `filterOnControl ()`
- void `filterOnMale ()`
- void `filterOnFemale ()`

- void `filterOnFounder ()`
- void `filterOnNonFounder ()`
- void `attribFilterSNP ()`
- void `attribFilterInd ()`
- void `zeroOnCluster ()`
- void `setObligMissing ()`
- int `deleteSNPs (vector< bool > &)`
- int `deleteSNPs (set< string > &)`
- int `deleteSNPs (set< Locus *> &)`
- int `deleteIndividuals (vector< bool > &)`
- int `deleteIndividualsNumeric (vector< bool > &)`
- int `deleteIndividuals (set< Individual *> &)`
- void `thinSNPs ()`
- int `keepSNPs (set< string > &)`
- int `keepSNPs (set< Locus *> &)`
- int `keepIndividuals (set< Individual *> &)`
- void `flipStrand ()`
- void `alleleRecoding ()`
- void `display_recoded_PEDFILE ()`
- void `display_recoded_PEDFILE_transpose ()`
- void `display_recoded_PEDFILE_AD ()`
- void `display_recoded_LONG ()`
- void `display_recoded_MUTLIST ()`
- void `output_fastphase_format ()`
- void `output_bim bam_format ()`
- void `output_structure_format ()`
- void `display_listByAllele ()`
- void `display_twolocus ()`
- void `display_pairList ()`
- void `display_indivReport ()`
- void `write_BITFILE ()`
- void `write_covariates ()`
- void `write_clusters ()`
- void `write_snplist ()`
- bool `openBinaryFile (string, ifstream &)`
- void `setTable ()`
- void `writeSetFile ()`
- void `tagMode ()`
- void `processDosageFile ()`
- void `displayGeneReport ()`
- void `annotateFile ()`
- void `metaAnalysis ()`
- void `webcheck (CArgs &)`
- void `lookup ()`
- void `lookup2 ()`
- void `cleanUp ()`
- void `setFlags (bool f)`
- void `permTestRareDistribution (Perm &)`
- void `elfBaseline ()`
- void `displayRareRange ()`
- `vector_t testRareDistribution (Perm &, bool, map< Range, int2 > &ranges)`

Public Attributes

- vector< Individual * > sample
- vector< CSNP * > SNP
- vector< Locus * > locus
- vector< Transcript * > transcript
- map< int, CInfo > scaffold
- vector< Variant * > gvar
- vector< Family * > family
- string phenoLabel
- vector< string > plistname
- vector< string > clistname
- vector< string > nlistname
- int n
- int cnt_f
- int npheno
- int np
- int nl_all
- int ngvar
- int nl
- int nk
- string phenotype_name
- ofstream OUTFILE
- ZOutput ZOUTFILE
- bool warnings
- map< string, int > kmap
- vector< string > kname
- vector< Cluster * > klist
- set< int2 > oblig_missing
- vector< int > conditioner
- vector< bool > conditioner_mask
- vector< bool > skip_pair
- string pairid
- vector< Z > Zlocus
- vector< Z > saved_IBDg
- vector< int > m1
- vector< int > m2
- vector< double > pos
- vector< vector< double > > pihat
- vector< Segment > segment
- set< Range > geneList
- map< Range, set< Segment > > gene2segment
- vector< int > indivSegmentGroup
- vector< double > pihat_G
- set< int2 > related
- vector< double > phenotype
- vector< int > pair1
- vector< int > pair2
- vector< int > in_anal
- double m_phenotype
- double v_phenotype

- double `prev_bt`
- vector< double > `m_pihat`
- vector< double > `v_pihat`
- double `E00`
- double `E10`
- double `E20`
- double `E01`
- double `E11`
- double `E21`
- double `E02`
- double `E12`
- double `E22`
- double `T00`
- double `T01`
- double `T02`
- double `T10`
- double `T11`
- double `T12`
- double `T20`
- double `T21`
- double `T22`
- vector< double > `maxr2`
- vector< string > `setname`
- vector< vector< int > > `snpset`
- vector< vector< double > > `original`
- vector< vector< double > > `mdist`
- double `pv`
- double `dst`
- double `pvlBS0`
- double `pvlBS2het`
- vector< bool > `epi1`
- vector< bool > `epi2`
- set< `Individual` * > `gset1`
- set< `Individual` * > `gset2`
- long int `diff_overlap`
- long int `diff_nonmissing_overlap`
- long int `diff_concordant_overlap`
- `vector_t` `tcnt`
- map< `int2`, double > `proxyLD`
- map< `indivPair`, int > `segmentCount`
- map< `indivPair`, double > `segmentLength`
- map< `indivPair`, double > `segmentCount2`
- map< `indivPair`, double > `segmentCount2Baseline`
- `vector_t` `expectedOverlap`
- `vector_t` `expectedOverlapBaseline`
- `Perm` * `pperm`
- `HaploPhase` * `haplo`
- `Model` * `model`
- `Chap` * `whap`
- `Set` * `pS`

6.91.1 Detailed Description

Definition at line 578 of file plink.h.

6.91.2 Constructor & Destructor Documentation

6.91.2.1 Plink()

```
Plink::Plink ( ) [inline]
```

Definition at line 581 of file plink.h.

6.91.3 Member Function Documentation

6.91.3.1 alleleRecoding()

```
void Plink::alleleRecoding ( )
```

Definition at line 2450 of file helper.cpp.

6.91.3.2 annotateFile()

```
void Plink::annotateFile ( )
```

Definition at line 29 of file annot.cpp.

6.91.3.3 attribFilterInd()

```
void Plink::attribFilterInd ( )
```

Definition at line 2318 of file helper.cpp.

6.91.3.4 attribFilterSNP()

```
void Plink::attribFilterSNP ( )
```

Definition at line 2244 of file helper.cpp.

6.91.3.5 buildCluster()

```
void Plink::buildCluster ( )
```

Definition at line 76 of file cluster.cpp.

6.91.3.6 buildT()

```
void Plink::buildT (
    double G,
    bool z2zero,
    double mA,
    double mB )
```

Definition at line 521 of file multi.cpp.

6.91.3.7 calcAssociationWithBootstrap()

```
void Plink::calcAssociationWithBootstrap ( )
```

Definition at line 862 of file legacy.cpp.

6.91.3.8 calcAssociationWithPermutation()

```
void Plink::calcAssociationWithPermutation (
    Perm & perm )
```

Definition at line 41 of file assoc.cpp.

6.91.3.9 calcEpistasis()

```
void Plink::calcEpistasis ( )
```

Definition at line 33 of file epi.cpp.

6.91.3.10 calcFlipScan()

```
void Plink::calcFlipScan ( )
```

Definition at line 185 of file flip.cpp.

6.91.3.11 calcFst()

```
void Plink::calcFst ( )
```

Definition at line 358 of file genome.cpp.

6.91.3.12 calcGenomeIBD()

```
Z Plink::calcGenomeIBD (
    Individual * p1,
    Individual * p2,
    Z IBSg )
```

Definition at line 832 of file genome.cpp.

6.91.3.13 calcGenomeIBM()

```
void Plink::calcGenomeIBM (
    Individual * p1,
    Individual * p2 )
```

Definition at line 1132 of file genome.cpp.

6.91.3.14 calcGenomeIBS()

```
Z Plink::calcGenomeIBS (
    Individual * p1,
    Individual * p2 )
```

Definition at line 568 of file genome.cpp.

6.91.3.15 calcGXE()

```
void Plink::calcGXE (
    Perm & )
```

6.91.3.16 calcHomog()

```
void Plink::calcHomog ( )
```

Definition at line 851 of file mh.cpp.

6.91.3.17 calcHotel()

```
vector< double > Plink::calcHotel (
    bool disp,
    Perm & perm,
    Set & S,
    int ncase,
    int ncontrol )
```

Definition at line 197 of file hotel.cpp.

6.91.3.18 calcInbreeding()

```
double Plink::calcInbreeding (
    Individual * p1,
    int m1,
    int m2,
    ofstream & HET )
```

Definition at line 429 of file genome.cpp.

6.91.3.19 calcInformative()

```
int Plink::calcInformative ( )
```

Definition at line 253 of file informative.cpp.

6.91.3.20 calcLDStatistics()

```
void Plink::calcLDStatistics ( )
```

Definition at line 2285 of file assoc.cpp.

6.91.3.21 calcLocusIBD()

```
vector< Z > Plink::calcLocusIBD (
    Individual * p1,
    Individual * p2,
    Z I )
```

Definition at line 20 of file locus.cpp.

6.91.3.22 calcMantelHaenszel_2x2xK()

```
vector< double > Plink::calcMantelHaenszel_2x2xK (
    Perm & perm,
    bool original )
```

Definition at line 207 of file mh.cpp.

6.91.3.23 calcMantelHaenszel_IxJxK()

```
vector< double > Plink::calcMantelHaenszel_IxJxK (
    vector< int > & X,
    vector< int > & Y,
    vector< int > & Z )
```

Definition at line 691 of file mh.cpp.

6.91.3.24 calcMantelHaenszel_ORD()

```
vector< double > Plink::calcMantelHaenszel_ORD (
    vector< int > & X,
    vector< int > & Y,
    vector< int > & Z )
```

Definition at line 1188 of file mh.cpp.

6.91.3.25 calcMH()

```
void Plink::calcMH ( )
```

Definition at line 26 of file mh.cpp.

6.91.3.26 calcMultiPoint()

```
vector< double > Plink::calcMultiPoint (
    vector< Z > & IBD,
    Z IBDg,
    ifstream & MP )
```

Definition at line 148 of file multi.cpp.

6.91.3.27 calcPairwiseLD()

```
void Plink::calcPairwiseLD ( )
```

Definition at line 1031 of file haplohelper.cpp.

6.91.3.28 calcPhenotypes()

```
short Plink::calcPhenotypes (
    vector< double > & l,
    Individual * p1,
    Individual * p2 )
```

Definition at line 309 of file legacy.cpp.

6.91.3.29 calcQTDT()

```
vector_t Plink::calcQTDT (
    vector< int > & C,
    ofstream & QOUT,
    bool permuting,
    Perm & perm,
    vector< int > & pbetween,
    vector< bool > & pwithin )
```

Definition at line 423 of file qfam.cpp.

6.91.3.30 calcRegression()

```
void Plink::calcRegression (
    int chr )
```

Definition at line 339 of file legacy.cpp.

6.91.3.31 calcSinglePoint()

```
vector< double > Plink::calcSinglePoint (
    vector< Z > & IBD,
    Z IBDg )
```

Definition at line 1134 of file legacy.cpp.

6.91.3.32 calcStratifiedAlleleFreqs()

```
void Plink::calcStratifiedAlleleFreqs ( )
```

Definition at line 30 of file genome.cpp.

6.91.3.33 calculateProfile()

```
void Plink::calculateProfile (
    map< int, double > & scores,
    map< int, bool > & allele1,
    vector_t & profile,
    matrix_t & set_profile,
    vector< int > & count,
    vector< int > & account )
```

Definition at line 428 of file profile.cpp.

6.91.3.34 checkMendel()

```
void Plink::checkMendel ( )
```

Definition at line 398 of file trio.cpp.

6.91.3.35 cleanUp()

```
void Plink::cleanUp ( )
```

Definition at line 19 of file plink.cpp.

6.91.3.36 conditionalHaplotypeTest()

```
vector_t Plink::conditionalHaplotypeTest (
    bool print_results,
    Perm & perm )
```

Definition at line 75 of file whap.cpp.

6.91.3.37 convertGenericVariantData()

```
void Plink::convertGenericVariantData ( )
```

Definition at line 1124 of file gvar.cpp.

6.91.3.38 correlation2SNP()

```
double Plink::correlation2SNP (
    int l1,
    int l2,
    bool squared,
    bool covariance,
    bool useFlag = false )
```

Definition at line 2445 of file assoc.cpp.

6.91.3.39 countCNVPerRegion()

```
void Plink::countCNVPerRegion (
    vector< int > & caseCount,
    vector< int > & controlCount )
```

Definition at line 1871 of file cnv.cpp.

6.91.3.40 deleteIndividuals() [1/2]

```
int Plink::deleteIndividuals (
    vector< bool > & del )
```

Definition at line 1983 of file helper.cpp.

6.91.3.41 deleteIndividuals() [2/2]

```
int Plink::deleteIndividuals (
    set< Individual *> & pset )
```

Definition at line 1843 of file helper.cpp.

6.91.3.42 deleteIndividualsNumeric()

```
int Plink::deleteIndividualsNumeric (
    vector< bool > & )
```

6.91.3.43 deleteSNPs() [1/3]

```
int Plink::deleteSNPs (
    vector< bool > & del )
```

Definition at line 1883 of file helper.cpp.

6.91.3.44 deleteSNPs() [2/3]

```
int Plink::deleteSNPs (
    set< string > & mset )
```

Definition at line 1823 of file helper.cpp.

6.91.3.45 deleteSNPs() [3/3]

```
int Plink::deleteSNPs (
    set< Locus *> & mset )
```

Definition at line 1833 of file helper.cpp.

6.91.3.46 display_indivReport()

```
void Plink::display_indivReport ( )
```

Definition at line 42 of file output.cpp.

6.91.3.47 display_listByAllele()

```
void Plink::display_listByAllele ( )
```

Definition at line 171 of file output.cpp.

6.91.3.48 display_pairList()

```
void Plink::display_pairList ( )
```

Definition at line 91 of file output.cpp.

6.91.3.49 display_recoded_LONG()

```
void Plink::display_recoded_LONG ( )
```

Definition at line 1093 of file output.cpp.

6.91.3.50 display_recoded_MUTLIST()

```
void Plink::display_recoded_MUTLIST ( )
```

Definition at line 969 of file output.cpp.

6.91.3.51 display_recoded_PEDFILE()

```
void Plink::display_recoded_PEDFILE ( )
```

Definition at line 849 of file output.cpp.

6.91.3.52 display_recoded_PEDFILE_AD()

```
void Plink::display_recoded_PEDFILE_AD ( )
```

Definition at line 1458 of file output.cpp.

6.91.3.53 display_recoded_PEDFILE_transpose()

```
void Plink::display_recoded_PEDFILE_transpose ( )
```

Definition at line 1401 of file output.cpp.

6.91.3.54 display_twolocus()

```
void Plink::display_twolocus ( )
```

Definition at line 357 of file output.cpp.

6.91.3.55 displayGeneReport()

```
void Plink::displayGeneReport ( )
```

Definition at line 27 of file greport.cpp.

6.91.3.56 displayGenomePV()

```
void Plink::displayGenomePV ( )
```

Definition at line 70 of file output.cpp.

6.91.3.57 displayGenomeWideInfo()

```
void Plink::displayGenomeWideInfo ( )
```

Definition at line 999 of file genome.cpp.

6.91.3.58 displayGMULTI()

```
void Plink::displayGMULTI (
    Individual * p1,
    Individual * p2,
    int l,
    ofstream & GMULTI )
```

Definition at line 1611 of file segment.cpp.

6.91.3.59 displayQTMeans()

```
void Plink::displayQTMeans (
    ofstream & QT_MEANS,
    int l )
```

Definition at line 1989 of file assoc.cpp.

6.91.3.60 displayRareRange()

```
void Plink::displayRareRange ( )
```

Definition at line 869 of file elf.cpp.

6.91.3.61 displaySegmentsBED()

```
void Plink::displaySegmentsBED ( )
```

Definition at line 2276 of file segment.cpp.

6.91.3.62 displaySegmentsLong()

```
void Plink::displaySegmentsLong ( )
```

Definition at line 1959 of file segment.cpp.

6.91.3.63 doRegression()

```
vector< double > Plink::doRegression (
    int npos,
    vector< double > & ph )
```

Definition at line 753 of file legacy.cpp.

6.91.3.64 driverSCREEPI()

```
void Plink::driverSCREEPI ( )
```

Definition at line 225 of file genepi.cpp.

6.91.3.65 dummyLoader()

```
void Plink::dummyLoader ( )
```

Definition at line 2386 of file helper.cpp.

6.91.3.66 elfBaseline()

```
void Plink::elfBaseline ( )
```

Definition at line 1082 of file elf.cpp.

6.91.3.67 extractExcludeSet()

```
void Plink::extractExcludeSet ( 
    bool exclude )
```

Definition at line 465 of file output.cpp.

6.91.3.68 filterOnCase()

```
void Plink::filterOnCase ( )
```

Definition at line 2140 of file helper.cpp.

6.91.3.69 filterOnControl()

```
void Plink::filterOnControl ( )
```

Definition at line 2158 of file helper.cpp.

6.91.3.70 filterOnCovariate()

```
void Plink::filterOnCovariate ( )
```

Definition at line 2067 of file helper.cpp.

6.91.3.71 filterOnFemale()

```
void Plink::filterOnFemale ( )
```

Definition at line 2193 of file helper.cpp.

6.91.3.72 filterOnFounder()

```
void Plink::filterOnFounder ( )
```

Definition at line 2210 of file helper.cpp.

6.91.3.73 filterOnMale()

```
void Plink::filterOnMale ( )
```

Definition at line 2176 of file helper.cpp.

6.91.3.74 filterOnNonFounder()

```
void Plink::filterOnNonFounder ( )
```

Definition at line 2227 of file helper.cpp.

6.91.3.75 filterQualGenotypes()

```
void Plink::filterQualGenotypes ( )
```

Definition at line 93 of file qualscores.cpp.

6.91.3.76 filterQualSNPs()

```
void Plink::filterQualSNPs ( )
```

Definition at line 29 of file qualscores.cpp.

6.91.3.77 filterSNPs()

```
void Plink::filterSNPs ( )
```

Definition at line 32 of file filters.cpp.

6.91.3.78 findAllHomozygousRuns()

```
void Plink::findAllHomozygousRuns (
    Perm & perm )
```

Definition at line 257 of file homozyg.cpp.

6.91.3.79 findHomoRuns()

```
void Plink::findHomoRuns (
    Individual * person,
    ofstream & HOM )
```

Definition at line 1805 of file homozyg.cpp.

6.91.3.80 findHomoWindow()

```
void Plink::findHomoWindow (
    Individual * person,
    ofstream & HOM )
```

Definition at line 2085 of file homozyg.cpp.

6.91.3.81 findIBSRuns()

```
void Plink::findIBSRuns (
    Individual * person1,
    Individual * person2,
    ofstream & IBS )
```

Definition at line 28 of file legacy.cpp.

6.91.3.82 findMissRuns()

```
void Plink::findMissRuns (
    Individual * person,
    ofstream & RUN )
```

Definition at line 224 of file genome.cpp.

6.91.3.83 findSegments()

```
void Plink::findSegments (
    int i1,
    int i2,
    vector_t & p,
    ofstream & SEG )
```

Definition at line 26 of file segment.cpp.

6.91.3.84 flipStrand()

```
void Plink::flipStrand ( )
```

Definition at line 29 of file flip.cpp.

6.91.3.85 fullModelAssoc()

```
vector< double > Plink::fullModelAssoc (
    bool print_results,
    Perm & perm )
```

Definition at line 1211 of file assoc.cpp.

6.91.3.86 generateMDS()

```
void Plink::generateMDS ( )
```

Definition at line 25 of file mds.cpp.

6.91.3.87 getSimpleSNPValue()

```
int Plink::getSimpleSNPValue (
    Individual * person,
    int snp )
```

Definition at line 2270 of file output.cpp.

6.91.3.88 glmAssoc()

```
vector_t Plink::glmAssoc (
    bool print_results,
    Perm & perm )
```

Definition at line 49 of file glm.cpp.

6.91.3.89 glmCNVBurdenModel()

```
vector_t Plink::glmCNVBurdenModel (
    Perm & perm,
    bool print )
```

Definition at line 2071 of file cnv.cpp.

6.91.3.90 glmHaplotypeTest()

```
vector_t Plink::glmHaplotypeTest (
    bool print,
    Perm & perm )
```

Definition at line 40 of file hapglm.cpp.

6.91.3.91 groupGenome()

```
void Plink::groupGenome ( )
```

Definition at line 1440 of file cluster.cpp.

6.91.3.92 groupSegmentsSpanning()

```
void Plink::groupSegmentsSpanning ( int l )
```

Definition at line 910 of file homozyg.cpp.

6.91.3.93 hardyWeinbergCheck()

```
void Plink::hardyWeinbergCheck ( )
```

6.91.3.94 homozygousSegmentPermutationTest()

```
void Plink::homozygousSegmentPermutationTest ( Perm & perm, string f, vector< int > & coverage_aff, vector< int > & coverage_unaff )
```

Definition at line 2343 of file homozyg.cpp.

6.91.3.95 Ind2SNP()

```
void Plink::Ind2SNP ( )
```

Definition at line 1722 of file helper.cpp.

6.91.3.96 indivSegmentSummary()

```
void Plink::indivSegmentSummary ( )
```

Definition at line 1781 of file segment.cpp.

6.91.3.97 indivSegmentSummaryCalc()

```
void Plink::indivSegmentSummaryCalc (
    map< indivPair, int > & segmentCount,
    map< indivPair, double > & segmentLength,
    bool countCases,
    bool countControls )
```

Definition at line 1687 of file segment.cpp.

6.91.3.98 initialiseGeneCountAssociation()

```
void Plink::initialiseGeneCountAssociation (
    Perm & perm )
```

Definition at line 1934 of file cnv.cpp.

6.91.3.99 keep2SetsForGenome()

```
void Plink::keep2SetsForGenome ( )
```

Definition at line 654 of file output.cpp.

6.91.3.100 keepIndividuals()

```
int Plink::keepIndividuals (
    set< Individual *> & pset )
```

Definition at line 1873 of file helper.cpp.

6.91.3.101 keepSNPs() [1/2]

```
int Plink::keepSNPs (
    set< string > & mset )
```

Definition at line 1853 of file helper.cpp.

6.91.3.102 keepSNPs() [2/2]

```
int Plink::keepSNPs (
    set< Locus *> & mset )
```

Definition at line 1863 of file helper.cpp.

6.91.3.103 linkRelateds()

```
void Plink::linkRelateds (
    map< Individual *, int > & idmap,
    map< string, Individual *> & fnd )
```

Definition at line 351 of file genedrop.cpp.

6.91.3.104 lookup()

```
void Plink::lookup ( )
```

Definition at line 27 of file lookup.cpp.

6.91.3.105 lookup2()

```
void Plink::lookup2 ( )
```

Definition at line 79 of file lookup2.cpp.

6.91.3.106 makeFounders()

```
void Plink::makeFounders ( )
```

Definition at line 992 of file trio.cpp.

6.91.3.107 makeMissingParents()

```
void Plink::makeMissingParents ( )
```

Definition at line 1174 of file trio.cpp.

6.91.3.108 makePhenotype()

```
void Plink::makePhenotype ( )
```

Definition at line 931 of file input.cpp.

6.91.3.109 mergeBinaryData()

```
void Plink::mergeBinaryData ( )
```

Definition at line 26 of file bmerge.cpp.

6.91.3.110 mergeData()

```
void Plink::mergeData ( )
```

Definition at line 119 of file merge.cpp.

6.91.3.111 mergeList()

```
void Plink::mergeList ( )
```

Definition at line 28 of file merge.cpp.

6.91.3.112 metaAnalysis()

```
void Plink::metaAnalysis ( )
```

Definition at line 93 of file metaanal.cpp.

6.91.3.113 missingGenotype()

```
bool Plink::missingGenotype (
    int i,
    int l )
```

Definition at line 1081 of file helper.cpp.

6.91.3.114 mkBlks()

```
map< Range, vector< int > > Plink::mkBlks (
    int null1,
    int null2 )
```

Definition at line 106 of file blox.cpp.

6.91.3.115 multcomp()

```
void Plink::multcomp (
    vector< double > & chi,
    string title )
```

Definition at line 73 of file multiple.cpp.

6.91.3.116 obligMissing()

```
bool Plink::obligMissing (
    int i,
    int l )
```

Definition at line 1074 of file helper.cpp.

6.91.3.117 openBinaryFile()

```
bool Plink::openBinaryFile (
    string s,
    ifstream & BIT )
```

Definition at line 414 of file binput.cpp.

6.91.3.118 output_bimbam_format()

```
void Plink::output_bimbam_format ( )
```

Definition at line 1280 of file output.cpp.

6.91.3.119 output_fastphase_format()

```
void Plink::output_fastphase_format ( )
```

Definition at line 1196 of file output.cpp.

6.91.3.120 output_structure_format()

```
void Plink::output_structure_format ( )
```

Definition at line 1347 of file output.cpp.

6.91.3.121 outputArffFile()

```
bool Plink::outputArffFile (
    string arfffilename )
```

Definition at line 2174 of file output.cpp.

6.91.3.122 outputDelimitedFile()

```
bool Plink::outputDelimitedFile (
    string delimitedFilename,
    string delimiter )
```

Definition at line 2293 of file output.cpp.

6.91.3.123 outputGenericVariantFile()

```
void Plink::outputGenericVariantFile ( )
```

Definition at line 1129 of file gvar.cpp.

6.91.3.124 outputNumericExtract()

```
bool Plink::outputNumericExtract (
    string attributeFilename )
```

Definition at line 2405 of file output.cpp.

6.91.3.125 outputNumericFiltered()

```
bool Plink::outputNumericFiltered (
    string newFilename,
    boolvec_t varFlags )
```

Definition at line 2376 of file output.cpp.

6.91.3.126 outputPermedPhenotypes()

```
void Plink::outputPermedPhenotypes (
    Perm & perm )
```

Definition at line 2995 of file helper.cpp.

6.91.3.127 outputSetFile()

```
void Plink::outputSetFile ( )
```

Definition at line 1960 of file `output.cpp`.

6.91.3.128 outputSifToGain()

```
bool Plink::outputSifToGain (
    string sifFilename )
```

Definition at line 2473 of file `output.cpp`.

6.91.3.129 parseTrios()

```
void Plink::parseTrios ( )
```

Definition at line 118 of file `trio.cpp`.

6.91.3.130 performMisHapTests()

```
void Plink::performMisHapTests ( )
```

Definition at line 25 of file `mishap.cpp`.

6.91.3.131 performProxyTests()

```
void Plink::performProxyTests (
    int l )
```

Definition at line 101 of file `proxy.cpp`.

6.91.3.132 perm_segmentIndividualTest()

```
vector_t Plink::perm_segmentIndividualTest (
    Perm & perm,
    bool display,
    int total_cases,
    int total_controls,
    map< Individual *, int > & ip )
```

Definition at line 249 of file segment.cpp.

6.91.3.133 perm_sharingIBSTest()

```
void Plink::perm_sharingIBSTest (
    Perm & perm )
```

Definition at line 23 of file sharing.cpp.

6.91.3.134 perm_testGXE2()

```
void Plink::perm_testGXE2 (
    Perm & perm )
```

Definition at line 26 of file gxe.cpp.

6.91.3.135 perm_testHotel()

```
void Plink::perm_testHotel (
    Perm & perm )
```

Definition at line 40 of file hotel.cpp.

6.91.3.136 perm_testQTDT()

```
void Plink::perm_testQTDT (
    Perm & perm )
```

Definition at line 241 of file qfam.cpp.

6.91.3.137 perm_testTDT()

```
void Plink::perm_testTDT (
    Perm & perm )
```

Definition at line 33 of file tdt.cpp.

6.91.3.138 perm_testTDT_POO()

```
void Plink::perm_testTDT_POO (
    Perm & perm )
```

Definition at line 33 of file poo.cpp.

6.91.3.139 permTestRareDistribution()

```
void Plink::permTestRareDistribution (
    Perm & perm )
```

Definition at line 507 of file elf.cpp.

6.91.3.140 permutationIBSTest()

```
void Plink::permutationIBSTest (
    Perm & perm )
```

Definition at line 1199 of file cluster.cpp.

6.91.3.141 positionPermuteSegments()

```
void Plink::positionPermuteSegments ( )
```

Definition at line 2001 of file cnv.cpp.

6.91.3.142 preCalcGenomeIBD()

```
void Plink::preCalcGenomeIBD ( )
```

Definition at line 729 of file genome.cpp.

6.91.3.143 preCalcMultiPoint()

```
void Plink::preCalcMultiPoint ( )
```

Definition at line 46 of file multi.cpp.

6.91.3.144 preCalcPhenotypes()

```
void Plink::preCalcPhenotypes ( )
```

Definition at line 244 of file legacy.cpp.

6.91.3.145 preCalcRegression_PHENO()

```
void Plink::preCalcRegression_PHENO (   
    vector< double > & pheno )
```

Definition at line 650 of file legacy.cpp.

6.91.3.146 preCalcRegression_PIHAT()

```
void Plink::preCalcRegression_PIHAT ( )
```

Definition at line 691 of file legacy.cpp.

6.91.3.147 preCalcSinglePoint()

```
void Plink::preCalcSinglePoint ( )
```

Definition at line 24 of file multi.cpp.

6.91.3.148 prettyPrintLengths()

```
void Plink::prettyPrintLengths ( )
```

Definition at line 1087 of file helper.cpp.

6.91.3.149 printLOG()

```
void Plink::printLOG ( string s )
```

Definition at line 32 of file output.cpp.

6.91.3.150 processCNVList()

```
void Plink::processCNVList ( )
```

Definition at line 1433 of file cnv.cpp.

6.91.3.151 processDosageFile()

```
void Plink::processDosageFile ( )
```

Definition at line 57 of file dosage.cpp.

6.91.3.152 processGVAR()

```
void Plink::processGVAR ( )
```

Definition at line 571 of file gvar.cpp.

6.91.3.153 proxyWrapper()

```
void Plink::proxyWrapper ( )
```

Definition at line 20 of file pdriver.cpp.

6.91.3.154 pruneLD()

```
void Plink::pruneLD ( )
```

Definition at line 1172 of file genome.cpp.

6.91.3.155 pseudoCaseControl()

```
void Plink::pseudoCaseControl ( )
```

Definition at line 1015 of file trio.cpp.

6.91.3.156 readBinData()

```
void Plink::readBinData ( )
```

Definition at line 28 of file binput.cpp.

6.91.3.157 readClusterFile()

```
bool Plink::readClusterFile (
    bool verbose = true )
```

Definition at line 631 of file input.cpp.

6.91.3.158 readCNVList()

```
void Plink::readCNVList ( )
```

Definition at line 151 of file cnv.cpp.

6.91.3.159 readConditioningList()

```
void Plink::readConditioningList ( )
```

Definition at line 1860 of file input.cpp.

6.91.3.160 readCovariateFile()

```
bool Plink::readCovariateFile ( )
```

Definition at line 1210 of file input.cpp.

6.91.3.161 readCovListFile()

```
bool Plink::readCovListFile ( )
```

Definition at line 1303 of file input.cpp.

6.91.3.162 readData()

```
void Plink::readData ( )
```

Definition at line 32 of file input.cpp.

6.91.3.163 readDataLongFormat()

```
void Plink::readDataLongFormat ( )
```

Definition at line 30 of file linput.cpp.

6.91.3.164 readFamFile()

```
void Plink::readFamFile (   
    string filename )
```

Definition at line 2148 of file input.cpp.

6.91.3.165 readGenericVariantData()

```
void Plink::readGenericVariantData ( )
```

Definition at line 59 of file gvar.cpp.

6.91.3.166 `readHomozygSegmentFile()`

```
void Plink::readHomozygSegmentFile (
    ifstream & SEG )
```

Definition at line 2329 of file input.cpp.

6.91.3.167 `readInformative()`

```
int Plink::readInformative ( )
```

Definition at line 33 of file informative.cpp.

6.91.3.168 `readMapFile()`

```
void Plink::readMapFile (
    string filename,
    vector< bool > & include,
    vector< int > & include_pos,
    int & nl_actual )
```

Definition at line 1891 of file input.cpp.

6.91.3.169 `readMultiplePhenoFile()`

```
bool Plink::readMultiplePhenoFile ( )
```

Definition at line 1515 of file input.cpp.

6.91.3.170 `readNumericFile()`

```
bool Plink::readNumericFile ( )
```

Definition at line 1043 of file input.cpp.

6.91.3.171 readPhenoFile()

```
bool Plink::readPhenoFile ( )
```

Definition at line 742 of file input.cpp.

6.91.3.172 readSegmentFile()

```
void Plink::readSegmentFile (  
    ifstream & SEG )
```

Definition at line 1716 of file input.cpp.

6.91.3.173 readSegmentFileMinimal()

```
void Plink::readSegmentFileMinimal (   
    ifstream & SEG )
```

Definition at line 1797 of file input.cpp.

6.91.3.174 readSet()

```
void Plink::readSet ( )
```

Definition at line 528 of file input.cpp.

6.91.3.175 readStdIn()

```
void Plink::readStdIn ( )
```

Definition at line 2409 of file input.cpp.

6.91.3.176 `readTransposedData()`

```
void Plink::readTransposedData ( )
```

Definition at line 29 of file `tinput.cpp`.

6.91.3.177 `reconcileMerge()`

```
bool Plink::reconcileMerge (
    int indn,
    int k,
    string one,
    string two,
    bool already_in,
    bool snp_exists,
    ofstream & MERD,
    map< string, int > & misstrand )
```

Definition at line 863 of file `merge.cpp`.

6.91.3.178 `removeIndividuals()`

```
void Plink::removeIndividuals (
    bool keep )
```

Definition at line 592 of file `output.cpp`.

6.91.3.179 `runTestCNVwithGLM()`

```
void Plink::runTestCNVwithGLM (
    Perm & )
```

6.91.3.180 `runTestCNVwithQT()`

```
void Plink::runTestCNVwithQT (
    Perm & perm )
```

Definition at line 30 of file `cnvqt.cpp`.

6.91.3.181 scoreIndividuals()

```
void Plink::scoreIndividuals ( )
```

Definition at line 40 of file profile.cpp.

6.91.3.182 segmentIndividualTest()

```
void Plink::segmentIndividualTest (   
    Perm & perm )
```

Definition at line 155 of file segment.cpp.

6.91.3.183 segmentPermutationTest()

```
void Plink::segmentPermutationTest (   
    Perm & perm,   
    bool ibd,   
    string f,   
    vector< int > & coverage_conc_aff,   
    vector< int > & coverage_disc,   
    vector< int > & coverage_conc_unaff )
```

Definition at line 652 of file segment.cpp.

6.91.3.184 setAssocSummary()

```
void Plink::setAssocSummary ( )
```

Definition at line 43 of file setscreen.cpp.

6.91.3.185 setFlags()

```
void Plink::setFlags (   
    bool f ) [inline]
```

Definition at line 1100 of file plink.h.

6.91.3.186 setFlagToCase()

```
void Plink::setFlagToCase ( )
```

Definition at line 2947 of file helper.cpp.

6.91.3.187 setFlagToControl()

```
void Plink::setFlagToControl ( )
```

Definition at line 2959 of file helper.cpp.

6.91.3.188 setMarkerRange()

```
void Plink::setMarkerRange ( )
```

Definition at line 1314 of file helper.cpp.

6.91.3.189 setObligMissing()

```
void Plink::setObligMissing ( )
```

Definition at line 783 of file output.cpp.

6.91.3.190 setQt1PhenoFromNumericIndex()

```
bool Plink::setQt1PhenoFromNumericIndex ( int index )
```

Definition at line 731 of file input.cpp.

6.91.3.191 setReferenceAllele()

```
void Plink::setReferenceAllele ( )
```

Definition at line 409 of file flip.cpp.

6.91.3.192 setTable()

```
void Plink::setTable ( )
```

Definition at line 2117 of file output.cpp.

6.91.3.193 setUpForCNVList()

```
void Plink::setUpForCNVList ( )
```

Definition at line 109 of file cnv.cpp.

6.91.3.194 sexCheck()

```
void Plink::sexCheck ( )
```

Definition at line 327 of file genome.cpp.

6.91.3.195 sharingIBSTest()

```
vector< double > Plink::sharingIBSTest (
    Perm & perm )
```

Definition at line 120 of file sharing.cpp.

6.91.3.196 simulateSNPs()

```
void Plink::simulateSNPs ( )
```

Definition at line 517 of file simul.cpp.

6.91.3.197 simulateSNPs_QT()

```
void Plink::simulateSNPs_QT ( )
```

Definition at line 1018 of file simul.cpp.

6.91.3.198 SNP2Ind()

```
void Plink::SNP2Ind ( )
```

Definition at line 1775 of file helper.cpp.

6.91.3.199 summariseHomoRuns()

```
void Plink::summariseHomoRuns ( )
```

Definition at line 940 of file homozyg.cpp.

6.91.3.200 summaryIBD()

```
void Plink::summaryIBD ( )
```

Definition at line 1513 of file segment.cpp.

6.91.3.201 summaryIBDsegments()

```
void Plink::summaryIBDsegments (
    Perm & perm )
```

Definition at line 1353 of file segment.cpp.

6.91.3.202 summaryIBSsegments()

```
void Plink::summaryIBSsegments (
    Perm & perm )
```

Definition at line 1181 of file segment.cpp.

6.91.3.203 tagMode()

```
void Plink::tagMode ( )
```

Definition at line 29 of file tag.cpp.

6.91.3.204 testAssoc()

```
vector< double > Plink::testAssoc (  
    int & aff,  
    int & unf,  
    vector< int > & a1,  
    vector< int > & a2,  
    vector< int > & a0,  
    vector< double > & odds,  
    vector< double > & exp_afffreq1,  
    vector< double > & exp_afffreq2,  
    vector< double > & exp_unffreq1,  
    vector< double > & exp_unffreq2,  
    Perm & perm,  
    bool display )
```

Definition at line 821 of file assoc.cpp.

6.91.3.205 testCNVwithGLM()

```
vector_t Plink::testCNVwithGLM (   
    bool ,  
    Perm & ,  
    vector< int > & )
```

6.91.3.206 testCNVwithQT()

```
vector_t Plink::testCNVwithQT (   
    double grandMean,  
    int validN,  
    int nt,  
    vector_t & count,  
    vector_t & m1,  
    vector_t & m0 )
```

Definition at line 231 of file cnvqt.cpp.

6.91.3.207 testGenomeIBDByCovariate()

```
void Plink::testGenomeIBDByCovariate (
    Perm & perm )
```

Definition at line 889 of file segment.cpp.

6.91.3.208 testMiss()

```
vector< double > Plink::testMiss (
    Perm & perm,
    bool display )
```

Definition at line 2130 of file assoc.cpp.

6.91.3.209 testQAssoc()

```
vector< double > Plink::testQAssoc (
    bool print_results,
    Perm & perm )
```

Definition at line 1726 of file assoc.cpp.

6.91.3.210 testQAssocGXE2()

```
vector< double > Plink::testQAssocGXE2 (
    bool print_results,
    Perm & perm )
```

Definition at line 94 of file gxe.cpp.

6.91.3.211 testRareDistribution()

```
vector_t Plink::testRareDistribution (
    Perm & perm,
    bool disp,
    map< Range, int2 > & ranges )
```

Definition at line 643 of file elf.cpp.

6.91.3.212 testSibTDT()

```
vector< double > Plink::testSibTDT (
    bool print_results,
    bool permute,
    Perm & perm,
    vector< bool > & flipA,
    vector< bool > & flipP )
```

Definition at line 30 of file dfam.cpp.

6.91.3.213 testTDT()

```
vector< double > Plink::testTDT (
    bool print_results,
    bool permute,
    Perm & perm,
    vector< bool > & flipA,
    vector< bool > & flipP )
```

Definition at line 443 of file tdt.cpp.

6.91.3.214 testTDT_POO()

```
vector< double > Plink::testTDT_POO (
    bool print_results,
    bool permute,
    Perm & perm,
    vector< bool > & flipA,
    vector< bool > & flipB )
```

Definition at line 226 of file poo.cpp.

6.91.3.215 thinSNPs()

```
void Plink::thinSNPs ( )
```

Definition at line 1155 of file filters.cpp.

6.91.3.216 updateAlleles()

```
void Plink::updateAlleles ( )
```

Definition at line 2531 of file input.cpp.

6.91.3.217 updateFamFile()

```
void Plink::updateFamFile ( )
```

Definition at line 2623 of file input.cpp.

6.91.3.218 updateMapFile()

```
void Plink::updateMapFile ( )
```

Definition at line 2416 of file input.cpp.

6.91.3.219 validateSegments()

```
void Plink::validateSegments ( )
```

Definition at line 1103 of file segment.cpp.

6.91.3.220 webcheck()

```
void Plink::webcheck (   
    CArgs & a )
```

Definition at line 33 of file webcheck.cpp.

6.91.3.221 write_BITFILE()

```
void Plink::write_BITFILE ( )
```

Definition at line 1766 of file output.cpp.

6.91.3.222 write_clusters()

```
void Plink::write_clusters ( )
```

Definition at line 1727 of file output.cpp.

6.91.3.223 write_covariates()

```
void Plink::write_covariates ( )
```

Definition at line 1611 of file output.cpp.

6.91.3.224 write_snplist()

```
void Plink::write_snplist ( )
```

Definition at line 1757 of file output.cpp.

6.91.3.225 writeInformative()

```
void Plink::writeInformative ( )
```

Definition at line 349 of file informative.cpp.

6.91.3.226 writeSetFile()

```
void Plink::writeSetFile ( )
```

Definition at line 2158 of file output.cpp.

6.91.3.227 zeroOnCluster()

```
void Plink::zeroOnCluster ( )
```

Definition at line 720 of file output.cpp.

6.91.4 Member Data Documentation

6.91.4.1 `clistname`

```
vector<string> Plink::clistname
```

Definition at line 628 of file plink.h.

6.91.4.2 `cnt_f`

```
int Plink::cnt_f
```

Definition at line 634 of file plink.h.

6.91.4.3 `conditioner`

```
vector<int> Plink::conditioner
```

Definition at line 660 of file plink.h.

6.91.4.4 `conditioner_mask`

```
vector<bool> Plink::conditioner_mask
```

Definition at line 661 of file plink.h.

6.91.4.5 `diff_concordant_overlap`

```
long int Plink::diff_concordant_overlap
```

Definition at line 742 of file plink.h.

6.91.4.6 diff_nonmissing_overlap

```
long int Plink::diff_nonmissing_overlap
```

Definition at line 741 of file plink.h.

6.91.4.7 diff_overlap

```
long int Plink::diff_overlap
```

Definition at line 740 of file plink.h.

6.91.4.8 dst

```
double Plink::dst
```

Definition at line 727 of file plink.h.

6.91.4.9 E00

```
double Plink::E00
```

Definition at line 705 of file plink.h.

6.91.4.10 E01

```
double Plink::E01
```

Definition at line 706 of file plink.h.

6.91.4.11 E02

```
double Plink::E02
```

Definition at line 707 of file plink.h.

6.91.4.12 E10

```
double Plink::E10
```

Definition at line 705 of file plink.h.

6.91.4.13 E11

```
double Plink::E11
```

Definition at line 706 of file plink.h.

6.91.4.14 E12

```
double Plink::E12
```

Definition at line 707 of file plink.h.

6.91.4.15 E20

```
double Plink::E20
```

Definition at line 705 of file plink.h.

6.91.4.16 E21

```
double Plink::E21
```

Definition at line 706 of file plink.h.

6.91.4.17 E22

```
double Plink::E22
```

Definition at line 707 of file plink.h.

6.91.4.18 epi1

```
vector<bool> Plink::epi1
```

Definition at line 732 of file plink.h.

6.91.4.19 epi2

```
vector<bool> Plink::epi2
```

Definition at line 733 of file plink.h.

6.91.4.20 expectedOverlap

```
vector_t Plink::expectedOverlap
```

Definition at line 758 of file plink.h.

6.91.4.21 expectedOverlapBaseline

```
vector_t Plink::expectedOverlapBaseline
```

Definition at line 759 of file plink.h.

6.91.4.22 family

```
vector<Family*> Plink::family
```

Definition at line 622 of file plink.h.

6.91.4.23 gene2segment

```
map<Range, set<Segment> > Plink::gene2segment
```

Definition at line 687 of file plink.h.

6.91.4.24 geneList

```
set<Range> Plink::geneList
```

Definition at line 686 of file plink.h.

6.91.4.25 gset1

```
set<Individual*> Plink::gset1
```

Definition at line 736 of file plink.h.

6.91.4.26 gset2

```
set<Individual*> Plink::gset2
```

Definition at line 737 of file plink.h.

6.91.4.27 gvar

```
vector<Variant*> Plink::gvar
```

Definition at line 619 of file plink.h.

6.91.4.28 haplo

```
HaploPhase* Plink::haplo
```

Definition at line 895 of file plink.h.

6.91.4.29 in_anal

```
vector<int> Plink::in_anal
```

Definition at line 695 of file plink.h.

6.91.4.30 indivSegmentGroup

```
vector<int> Plink::indivSegmentGroup
```

Definition at line 689 of file plink.h.

6.91.4.31 klist

```
vector<Cluster*> Plink::klist
```

Definition at line 654 of file plink.h.

6.91.4.32 kmap

```
map<string, int> Plink::kmap
```

Definition at line 652 of file plink.h.

6.91.4.33 kname

```
vector<string> Plink::kname
```

Definition at line 653 of file plink.h.

6.91.4.34 locus

```
vector<Locus*> Plink::locus
```

Definition at line 610 of file plink.h.

6.91.4.35 m1

```
vector<int> Plink::m1
```

Definition at line 677 of file plink.h.

6.91.4.36 m2

```
vector<int> Plink::m2
```

Definition at line 678 of file plink.h.

6.91.4.37 m_phenotype

```
double Plink::m_phenotype
```

Definition at line 698 of file plink.h.

6.91.4.38 m_pihat

```
vector<double> Plink::m_pihat
```

Definition at line 701 of file plink.h.

6.91.4.39 maxr2

```
vector<double> Plink::maxr2
```

Definition at line 715 of file plink.h.

6.91.4.40 mdist

```
vector<vector<double> > Plink::mdist
```

Definition at line 725 of file plink.h.

6.91.4.41 model

```
Model* Plink::model
```

Definition at line 898 of file plink.h.

6.91.4.42 n

```
int Plink::n
```

Definition at line 633 of file plink.h.

6.91.4.43 ngvar

```
int Plink::ngvar
```

Definition at line 638 of file plink.h.

6.91.4.44 nk

```
int Plink::nk
```

Definition at line 640 of file plink.h.

6.91.4.45 nl

```
int Plink::nl
```

Definition at line 639 of file plink.h.

6.91.4.46 nl_all

```
int Plink::nl_all
```

Definition at line 637 of file plink.h.

6.91.4.47 nlistname

```
vector<string> Plink::nlistname
```

Definition at line 630 of file plink.h.

6.91.4.48 np

```
int Plink::np
```

Definition at line 636 of file plink.h.

6.91.4.49 npheno

```
int Plink::npheno
```

Definition at line 635 of file plink.h.

6.91.4.50 oblig_missing

```
set<int2> Plink::oblig_missing
```

Definition at line 657 of file plink.h.

6.91.4.51 original

```
vector<vector<double>> Plink::original
```

Definition at line 722 of file plink.h.

6.91.4.52 OUTFILE

```
ofstream Plink::OUTFILE
```

Definition at line 645 of file plink.h.

6.91.4.53 pair1

```
vector<int> Plink::pair1
```

Definition at line 693 of file plink.h.

6.91.4.54 pair2

```
vector<int> Plink::pair2
```

Definition at line 694 of file plink.h.

6.91.4.55 pairid

```
string Plink::pairid
```

Definition at line 667 of file plink.h.

6.91.4.56 phenoLabel

```
string Plink::phenoLabel
```

Definition at line 625 of file plink.h.

6.91.4.57 phenotype

```
vector<double> Plink::phenotype
```

Definition at line 692 of file plink.h.

6.91.4.58 phenotype_name

```
string Plink::phenotype_name
```

Definition at line 642 of file plink.h.

6.91.4.59 pihat

```
vector< vector<double> > Plink::pihat
```

Definition at line 682 of file plink.h.

6.91.4.60 pihat_G

```
vector<double> Plink::pihat_G
```

Definition at line 690 of file plink.h.

6.91.4.61 plistname

```
vector<string> Plink::plistname
```

Definition at line 626 of file plink.h.

6.91.4.62 pos

```
vector<double> Plink::pos
```

Definition at line 679 of file plink.h.

6.91.4.63 pperm

```
Perm* Plink::pperm
```

Definition at line 762 of file plink.h.

6.91.4.64 prev_bt

```
double Plink::prev_bt
```

Definition at line 700 of file plink.h.

6.91.4.65 proxyLD

```
map<int2, double> Plink::proxyLD
```

Definition at line 748 of file plink.h.

6.91.4.66 pS

```
Set* Plink::pS
```

Definition at line 904 of file plink.h.

6.91.4.67 pv

```
double Plink::pv
```

Definition at line 726 of file plink.h.

6.91.4.68 pvlBS0

```
double Plink::pvlBS0
```

Definition at line 728 of file plink.h.

6.91.4.69 pvlBS2het

```
double Plink::pvlBS2het
```

Definition at line 729 of file plink.h.

6.91.4.70 related

```
set<int2> Plink::related
```

Definition at line 691 of file plink.h.

6.91.4.71 sample

```
vector<Individual*> Plink::sample
```

Definition at line 604 of file plink.h.

6.91.4.72 saved_IBDg

```
vector<Z> Plink::saved_IBDg
```

Definition at line 674 of file plink.h.

6.91.4.73 scaffold

```
map<int, CInfo> Plink::scaffold
```

Definition at line 616 of file plink.h.

6.91.4.74 segment

```
vector<Segment> Plink::segment
```

Definition at line 685 of file plink.h.

6.91.4.75 segmentCount

```
map<indivPair, int> Plink::segmentCount
```

Definition at line 751 of file plink.h.

6.91.4.76 segmentCount2

```
map<indivPair, double> Plink::segmentCount2
```

Definition at line 754 of file plink.h.

6.91.4.77 segmentCount2Baseline

```
map<indivPair, double> Plink::segmentCount2Baseline
```

Definition at line 755 of file plink.h.

6.91.4.78 segmentLength

```
map<indivPair, double> Plink::segmentLength
```

Definition at line 752 of file plink.h.

6.91.4.79 setname

```
vector<string> Plink::setname
```

Definition at line 718 of file plink.h.

6.91.4.80 skip_pair

```
vector<bool> Plink::skip_pair
```

Definition at line 664 of file plink.h.

6.91.4.81 SNP

```
vector<CSNP\*> Plink::SNP
```

Definition at line 607 of file plink.h.

6.91.4.82 snpset

```
vector<vector<int> > Plink::snpset
```

Definition at line 719 of file plink.h.

6.91.4.83 T00

```
double Plink::T00
```

Definition at line 710 of file plink.h.

6.91.4.84 T01

```
double Plink::T01
```

Definition at line 710 of file plink.h.

6.91.4.85 T02

```
double Plink::T02
```

Definition at line 710 of file plink.h.

6.91.4.86 T10

```
double Plink::T10
```

Definition at line 711 of file plink.h.

6.91.4.87 T11

```
double Plink::T11
```

Definition at line 711 of file plink.h.

6.91.4.88 T12

```
double Plink::T12
```

Definition at line 711 of file plink.h.

6.91.4.89 T20

```
double Plink::T20
```

Definition at line 712 of file plink.h.

6.91.4.90 T21

```
double Plink::T21
```

Definition at line 712 of file plink.h.

6.91.4.91 T22

```
double Plink::T22
```

Definition at line 712 of file plink.h.

6.91.4.92 tcnt

```
vector_t Plink::tcnt
```

Definition at line 745 of file plink.h.

6.91.4.93 transcript

```
vector<Transcript*> Plink::transcript
```

Definition at line 613 of file plink.h.

6.91.4.94 v_phenotype

```
double Plink::v_phenotype
```

Definition at line 699 of file plink.h.

6.91.4.95 v_pihat

```
vector<double> Plink::v_pihat
```

Definition at line 702 of file plink.h.

6.91.4.96 warnings

```
bool Plink::warnings
```

Definition at line 649 of file plink.h.

6.91.4.97 whap

```
Chap* Plink::whap
```

Definition at line 901 of file plink.h.

6.91.4.98 Zlocus

```
vector<Z> Plink::Zlocus
```

Definition at line 671 of file plink.h.

6.91.4.99 ZOUTFILE

```
ZOutput Plink::ZOUTFILE
```

Definition at line 646 of file plink.h.

The documentation for this class was generated from the following files:

- src/[plink.h](#)
- src/[annot.cpp](#)
- src/[assoc.cpp](#)
- src/[binput.cpp](#)
- src/[blox.cpp](#)
- src/[bmerge.cpp](#)
- src/[cluster.cpp](#)
- src/[cnv.cpp](#)
- src/[cnvqt.cpp](#)
- src/[dfam.cpp](#)
- src/[dosage.cpp](#)
- src/[elf.cpp](#)
- src/[epi.cpp](#)
- src/[filters.cpp](#)
- src/[flip.cpp](#)

- src/[genedrop.cpp](#)
- src/[genepi.cpp](#)
- src/[genome.cpp](#)
- src/[glm.cpp](#)
- src/[greport.cpp](#)
- src/[gvar.cpp](#)
- src/[gxe.cpp](#)
- src/[hapglm.cpp](#)
- src/[haplohelper.cpp](#)
- src/[helper.cpp](#)
- src/[homozyg.cpp](#)
- src/[hotel.cpp](#)
- src/[informative.cpp](#)
- src/[input.cpp](#)
- src/[legacy.cpp](#)
- src/[linput.cpp](#)
- src/[locus.cpp](#)
- src/[lookup.cpp](#)
- src/[lookup2.cpp](#)
- src/[mds.cpp](#)
- src/[merge.cpp](#)
- src/[metaanal.cpp](#)
- src/[mh.cpp](#)
- src/[mishap.cpp](#)
- src/[multi.cpp](#)
- src/[multiple.cpp](#)
- src/[output.cpp](#)
- src/[pdriver.cpp](#)
- src/[plink.cpp](#)
- src/[poo.cpp](#)
- src/[profile.cpp](#)
- src/[proxy.cpp](#)
- src/[qfam.cpp](#)
- src/[qualscores.cpp](#)
- src/[segment.cpp](#)
- src/[setscreen.cpp](#)
- src/[sharing.cpp](#)
- src/[simul.cpp](#)
- src/[tag.cpp](#)
- src/[tdt.cpp](#)
- src/[tinput.cpp](#)
- src/[trio.cpp](#)
- src/[webcheck.cpp](#)
- src/[whap.cpp](#)

6.92 PlinkInternalsDataset Class Reference

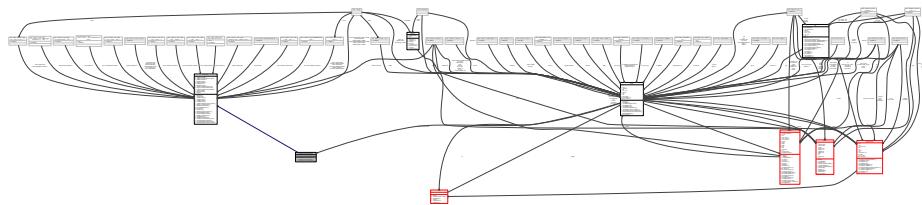
[Plink](#) internals adapter class.

```
#include <PlinkInternalsDataset.h>
```

Inheritance diagram for PlinkInternalsDataset:



Collaboration diagram for PlinkInternalsDataset:



Public Member Functions

- [PlinkInternalsDataset \(Plink *plinkPtr\)](#)
- [bool LoadDatasetFromPlink \(\)](#)
- [~PlinkInternalsDataset \(\)](#)

Private Attributes

- [Plink * plinkInternalsPtr](#)

Additional Inherited Members

6.92.1 Detailed Description

Plink internals adapter class.

See also

[Dataset](#)

Author

Bill White

Version

1.0

Contact: bill.c.white@gmail.com Created on: 2/24/11

Modified for inclusion in inbix 8/3/16

Definition at line 26 of file PlinkInternalsDataset.h.

6.92.2 Constructor & Destructor Documentation

6.92.2.1 PlinkInternalsDataset()

```
PlinkInternalsDataset::PlinkInternalsDataset (   
    Plink * plinkPtr )
```

Definition at line 18 of file PlinkInternalsDataset.cpp.

6.92.2.2 ~PlinkInternalsDataset()

```
PlinkInternalsDataset::~PlinkInternalsDataset ( )
```

Definition at line 23 of file PlinkInternalsDataset.cpp.

6.92.3 Member Function Documentation

6.92.3.1 LoadDatasetFromPlink()

```
bool PlinkInternalsDataset::LoadDatasetFromPlink ( )
```

Definition at line 26 of file PlinkInternalsDataset.cpp.

6.92.4 Member Data Documentation

6.92.4.1 plinkInternalsPtr

```
Plink* PlinkInternalsDataset::plinkInternalsPtr [private]
```

Definition at line 33 of file PlinkInternalsDataset.h.

The documentation for this class was generated from the following files:

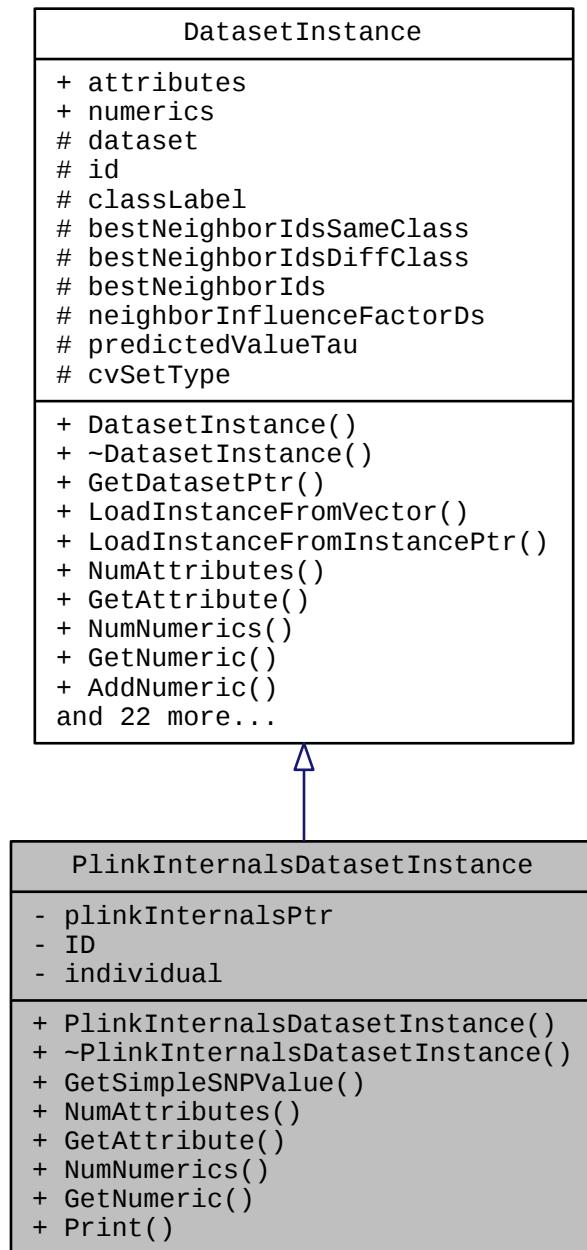
- src/[PlinkInternalsDataset.h](#)
- src/[PlinkInternalsDataset.cpp](#)

6.93 PlinkInternalsDatasetInstance Class Reference

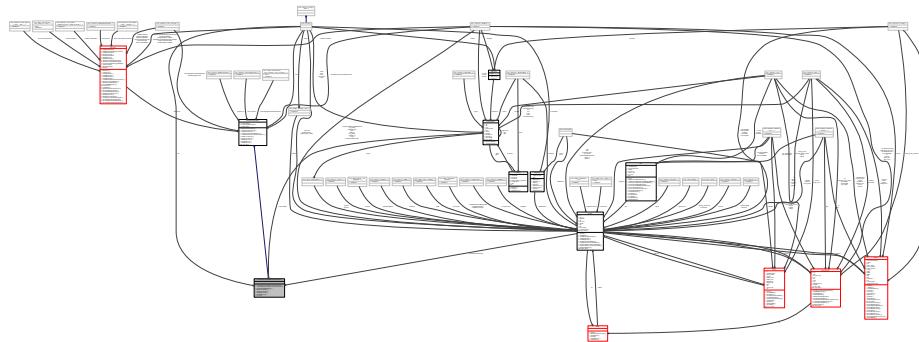
Class to hold dataset instances (rows of attributes).

```
#include <PlinkInternalsDatasetInstance.h>
```

Inheritance diagram for PlinkInternalsDatasetInstance:



Collaboration diagram for PlinkInternalsDatasetInstance:



Public Member Functions

- **PlinkInternalsDatasetInstance** (`Dataset *ds, std::string instanceID, Plink *plinkPtr, Individual *plinkInd`)

Construct an data set instance object.
- **~PlinkInternalsDatasetInstance ()**
- **double GetSimpleSNPValue (int snp)**
- **unsigned int NumAttributes () override**

return the number of discrete attributes
- **AttributeLevel GetAttribute (unsigned int index) override**

Get and return an attribute value at index.
- **unsigned int NumNumerics () override**

return the number of continuous attributes
- **NumericLevel GetNumeric (unsigned int index) override**

Get and return numeric value at index.
- **void Print () override**

Print the attributes, numerics and class name of this instance to stdout.

Private Attributes

- **Plink * plinkInternalsPtr**
- **string ID**
- **Individual * individual**

Additional Inherited Members

6.93.1 Detailed Description

Class to hold dataset instances (rows of attributes).

Reworked entirely for McKinney Lab work - 2/28/11

Author

Bill White

Version

1.0

Contact: bill.c.white@gmail.com Created on: 6/14/05 Modified for inclusion in inbix 8/3/16

Definition at line 30 of file PlinkInternalsDatasetInstance.h.

6.93.2 Constructor & Destructor Documentation

6.93.2.1 PlinkInternalsDatasetInstance()

```
PlinkInternalsDatasetInstance::PlinkInternalsDatasetInstance (
    Dataset * ds,
    std::string instanceID,
    Plink * plinkPtr,
    Individual * plinkInd )
```

Construct an data set instance object.

Parameters

in	ds	pointer to a Dataset object
----	----	---

Definition at line 28 of file PlinkInternalsDatasetInstance.cpp.

6.93.2.2 ~PlinkInternalsDatasetInstance()

```
PlinkInternalsDatasetInstance::~PlinkInternalsDatasetInstance ( )
```

Definition at line 38 of file PlinkInternalsDatasetInstance.cpp.

6.93.3 Member Function Documentation

6.93.3.1 GetAttribute()

```
AttributeLevel PlinkInternalsDatasetInstance::GetAttribute (
    unsigned int index ) [override], [virtual]
```

Get and return an attribute value at index.

Parameters

in	<i>index</i>	attribute index
----	--------------	-----------------

Returns

attribute value at index

Reimplemented from [DatasetInstance](#).

Definition at line 68 of file PlinkInternalsDatasetInstance.cpp.

6.93.3.2 GetNumeric()

```
double PlinkInternalsDatasetInstance::GetNumeric (
    unsigned int index ) [override], [virtual]
```

Get and return numeric value at index.

Parameters

in	<i>index</i>	numeric index
----	--------------	---------------

Returns

numeric value at index

Reimplemented from [DatasetInstance](#).

Definition at line 84 of file PlinkInternalsDatasetInstance.cpp.

6.93.3.3 GetSimpleSNPValue()

```
double PlinkInternalsDatasetInstance::GetSimpleSNPValue (
    int snp )
```

Definition at line 41 of file PlinkInternalsDatasetInstance.cpp.

6.93.3.4 NumAttributes()

```
unsigned int PlinkInternalsDatasetInstance::NumAttributes ( ) [override], [virtual]
```

return the number of discrete attributes

Reimplemented from [DatasetInstance](#).

Definition at line 64 of file PlinkInternalsDatasetInstance.cpp.

6.93.3.5 NumNumerics()

```
unsigned int PlinkInternalsDatasetInstance::NumNumerics ( ) [override], [virtual]
```

return the number of continuous attributes

Reimplemented from [DatasetInstance](#).

Definition at line 80 of file PlinkInternalsDatasetInstance.cpp.

6.93.3.6 Print()

```
void PlinkInternalsDatasetInstance::Print ( ) [override], [virtual]
```

Print the attributes, numerics and class name of this instance to stdout.

Reimplemented from [DatasetInstance](#).

Definition at line 97 of file PlinkInternalsDatasetInstance.cpp.

6.93.4 Member Data Documentation

6.93.4.1 ID

```
string PlinkInternalsDatasetInstance::ID [private]
```

Definition at line 60 of file PlinkInternalsDatasetInstance.h.

6.93.4.2 individual

```
Individual* PlinkInternalsDatasetInstance::individual [private]
```

Definition at line 61 of file PlinkInternalsDatasetInstance.h.

6.93.4.3 plinkInternalsPtr

```
Plink* PlinkInternalsDatasetInstance::plinkInternalsPtr [private]
```

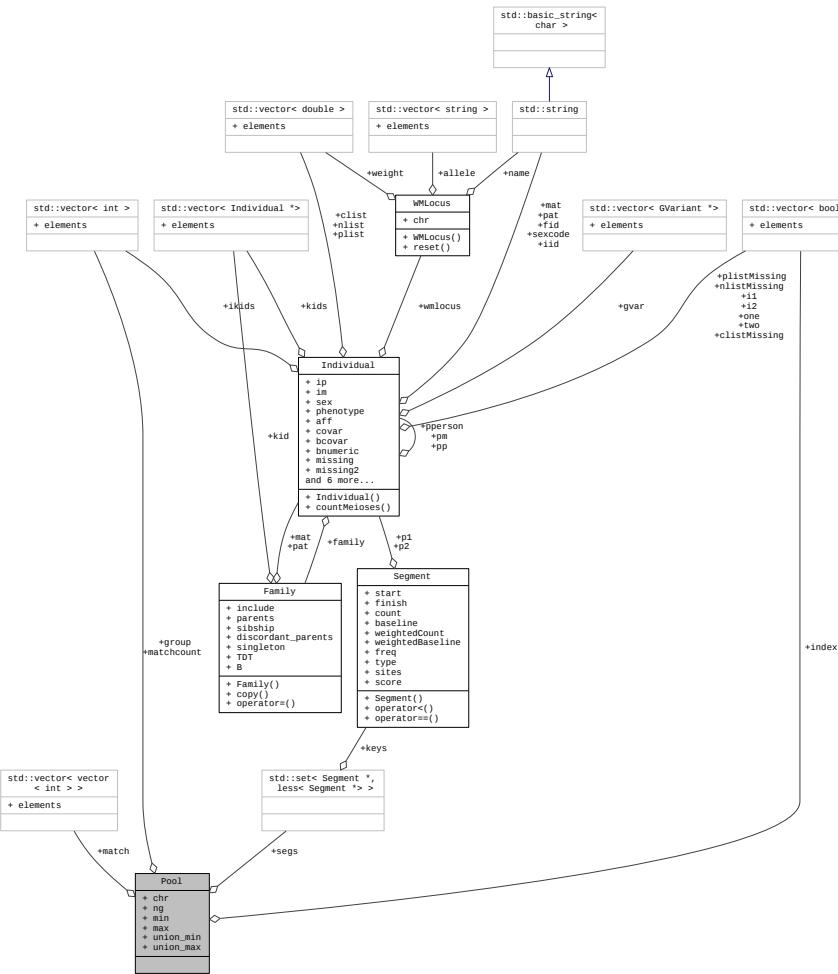
Definition at line 59 of file PlinkInternalsDatasetInstance.h.

The documentation for this class was generated from the following files:

- src/[PlinkInternalsDatasetInstance.h](#)
- src/[PlinkInternalsDatasetInstance.cpp](#)

6.94 Pool Class Reference

Collaboration diagram for Pool:



Public Attributes

- set< Segment *, less< Segment * > > segs
 - vector< vector< int > > match
 - vector< int > matchcount
 - vector< int > group
 - vector< bool > index
 - int chr
 - int ng
 - int min
 - int max
 - int union_min
 - int union_max

6.94.1 Detailed Description

Definition at line 51 of file homozyg.cpp.

6.94.2 Member Data Documentation

6.94.2.1 chr

```
int Pool::chr
```

Definition at line 59 of file homozyg.cpp.

6.94.2.2 group

```
vector<int> Pool::group
```

Definition at line 57 of file homozyg.cpp.

6.94.2.3 index

```
vector<bool> Pool::index
```

Definition at line 58 of file homozyg.cpp.

6.94.2.4 match

```
vector<vector<int>> Pool::match
```

Definition at line 55 of file homozyg.cpp.

6.94.2.5 matchcount

```
vector<int> Pool::matchcount
```

Definition at line 56 of file homozyg.cpp.

6.94.2.6 max

```
int Pool::max
```

Definition at line 62 of file homozyg.cpp.

6.94.2.7 min

```
int Pool::min
```

Definition at line 61 of file homozyg.cpp.

6.94.2.8 ng

```
int Pool::ng
```

Definition at line 60 of file homozyg.cpp.

6.94.2.9 segs

```
set<Segment\*, less<Segment\*>> Pool::segs
```

Definition at line 54 of file homozyg.cpp.

6.94.2.10 union_max

```
int Pool::union_max
```

Definition at line 64 of file homozyg.cpp.

6.94.2.11 union_min

```
int Pool::union_min
```

Definition at line 63 of file homozyg.cpp.

The documentation for this class was generated from the following file:

- [src/homozyg.cpp](#)

6.95 probabilisticGenotype Class Reference

Collaboration diagram for probabilisticGenotype:

probabilisticGenotype
+ calculated + AA + AB + BA + BB + phased + genotyped + genotype + phased_genotype
+ probabilisticGenotype()

Public Member Functions

- [probabilisticGenotype \(\)](#)

Public Attributes

- bool [calculated](#)
- double [AA](#)
- double [AB](#)
- double [BA](#)
- double [BB](#)
- bool [phased](#)
- bool [genotyped](#)
- int [genotype](#)
- int [phased_genotype](#)

6.95.1 Detailed Description

Definition at line 35 of file impute.cpp.

6.95.2 Constructor & Destructor Documentation

6.95.2.1 probabilisticGenotype()

```
probabilisticGenotype::probabilisticGenotype () [inline]
```

Definition at line 37 of file impute.cpp.

6.95.3 Member Data Documentation

6.95.3.1 AA

```
double probabilisticGenotype::AA
```

Definition at line 46 of file impute.cpp.

6.95.3.2 AB

```
double probabilisticGenotype::AB
```

Definition at line 46 of file impute.cpp.

6.95.3.3 BA

```
double probabilisticGenotype::BA
```

Definition at line 46 of file impute.cpp.

6.95.3.4 BB

```
double probabilisticGenotype::BB
```

Definition at line 46 of file impute.cpp.

6.95.3.5 calculated

```
bool probabilisticGenotype::calculated
```

Definition at line 45 of file impute.cpp.

6.95.3.6 genotype

```
int probabilisticGenotype::genotype
```

Definition at line 49 of file impute.cpp.

6.95.3.7 genotyped

```
bool probabilisticGenotype::genotyped
```

Definition at line 48 of file impute.cpp.

6.95.3.8 phased

```
bool probabilisticGenotype::phased
```

Definition at line 47 of file impute.cpp.

6.95.3.9 phased_genotype

```
int probabilisticGenotype::phased_genotype
```

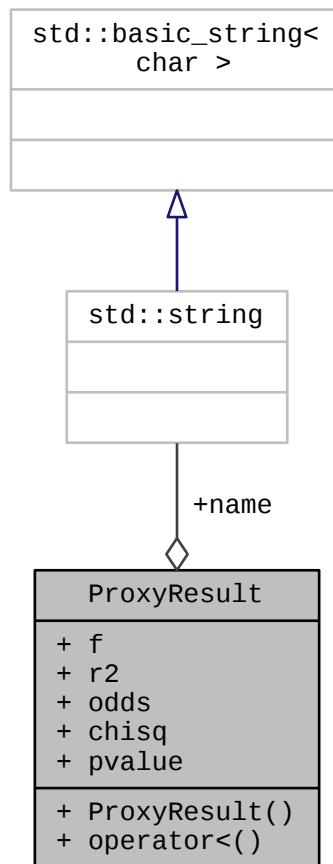
Definition at line 50 of file impute.cpp.

The documentation for this class was generated from the following file:

- [src/impute.cpp](#)

6.96 ProxyResult Class Reference

Collaboration diagram for ProxyResult:



Public Member Functions

- `ProxyResult (string n, double frq, double r, double o, double c, double p)`
- `bool operator< (const ProxyResult &b) const`

Public Attributes

- string `name`
- double `f`
- double `r2`
- double `odds`
- double `chisq`
- double `pvalue`

6.96.1 Detailed Description

Definition at line 32 of file proxy.cpp.

6.96.2 Constructor & Destructor Documentation

6.96.2.1 ProxyResult()

```
ProxyResult::ProxyResult (
    string n,
    double frq,
    double r,
    double o,
    double c,
    double p ) [inline]
```

Definition at line 42 of file proxy.cpp.

6.96.3 Member Function Documentation

6.96.3.1 operator<()

```
bool ProxyResult::operator< (
    const ProxyResult & b ) const [inline]
```

Definition at line 45 of file proxy.cpp.

6.96.4 Member Data Documentation

6.96.4.1 chisq

```
double ProxyResult::chisq
```

Definition at line 39 of file proxy.cpp.

6.96.4.2 f

```
double ProxyResult::f
```

Definition at line 36 of file proxy.cpp.

6.96.4.3 name

```
string ProxyResult::name
```

Definition at line 35 of file proxy.cpp.

6.96.4.4 odds

```
double ProxyResult::odds
```

Definition at line 38 of file proxy.cpp.

6.96.4.5 pvalue

```
double ProxyResult::pvalue
```

Definition at line 40 of file proxy.cpp.

6.96.4.6 r2

```
double ProxyResult::r2
```

Definition at line 37 of file proxy.cpp.

The documentation for this class was generated from the following file:

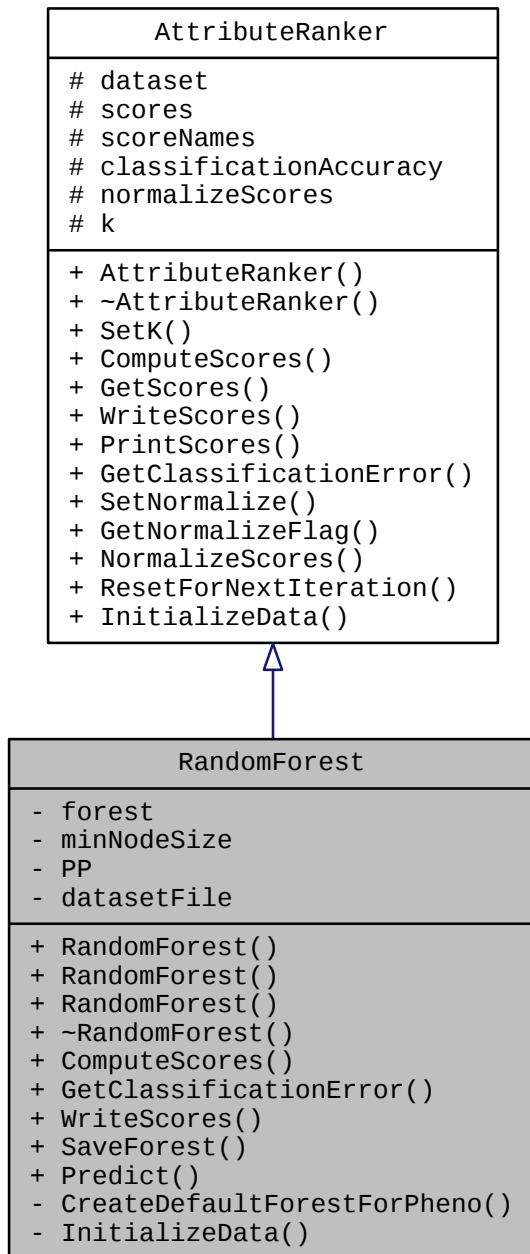
- [src/proxy.cpp](#)

6.97 RandomForest Class Reference

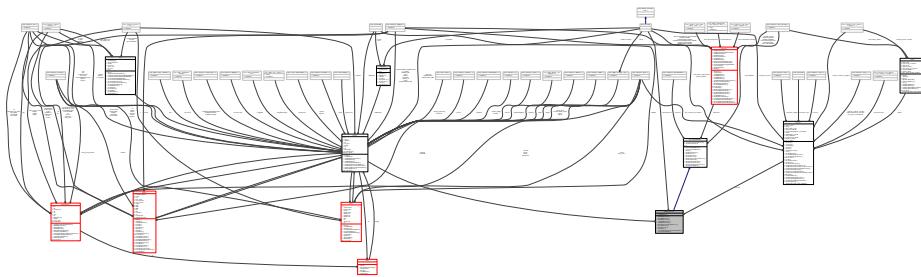
[RandomForest](#) attribute ranking algorithm.

```
#include <RandomForest.h>
```

Inheritance diagram for RandomForest:



Collaboration diagram for RandomForest:



Public Member Functions

- **RandomForest (Dataset *ds, Plink *plinkPtr, bool doPrediction=false, bool computeImportance=true)**
Construct an [RandomForest](#) algorithm object.
- **RandomForest (Dataset *ds, std::vector< std::string > bestAttributeNames, bool doPrediction=false)**
Construct an [RandomForest](#) algorithm object.
- **RandomForest (Dataset *ds, std::string datasetFilename, std::vector< std::string > bestAttributeNames, bool doPrediction=false)**
Construct an [RandomForest](#) algorithm object.
- **virtual ~RandomForest ()**
Deconstruct an [RandomForest](#) algorithm object.
- **AttributeScores ComputeScores () override**
Compute the attribute scores for the current set of attributes.
- **double GetClassificationError () override**
Error from using ranked attributes in a classifier.
- **void WriteScores (std::string baseFilename) override**
Write the scores and attribute names to file.
- **void SaveForest ()**
- **double Predict ()**

Private Member Functions

- **bool CreateDefaultForestForPheno ()**
- **bool InitializeData (bool doPrediction, bool useMaske, bool dolImportance) override**

Private Attributes

- **Forest *forest**
- **unsigned int minNodeSize**
- **Plink * PP**
- **std::string datasetFile**

Additional Inherited Members

6.97.1 Detailed Description

[RandomForest](#) attribute ranking algorithm.

Adapter class to map EC call for Random Jungle importance scores to Random Jungle library functions.

Author

Bill White

Version

1.0

Contact: bill.c.white@gmail.com Created on: 10/16/11

Definition at line 28 of file [RandomForest.h](#).

6.97.2 Constructor & Destructor Documentation

6.97.2.1 [RandomForest\(\)](#) [1/3]

```
RandomForest::RandomForest (
    Dataset * ds,
    PLINK * plinkPtr,
    bool doPrediction = false,
    bool computeImportance = true )
```

Construct an [RandomForest](#) algorithm object.

Parameters

in	<i>ds</i>	pointer to a Dataset object
in	<i>plinkPtr</i>	pointer to a PLINK environment

Definition at line 34 of file [RandomForest.cpp](#).

6.97.2.2 RandomForest() [2/3]

```
RandomForest::RandomForest (
    Dataset * ds,
    std::vector< std::string > bestAttributeNames,
    bool doPrediction = false )
```

Construct an [RandomForest](#) algorithm object.

Parameters

in	<i>ds</i>	pointer to a Dataset object
in	<i>bestAttributeNames</i>	best attribute names in the data set to use

Definition at line 56 of file RandomForest.cpp.

6.97.2.3 RandomForest() [3/3]

```
RandomForest::RandomForest (
    Dataset * ds,
    std::string datasetFilename,
    std::vector< std::string > bestAttributeNames,
    bool doPrediction = false )
```

Construct an [RandomForest](#) algorithm object.

Parameters

in	<i>datasetFilename</i>	dataset filename
in	<i>bestAttributeNames</i>	best attribute names in the data set to use

Definition at line 78 of file RandomForest.cpp.

6.97.2.4 ~RandomForest()

```
RandomForest::~RandomForest ( ) [virtual]
```

Deconstruct an [RandomForest](#) algorithm object.

Definition at line 156 of file RandomForest.cpp.

6.97.3 Member Function Documentation

6.97.3.1 ComputeScores()

```
AttributeScores RandomForest::ComputeScores () [override], [virtual]
```

Compute the attribute scores for the current set of attributes.

Implements [AttributeRanker](#).

Definition at line 160 of file RandomForest.cpp.

6.97.3.2 CreateDefaultForestForPheno()

```
bool RandomForest::CreateDefaultForestForPheno () [private]
```

Definition at line 228 of file RandomForest.cpp.

6.97.3.3 GetClassificationError()

```
double RandomForest::GetClassificationError () [override], [virtual]
```

Error from using ranked attributes in a classifier.

Reimplemented from [AttributeRanker](#).

Definition at line 200 of file RandomForest.cpp.

6.97.3.4 InitializeData()

```
bool RandomForest::InitializeData (
    bool doPrediction,
    bool useMaske,
    bool doImportance ) [override], [private], [virtual]
```

Reimplemented from [AttributeRanker](#).

Definition at line 100 of file RandomForest.cpp.

6.97.3.5 Predict()

```
double RandomForest::Predict ( )
```

Definition at line 180 of file RandomForest.cpp.

6.97.3.6 SaveForest()

```
void RandomForest::SaveForest ( )
```

Definition at line 218 of file RandomForest.cpp.

6.97.3.7 WriteScores()

```
void RandomForest::WriteScores (
    std::string baseFilename ) [override], [virtual]
```

Write the scores and attribute names to file.

Parameters

in	<i>baseFilename</i>	filename to write score-attribute name pairs
----	---------------------	--

Reimplemented from [AttributeRanker](#).

Definition at line 208 of file RandomForest.cpp.

6.97.4 Member Data Documentation

6.97.4.1 datasetFile

```
std::string RandomForest::datasetFile [private]
```

Definition at line 72 of file RandomForest.h.

6.97.4.2 forest

```
Forest* RandomForest::forest [private]
```

Definition at line 69 of file RandomForest.h.

6.97.4.3 minNodeSize

```
unsigned int RandomForest::minNodeSize [private]
```

Definition at line 70 of file RandomForest.h.

6.97.4.4 PP

```
Plink* RandomForest::PP [private]
```

Definition at line 71 of file RandomForest.h.

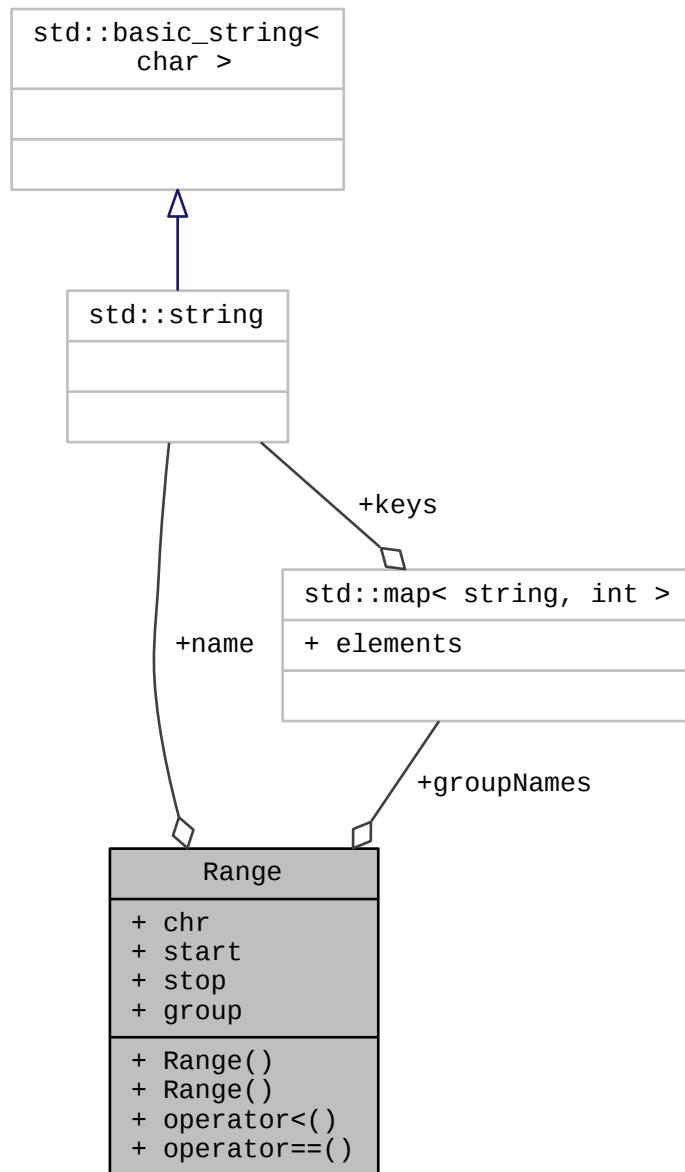
The documentation for this class was generated from the following files:

- src/[RandomForest.h](#)
- src/[RandomForest.cpp](#)

6.98 Range Class Reference

```
#include <plink.h>
```

Collaboration diagram for Range:



Public Member Functions

- `Range()`
- `Range(int p1, int p2, int p3, string p4)`
- `bool operator<(const Range &b) const`
- `bool operator==(const Range &b) const`

Public Attributes

- int `chr`
- int `start`
- int `stop`
- string `name`
- int `group`

Static Public Attributes

- static map< string, int > `groupNames`

6.98.1 Detailed Description

Definition at line 162 of file plink.h.

6.98.2 Constructor & Destructor Documentation

6.98.2.1 Range() [1/2]

```
Range::Range ( ) [inline]
```

Definition at line 171 of file plink.h.

6.98.2.2 Range() [2/2]

```
Range::Range ( 
    int p1,
    int p2,
    int p3,
    string p4 ) [inline]
```

Definition at line 174 of file plink.h.

6.98.3 Member Function Documentation

6.98.3.1 operator<()

```
bool Range::operator< (
    const Range & b ) const [inline]
```

Definition at line 181 of file plink.h.

6.98.3.2 operator==()

```
bool Range::operator== (
    const Range & b ) const [inline]
```

Definition at line 193 of file plink.h.

6.98.4 Member Data Documentation

6.98.4.1 chr

```
int Range::chr
```

Definition at line 164 of file plink.h.

6.98.4.2 group

```
int Range::group
```

Definition at line 168 of file plink.h.

6.98.4.3 groupNames

```
map< string, int > Range::groupNames [static]
```

Definition at line 169 of file plink.h.

6.98.4.4 name

```
string Range::name
```

Definition at line 167 of file plink.h.

6.98.4.5 start

```
int Range::start
```

Definition at line 165 of file plink.h.

6.98.4.6 stop

```
int Range::stop
```

Definition at line 166 of file plink.h.

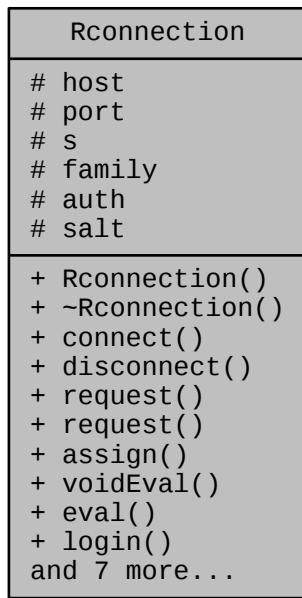
The documentation for this class was generated from the following files:

- src/[plink.h](#)
- src/[inbix.cpp](#)

6.99 Rconnection Class Reference

```
#include <Rconnection.h>
```

Collaboration diagram for Rconnection:



Public Member Functions

- **Rconnection** (const char ***host**="127.0.0.1", int **port**=**default_Rsrv_port**)

host - either host name or unix socket path port - either TCP port or -1 if unix sockets should be used
- virtual **~Rconnection** ()
- int **connect** ()
- int **disconnect** ()
- int **request** (**Rmessage** *msg, int cmd, int len=0, void *par=0)

— low-level functions (should not be used directly) —
- int **request** (**Rmessage** *targetMsg, **Rmessage** *contents)
- int **assign** (const char *symbol, **Rexp** *exp)

— high-level functions —
- int **voidEval** (const char *cmd)
- **Rexp** * **eval** (const char *cmd, int *status=0, int opt=0)
- int **login** (const char *user, const char *pwd)
- int **shutdown** (const char *key)

— high-level functions —
- int **openFile** (const char *fn)
- int **createFile** (const char *fn)
- int **readFile** (char *buf, unsigned int len)
- int **writeFile** (const char *buf, unsigned int len)
- int **closeFile** ()
- int **removeFile** (const char *fn)

Protected Attributes

- char * `host`
- int `port`
- SOCKET `s`
- int `family`
- int `auth`
- char `salt` [2]

6.99.1 Detailed Description

Definition at line 362 of file Rconnection.h.

6.99.2 Constructor & Destructor Documentation

6.99.2.1 Rconnection()

```
Rconnection::Rconnection (
    const char * host = "127.0.0.1",
    int port = default_Rsrv_port )
```

host - either host name or unix socket path
port - either TCP port or -1 if unix sockets should be used

Definition at line 555 of file Rconnection.cpp.

6.99.2.2 ~Rconnection()

```
Rconnection::~Rconnection ( ) [virtual]
```

Definition at line 566 of file Rconnection.cpp.

6.99.3 Member Function Documentation

6.99.3.1 assign()

```
int Rconnection::assign (
    const char * symbol,
    Rexp * exp )
```

— high-level functions —

Definition at line 688 of file Rconnection.cpp.

6.99.3.2 closeFile()

```
int Rconnection::closeFile ( )
```

Definition at line 803 of file Rconnection.cpp.

6.99.3.3 connect()

```
int Rconnection::connect ( )
```

Definition at line 572 of file Rconnection.cpp.

6.99.3.4 createFile()

```
int Rconnection::createFile (
    const char * fn )
```

Definition at line 758 of file Rconnection.cpp.

6.99.3.5 disconnect()

```
int Rconnection::disconnect ( )
```

Definition at line 640 of file Rconnection.cpp.

6.99.3.6 eval()

```
Rexp * Rconnection::eval (
    const char * cmd,
    int * status = 0,
    int opt = 0 )
```

Definition at line 724 of file Rconnection.cpp.

6.99.3.7 login()

```
int Rconnection::login (
    const char * user,
    const char * pwd )
```

Definition at line 827 of file Rconnection.cpp.

6.99.3.8 openFile()

```
int Rconnection::openFile (
    const char * fn )
```

Definition at line 748 of file Rconnection.cpp.

6.99.3.9 readFile()

```
int Rconnection::readFile (
    char * buf,
    unsigned int len )
```

Definition at line 768 of file Rconnection.cpp.

6.99.3.10 removeFile()

```
int Rconnection::removeFile (
    const char * fn )
```

Definition at line 817 of file Rconnection.cpp.

6.99.3.11 request() [1/2]

```
int Rconnection::request (
    Rmessage * msg,
    int cmd,
    int len = 0,
    void * par = 0 )
```

— low-level functions (should not be used directly) —

— low-level functions –

Definition at line 650 of file Rconnection.cpp.

6.99.3.12 request() [2/2]

```
int Rconnection::request (
    Rmessage * targetMsg,
    Rmessage * contents )
```

Definition at line 668 of file Rconnection.cpp.

6.99.3.13 shutdown()

```
int Rconnection::shutdown (
    const char * key )
```

— high-level functions –

Definition at line 679 of file Rconnection.cpp.

6.99.3.14 voidEval()

```
int Rconnection::voidEval (
    const char * cmd )
```

Definition at line 718 of file Rconnection.cpp.

6.99.3.15 writeFile()

```
int Rconnection::writeFile (
    const char * buf,
    unsigned int len )
```

Definition at line 789 of file Rconnection.cpp.

6.99.4 Member Data Documentation

6.99.4.1 auth

```
int Rconnection::auth [protected]
```

Definition at line 368 of file Rconnection.h.

6.99.4.2 family

```
int Rconnection::family [protected]
```

Definition at line 367 of file Rconnection.h.

6.99.4.3 host

```
char* Rconnection::host [protected]
```

Definition at line 364 of file Rconnection.h.

6.99.4.4 port

```
int Rconnection::port [protected]
```

Definition at line 365 of file Rconnection.h.

6.99.4.5 s

```
SOCKET Rconnection::s [protected]
```

Definition at line 366 of file Rconnection.h.

6.99.4.6 salt

```
char Rconnection::salt[2] [protected]
```

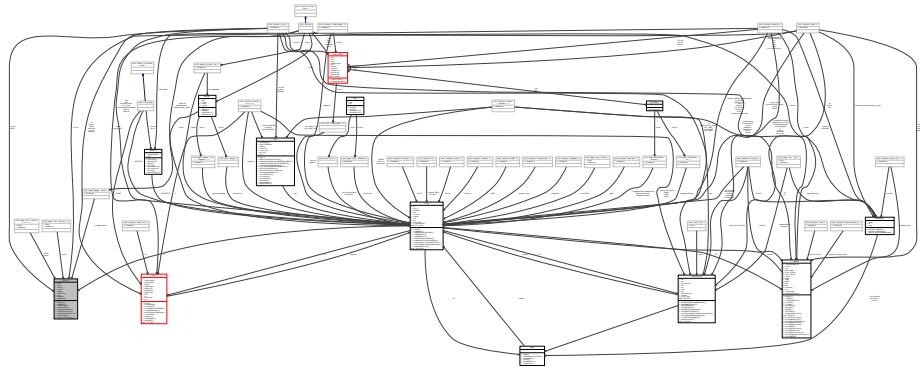
Definition at line 369 of file Rconnection.h.

The documentation for this class was generated from the following files:

- src/[Rconnection.h](#)
- src/[Rconnection.cpp](#)

6.100 RCount Class Reference

Collaboration diagram for RCount:



Public Member Functions

- [RCount \(Plink *p_, map< Range, int2 > *rl_\)](#)
- [bool addSNP \(int l\)](#)
- [bool removeSNP \(int l\)](#)
- [bool setWindow \(int chr, int bp\)](#)
- [void displayWindow \(\)](#)
- [void loadCovariate \(\)](#)
- [void loadPCACovariate \(\)](#)
- [void mainStats \(\)](#)
- [bool ignorePosition \(\)](#)

Public Attributes

- `Plink * P`
- `map< Range, int2 > * rangeLookup`
- `vector_t rval`
- `vector< int > gval`
- `vector< int > aval`
- `map< int, vector_t > rwin`
- `map< int, vector< int > > gwin`
- `map< int, vector< int > > awin`
- `double acnt`
- `double ucnt`
- `int acnt2`
- `int ucnt2`
- `int nsnps`
- `int nalleles`
- `int npc`
- `bool pcMode`
- `bool domModel`

6.100.1 Detailed Description

Definition at line 28 of file elf.cpp.

6.100.2 Constructor & Destructor Documentation

6.100.2.1 RCount()

```
RCount::RCount (
    Plink * p_,
    map< Range, int2 > * rl_ ) [inline]
```

Definition at line 32 of file elf.cpp.

6.100.3 Member Function Documentation

6.100.3.1 addSNP()

```
bool RCount::addSNP ( int l )
```

Definition at line 358 of file elf.cpp.

6.100.3.2 displayWindow()

```
void RCount::displayWindow ( )
```

Definition at line 267 of file elf.cpp.

6.100.3.3 ignorePosition()

```
bool RCount::ignorePosition ( )
```

Definition at line 79 of file elf.cpp.

6.100.3.4 loadCovariate()

```
void RCount::loadCovariate ( )
```

Definition at line 86 of file elf.cpp.

6.100.3.5 loadPCACovariate()

```
void RCount::loadPCACovariate ( )
```

Definition at line 98 of file elf.cpp.

6.100.3.6 mainStats()

```
void RCount::mainStats ( )
```

Definition at line 303 of file elf.cpp.

6.100.3.7 removeSNP()

```
bool RCount::removeSNP (
    int l )
```

Definition at line 442 of file elf.cpp.

6.100.3.8 setWindow()

```
bool RCount::setWindow (
    int chr,
    int bp )
```

Definition at line 207 of file elf.cpp.

6.100.4 Member Data Documentation

6.100.4.1 acnt

```
double RCount::acnt
```

Definition at line 60 of file elf.cpp.

6.100.4.2 acnt2

```
int RCount::acnt2
```

Definition at line 61 of file elf.cpp.

6.100.4.3 aval

```
vector<int> RCount::aval
```

Definition at line 53 of file elf.cpp.

6.100.4.4 awin

```
map<int, vector<int> > RCount::awin
```

Definition at line 58 of file elf.cpp.

6.100.4.5 domModel

```
bool RCount::domModel
```

Definition at line 67 of file elf.cpp.

6.100.4.6 gval

```
vector<int> RCount::gval
```

Definition at line 52 of file elf.cpp.

6.100.4.7 gwin

```
map<int, vector<int> > RCount::gwin
```

Definition at line 57 of file elf.cpp.

6.100.4.8 nalleles

```
int RCount::nalleles
```

Definition at line 63 of file elf.cpp.

6.100.4.9 npc

```
int RCount::npc
```

Definition at line 64 of file elf.cpp.

6.100.4.10 nsnps

```
int RCount::nsnps
```

Definition at line 62 of file elf.cpp.

6.100.4.11 P

```
PLink* RCount::P
```

Definition at line 48 of file elf.cpp.

6.100.4.12 pcMode

```
bool RCount::pcMode
```

Definition at line 66 of file elf.cpp.

6.100.4.13 rangeLookup

```
map<Range, int2>* RCount::rangeLookup
```

Definition at line 49 of file elf.cpp.

6.100.4.14 rval

```
vector_t RCount::rval
```

Definition at line 51 of file elf.cpp.

6.100.4.15 rwin

```
map<int, vector_t > RCount::rwin
```

Definition at line 56 of file elf.cpp.

6.100.4.16 ucnt

```
double RCount::ucnt
```

Definition at line 60 of file elf.cpp.

6.100.4.17 ucnt2

```
int RCount::ucnt2
```

Definition at line 61 of file elf.cpp.

The documentation for this class was generated from the following file:

- src/[elf.cpp](#)

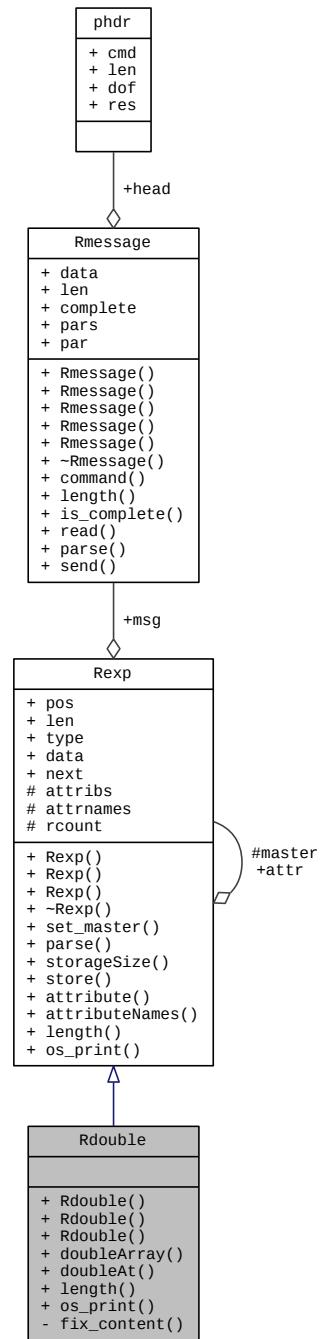
6.101 Rdouble Class Reference

```
#include <Rconnection.h>
```

Inheritance diagram for Rdouble:



Collaboration diagram for Rdouble:



Public Member Functions

- [Rdouble \(Rmessage *msg\)](#)

- `Rdouble` (`unsigned int *ipos, Rmessage *imsg`)
- `Rdouble` (`double *array, int count`)
- `double * doubleArray ()`
- `double doubleAt (int pos)`
- `virtual Rsize_t length ()`
- `virtual std::ostream & os_print (std::ostream &os)`

Private Member Functions

- `void fix_content ()`

Additional Inherited Members

6.101.1 Detailed Description

Definition at line 177 of file Rconnection.h.

6.101.2 Constructor & Destructor Documentation

6.101.2.1 `Rdouble()` [1/3]

```
Rdouble::Rdouble (
    Rmessage * msg )  [inline]
```

Definition at line 179 of file Rconnection.h.

6.101.2.2 `Rdouble()` [2/3]

```
Rdouble::Rdouble (
    unsigned int * ipos,
    Rmessage * imsg )  [inline]
```

Definition at line 180 of file Rconnection.h.

6.101.2.3 Rdouble() [3/3]

```
Rdouble::Rdouble (
    double * array,
    int count ) [inline]
```

Definition at line 181 of file Rconnection.h.

6.101.3 Member Function Documentation

6.101.3.1 doubleArray()

```
double* Rdouble::doubleArray () [inline]
```

Definition at line 183 of file Rconnection.h.

6.101.3.2 doubleAt()

```
double Rdouble::doubleAt (
    int pos ) [inline]
```

Definition at line 184 of file Rconnection.h.

6.101.3.3 fix_content()

```
void Rdouble::fix_content () [private]
```

Definition at line 372 of file Rconnection.cpp.

6.101.3.4 length()

```
virtual Rsize_t Rdouble::length () [inline], [virtual]
```

Reimplemented from [Rexp](#).

Definition at line 185 of file Rconnection.h.

6.101.3.5 os_print()

```
virtual std::ostream& Rdouble::os_print (
    std::ostream & os ) [inline], [virtual]
```

Reimplemented from [Rexp](#).

Definition at line 187 of file [Rconnection.h](#).

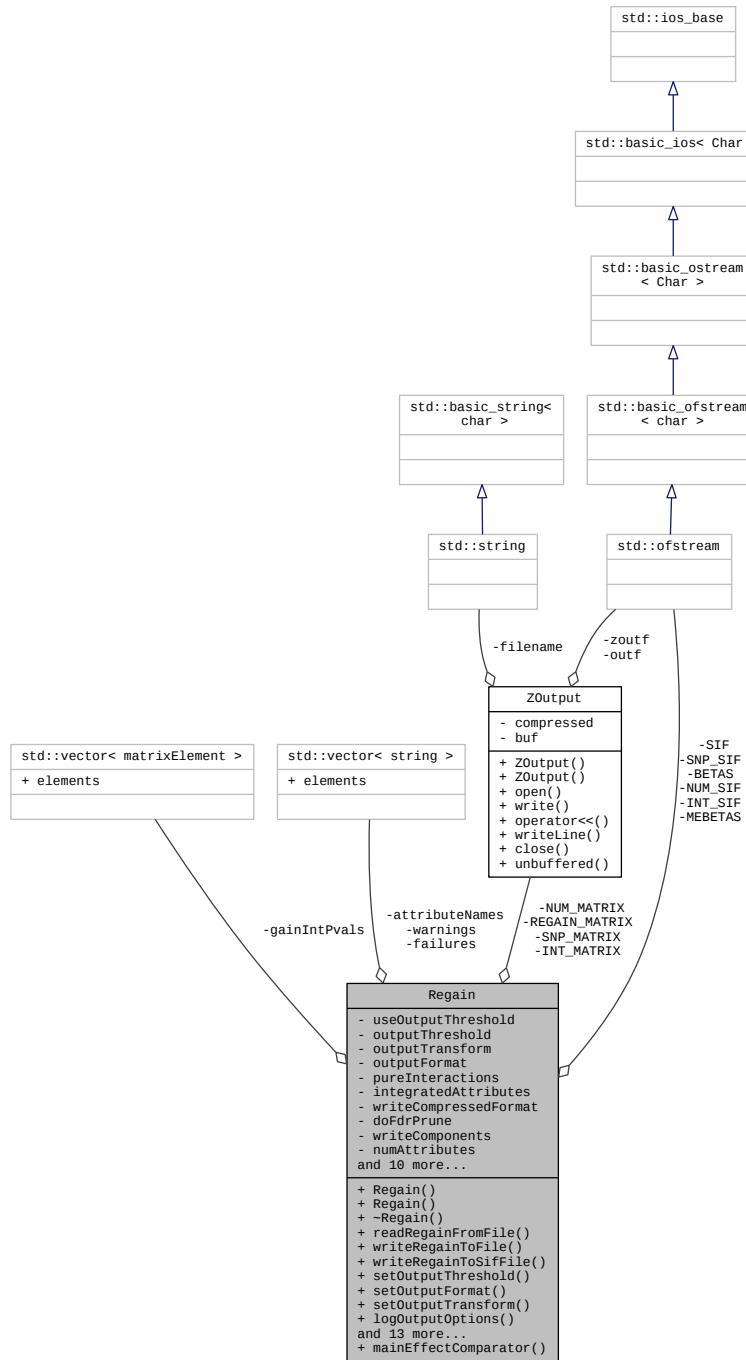
The documentation for this class was generated from the following files:

- src/[Rconnection.h](#)
- src/[Rconnection.cpp](#)

6.102 Regain Class Reference

```
#include <regain.h>
```

Collaboration diagram for Regain:



Public Member Functions

- [Regain](#) (bool compr, double sifthr, bool compo)

- `Regain` (bool compr, double sifthr, bool integrative, bool compo, bool fdrpr=false, bool initMatrixFromData=true)
- `~Regain ()`
- `bool readRegainFromFile` (string regainFilename)
- `bool writeRegainToFile` (string newRegainFilename)
- `bool writeRegainToSifFile` (string newSifFilename)
- `bool setOutputThreshold` (double threshold)
- `bool setOutputFormat` (`RegainOutputFormat` format)
- `bool setOutputTransform` (`RegainOutputTransform` transform)
- `void logOutputOptions ()`
- `void run ()`
- `void mainEffect` (int varIndex, bool varIsNumeric)
- `void addCovariates` (`Model` &m)
- `void interactionEffect` (int varIndex1, bool var1IsNumeric, int varIndex2, bool var2IsNumeric)
- `void pureInteractionEffect` (int varIndex1, bool var1IsNumeric, int varIndex2, bool var2IsNumeric)
- `void writeRegain` (bool pvalues, bool fdrprune=false)
- `void fdrPrune` (double fdr)
- `void writeRcomm` (double T, double fdr)
- `void performPureInteraction` (bool flag)
- `void setFailureValue` (double fValue)
- `bool updateStats ()`
- `bool logMatrixStats ()`
- `double ** getRawMatrix ()`

Static Public Member Functions

- static `bool mainEffectComparator` (const `matrixElement` &l, const `matrixElement` &r)

Private Attributes

- `bool useOutputThreshold`
- `double outputThreshold`
- `RegainOutputTransform outputTransform`
- `RegainOutputFormat outputFormat`
- `bool pureInteractions`
- `bool integratedAttributes`
- `bool writeCompressedFormat`
- `bool doFdrPrune`
- `bool writeComponents`
- `int numAttributes`
- `vector< string > attributeNameNames`
- `double sifThresh`
- `ZOutput REGAIN_MATRIX`
- `ZOutput SNP_MATRIX`
- `ZOutput NUM_MATRIX`
- `ZOutput INT_MATRIX`
- `ofstream MEBETAS`
- `ofstream BETAS`
- `ofstream SIF`
- `ofstream SNP_SIF`

- ofstream `NUM_SIF`
- ofstream `INT_SIF`
- double ** `regainMatrix`
- double ** `regainPMatrix`
- vector< `matrixElement` > `gainIntPvals`
- vector< string > `warnings`
- vector< string > `failures`
- double `failureValue`
- int `nanCount`
- int `infCount`
- double `minMainEffect`
- double `maxMainEffect`
- double `minInteraction`
- double `maxInteraction`

6.102.1 Detailed Description

Definition at line 38 of file `regain.h`.

6.102.2 Constructor & Destructor Documentation

6.102.2.1 Regain() [1/2]

```
Regain::Regain (
    bool compr,
    double sifthr,
    bool compo )
```

Definition at line 39 of file `regain.cpp`.

6.102.2.2 Regain() [2/2]

```
Regain::Regain (
    bool compr,
    double sifthr,
    bool integrative,
    bool compo,
    bool fdrpr = false,
    bool initMatrixFromData = true )
```

Definition at line 61 of file `regain.cpp`.

6.102.2.3 ~Regain()

```
Regain::~Regain ( )
```

Definition at line 225 of file regain.cpp.

6.102.3 Member Function Documentation

6.102.3.1 addCovariates()

```
void Regain::addCovariates (
    Model & m )
```

Definition at line 624 of file regain.cpp.

6.102.3.2 fdrPrune()

```
void Regain::fdrPrune (
    double fdr )
```

Definition at line 1311 of file regain.cpp.

6.102.3.3 getRawMatrix()

```
double** Regain::getRawMatrix ( ) [inline]
```

Definition at line 92 of file regain.h.

6.102.3.4 interactionEffect()

```
void Regain::interactionEffect (
    int varIndex1,
    bool var1IsNumeric,
    int varIndex2,
    bool var2IsNumeric )
```

Definition at line 632 of file regain.cpp.

6.102.3.5 logMatrixStats()

```
bool Regain::logMatrixStats ( )
```

Definition at line 1430 of file regain.cpp.

6.102.3.6 logOutputOptions()

```
void Regain::logOutputOptions ( )
```

Definition at line 199 of file regain.cpp.

6.102.3.7 mainEffect()

```
void Regain::mainEffect (
    int varIndex,
    bool varIsNumeric )
```

Definition at line 471 of file regain.cpp.

6.102.3.8 mainEffectComparator()

```
bool Regain::mainEffectComparator (
    const matrixElement & l,
    const matrixElement & r ) [static]
```

Definition at line 1399 of file regain.cpp.

6.102.3.9 performPureInteraction()

```
void Regain::performPureInteraction (
    bool flag )
```

Definition at line 179 of file regain.cpp.

6.102.3.10 pureInteractionEffect()

```
void Regain::pureInteractionEffect (
    int varIndex1,
    bool var1IsNumeric,
    int varIndex2,
    bool var2IsNumeric )
```

Definition at line 888 of file regain.cpp.

6.102.3.11 readRegainFromFile()

```
bool Regain::readRegainFromFile (
    string regainFilename )
```

Definition at line 245 of file regain.cpp.

6.102.3.12 run()

```
void Regain::run ( )
```

Definition at line 398 of file regain.cpp.

6.102.3.13 setFailureValue()

```
void Regain::setFailureValue (
    double fValue )
```

Definition at line 175 of file regain.cpp.

6.102.3.14 setOutputFormat()

```
bool Regain::setOutputFormat (
    RegainOutputFormat format )
```

Definition at line 189 of file regain.cpp.

6.102.3.15 setOutputThreshold()

```
bool Regain::setOutputThreshold (
    double threshold )
```

Definition at line 183 of file regain.cpp.

6.102.3.16 setOutputTransform()

```
bool Regain::setOutputTransform (
    RegainOutputTransform transform )
```

Definition at line 194 of file regain.cpp.

6.102.3.17 updateStats()

```
bool Regain::updateStats ( )
```

Definition at line 1404 of file regain.cpp.

6.102.3.18 writeRcomm()

```
void Regain::writeRcomm (
    double T,
    double fdr )
```

Definition at line 1358 of file regain.cpp.

6.102.3.19 writeRegain()

```
void Regain::writeRegain (
    bool pvalues,
    bool fdrprune = false )
```

Definition at line 1146 of file regain.cpp.

6.102.3.20 writeRegainToFile()

```
bool Regain::writeRegainToFile (
    string newRegainFilename )
```

Definition at line 309 of file regain.cpp.

6.102.3.21 writeRegainToSifFile()

```
bool Regain::writeRegainToSifFile (
    string newSifFilename )
```

Definition at line 374 of file regain.cpp.

6.102.4 Member Data Documentation

6.102.4.1 attributeNames

```
vector<string> Regain::attributeNames [private]
```

Definition at line 112 of file regain.h.

6.102.4.2 BETAS

```
ofstream Regain::BETAS [private]
```

Definition at line 122 of file regain.h.

6.102.4.3 doFdrPrune

```
bool Regain::doFdrPrune [private]
```

Definition at line 106 of file regain.h.

6.102.4.4 failures

```
vector<string> Regain::failures [private]
```

Definition at line 135 of file regain.h.

6.102.4.5 failureValue

```
double Regain::failureValue [private]
```

Definition at line 137 of file regain.h.

6.102.4.6 gainIntPvals

```
vector<matrixElement> Regain::gainIntPvals [private]
```

Definition at line 131 of file regain.h.

6.102.4.7 infCount

```
int Regain::infCount [private]
```

Definition at line 139 of file regain.h.

6.102.4.8 INT_MATRIX

```
ZOutput Regain::INT_MATRIX [private]
```

Definition at line 119 of file regain.h.

6.102.4.9 INT_SIF

```
ofstream Regain::INT_SIF [private]
```

Definition at line 126 of file regain.h.

6.102.4.10 integratedAttributes

```
bool Regain::integratedAttributes [private]
```

Definition at line 102 of file regain.h.

6.102.4.11 maxInteraction

```
double Regain::maxInteraction [private]
```

Definition at line 144 of file regain.h.

6.102.4.12 maxMainEffect

```
double Regain::maxMainEffect [private]
```

Definition at line 142 of file regain.h.

6.102.4.13 MEBETAS

```
ofstream Regain::MEBETAS [private]
```

Definition at line 121 of file regain.h.

6.102.4.14 minInteraction

```
double Regain::minInteraction [private]
```

Definition at line 143 of file regain.h.

6.102.4.15 minMainEffect

```
double Regain::minMainEffect [private]
```

Definition at line 141 of file regain.h.

6.102.4.16 nanCount

```
int Regain::nanCount [private]
```

Definition at line 138 of file regain.h.

6.102.4.17 NUM_MATRIX

```
ZOutput Regain::NUM_MATRIX [private]
```

Definition at line 118 of file regain.h.

6.102.4.18 NUM_SIF

```
ofstream Regain::NUM_SIF [private]
```

Definition at line 125 of file regain.h.

6.102.4.19 numAttributes

```
int Regain::numAttributes [private]
```

Definition at line 110 of file regain.h.

6.102.4.20 outputFormat

```
RegainOutputFormat Regain::outputFormat [private]
```

Definition at line 98 of file regain.h.

6.102.4.21 outputThreshold

```
double Regain::outputThreshold [private]
```

Definition at line 96 of file regain.h.

6.102.4.22 outputTransform

```
ZOutput Regain::outputTransform [private]
```

Definition at line 97 of file regain.h.

6.102.4.23 pureInteractions

```
bool Regain::pureInteractions [private]
```

Definition at line 100 of file regain.h.

6.102.4.24 REGAIN_MATRIX

```
ZOutput Regain::REGAIN_MATRIX [private]
```

Definition at line 116 of file regain.h.

6.102.4.25 regainMatrix

```
double** Regain::regainMatrix [private]
```

Definition at line 128 of file regain.h.

6.102.4.26 regainPMatrix

```
double** Regain::regainPMatrix [private]
```

Definition at line 129 of file regain.h.

6.102.4.27 SIF

```
ofstream Regain::SIF [private]
```

Definition at line 123 of file regain.h.

6.102.4.28 sifThresh

```
double Regain::sifThresh [private]
```

Definition at line 114 of file regain.h.

6.102.4.29 SNP_MATRIX

```
ZOutput Regain::SNP_MATRIX [private]
```

Definition at line 117 of file regain.h.

6.102.4.30 SNP_SIF

```
ofstream Regain::SNP_SIF [private]
```

Definition at line 124 of file regain.h.

6.102.4.31 useOutputThreshold

```
bool Regain::useOutputThreshold [private]
```

Definition at line 95 of file regain.h.

6.102.4.32 warnings

```
vector<string> Regain::warnings [private]
```

Definition at line 133 of file regain.h.

6.102.4.33 writeComponents

```
bool Regain::writeComponents [private]
```

Definition at line 108 of file regain.h.

6.102.4.34 writeCompressedFormat

```
bool Regain::writeCompressedFormat [private]
```

Definition at line 104 of file regain.h.

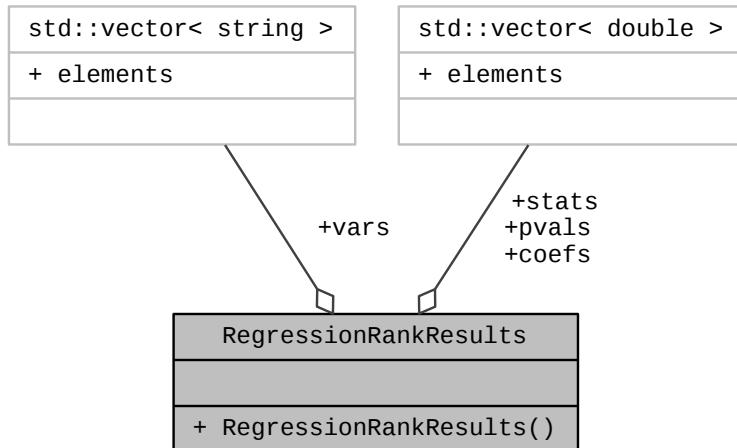
The documentation for this class was generated from the following files:

- [src/regain.h](#)
- [src/regain.cpp](#)

6.103 RegressionRankResults Class Reference

```
#include <plink.h>
```

Collaboration diagram for RegressionRankResults:



Public Member Functions

- [RegressionRankResults \(\)](#)

Public Attributes

- `vector< string > vars`
- `vector_t coefs`
- `vector_t pvals`
- `vector_t stats`

6.103.1 Detailed Description

Definition at line 78 of file plink.h.

6.103.2 Constructor & Destructor Documentation

6.103.2.1 RegressionRankResults()

```
RegressionRankResults::RegressionRankResults () [inline]
```

Definition at line 85 of file plink.h.

6.103.3 Member Data Documentation

6.103.3.1 coefs

```
vector_t RegressionRankResults::coefs
```

Definition at line 81 of file plink.h.

6.103.3.2 pvals

```
vector_t RegressionRankResults::pvals
```

Definition at line 82 of file plink.h.

6.103.3.3 stats

```
vector_t RegressionRankResults::stats
```

Definition at line 83 of file plink.h.

6.103.3.4 vars

```
vector<string> RegressionRankResults::vars
```

Definition at line 80 of file plink.h.

The documentation for this class was generated from the following file:

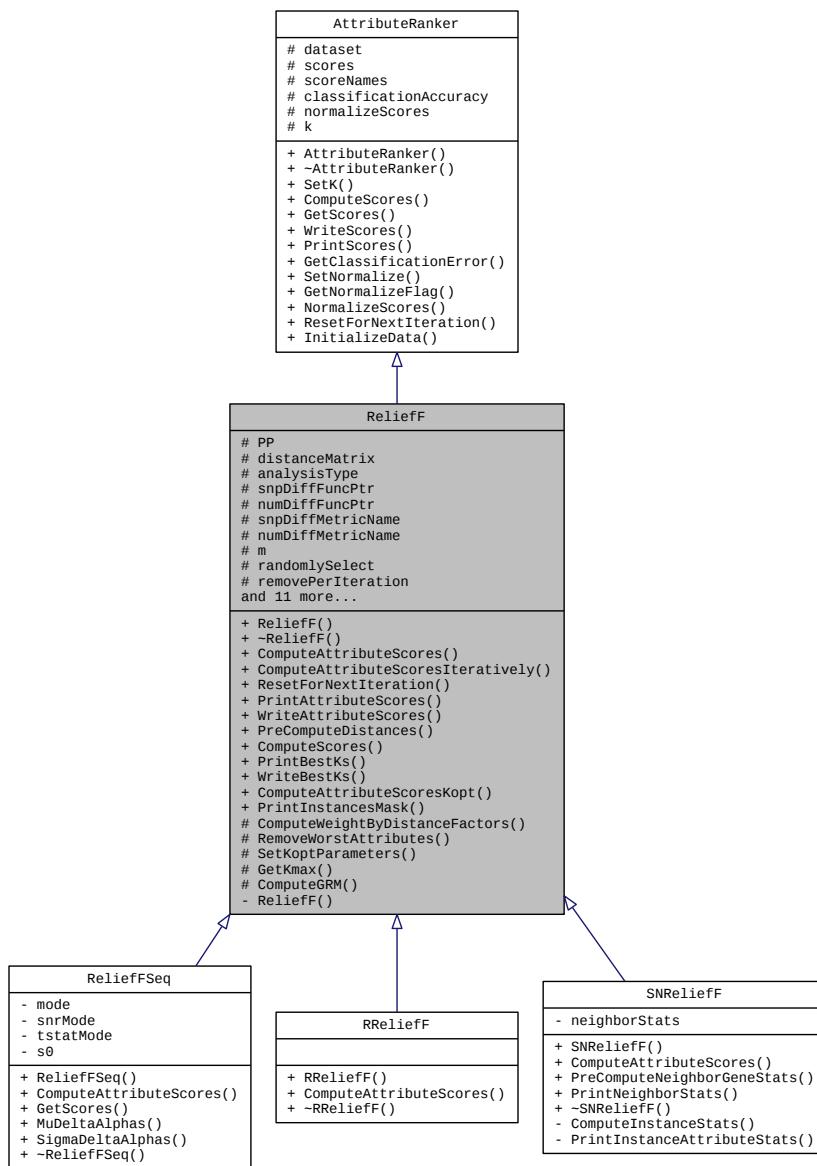
- src/[plink.h](#)

6.104 ReliefF Class Reference

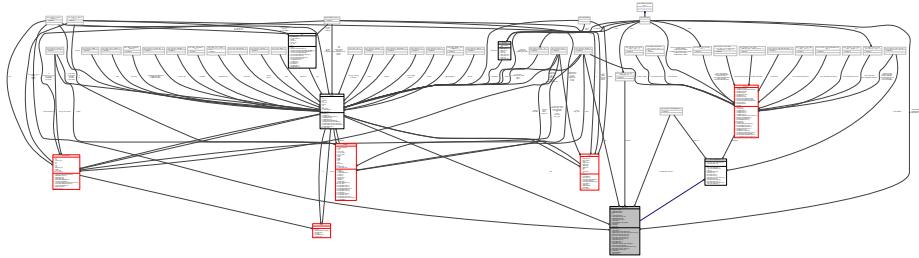
[ReliefF](#) attribute ranking algorithm.

```
#include <ReliefF.h>
```

Inheritance diagram for ReliefF:



Collaboration diagram for ReliefF:



Public Member Functions

- **ReliefF (Dataset *ds, Plink *plinkPtr, AnalysisType anaType)**
Construct an [ReliefF](#) algorithm object.
- **virtual ~ReliefF ()**
- **virtual bool ComputeAttributeScores ()**
Compute the [ReliefF](#) scores for the current set of attributes.
- **bool ComputeAttributeScoresIteratively ()**
Compute the [ReliefF](#) scores by iteratively removing worst attributes.
- **bool ResetForNextIteration () override**
Resets some data structures for the next iteration of [ReliefF](#).
- **void PrintAttributeScores (std::ofstream &outStream)**
Write the scores and attribute names to stream.
- **void WriteAttributeScores (std::string baseFilename)**
Write the scores and attribute names to file.
- **bool PreComputeDistances ()**
Pre-compute all pairwise instance-to-instance distances.
- **AttributeScores ComputeScores () override**
Implements [AttributeRanker](#) interface.
- **void PrintBestKs ()**
- **void WriteBestKs (std::string baseFilename)**
Write the best k-nearest neighbors best k and attribute names to file.
- **bool ComputeAttributeScoresKopt ()**
Compute scores based on optimum k.
- **void PrintInstancesMask ()**
Print the ID to instance index map to stdout.

Protected Member Functions

- **bool ComputeWeightByDistanceFactors ()**
Compute the const AttributeScores& [ComputeScores\(\)](#); weight by distance factors for nearest neighbors.
- **bool RemoveWorstAttributes (unsigned int numToRemove=1)**
Remove the worst attribute based on free energy scores.
- **bool SetKoptParameters ()**
- **unsigned int GetKmax ()**
Determine the maximum k value for optimization.
- **bool ComputeGRM ()**

Protected Attributes

- `Plink * PP`
- `matrix_t distanceMatrix`
- `AnalysisType analysisType`
`type of analysis to perform`
- `double(* snpDiffFuncPtr)(unsigned int attributeIndex, DatasetInstance *dsi1, DatasetInstance *dsi2)`
`Compute the discrete difference in an attribute between two instances.`
- `double(* numDiffFuncPtr)(unsigned int attributeIndex, DatasetInstance *dsi1, DatasetInstance *dsi2)`
`Compute the continuous difference in an attribute between two instances.`
- `std::string snpDiffMetricName`
`the name of discrete diff(erence) function`
- `std::string numDiffMetricName`
`the name of continuous diff(erence) function`
- `unsigned int m`
`number of instances to sample`
- `bool randomlySelect`
`are instances being randomly selected?`
- `unsigned int removePerIteration`
`number of attributes to remove each iteration if running iteratively`
- `bool doRemovePercent`
`are we removing a percentage per iteration?`
- `double removePercentage`
`percentage of attributes to remove per iteration if running iteratively`
- `unsigned int numTarget`
`number of target attributes`
- `std::string weightByDistanceMethod`
`name of the weight-by-distance method`
- `double weightByDistanceSigma`
`sigma value used in exponential decay weight-by-distance`
- `std::vector< double > W`
`RAW attribute scores/weights - ie, no normalization.`
- `AttributeScores removedAttributes`
`attributes that have been evaporated so far`
- `unsigned int koptBegin`
`optimize k begin value`
- `unsigned int koptEnd`
`optimize k end value`
- `unsigned int koptStep`
`optimize k step value`
- `std::map< std::string, unsigned int > bestKs`
`best k by attribute`

Private Member Functions

- `ReliefF()`
`no default constructor`

6.104.1 Detailed Description

[ReliefF](#) attribute ranking algorithm.

Totally redone for the McKinney in silico lab in 2011. Large refactoring to move all attribute elimination handling to the [Dataset](#) and its subclasses. 9/11/11

See also

[RReliefF](#)

Author

Bill White

Version

1.0

Contact: bill.c.white@gmail.com Created on: 7/16/05

Definition at line 31 of file ReliefF.h.

6.104.2 Constructor & Destructor Documentation

6.104.2.1 ReliefF() [1/2]

```
ReliefF::ReliefF (
    Dataset * ds,
    Plink * plinkPtr,
    AnalysisType anaType )
```

Construct an [ReliefF](#) algorithm object.

Parameters

in	<i>ds</i>	pointer to a Dataset object
in	<i>anaType</i>	analysis type

set the SNP metric function pointer based on command line params or defaults

Definition at line 64 of file ReliefF.cpp.

6.104.2.2 ~ReliefF()

```
ReliefF::~ReliefF () [virtual]
```

Definition at line 198 of file ReliefF.cpp.

6.104.2.3 ReliefF() [2/2]

```
ReliefF::ReliefF () [private]
```

no default constructor

6.104.3 Member Function Documentation

6.104.3.1 ComputeAttributeScores()

```
bool ReliefF::ComputeAttributeScores () [virtual]
```

Compute the [ReliefF](#) scores for the current set of attributes.

Implements [ReliefF](#) algorithm: Marko Robnik-Sikonja, Igor Kononenko: Theoretical and Empirical Analysis of [ReliefF](#) and [RReliefF](#). Machine Learning Journal, 53:23-69, 2003 <http://lkm.fri.uni-lj.si/rmarko/papers/robnik03-mlj.pdf> algorithm line 1

algorithm line 2

algorithm lines 4, 5 and 6

algorithm line 7

algorithm line 8

algorithm line 9

Reimplemented in [SNReliefF](#), [ReliefFSeq](#), and [RReliefF](#).

Definition at line 201 of file ReliefF.cpp.

6.104.3.2 ComputeAttributeScoresIteratively()

```
bool ReliefF::ComputeAttributeScoresIteratively ( )
```

Compute the [ReliefF](#) scores by iteratively removing worst attributes.

Definition at line 383 of file ReliefF.cpp.

6.104.3.3 ComputeAttributeScoresKopt()

```
bool ReliefF::ComputeAttributeScoresKopt ( )
```

Compute scores based on optimum k.

Definition at line 460 of file ReliefF.cpp.

6.104.3.4 ComputeGRM()

```
bool ReliefF::ComputeGRM ( ) [protected]
```

Definition at line 592 of file ReliefF.cpp.

6.104.3.5 ComputeScores()

```
AttributeScores ReliefF::ComputeScores ( ) [override], [virtual]
```

Implements [AttributeRanker](#) interface.

Implements [AttributeRanker](#).

Definition at line 799 of file ReliefF.cpp.

6.104.3.6 ComputeWeightByDistanceFactors()

```
bool ReliefF::ComputeWeightByDistanceFactors ( ) [protected]
```

Compute the const AttributeScores& [ComputeScores\(\)](#); weight by distance factors for nearest neighbors.

Definition at line 804 of file ReliefF.cpp.

6.104.3.7 GetKmax()

```
unsigned int ReliefF::GetKmax ( ) [protected]
```

Determine the maximum k value for optimization.

Definition at line 883 of file ReliefF.cpp.

6.104.3.8 PreComputeDistances()

```
bool ReliefF::PreComputeDistances ( )
```

Pre-compute all pairwise instance-to-instance distances.

be sure to call [Dataset::ComputeInstanceToInstanceDistance](#)

Definition at line 668 of file ReliefF.cpp.

6.104.3.9 PrintAttributeScores()

```
void ReliefF::PrintAttributeScores ( std::ofstream & outStream )
```

Write the scores and attribute names to stream.

Parameters

in	outStream	stream to write score-attribute name pairs
----	-----------	--

Definition at line 547 of file ReliefF.cpp.

6.104.3.10 PrintBestKs()

```
void ReliefF::PrintBestKs ( )
```

Definition at line 898 of file ReliefF.cpp.

6.104.3.11 PrintInstancesMask()

```
void ReliefF::PrintInstancesMask ( )
```

Print the ID to instance index map to stdout.

Definition at line 660 of file ReliefF.cpp.

6.104.3.12 RemoveWorstAttributes()

```
bool ReliefF::RemoveWorstAttributes (
    unsigned int numToRemove = 1 ) [protected]
```

Remove the worst attribute based on free energy scores.

Parameters

in	<i>numToRemove</i>	number of attributes to remove/evaporate
----	--------------------	--

Returns

distance

Definition at line 923 of file ReliefF.cpp.

6.104.3.13 ResetForNextIteration()

```
bool ReliefF::ResetForNextIteration ( ) [override], [virtual]
```

Resets some data structures for the next iteration of [ReliefF](#).

Reimplemented from [AttributeRanker](#).

Definition at line 541 of file ReliefF.cpp.

6.104.3.14 SetKoptParameters()

```
bool ReliefF::SetKoptParameters ( ) [protected]
```

Definition at line 847 of file ReliefF.cpp.

6.104.3.15 WriteAttributeScores()

```
void ReliefF::WriteAttributeScores (  
    std::string basefilename )
```

Write the scores and attribute names to file.

Parameters

in	<i>basefilename</i>	filename to write score-attribute name pairs
----	---------------------	--

Definition at line 575 of file ReliefF.cpp.

6.104.3.16 WriteBestKs()

```
void ReliefF::WriteBestKs (
    std::string basefilename )
```

Write the best k-nearest neighbors best k and attribute names to file.

Parameters

in	<i>basefilename</i>	filename to write best-k-attribute name pairs
----	---------------------	---

Definition at line 905 of file ReliefF.cpp.

6.104.4 Member Data Documentation**6.104.4.1 analysisType**

[AnalysisType](#) ReliefF::analysisType [protected]

type of analysis to perform

Definition at line 90 of file ReliefF.h.

6.104.4.2 bestKs

std::map<std::string, unsigned int> ReliefF::bestKs [protected]

best k by attribute

Definition at line 155 of file ReliefF.h.

6.104.4.3 distanceMatrix

```
matrix_t ReliefF::distanceMatrix [protected]
```

Definition at line 85 of file ReliefF.h.

6.104.4.4 doRemovePercent

```
bool ReliefF::doRemovePercent [protected]
```

are we removing a percentage per iteration?

Definition at line 122 of file ReliefF.h.

6.104.4.5 koptBegin

```
unsigned int ReliefF::koptBegin [protected]
```

optimize k begin value

Definition at line 149 of file ReliefF.h.

6.104.4.6 koptEnd

```
unsigned int ReliefF::koptEnd [protected]
```

optimize k end value

Definition at line 151 of file ReliefF.h.

6.104.4.7 koptStep

```
unsigned int ReliefF::koptStep [protected]
```

optimize k step value

Definition at line 153 of file ReliefF.h.

6.104.4.8 m

```
unsigned int ReliefF::m [protected]
```

number of instances to sample

Definition at line 116 of file ReliefF.h.

6.104.4.9 numDiffFuncPtr

```
double(* ReliefF::numDiffFuncPtr) (unsigned int attributeIndex, DatasetInstance *dsi1, DatasetInstance *dsi2) [protected]
```

Compute the continuous difference in an attribute between two instances.

Parameters

in	<i>attributeIndex</i>	index into vector of all attributes
in	<i>dsi1</i>	pointer to DatasetInstance 1
in	<i>dsi2</i>	pointer to DatasetInstance 2

Returns

[diff\(ference\)](#)

Definition at line 108 of file ReliefF.h.

6.104.4.10 numDiffMetricName

```
std::string ReliefF::numDiffMetricName [protected]
```

the name of continuous diff(ference) function

Definition at line 114 of file ReliefF.h.

6.104.4.11 numTarget

```
unsigned int ReliefF::numTarget [protected]
```

number of target attributes

Definition at line 126 of file ReliefF.h.

6.104.4.12 PP

`Plink*` `ReliefF::PP` [protected]

Definition at line 83 of file ReliefF.h.

6.104.4.13 randomlySelect

`bool` `ReliefF::randomlySelect` [protected]

are instances being randomly selected?

Definition at line 118 of file ReliefF.h.

6.104.4.14 removedAttributes

`AttributeScores` `ReliefF::removedAttributes` [protected]

attributes that have been evaporated so far

Definition at line 147 of file ReliefF.h.

6.104.4.15 removePercentage

`double` `ReliefF::removePercentage` [protected]

percentage of attributes to remove per iteration if running iteratively

Definition at line 124 of file ReliefF.h.

6.104.4.16 removePerIteration

`unsigned int` `ReliefF::removePerIteration` [protected]

number of attributes to remove each iteration if running iteratively

Definition at line 120 of file ReliefF.h.

6.104.4.17 snpDiffFuncPtr

`double(*)` `ReliefF::snpDiffFuncPtr` (`unsigned int` `attributeIndex`, `DatasetInstance` *`dsil`, `DatasetInstance` *`dsi2`) [protected]

Compute the discrete difference in an attribute between two instances.

Parameters

in	<i>attributeIndex</i>	index into vector of all attributes
in	<i>dsi1</i>	pointer to DatasetInstance 1
in	<i>dsi2</i>	pointer to DatasetInstance 2

Returns

`diff(erence)`

Definition at line 98 of file ReliefF.h.

6.104.4.18 snpDiffMetricName

`std::string ReliefF::snpDiffMetricName [protected]`

the name of discrete `diff(erence)` function

Definition at line 112 of file ReliefF.h.

6.104.4.19 W

`std::vector<double> ReliefF::W [protected]`

RAW attribute scores/weights - ie, no normalization.

Definition at line 132 of file ReliefF.h.

6.104.4.20 weightByDistanceMethod

`std::string ReliefF::weightByDistanceMethod [protected]`

name of the weight-by-distance method

Definition at line 128 of file ReliefF.h.

6.104.4.21 weightByDistanceSigma

```
double ReliefF::weightByDistanceSigma [protected]
```

sigma value used in exponential decay weight-by-distance

Definition at line 130 of file ReliefF.h.

The documentation for this class was generated from the following files:

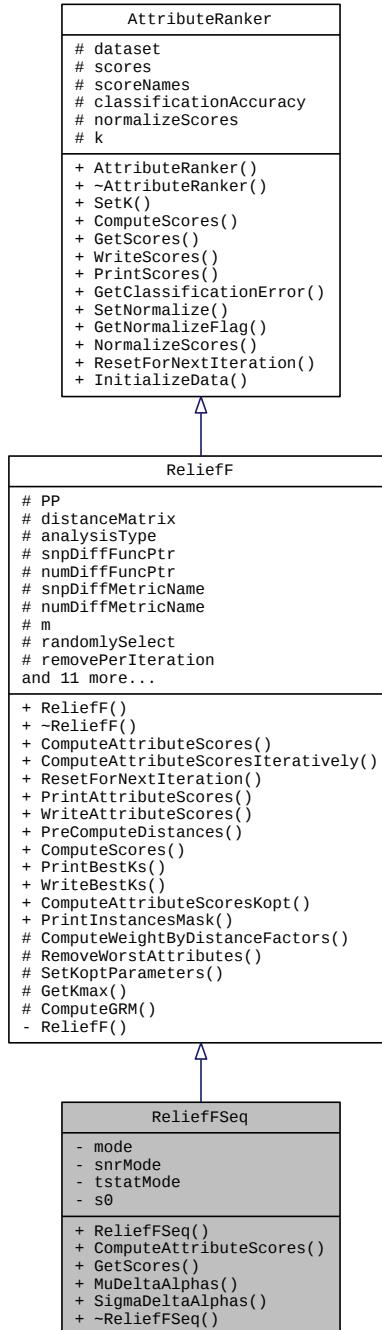
- src/[ReliefF.h](#)
- src/[ReliefF.cpp](#)

6.105 ReliefFSeq Class Reference

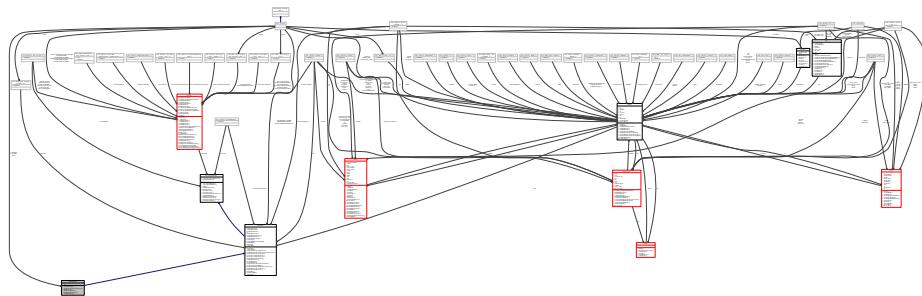
[ReliefFSeq](#) attribute ranking algorithm.

```
#include <ReliefFSeq.h>
```

Inheritance diagram for ReliefFSeq:



Collaboration diagram for ReliefFSeq:



Public Member Functions

- **ReliefFSeq (Dataset *ds, Plink *plinkPtr)**
Construct an [ReliefFSeq](#) algorithm object.
- **bool ComputeAttributeScores () override**
Compute the [ReliefF](#) scores for the current set of attributes.
- **AttributeScores GetScores () override**
Get the (importance) scores as a vector of pairs: score, attribute name.
- **std::pair< double, double > MuDeltaAlphas (unsigned int alpha)**
std::pair< double, double > SigmaDeltaAlphas (unsigned int alpha, double muDeltaHit, double muDeltaMiss)
standard deviations of hit and miss diffs for gene alpha
- **virtual ~ReliefFSeq ()**

Private Attributes

- **std::string mode**
ReliefSeq mode: signal-to-noise ratio (snr) or t-statistic (tstat)
- **std::string snrMode**
ReliefSeq signal-to-noise ratio mode: signal-to-noise ratio (snr) or [ReliefF](#) (relieff)
- **std::string tstatMode**
[ReliefFSeq](#) t-statistic mode: 1-pvalue (pval) or the absolute value of the t-statistic (abst)
- **double s0**
variance denominator adjustment s0

Additional Inherited Members

6.105.1 Detailed Description

[ReliefFSeq](#) attribute ranking algorithm.

Designed to handle digital gene expression (DGE) data sets, particularly RNA-Seq high-throughput count data, by accounting for variable-specific variance in counts. [Data](#) is known to follow a Poisson or negative binomial distribution. This algorithm is a more computationally practical approach than others that use more sophisticated statistical methods and models. Our approach keeps the [ReliefF](#) algorithm general while addressing the variance "dispersion" problem as a special case.

See also[ReliefF](#)**Author**

Bill White

Version

1.0

Contact: bill.c.white@gmail.com Created on: 7/23/12

Definition at line 38 of file ReliefFSeq.h.

6.105.2 Constructor & Destructor Documentation**6.105.2.1 ReliefFSeq()**

```
ReliefFSeq::ReliefFSeq (
    Dataset * ds,
    Plink * plinkPtr )
```

Construct an [ReliefFSeq](#) algorithm object.**Parameters**

in	ds	pointer to a Dataset object
----	----	---

set the various algorithm modes to one of four combinations: snr-snr, snr-relieff, tstat-pval, tstat-abst

set the s0 value

Definition at line 30 of file ReliefFSeq.cpp.

6.105.2.2 ~ReliefFSeq()

```
ReliefFSeq::~ReliefFSeq ( ) [virtual]
```

Definition at line 72 of file ReliefFSeq.cpp.

6.105.3 Member Function Documentation

6.105.3.1 ComputeAttributeScores()

```
bool ReliefFSeq::ComputeAttributeScores ( ) [override], [virtual]
```

Compute the [ReliefF](#) scores for the current set of attributes.

Implements [ReliefF](#) algorithm: Marko Robnik-Sikonja, Igor Kononenko: Theoretical and Empirical Analysis of [ReliefF](#) and [RReliefF](#). Machine Learning Journal, 53:23-69, 2003 <http://lkm.fri.uni-lj.si/rmarko/papers/robnik03-mlj.pdf> run this loop on as many cores as possible through OpenMP

assign a weight to this variable index

algorithm line 1

algorithm line 2

algorithm lines 4, 5 and 6

algorithm line 7

algorithm line 8

algorithm line 9

Reimplemented from [ReliefF](#).

Definition at line 75 of file ReliefFSeq.cpp.

6.105.3.2 GetScores()

```
AttributeScores ReliefFSeq::GetScores ( ) [override], [virtual]
```

Get the (importance) scores as a vector of pairs: score, attribute name.

Returns

vector of pairs

Reimplemented from [AttributeRanker](#).

Definition at line 209 of file ReliefFSeq.cpp.

6.105.3.3 MuDeltaAlphas()

```
pair< double, double > ReliefFSeq::MuDeltaAlphas (  
    unsigned int alpha )
```

Definition at line 219 of file ReliefFSeq.cpp.

6.105.3.4 SigmaDeltaAlphas()

```
pair< double, double > ReliefFSeq::SigmaDeltaAlphas (  
    unsigned int alpha,  
    double muDeltaHit,  
    double muDeltaMiss )
```

standard deviations of hit and miss diffs for gene alpha

get hits and misses for this instance

Definition at line 250 of file ReliefFSeq.cpp.

6.105.4 Member Data Documentation

6.105.4.1 mode

```
std::string ReliefFSeq::mode [private]
```

ReliefSeq mode: signal-to-noise ratio (snr) or t-statistic (tstat)

Definition at line 56 of file ReliefFSeq.h.

6.105.4.2 s0

```
double ReliefFSeq::s0 [private]
```

variance denominator adjustment s0

Definition at line 64 of file ReliefFSeq.h.

6.105.4.3 snrMode

```
std::string ReliefFSeq::snrMode [private]
```

ReliefSeq signal-to-noise ratio mode: signal-to-noise ratio (snr) or [ReliefF](#) (relieff)

Definition at line 59 of file ReliefFSeq.h.

6.105.4.4 tstatMode

```
std::string ReliefFSeq::tstatMode [private]
```

[ReliefFSeq](#) t-statistic mode: 1-pvalue (pval) or the absolute value of the t-statistic (abst)

Definition at line 62 of file ReliefFSeq.h.

The documentation for this class was generated from the following files:

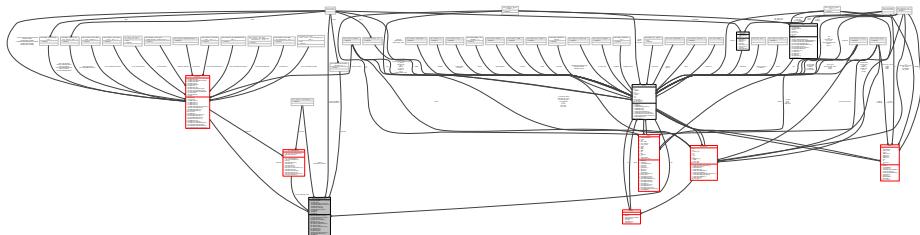
- src/[ReliefFSeq.h](#)
- src/[ReliefFSeq.cpp](#)

6.106 ReliefSeqController Class Reference

Interface for running ReliefSeq algorithms.

```
#include <ReliefSeqController.h>
```

Collaboration diagram for ReliefSeqController:



Public Member Functions

- **ReliefSeqController** (*Dataset* *ds, *Plink* *plinkPtr, *AnalysisType* anaType=**SNP_ONLY_ANALYSIS**)
Construct an [ReliefSeqController](#) object.
- **virtual ~ReliefSeqController ()**
- **bool ComputeScores ()**
Compute the scores based on the current set of attributes.
- **bool ComputeScoresKopt ()**
Compute scores based on optimum k.
- **AttributeScores & GetScores ()**
Get the last computed scores.
- **std::string GetAlgorithmMode ()**
Return the algorithm mode.
- **void WriteAttributeScores (std::string baseFilename)**
Write the scores and attribute names to file.
- **void WriteBestKs (std::string baseFilename)**
Write the best k-nearest neighbors best k and attribute names to file.
- **void PrintAttributeScores (std::ofstream &outStream)**
Write the scores and attribute names to stream.
- **void PrintScores ()**
- **void PrintBestKs ()**

Private Member Functions

- **bool RunReliefF ()**
Run the [ReliefF](#) algorithm.
- **bool RemoveWorstAttributes (unsigned int numToRemove=1)**
Remove the worst attribute based on free energy scores.
- **bool SetKoptParameters ()**
Set the k-optimization parameters from the command line.
- **unsigned int GetKmax ()**
Determine the maximum k value for optimization.

Private Attributes

- **Plink *** *PP*
- **Dataset *** *dataset*
pointer to a [Dataset](#) object
- **std::string** *outFilesPrefix*
prefix for all output files
- **AnalysisType** *analysisType*
type of analysis to perform
- **std::string** *algorithmMode*
algorithm steps to perform
- **AttributeRanker *** *reliefseqAlgorithm*
pointer to an interaction ranker algorithm object

- `unsigned int numThreads`
`number of attributes to remove per iteration`
- `unsigned int numToRemoveNextIteration`
`number of attributes to remove next iteration`
- `bool doRemovePercent`
`number of target attributes`
- `double removePercentage`
- `unsigned int numTargetAttributes`
- `AttributeScores removedAttributes`
`attributes that have been evaporated so far`
- `AttributeScores scores`
`current set of scores`
- `unsigned int koptBegin`
`optimize k begin value`
- `unsigned int koptEnd`
`optimize k end value`
- `unsigned int koptStep`
`optimize k step value`
- `std::map< std::string, unsigned int > bestKs`
`best k by attribute`

6.106.1 Detailed Description

Interface for running ReliefSeq algorithms.

Definition at line 18 of file ReliefSeqController.h.

6.106.2 Constructor & Destructor Documentation

6.106.2.1 ReliefSeqController()

```
ReliefSeqController::ReliefSeqController (
    Dataset * ds,
    Plink * plinkPtr,
    AnalysisType anaType = SNP_ONLY_ANALYSIS )
```

Construct an `ReliefSeqController` object.

Parameters

in	<code>ds</code>	pointer to a <code>Dataset</code> object
in	<code>vm</code>	reference to a Boost map of command line options
in	<code>anaType</code>	analysis type

set the relief algorithm

Definition at line 33 of file ReliefSeqController.cpp.

6.106.2.2 ~ReliefSeqController()

```
ReliefSeqController::~ReliefSeqController () [virtual]
```

Definition at line 108 of file ReliefSeqController.cpp.

6.106.3 Member Function Documentation

6.106.3.1 ComputeScores()

```
bool ReliefSeqController::ComputeScores ()
```

Compute the scores based on the current set of attributes.

Definition at line 114 of file ReliefSeqController.cpp.

6.106.3.2 ComputeScoresKopt()

```
bool ReliefSeqController::ComputeScoresKopt ()
```

Compute scores based on optimum k.

Definition at line 263 of file ReliefSeqController.cpp.

6.106.3.3 GetAlgorithmMode()

```
string ReliefSeqController::GetAlgorithmMode ()
```

Return the algorithm mode.

Definition at line 346 of file ReliefSeqController.cpp.

6.106.3.4 GetKmax()

```
unsigned int ReliefSeqController::GetKmax ( ) [private]
```

Determine the maximum k value for optimization.

Definition at line 248 of file ReliefSeqController.cpp.

6.106.3.5 GetScores()

```
AttributeScores & ReliefSeqController::GetScores ( )
```

Get the last computed scores.

Definition at line 342 of file ReliefSeqController.cpp.

6.106.3.6 PrintAttributeScores()

```
void ReliefSeqController::PrintAttributeScores ( std::ofstream & outStream )
```

Write the scores and attribute names to stream.

Parameters

in	<i>outStream</i>	stream to write score-attribute name pairs
----	------------------	--

Definition at line 350 of file ReliefSeqController.cpp.

6.106.3.7 PrintBestKs()

```
void ReliefSeqController::PrintBestKs ( )
```

Definition at line 364 of file ReliefSeqController.cpp.

6.106.3.8 PrintScores()

```
void ReliefSeqController::PrintScores ( )
```

Definition at line 357 of file ReliefSeqController.cpp.

6.106.3.9 RemoveWorstAttributes()

```
bool ReliefSeqController::RemoveWorstAttributes (
    unsigned int numToRemove = 1 ) [private]
```

Remove the worst attribute based on free energy scores.

Parameters

in	<i>numToRemove</i>	number of attributes to remove/evaporate
----	--------------------	--

Returns

distance

Definition at line 462 of file ReliefSeqController.cpp.

6.106.3.10 RunReliefF()

```
bool ReliefSeqController::RunReliefF ( ) [private]
```

Run the [ReliefF](#) algorithm.

Definition at line 407 of file ReliefSeqController.cpp.

6.106.3.11 SetKoptParameters()

```
bool ReliefSeqController::SetKoptParameters ( ) [private]
```

Set the k-optimization parameters from the command line.

Definition at line 212 of file ReliefSeqController.cpp.

6.106.3.12 WriteAttributeScores()

```
void ReliefSeqController::WriteAttributeScores (
    std::string basename )
```

Write the scores and attribute names to file.

Parameters

in	<i>baseFilename</i>	filename to write score-attribute name pairs
----	---------------------	--

Definition at line 371 of file ReliefSeqController.cpp.

6.106.3.13 WriteBestKs()

```
void ReliefSeqController::WriteBestKs (
    std::string baseFilename )
```

Write the best k-nearest neighbors best k and attribute names to file.

Parameters

in	<i>baseFilename</i>	filename to write best-k-attribute name pairs
----	---------------------	---

Definition at line 387 of file ReliefSeqController.cpp.

6.106.4 Member Data Documentation**6.106.4.1 algorithmMode**

`std::string ReliefSeqController::algorithmMode [private]`

algorithm steps to perform

Definition at line 80 of file ReliefSeqController.h.

6.106.4.2 analysisType

`AnalysisType ReliefSeqController::analysisType [private]`

type of analysis to perform

See also

[ReliefF](#)

Definition at line 78 of file ReliefSeqController.h.

6.106.4.3 bestKs

```
std::map<std::string, unsigned int> ReliefSeqController::bestKs [private]
```

best k by attribute

Definition at line 108 of file ReliefSeqController.h.

6.106.4.4 dataset

```
Dataset* ReliefSeqController::dataset [private]
```

pointer to a [Dataset](#) object

Definition at line 72 of file ReliefSeqController.h.

6.106.4.5 doRemovePercent

```
bool ReliefSeqController::doRemovePercent [private]
```

number of target attributes

Definition at line 93 of file ReliefSeqController.h.

6.106.4.6 koptBegin

```
unsigned int ReliefSeqController::koptBegin [private]
```

optimize k begin value

Definition at line 102 of file ReliefSeqController.h.

6.106.4.7 koptEnd

```
unsigned int ReliefSeqController::koptEnd [private]
```

optimize k end value

Definition at line 104 of file ReliefSeqController.h.

6.106.4.8 koptStep

```
unsigned int ReliefSeqController::koptStep [private]
```

optimize k step value

Definition at line 106 of file ReliefSeqController.h.

6.106.4.9 numTargetAttributes

```
unsigned int ReliefSeqController::numTargetAttributes [private]
```

Definition at line 95 of file ReliefSeqController.h.

6.106.4.10 numThreads

```
unsigned int ReliefSeqController::numThreads [private]
```

Definition at line 86 of file ReliefSeqController.h.

6.106.4.11 numToRemoveNextIteration

```
unsigned int ReliefSeqController::numToRemoveNextIteration [private]
```

number of attributes to remove next iteration

Definition at line 90 of file ReliefSeqController.h.

6.106.4.12 numToRemovePerIteration

```
unsigned int ReliefSeqController::numToRemovePerIteration [private]
```

number of attributes to remove per iteration

Definition at line 88 of file ReliefSeqController.h.

6.106.4.13 outFilesPrefix

```
std::string ReliefSeqController::outFilesPrefix [private]
```

prefix for all output files

Definition at line 74 of file ReliefSeqController.h.

6.106.4.14 PP

```
Plink* ReliefSeqController::PP [private]
```

Definition at line 69 of file ReliefSeqController.h.

6.106.4.15 reliefseqAlgorithm

```
AttributeRanker* ReliefSeqController::reliefseqAlgorithm [private]
```

pointer to an interaction ranker algorithm object

Definition at line 83 of file ReliefSeqController.h.

6.106.4.16 removedAttributes

```
AttributeScores ReliefSeqController::removedAttributes [private]
```

attributes that have been evaporated so far

Definition at line 97 of file ReliefSeqController.h.

6.106.4.17 removePercentage

```
double ReliefSeqController::removePercentage [private]
```

Definition at line 94 of file ReliefSeqController.h.

6.106.4.18 scores

```
AttributeScores ReliefSeqController::scores [private]
```

current set of scores

Definition at line 99 of file ReliefSeqController.h.

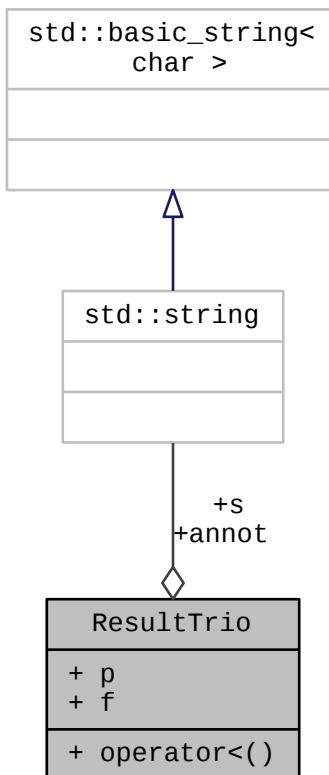
The documentation for this class was generated from the following files:

- src/[ReliefSeqController.h](#)
- src/[ReliefSeqController.cpp](#)

6.107 ResultTrio Class Reference

```
#include <clumpld.h>
```

Collaboration diagram for ResultTrio:



Public Member Functions

- bool `operator<` (const `ResultTrio` &`p2`) const

Public Attributes

- double `p`
- string `annot`
- int `f`
- string `s`

6.107.1 Detailed Description

Definition at line 26 of file `clumpld.h`.

6.107.2 Member Function Documentation

6.107.2.1 `operator<()`

```
bool ResultTrio::operator< (
    const ResultTrio & p2 ) const [inline]
```

Definition at line 35 of file `clumpld.h`.

6.107.3 Member Data Documentation

6.107.3.1 `annot`

```
string ResultTrio::annot
```

Definition at line 31 of file `clumpld.h`.

6.107.3.2 f

```
int ResultTrio::f
```

Definition at line 32 of file clumpId.h.

6.107.3.3 p

```
double ResultTrio::p
```

Definition at line 30 of file clumpld.h.

6.107.3.4 s

string ResultTrio::s

Definition at line 33 of file clumpld.h.

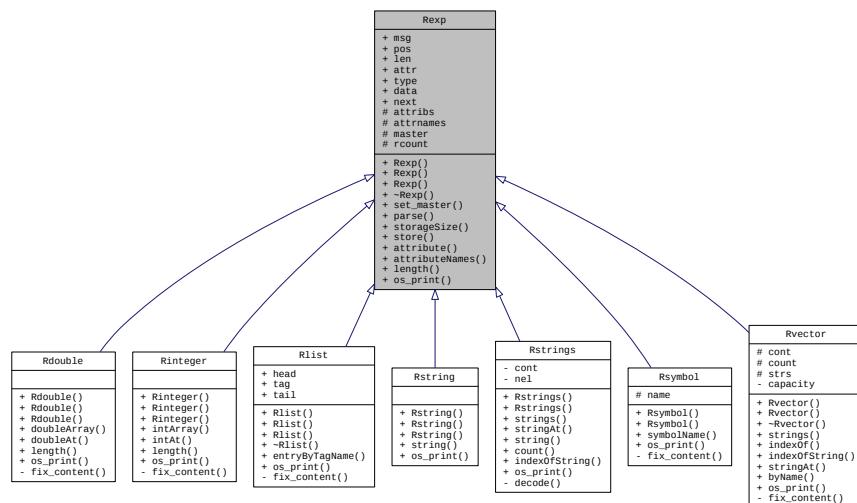
The documentation for this class was generated from the following file:

- `src/clumpld.h`

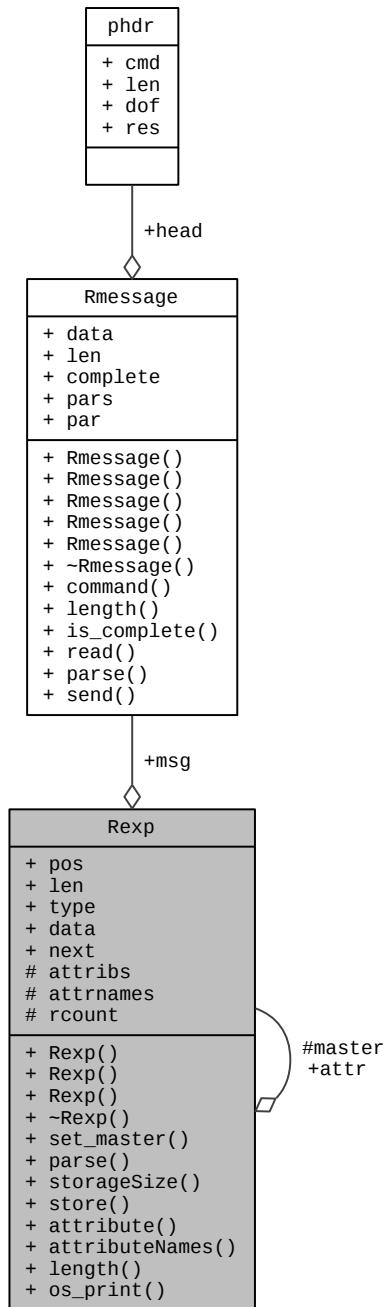
6.108 Rexp Class Reference

```
#include <Rconnection.h>
```

Inheritance diagram for Rexp:



Collaboration diagram for Rexp:



Public Member Functions

- [Rexp \(Rmessage *msg\)](#)

- [Rexp \(unsigned int *pos, Rmessage *msg=0\)](#)
- [Rexp \(int type, const char *data=0, int len=0, Rexp *attr=0\)](#)
- virtual [~Rexp \(\)](#)
- void [set_master \(Rexp *m\)](#)
- char * [parse \(unsigned int *pos\)](#)
- virtual [Rsize_t storageSize \(\)](#)
- virtual void [store \(char *buf\)](#)
- [Rexp * attribute \(const char *name\)](#)
- char ** [attributeNames \(\)](#)
- virtual [Rsize_t length \(\)](#)
- virtual std::ostream & [os_print \(std::ostream &os\)](#)

Public Attributes

- [Rmessage * msg](#)
- unsigned int * [pos](#)
- [Rsize_t len](#)
- [Rexp * attr](#)
- int [type](#)
- char * [data](#)
- char * [next](#)

Protected Attributes

- int [attribs](#)
- char ** [attrnames](#)
- [Rexp * master](#)
- int [rcount](#)

Friends

- std::ostream & [operator<< \(std::ostream &os, const Rexp &exp\)](#)
- std::ostream & [operator<< \(std::ostream &os, const Rexp *exp\)](#)

6.108.1 Detailed Description

Definition at line 100 of file Rconnection.h.

6.108.2 Constructor & Destructor Documentation

6.108.2.1 Rexp() [1/3]

```
Rexp::Rexp (
    Rmessage * msg )
```

Definition at line 234 of file Rconnection.cpp.

6.108.2.2 Rexp() [2/3]

```
Rexp::Rexp (
    unsigned int * pos,
    Rmessage * msg = 0 )
```

Definition at line 250 of file Rconnection.cpp.

6.108.2.3 Rexp() [3/3]

```
Rexp::Rexp (
    int type,
    const char * data = 0,
    int len = 0,
    Rexp * attr = 0 )
```

Definition at line 258 of file Rconnection.cpp.

6.108.2.4 ~Rexp()

```
Rexp::~Rexp ( ) [virtual]
```

Definition at line 271 of file Rconnection.cpp.

6.108.3 Member Function Documentation

6.108.3.1 attribute()

```
Rexp * Rexp::attribute (
    const char * name )
```

Definition at line 337 of file Rconnection.cpp.

6.108.3.2 attributeNames()

```
char ** Rexp::attributeNames ( )
```

Definition at line 341 of file Rconnection.cpp.

6.108.3.3 length()

```
virtual Rsize_t Rexp::length ( ) [inline], [virtual]
```

Reimplemented in [Rdouble](#), and [Rinteger](#).

Definition at line 140 of file Rconnection.h.

6.108.3.4 os_print()

```
virtual std::ostream& Rexp::os_print (
    std::ostream & os ) [inline], [virtual]
```

Reimplemented in [Rvector](#), [Rlist](#), [Rstring](#), [Rstrings](#), [Rsymbol](#), [Rdouble](#), and [Rinteger](#).

Definition at line 150 of file Rconnection.h.

6.108.3.5 parse()

```
char * Rexp::parse (
    unsigned int * pos )
```

Definition at line 300 of file Rconnection.cpp.

6.108.3.6 set_master()

```
void Rexp::set_master (
    Rexp * m )
```

Definition at line 294 of file Rconnection.cpp.

6.108.3.7 storageSize()

```
virtual Rsize_t Rexp::storageSize ( ) [inline], [virtual]
```

Definition at line 134 of file Rconnection.h.

6.108.3.8 store()

```
void Rexp::store (
    char * buf ) [virtual]
```

Definition at line 324 of file Rconnection.cpp.

6.108.4 Friends And Related Function Documentation

6.108.4.1 operator<< [1/2]

```
std::ostream& operator<< (
    std::ostream & os,
    const Rexp & exp ) [friend]
```

Definition at line 142 of file Rconnection.h.

6.108.4.2 operator<< [2/2]

```
std::ostream& operator<< (
    std::ostream & os,
    const Rexp * exp ) [friend]
```

Definition at line 146 of file Rconnection.h.

6.108.5 Member Data Documentation

6.108.5.1 attr

```
Rexp* Rexp::attr
```

Definition at line 105 of file Rconnection.h.

6.108.5.2 attribs

```
int Rexp::attribs [protected]
```

Definition at line 118 of file Rconnection.h.

6.108.5.3 attrnames

```
char** Rexp::attrnames [protected]
```

Definition at line 119 of file Rconnection.h.

6.108.5.4 data

```
char* Rexp::data
```

Definition at line 114 of file Rconnection.h.

6.108.5.5 len

```
Rsize_t Rexp::len
```

Definition at line 104 of file Rconnection.h.

6.108.5.6 master

`Rexp* Rexp::master [protected]`

Definition at line 121 of file Rconnection.h.

6.108.5.7 msg

`Rmessage* Rexp::msg`

Definition at line 102 of file Rconnection.h.

6.108.5.8 next

`char * Rexp::next`

Definition at line 114 of file Rconnection.h.

6.108.5.9 pos

`unsigned int* Rexp::pos`

Definition at line 103 of file Rconnection.h.

6.108.5.10 rcount

`int Rexp::rcount [protected]`

Definition at line 122 of file Rconnection.h.

6.108.5.11 type

```
int Rexp::type
```

Definition at line 106 of file Rconnection.h.

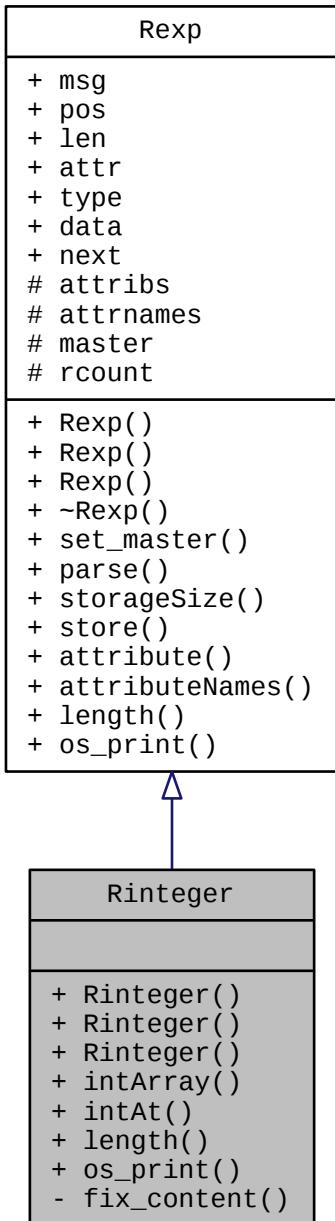
The documentation for this class was generated from the following files:

- src/[Rconnection.h](#)
- src/[Rconnection.cpp](#)

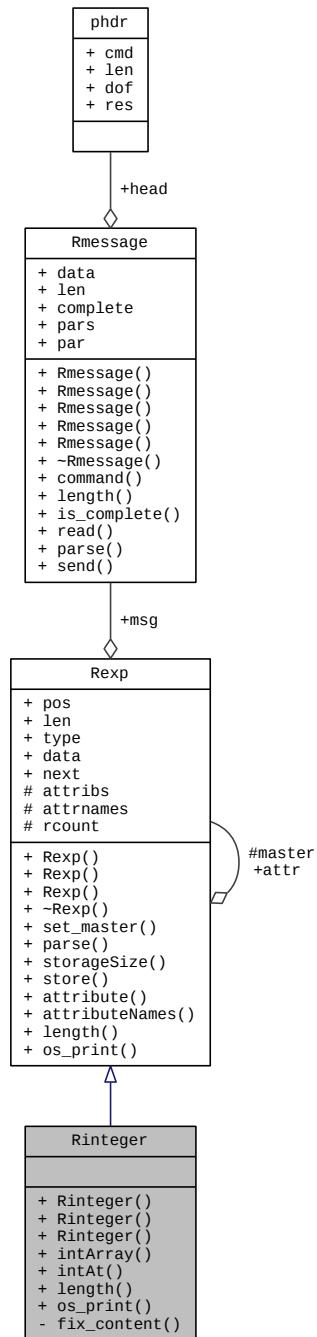
6.109 Rinteger Class Reference

```
#include <Rconnection.h>
```

Inheritance diagram for Rinteger:



Collaboration diagram for Rinteger:



Public Member Functions

- [Rinteger \(Rmessage *msg\)](#)

- `Rinteger` (`unsigned int *ipos, Rmessage *imsg`)
- `Rinteger` (`int *array, int count`)
- `int * intArray ()`
- `int intAt (int pos)`
- virtual `Rsize_t length ()`
- virtual `std::ostream & os_print (std::ostream &os)`

Private Member Functions

- void `fix_content ()`

Additional Inherited Members

6.109.1 Detailed Description

Definition at line 157 of file Rconnection.h.

6.109.2 Constructor & Destructor Documentation

6.109.2.1 `Rinteger()` [1/3]

```
Rinteger::Rinteger (
    Rmessage * msg )  [inline]
```

Definition at line 159 of file Rconnection.h.

6.109.2.2 `Rinteger()` [2/3]

```
Rinteger::Rinteger (
    unsigned int * ipos,
    Rmessage * imsg )  [inline]
```

Definition at line 160 of file Rconnection.h.

6.109.2.3 Rinteger() [3/3]

```
Rinteger::Rinteger (
    int * array,
    int count ) [inline]
```

Definition at line 161 of file Rconnection.h.

6.109.3 Member Function Documentation

6.109.3.1 fix_content()

```
void Rinteger::fix_content () [private]
```

Definition at line 363 of file Rconnection.cpp.

6.109.3.2 intArray()

```
int* Rinteger::intArray () [inline]
```

Definition at line 163 of file Rconnection.h.

6.109.3.3 intAt()

```
int Rinteger::intAt (
    int pos ) [inline]
```

Definition at line 164 of file Rconnection.h.

6.109.3.4 length()

```
virtual Rsize_t Rinteger::length () [inline], [virtual]
```

Reimplemented from [Rexp](#).

Definition at line 165 of file Rconnection.h.

6.109.3.5 os_print()

```
virtual std::ostream& Rinteger::os_print (
    std::ostream & os ) [inline], [virtual]
```

Reimplemented from [Rexp](#).

Definition at line 167 of file [Rconnection.h](#).

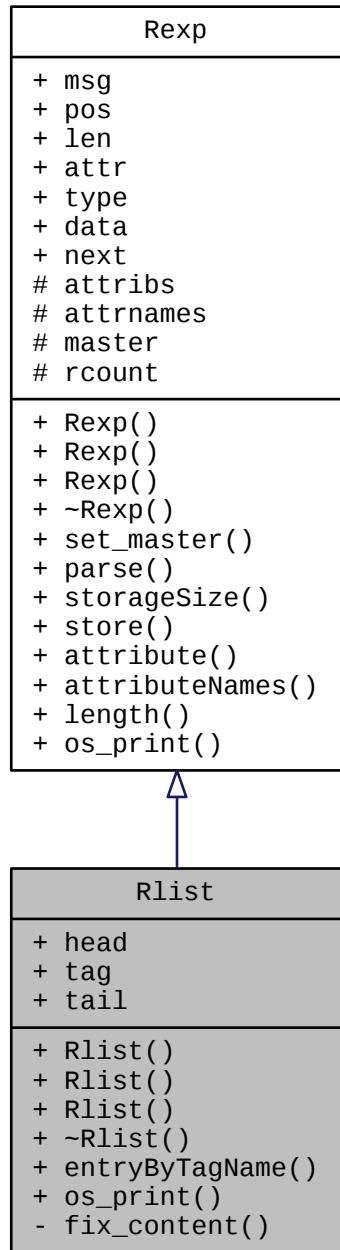
The documentation for this class was generated from the following files:

- src/[Rconnection.h](#)
- src/[Rconnection.cpp](#)

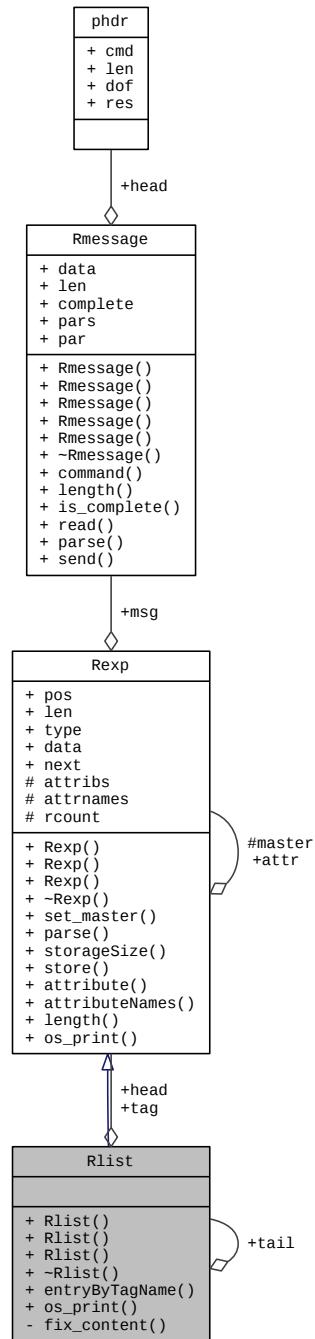
6.110 Rlist Class Reference

```
#include <Rconnection.h>
```

Inheritance diagram for Rlist:



Collaboration diagram for Rlist:



Public Member Functions

- [Rlist \(Rmessage *msg\)](#)

- `Rlist (unsigned int *ipos, Rmessage *imsg)`
- `Rlist (int type, Rexp *head, Rexp *tag, char *next, Rmessage *imsg)`
- `virtual ~Rlist ()`
- `Rexp * entryByTagName (const char *tagName)`
- `virtual std::ostream & os_print (std::ostream &os)`

Public Attributes

- `Rexp * head`
- `Rexp * tag`
- `Rlist * tail`

Private Member Functions

- `void fix_content ()`

Additional Inherited Members

6.110.1 Detailed Description

Definition at line 278 of file Rconnection.h.

6.110.2 Constructor & Destructor Documentation

6.110.2.1 Rlist() [1/3]

```
Rlist::Rlist (
    Rmessage * msg ) [inline]
```

Definition at line 283 of file Rconnection.h.

6.110.2.2 Rlist() [2/3]

```
Rlist::Rlist (
    unsigned int * ipos,
    Rmessage * imsg ) [inline]
```

Definition at line 286 of file Rconnection.h.

6.110.2.3 Rlist() [3/3]

```
Rlist::Rlist (
    int type,
    Rexp * head,
    Rexp * tag,
    char * next,
    Rmessage * imsg ) [inline]
```

Definition at line 293 of file Rconnection.h.

6.110.2.4 ~Rlist()

```
Rlist::~Rlist ( ) [virtual]
```

Definition at line 389 of file Rconnection.cpp.

6.110.3 Member Function Documentation**6.110.3.1 entryByTagName()**

```
Rexp* Rlist::entryByTagName (
    const char * tagName ) [inline]
```

Definition at line 297 of file Rconnection.h.

6.110.3.2 fix_content()

```
void Rlist::fix_content ( ) [private]
```

Definition at line 395 of file Rconnection.cpp.

6.110.3.3 os_print()

```
virtual std::ostream& Rlist::os_print (
    std::ostream & os ) [inline], [virtual]
```

Reimplemented from [Rexp](#).

Definition at line 303 of file Rconnection.h.

6.110.4 Member Data Documentation

6.110.4.1 head

```
Rexp* Rlist::head
```

Definition at line 280 of file Rconnection.h.

6.110.4.2 tag

```
Rexp * Rlist::tag
```

Definition at line 280 of file Rconnection.h.

6.110.4.3 tail

```
Rlist* Rlist::tail
```

Definition at line 281 of file Rconnection.h.

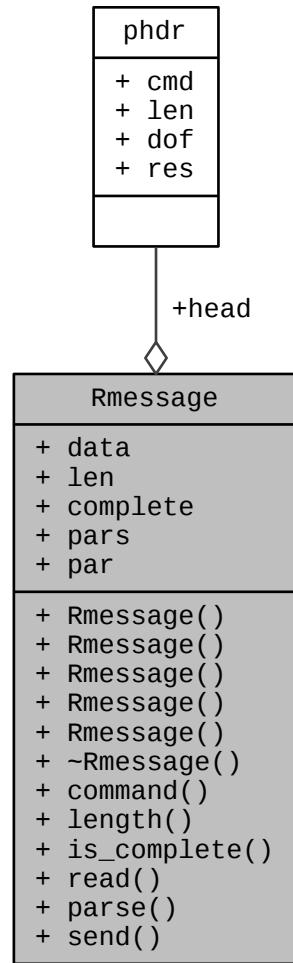
The documentation for this class was generated from the following files:

- src/Rconnection.h
- src/Rconnection.cpp

6.111 Rmessage Class Reference

```
#include <Rconnection.h>
```

Collaboration diagram for Rmessage:



Public Member Functions

- [Rmessage \(\)](#)
- [Rmessage \(int cmd\)](#)
- [Rmessage \(int cmd, const char *txt\)](#)
- [Rmessage \(int cmd, int i\)](#)
- [Rmessage \(int cmd, const void *buf, int len, int raw_data=0\)](#)

- virtual ~Rmessage ()
- int command ()
- Rsize_t length ()
- int is_complete ()
- int read (int s)
- void parse ()
- int send (int s)

Public Attributes

- struct phdr head
- char * data
- Rsize_t len
- int complete
- int pars
- unsigned int * par [16]

6.111.1 Detailed Description

Definition at line 71 of file Rconnection.h.

6.111.2 Constructor & Destructor Documentation

6.111.2.1 Rmessage() [1/5]

```
Rmessage::Rmessage ( )
```

Definition at line 91 of file Rconnection.cpp.

6.111.2.2 Rmessage() [2/5]

```
Rmessage::Rmessage (
    int cmd )
```

Definition at line 97 of file Rconnection.cpp.

6.111.2.3 Rmessage() [3/5]

```
Rmessage::Rmessage (
    int cmd,
    const char * txt )
```

Definition at line 105 of file Rconnection.cpp.

6.111.2.4 Rmessage() [4/5]

```
Rmessage::Rmessage (
    int cmd,
    int i )
```

Definition at line 132 of file Rconnection.cpp.

6.111.2.5 Rmessage() [5/5]

```
Rmessage::Rmessage (
    int cmd,
    const void * buf,
    int len,
    int raw_data = 0 )
```

Definition at line 120 of file Rconnection.cpp.

6.111.2.6 ~Rmessage()

```
Rmessage::~Rmessage ( ) [virtual]
```

Definition at line 143 of file Rconnection.cpp.

6.111.3 Member Function Documentation

6.111.3.1 command()

```
int Rmessage::command ( ) [inline]
```

Definition at line 89 of file Rconnection.h.

6.111.3.2 is_complete()

```
int Rmessage::is_complete ( ) [inline]
```

Definition at line 91 of file Rconnection.h.

6.111.3.3 length()

```
Rsize_t Rmessage::length ( ) [inline]
```

Definition at line 90 of file Rconnection.h.

6.111.3.4 parse()

```
void Rmessage::parse ( )
```

Definition at line 192 of file Rconnection.cpp.

6.111.3.5 read()

```
int Rmessage::read ( int s )
```

Definition at line 148 of file Rconnection.cpp.

6.111.3.6 send()

```
int Rmessage::send ( int s )
```

Definition at line 217 of file Rconnection.cpp.

6.111.4 Member Data Documentation

6.111.4.1 complete

```
int Rmessage::complete
```

Definition at line 76 of file Rconnection.h.

6.111.4.2 data

```
char* Rmessage::data
```

Definition at line 74 of file Rconnection.h.

6.111.4.3 head

```
struct phdr Rmessage::head
```

Definition at line 73 of file Rconnection.h.

6.111.4.4 len

```
Rsize_t Rmessage::len
```

Definition at line 75 of file Rconnection.h.

6.111.4.5 par

```
unsigned int* Rmessage::par[16]
```

Definition at line 80 of file Rconnection.h.

6.111.4.6 pars

```
int Rmessage::pars
```

Definition at line 79 of file Rconnection.h.

The documentation for this class was generated from the following files:

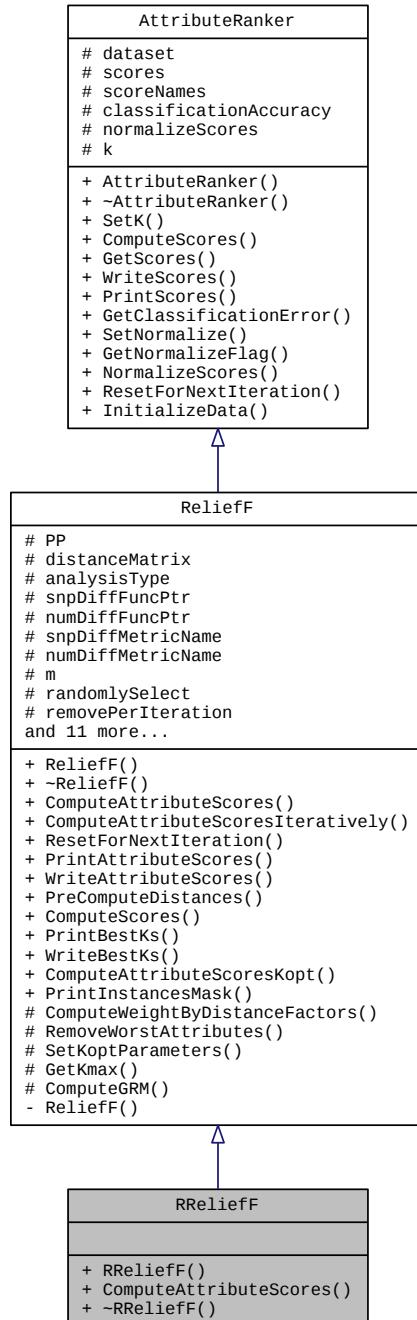
- src/[Rconnection.h](#)
- src/[Rconnection.cpp](#)

6.112 RReliefF Class Reference

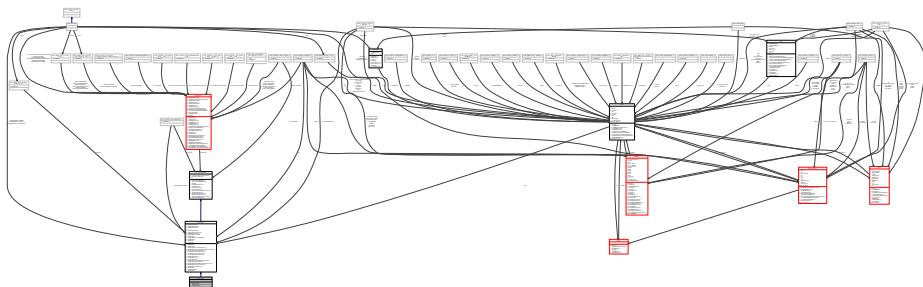
Regression [ReliefF](#) attribute ranking algorithm.

```
#include <RReliefF.h>
```

Inheritance diagram for RReliefF:



Collaboration diagram for RReliefF:



Public Member Functions

- **RReliefF (Dataset *ds, Plink *plinkPtr)**
Construct an [ReliefF](#) algorithm object.
- **bool ComputeAttributeScores () override**
Compute the [ReliefF](#) scores for the current set of attributes.
- **virtual ~RReliefF ()**

Additional Inherited Members

6.112.1 Detailed Description

Regression [ReliefF](#) attribute ranking algorithm.

Totally redone for the McKinney insilico lab in 2011. Large refactoring to move all attribute elimination handling to the [Dataset](#) and its subclasses. 9/11/11

See also

[ReliefF](#)

Author

Bill White

Version

1.0

Contact: bill.c.white@gmail.com Created on: 9/27/11

Definition at line 32 of file RReliefF.h.

6.112.2 Constructor & Destructor Documentation

6.112.2.1 RReliefF()

```
RReliefF::RReliefF (
    Dataset * ds,
    Plink * plinkPtr )
```

Construct an [ReliefF](#) algorithm object.

Parameters

in	<i>ds</i>	pointer to a Dataset object
----	-----------	---

Definition at line 22 of file RReliefF.cpp.

6.112.2.2 ~RReliefF()

```
RReliefF::~RReliefF ( ) [virtual]
```

Definition at line 30 of file RReliefF.cpp.

6.112.3 Member Function Documentation

6.112.3.1 ComputeAttributeScores()

```
bool RReliefF::ComputeAttributeScores ( ) [override], [virtual]
```

Compute the [ReliefF](#) scores for the current set of attributes.

Implements [ReliefF](#) algorithm: Marko Robnik-Sikonja, Igor Kononenko: Theoretical and Empirical Analysis of [ReliefF](#) and [RReliefF](#). Machine Learning Journal, 53:23-69, 2003 <http://lkm.fri.uni-lj.si/rmarko/papers/robnik03-mlj.pdf> Used to hold the probability of different value of an attribute given nearest instances (numeric class case)

Used to hold the probability of a different class value and different attribute value given nearest instances (numeric class case)

algorithm line 1

algorithm line 2

algorithm lines 4, 5 and 6

algorithm line 7

algorithm line 8

algorithm line 9

Reimplemented from [ReliefF](#).

Definition at line 33 of file [RReliefF.cpp](#).

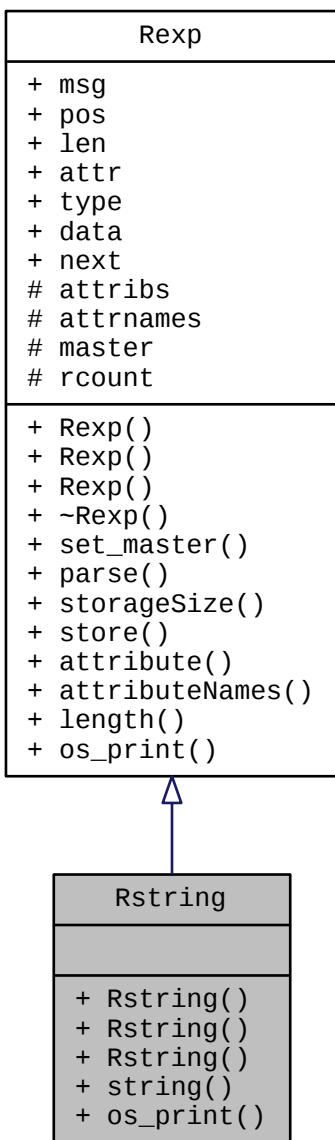
The documentation for this class was generated from the following files:

- src/[RReliefF.h](#)
- src/[RReliefF.cpp](#)

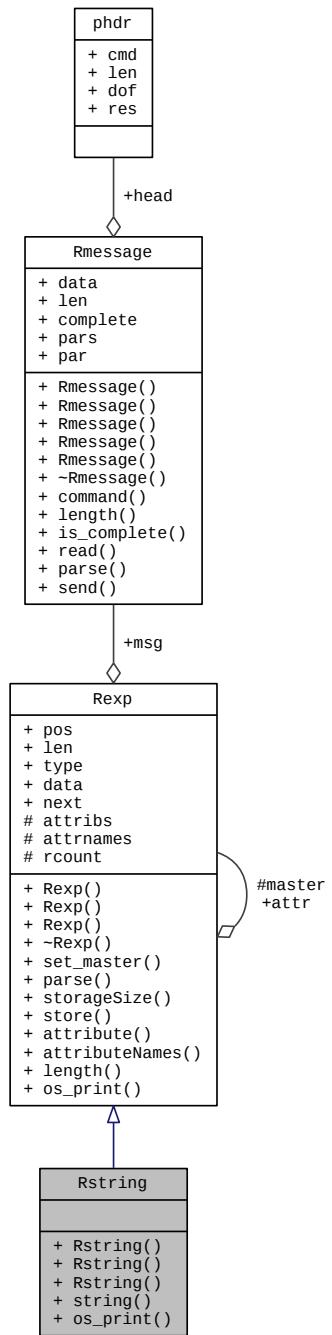
6.113 Rstring Class Reference

```
#include <Rconnection.h>
```

Inheritance diagram for Rstring:



Collaboration diagram for Rstring:



Public Member Functions

- [Rstring \(Rmessage *msg\)](#)

- [Rstring \(unsigned int *ipos, Rmessage *imsg\)](#)
- [Rstring \(const char *str\)](#)
- [char * string \(\)](#)
- [virtual std::ostream & os_print \(std::ostream &os\)](#)

Additional Inherited Members

6.113.1 Detailed Description

Definition at line 261 of file Rconnection.h.

6.113.2 Constructor & Destructor Documentation

6.113.2.1 Rstring() [1/3]

```
Rstring::Rstring (
    Rmessage * msg ) [inline]
```

Definition at line 263 of file Rconnection.h.

6.113.2.2 Rstring() [2/3]

```
Rstring::Rstring (
    unsigned int * ipos,
    Rmessage * imsg ) [inline]
```

Definition at line 264 of file Rconnection.h.

6.113.2.3 Rstring() [3/3]

```
Rstring::Rstring (
    const char * str ) [inline]
```

Definition at line 265 of file Rconnection.h.

6.113.3 Member Function Documentation

6.113.3.1 os_print()

```
virtual std::ostream& Rstring::os_print (
    std::ostream & os ) [inline], [virtual]
```

Reimplemented from [Rexp](#).

Definition at line 269 of file Rconnection.h.

6.113.3.2 string()

```
char* Rstring::string ( ) [inline]
```

Definition at line 267 of file Rconnection.h.

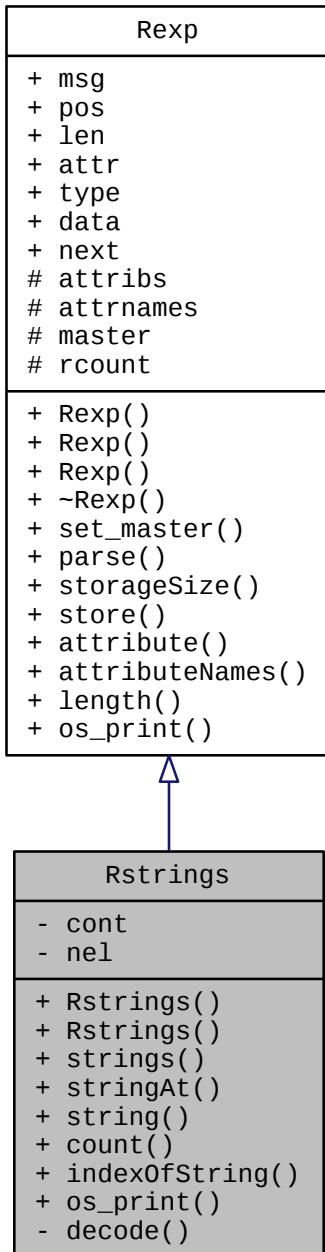
The documentation for this class was generated from the following file:

- src/[Rconnection.h](#)

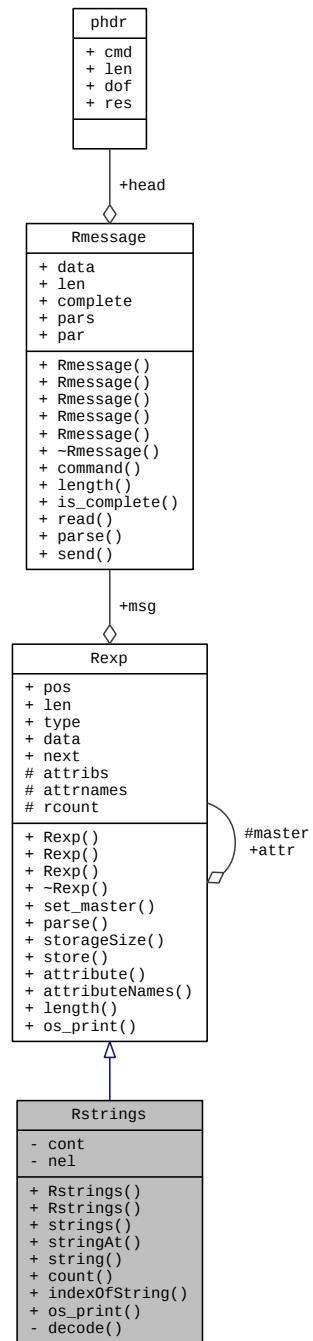
6.114 Rstrings Class Reference

```
#include <Rconnection.h>
```

Inheritance diagram for Rstrings:



Collaboration diagram for Rstrings:



Public Member Functions

- [Rstrings \(Rmessage *msg\)](#)

- `Rstrings` (`unsigned int *ipos, Rmessage *imsg`)
- `char ** strings ()`
- `char * stringAt (int i)`
- `char * string ()`
- `unsigned int count ()`
- `int indexOfString (const char *str)`
- `virtual std::ostream & os_print (std::ostream &os)`

Private Member Functions

- `void decode ()`

Private Attributes

- `char ** cont`
- `unsigned int nel`

Additional Inherited Members

6.114.1 Detailed Description

Definition at line 223 of file Rconnection.h.

6.114.2 Constructor & Destructor Documentation

6.114.2.1 `Rstrings()` [1/2]

```
Rstrings::Rstrings (
    Rmessage * msg )  [inline]
```

Definition at line 227 of file Rconnection.h.

6.114.2.2 `Rstrings()` [2/2]

```
Rstrings::Rstrings (
    unsigned int * ipos,
    Rmessage * imsg )  [inline]
```

Definition at line 228 of file Rconnection.h.

6.114.3 Member Function Documentation

6.114.3.1 count()

```
unsigned int Rstrings::count ( ) [inline]
```

Definition at line 234 of file Rconnection.h.

6.114.3.2 decode()

```
void Rstrings::decode ( ) [inline], [private]
```

Definition at line 241 of file Rconnection.h.

6.114.3.3 indexOfString()

```
int Rstrings::indexOfString (
    const char * str )
```

Definition at line 508 of file Rconnection.cpp.

6.114.3.4 os_print()

```
virtual std::ostream& Rstrings::os_print (
    std::ostream & os ) [inline], [virtual]
```

Reimplemented from [Rexp](#).

Definition at line 237 of file Rconnection.h.

6.114.3.5 string()

```
char* Rstrings::string ( ) [inline]
```

Definition at line 233 of file Rconnection.h.

6.114.3.6 stringAt()

```
char* Rstrings::stringAt (
    int i ) [inline]
```

Definition at line 232 of file Rconnection.h.

6.114.3.7 strings()

```
char** Rstrings::strings ( ) [inline]
```

Definition at line 231 of file Rconnection.h.

6.114.4 Member Data Documentation

6.114.4.1 cont

```
char** Rstrings::cont [private]
```

Definition at line 224 of file Rconnection.h.

6.114.4.2 nel

```
unsigned int Rstrings::nel [private]
```

Definition at line 225 of file Rconnection.h.

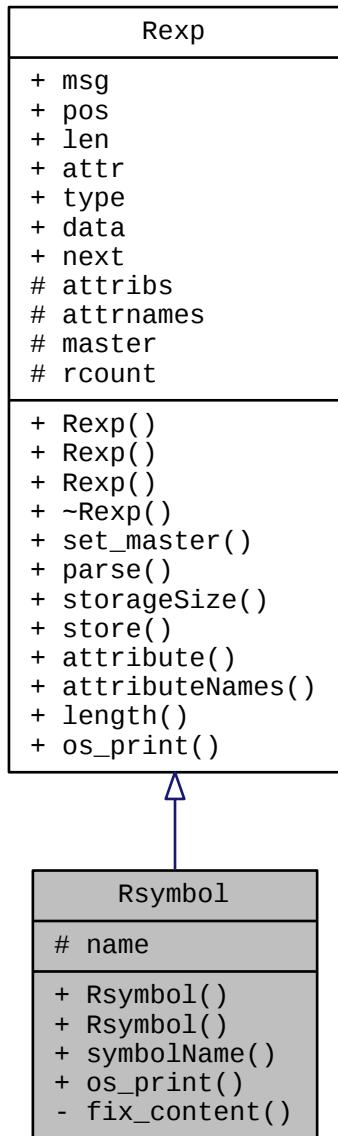
The documentation for this class was generated from the following files:

- src/[Rconnection.h](#)
- src/[Rconnection.cpp](#)

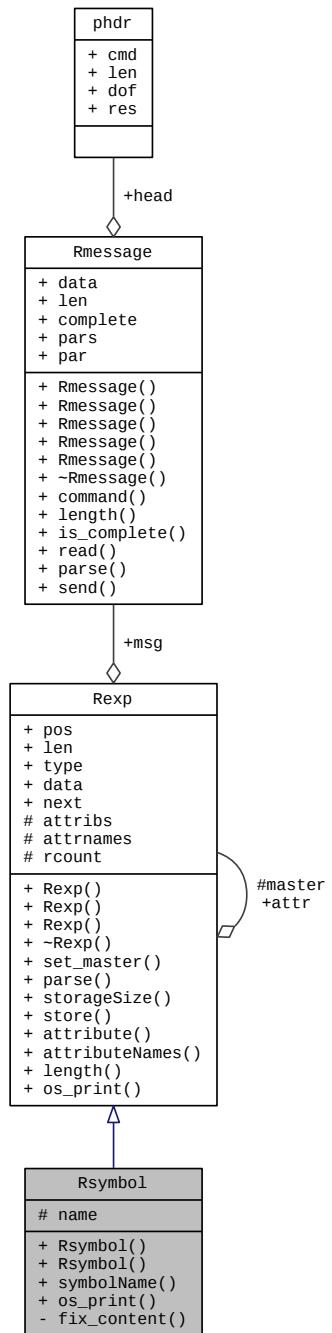
6.115 Rsymbol Class Reference

```
#include <Rconnection.h>
```

Inheritance diagram for Rsymbol:



Collaboration diagram for Rsymbol:



Public Member Functions

- [Rsymbol \(Rmessage *msg\)](#)

- `Rsymbol` (`unsigned int *ipos, Rmessage *imsg`)
- `char * symbolName ()`
- `virtual std::ostream & os_print (std::ostream &os)`

Protected Attributes

- `char * name`

Private Member Functions

- `void fix_content ()`

Additional Inherited Members

6.115.1 Detailed Description

Definition at line 197 of file Rconnection.h.

6.115.2 Constructor & Destructor Documentation

6.115.2.1 Rsymbol() [1/2]

```
Rsymbol::Rsymbol (
    Rmessage * msg ) [inline]
```

Definition at line 202 of file Rconnection.h.

6.115.2.2 Rsymbol() [2/2]

```
Rsymbol::Rsymbol (
    unsigned int * ipos,
    Rmessage * imsg ) [inline]
```

Definition at line 205 of file Rconnection.h.

6.115.3 Member Function Documentation

6.115.3.1 fix_content()

```
void Rsymbol::fix_content ( ) [private]
```

Definition at line 381 of file Rconnection.cpp.

6.115.3.2 os_print()

```
virtual std::ostream& Rsymbol::os_print (
    std::ostream & os ) [inline], [virtual]
```

Reimplemented from [Rexp](#).

Definition at line 210 of file Rconnection.h.

6.115.3.3 symbolName()

```
char* Rsymbol::symbolName ( ) [inline]
```

Definition at line 208 of file Rconnection.h.

6.115.4 Member Data Documentation

6.115.4.1 name

```
char* Rsymbol::name [protected]
```

Definition at line 199 of file Rconnection.h.

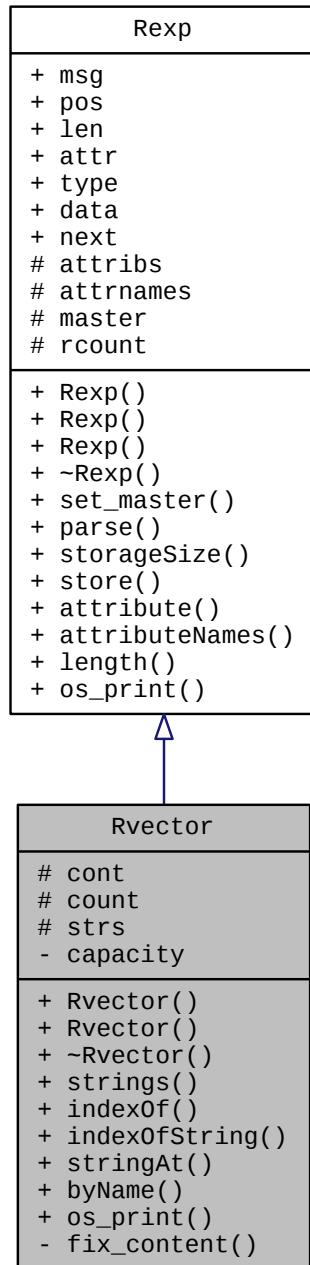
The documentation for this class was generated from the following files:

- src/[Rconnection.h](#)
- src/[Rconnection.cpp](#)

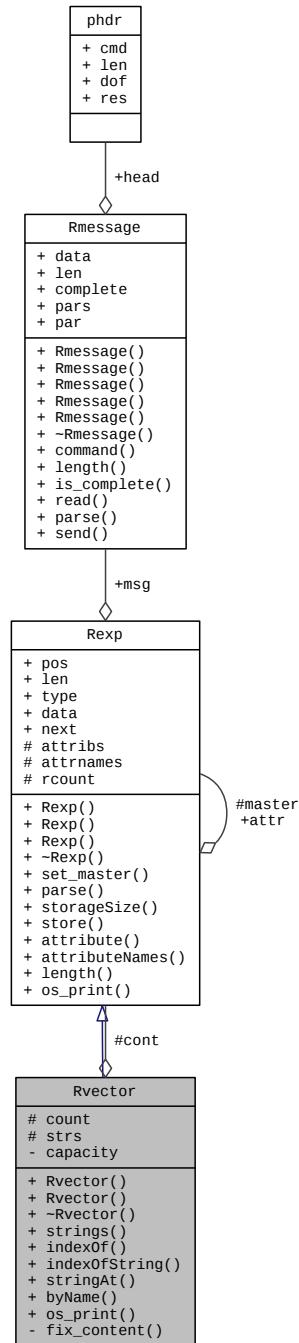
6.116 Rvector Class Reference

```
#include <Rconnection.h>
```

Inheritance diagram for Rvector:



Collaboration diagram for Rvector:



Public Member Functions

- [Rvector \(Rmessage *msg\)](#)

- `Rvector` (`unsigned int *ipos, Rmessage *imsg`)
- `virtual ~Rvector ()`
- `char ** strings ()`
- `int indexOf (Rexp *exp)`
- `int indexOfString (const char *str)`
- `char * stringAt (int i)`
- `Rexp * byName (const char *name)`
- `virtual std::ostream & os_print (std::ostream &os)`

Protected Attributes

- `Rexp ** cont`
- `int count`
- `char ** strs`

Private Member Functions

- `void fix_content ()`

Private Attributes

- `int capacity`

Additional Inherited Members

6.116.1 Detailed Description

Definition at line 318 of file Rconnection.h.

6.116.2 Constructor & Destructor Documentation

6.116.2.1 Rvector() [1/2]

```
Rvector::Rvector (
    Rmessage * msg )  [inline]
```

Definition at line 326 of file Rconnection.h.

6.116.2.2 Rvector() [2/2]

```
Rvector::Rvector (
    unsigned int * ipos,
    Rmessage * imsg ) [inline]
```

Definition at line 329 of file Rconnection.h.

6.116.2.3 ~Rvector()

```
Rvector::~Rvector () [virtual]
```

Definition at line 462 of file Rconnection.cpp.

6.116.3 Member Function Documentation

6.116.3.1 byName()

```
Rexp * Rvector::byName (
    const char * name )
```

Definition at line 517 of file Rconnection.cpp.

6.116.3.2 fix_content()

```
void Rvector::fix_content () [private]
```

Definition at line 537 of file Rconnection.cpp.

6.116.3.3 indexOf()

```
int Rvector::indexOf (
    Rexp * exp )
```

Definition at line 490 of file Rconnection.cpp.

6.116.3.4 indexOfString()

```
int Rvector::indexOfString (
    const char * str )
```

Definition at line 499 of file Rconnection.cpp.

6.116.3.5 os_print()

```
virtual std::ostream& Rvector::os_print (
    std::ostream & os ) [inline], [virtual]
```

Reimplemented from [Rexp](#).

Definition at line 345 of file Rconnection.h.

6.116.3.6 stringAt()

```
char* Rvector::stringAt (
    int i ) [inline]
```

Definition at line 338 of file Rconnection.h.

6.116.3.7 strings()

```
char ** Rvector::strings ( )
```

Definition at line 472 of file Rconnection.cpp.

6.116.4 Member Data Documentation

6.116.4.1 capacity

```
int Rvector::capacity [private]
```

Definition at line 356 of file Rconnection.h.

6.116.4.2 cont

```
Rexp** Rvector::cont [protected]
```

Definition at line 320 of file Rconnection.h.

6.116.4.3 count

```
int Rvector::count [protected]
```

Definition at line 321 of file Rconnection.h.

6.116.4.4 strs

```
char** Rvector::strs [protected]
```

Definition at line 324 of file Rconnection.h.

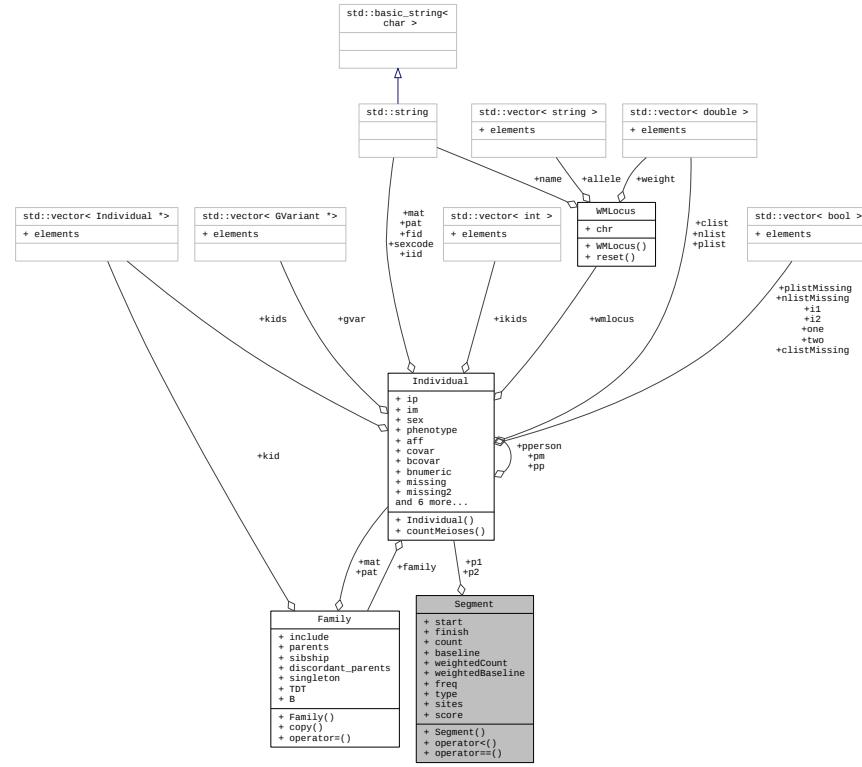
The documentation for this class was generated from the following files:

- src/Rconnection.h
- src/Rconnection.cpp

6.117 Segment Class Reference

```
#include <plink.h>
```

Collaboration diagram for Segment:



Public Member Functions

- Segment ()
 - bool operator< (const Segment &b) const
 - bool operator==(const Segment &b) const

Public Attributes

- int `start`
 - int `finish`
 - Individual * `p1`
 - Individual * `p2`
 - int `count`
 - int `baseline`
 - double `weightedCount`
 - double `weightedBaseline`
 - int `freq`
 - int `type`
 - int `sites`
 - double `score`

6.117.1 Detailed Description

Definition at line 511 of file plink.h.

6.117.2 Constructor & Destructor Documentation

6.117.2.1 Segment()

```
Segment::Segment ( ) [inline]
```

Definition at line 514 of file plink.h.

6.117.3 Member Function Documentation

6.117.3.1 operator<()

```
bool Segment::operator< (
    const Segment & b ) const [inline]
```

Definition at line 541 of file plink.h.

6.117.3.2 operator==()

```
bool Segment::operator== (
    const Segment & b ) const [inline]
```

Definition at line 550 of file plink.h.

6.117.4 Member Data Documentation

6.117.4.1 baseline

```
int Segment::baseline
```

Definition at line 529 of file plink.h.

6.117.4.2 count

```
int Segment::count
```

Definition at line 528 of file plink.h.

6.117.4.3 finish

```
int Segment::finish
```

Definition at line 522 of file plink.h.

6.117.4.4 freq

```
int Segment::freq
```

Definition at line 534 of file plink.h.

6.117.4.5 p1

```
Individual* Segment::p1
```

Definition at line 523 of file plink.h.

6.117.4.6 p2

```
Individual* Segment::p2
```

Definition at line 524 of file plink.h.

6.117.4.7 score

```
double Segment::score
```

Definition at line 537 of file plink.h.

6.117.4.8 sites

```
int Segment::sites
```

Definition at line 536 of file plink.h.

6.117.4.9 start

```
int Segment::start
```

Definition at line 521 of file plink.h.

6.117.4.10 type

```
int Segment::type
```

Definition at line 535 of file plink.h.

6.117.4.11 weightedBaseline

```
double Segment::weightedBaseline
```

Definition at line 532 of file plink.h.

6.117.4.12 weightedCount

```
double Segment::weightedCount
```

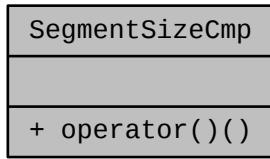
Definition at line 531 of file plink.h.

The documentation for this class was generated from the following file:

- [src/plink.h](#)

6.118 SegmentSizeCmp Class Reference

Collaboration diagram for SegmentSizeCmp:



Public Member Functions

- bool [operator\(\)](#) (const [Segment](#) &s1, const [Segment](#) &s2) const

6.118.1 Detailed Description

Definition at line 1934 of file segment.cpp.

6.118.2 Member Function Documentation

6.118.2.1 operator()()

```
bool SegmentSizeCmp::operator() (
    const Segment & s1,
    const Segment & s2 ) const [inline]
```

Definition at line 1936 of file segment.cpp.

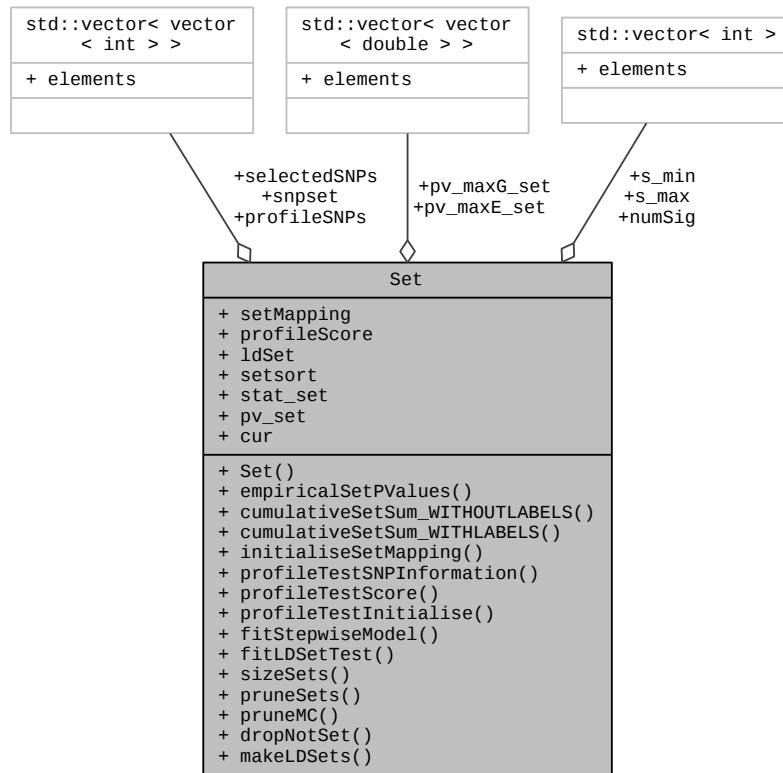
The documentation for this class was generated from the following file:

- [src/segment.cpp](#)

6.119 Set Class Reference

```
#include <sets.h>
```

Collaboration diagram for Set:



Public Member Functions

- `Set (vector<vector< int >> &)`
- `void empiricalSetPValues ()`
- `void cumulativeSetSum_WITHOUTLABELS (vector< double > &, int)`
- `void cumulativeSetSum_WITHLABELS (Plink &, vector< double > &)`
- `void initialiseSetMapping ()`
- `void profileTestSNPInformation (int, double)`
- `vector_t profileTestScore ()`
- `void profileTestInitialise ()`
- `vector_t fitStepwiseModel ()`
- `vector_t fitLDSetTest (vector_t &, bool)`
- `void sizeSets ()`
- `void pruneSets (Plink &)`
- `void pruneMC (Plink &, bool, double)`
- `void dropNotSet (Plink &)`
- `void makeLDSets ()`

Public Attributes

- vector< int > [s_min](#)
- vector< int > [s_max](#)
- vector< vector< int > > & [snpset](#)
- map< int, set< int > > [setMapping](#)
- vector< vector< int > > [profileSNPs](#)
- vector< vector_t > [profileScore](#)
- vector< vector< set< int > > > [ldSet](#)
- vector< int > [numSig](#)
- vector< vector< int > > [selectedSNPs](#)
- vector< vector< string > > [setsort](#)
- vector< vector< vector< double > > > [stat_set](#)
- vector< vector< vector< double > > > [pv_set](#)
- vector< vector< double > > [pv_maxG_set](#)
- vector< vector< double > > [pv_maxE_set](#)
- vector< vector< bool > > [cur](#)

6.119.1 Detailed Description

Definition at line 27 of file sets.h.

6.119.2 Constructor & Destructor Documentation

6.119.2.1 Set()

```
Set::Set (
    vector< vector< int > > & ss )
```

Definition at line 28 of file sets.cpp.

6.119.3 Member Function Documentation

6.119.3.1 cumulativeSetSum_WITHLABELS()

```
void Set::cumulativeSetSum_WITHLABELS (
    PLink & P,
    vector< double > & original )
```

Definition at line 445 of file sets.cpp.

6.119.3.2 cumulativeSetSum_WITHOUTLABELS()

```
void Set::cumulativeSetSum_WITHOUTLABELS (
    vector< double > & perm,
    int p )
```

Definition at line 524 of file sets.cpp.

6.119.3.3 dropNotSet()

```
void Set::dropNotSet (
    Plink & P )
```

Definition at line 209 of file sets.cpp.

6.119.3.4 empiricalSetPValues()

```
void Set::empiricalSetPValues ( )
```

Definition at line 603 of file sets.cpp.

6.119.3.5 fitLDSetTest()

```
vector_t Set::fitLDSetTest (
    vector_t & singleSNP,
    bool save )
```

Definition at line 869 of file sets.cpp.

6.119.3.6 fitStepwiseModel()

```
vector_t Set::fitStepwiseModel ( )
```

Definition at line 1021 of file sets.cpp.

6.119.3.7 initialiseSetMapping()

```
void Set::initialiseSetMapping ( )
```

Definition at line 407 of file sets.cpp.

6.119.3.8 makeLDSets()

```
void Set::makeLDSets ( )
```

Definition at line 253 of file sets.cpp.

6.119.3.9 profileTestInitialise()

```
void Set::profileTestInitialise ( )
```

Definition at line 812 of file sets.cpp.

6.119.3.10 profileTestScore()

```
vector_t Set::profileTestScore ( )
```

Definition at line 693 of file sets.cpp.

6.119.3.11 profileTestSNPInformation()

```
void Set::profileTestSNPInformation (
    int l,
    double odds )
```

Definition at line 661 of file sets.cpp.

6.119.3.12 pruneMC()

```
void Set::pruneMC (
    PLink & P,
    bool disp,
    double VIF_threshold )
```

Definition at line 124 of file sets.cpp.

6.119.3.13 pruneSets()

```
void Set::pruneSets (
    PLink & P )
```

Definition at line 94 of file sets.cpp.

6.119.3.14 sizeSets()

```
void Set::sizeSets ( )
```

Definition at line 33 of file sets.cpp.

6.119.4 Member Data Documentation

6.119.4.1 cur

```
vector<vector<bool> > Set::cur
```

Definition at line 84 of file sets.h.

6.119.4.2 ldSet

```
vector< vector<set<int> > > Set::ldSet
```

Definition at line 61 of file sets.h.

6.119.4.3 numSig

```
vector<int> Set::numSig
```

Definition at line 62 of file sets.h.

6.119.4.4 profileScore

```
vector<vector_t> Set::profileScore
```

Definition at line 54 of file sets.h.

6.119.4.5 profileSNPs

```
vector<vector<int> > Set::profileSNPs
```

Definition at line 53 of file sets.h.

6.119.4.6 pv_maxE_set

```
vector<vector<double> > Set::pv_maxE_set
```

Definition at line 81 of file sets.h.

6.119.4.7 pv_maxG_set

```
vector<vector<double> > Set::pv_maxG_set
```

Definition at line 80 of file sets.h.

6.119.4.8 pv_set

```
vector<vector<vector<double> > > Set::pv_set
```

Definition at line 79 of file sets.h.

6.119.4.9 s_max

```
vector<int> Set::s_max
```

Definition at line 33 of file sets.h.

6.119.4.10 s_min

```
vector<int> Set::s_min
```

Definition at line 32 of file sets.h.

6.119.4.11 selectedSNPs

```
vector<vector<int> > Set::selectedSNPs
```

Definition at line 63 of file sets.h.

6.119.4.12 setMapping

```
map<int, set<int> > Set::setMapping
```

Definition at line 39 of file sets.h.

6.119.4.13 setsort

```
vector<vector<string> > Set::setsort
```

Definition at line 73 of file sets.h.

6.119.4.14 snpset

```
vector<vector<int> >& Set::snpset
```

Definition at line 37 of file sets.h.

6.119.4.15 stat_set

```
vector<vector<vector<double> > > Set::stat_set
```

Definition at line 76 of file sets.h.

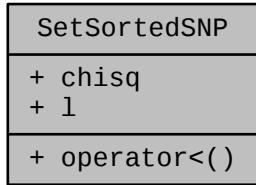
The documentation for this class was generated from the following files:

- src/[sets.h](#)
- src/[sets.cpp](#)

6.120 SetSortedSNP Class Reference

```
#include <sets.h>
```

Collaboration diagram for SetSortedSNP:



Public Member Functions

- bool [operator<](#) (const SetSortedSNP &s2) const

Public Attributes

- double [chisq](#)
- int [l](#)

6.120.1 Detailed Description

Definition at line 17 of file sets.h.

6.120.2 Member Function Documentation

6.120.2.1 operator<()

```
bool SetSortedSNP::operator< (
    const SetSortedSNP & s2 ) const [inline]
```

Definition at line 22 of file sets.h.

6.120.3 Member Data Documentation

6.120.3.1 chisq

```
double SetSortedSNP::chisq
```

Definition at line 20 of file sets.h.

6.120.3.2 l

```
int SetSortedSNP::l
```

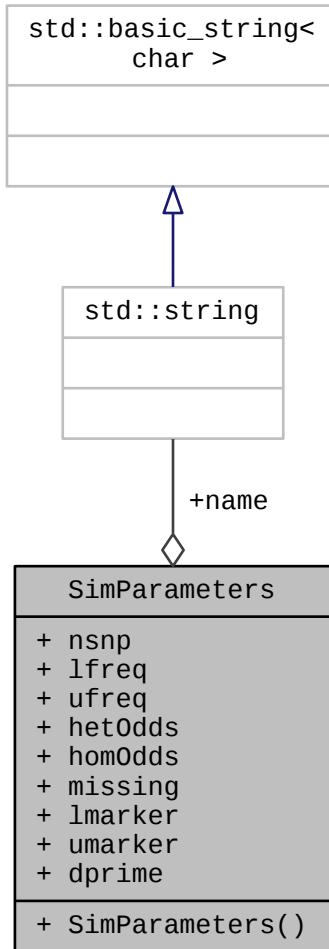
Definition at line 21 of file sets.h.

The documentation for this class was generated from the following file:

- src/[sets.h](#)

6.121 SimParameters Class Reference

Collaboration diagram for SimParameters:



Public Member Functions

- [SimParameters \(\)](#)

Public Attributes

- int `nsnp`
- double `lfreq`

- double `ufreq`
- double `hetOdds`
- double `homOdds`
- double `missing`
- string `name`
- double `lmarker`
- double `umarker`
- double `dprime`

6.121.1 Detailed Description

Definition at line 30 of file simul.cpp.

6.121.2 Constructor & Destructor Documentation

6.121.2.1 SimParameters()

```
SimParameters::SimParameters ( ) [inline]
```

Definition at line 45 of file simul.cpp.

6.121.3 Member Data Documentation

6.121.3.1 dprime

```
double SimParameters::dprime
```

Definition at line 43 of file simul.cpp.

6.121.3.2 hetOdds

```
double SimParameters::hetOdds
```

Definition at line 36 of file simul.cpp.

6.121.3.3 homOdds

```
double SimParameters::homOdds
```

Definition at line 37 of file simul.cpp.

6.121.3.4 lfreq

```
double SimParameters::lfreq
```

Definition at line 34 of file simul.cpp.

6.121.3.5 lmarker

```
double SimParameters::lmarker
```

Definition at line 41 of file simul.cpp.

6.121.3.6 missing

```
double SimParameters::missing
```

Definition at line 38 of file simul.cpp.

6.121.3.7 name

```
string SimParameters::name
```

Definition at line 39 of file simul.cpp.

6.121.3.8 nsnp

```
int SimParameters::nsnp
```

Definition at line 33 of file simul.cpp.

6.121.3.9 ufreq

```
double SimParameters::ufreq
```

Definition at line 35 of file simul.cpp.

6.121.3.10 umarker

```
double SimParameters::umarker
```

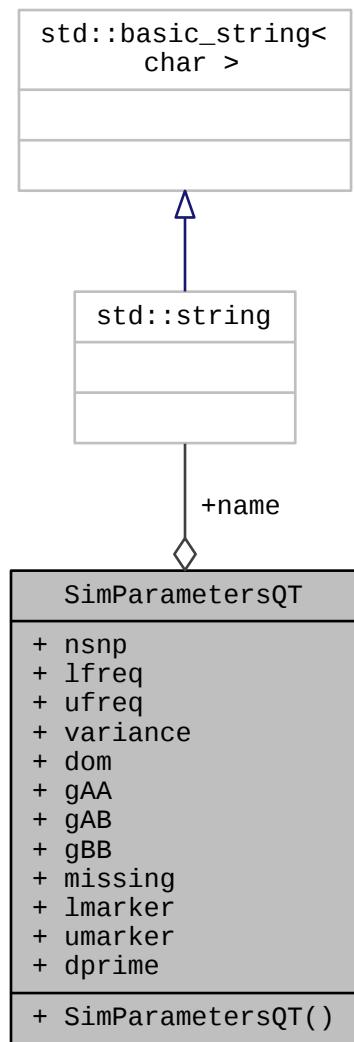
Definition at line 42 of file simul.cpp.

The documentation for this class was generated from the following file:

- src/[simul.cpp](#)

6.122 SimParametersQT Class Reference

Collaboration diagram for SimParametersQT:



Public Member Functions

- [SimParametersQT \(\)](#)

Public Attributes

- int `nsnp`
- double `lfreq`
- double `ufreq`
- double `variance`
- double `dom`
- double `gAA`
- double `gAB`
- double `gBB`
- double `missing`
- string `name`
- double `lmarker`
- double `umarker`
- double `dprime`

6.122.1 Detailed Description

Definition at line 57 of file simul.cpp.

6.122.2 Constructor & Destructor Documentation

6.122.2.1 SimParametersQT()

```
SimParametersQT::SimParametersQT ( ) [inline]
```

Definition at line 76 of file simul.cpp.

6.122.3 Member Data Documentation

6.122.3.1 dom

```
double SimParametersQT::dom
```

Definition at line 65 of file simul.cpp.

6.122.3.2 dprime

```
double SimParametersQT::dprime
```

Definition at line 73 of file simul.cpp.

6.122.3.3 gAA

```
double SimParametersQT::gAA
```

Definition at line 66 of file simul.cpp.

6.122.3.4 gAB

```
double SimParametersQT::gAB
```

Definition at line 66 of file simul.cpp.

6.122.3.5 gBB

```
double SimParametersQT::gBB
```

Definition at line 66 of file simul.cpp.

6.122.3.6 lfreq

```
double SimParametersQT::lfreq
```

Definition at line 61 of file simul.cpp.

6.122.3.7 lmarker

```
double SimParametersQT::lmarker
```

Definition at line 71 of file simul.cpp.

6.122.3.8 missing

```
double SimParametersQT::missing
```

Definition at line 68 of file simul.cpp.

6.122.3.9 name

```
string SimParametersQT::name
```

Definition at line 69 of file simul.cpp.

6.122.3.10 nsnp

```
int SimParametersQT::nsnp
```

Definition at line 60 of file simul.cpp.

6.122.3.11 ufreq

```
double SimParametersQT::ufreq
```

Definition at line 62 of file simul.cpp.

6.122.3.12 umarker

```
double SimParametersQT::umarker
```

Definition at line 72 of file simul.cpp.

6.122.3.13 variance

```
double SimParametersQT::variance
```

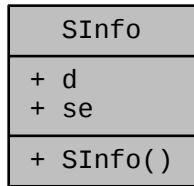
Definition at line 64 of file simul.cpp.

The documentation for this class was generated from the following file:

- [src/simul.cpp](#)

6.123 SInfo Class Reference

Collaboration diagram for SInfo:



Public Member Functions

- [SInfo \(double d, double se\)](#)

Public Attributes

- double [d](#)
- double [se](#)

6.123.1 Detailed Description

Definition at line 31 of file metaanal.cpp.

6.123.2 Constructor & Destructor Documentation

6.123.2.1 SInfo()

```
SInfo::SInfo (
    double d,
    double se ) [inline]
```

Definition at line 33 of file metaanal.cpp.

6.123.3 Member Data Documentation

6.123.3.1 d

```
double SInfo::d
```

Definition at line 34 of file metaanal.cpp.

6.123.3.2 se

```
double SInfo::se
```

Definition at line 35 of file metaanal.cpp.

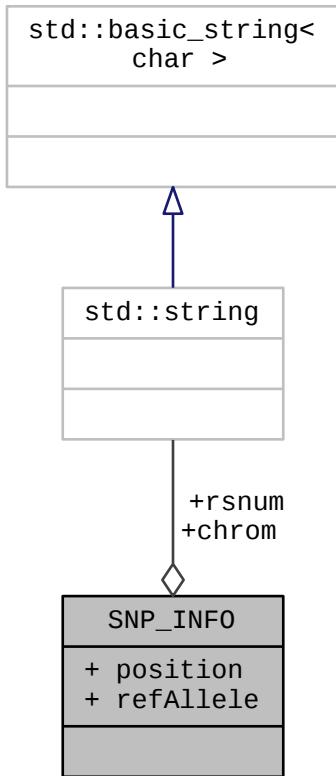
The documentation for this class was generated from the following file:

- [src/metaanal.cpp](#)

6.124 SNP_INFO Struct Reference

```
#include <DcVar.h>
```

Collaboration diagram for SNP_INFO:



Public Attributes

- `std::string chrom`
- `uint position`
- `std::string rsnum`
- `char refAllele`

6.124.1 Detailed Description

Definition at line 32 of file DcVar.h.

6.124.2 Member Data Documentation

6.124.2.1 chrom

```
std::string SNP_INFO::chrom
```

Definition at line 33 of file DcVar.h.

6.124.2.2 position

```
uint SNP_INFO::position
```

Definition at line 34 of file DcVar.h.

6.124.2.3 refAllele

```
char SNP_INFO::refAllele
```

Definition at line 36 of file DcVar.h.

6.124.2.4 rsnum

```
std::string SNP_INFO::rsnum
```

Definition at line 35 of file DcVar.h.

The documentation for this struct was generated from the following file:

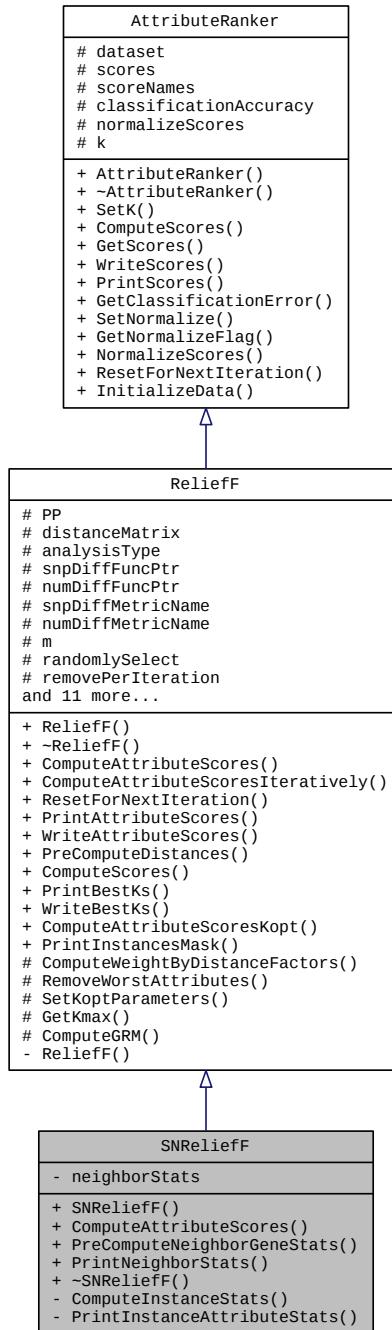
- src/[DcVar.h](#)

6.125 SNReliefF Class Reference

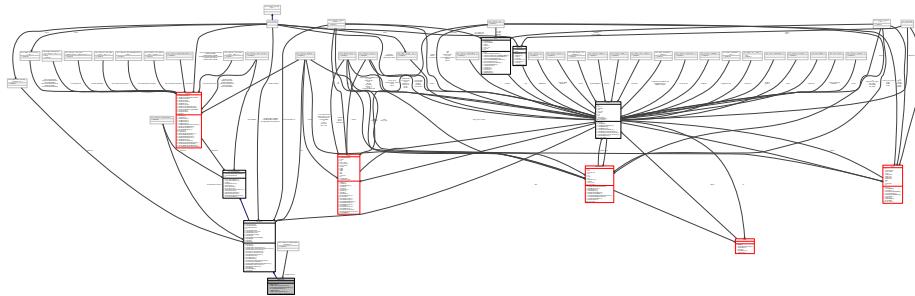
Signal-to-Noise ReliefF attribute ranking algorithm.

```
#include <SNReliefF.h>
```

Inheritance diagram for SNReliefF:



Collaboration diagram for SNReliefF:



Public Member Functions

- `SNReliefF (Dataset *ds, Plink *plinkPtr)`
- `bool ComputeAttributeScores () override`
Compute the ReliefF scores for the current set of attributes.
- `bool PreComputeNeighborGeneStats ()`
Precompute nearest neighbor gene statistics for all instances.
- `void PrintNeighborStats ()`
Print the neighbor statistics data structure.
- `virtual ~SNReliefF ()`

Private Member Functions

- `bool ComputeInstanceStats (DatasetInstance *dsi, std::vector< unsigned int > hitIndices, std::vector< unsigned int > missIndices, InstanceHitMissStats &hitMissStats)`
Computes the nearest neighbor statistics for a particular instance.
- `void PrintInstanceAttributeStats (InstanceAttributeStats stats)`
Prints all attribute stats to stdout.

Private Attributes

- `NeighborStats neighborStats`
nearest neighbor attribute averages and standard deviations

Additional Inherited Members

6.125.1 Detailed Description

Signal-to-Noise ReliefF attribute ranking algorithm.

Designed to handle digital gene expression (DGE) data sets, particularly RNA-Seq high-throughput count data, by accounting for variable-specific variance in counts. Data is known to follow a Poisson or negative binomial distribution. This algorithm is a more computationally practical approach than others that use more sophisticated statistical methods and models. Our approach keeps the ReliefF algorithm general while addressing the variance "dispersion" problem as a special case.

See also[ReliefF](#)**Author**

Bill White

Version

1.0

Contact: bill.c.white@gmail.com Created on: 7/21/12

Definition at line 51 of file SNReliefF.h.

6.125.2 Constructor & Destructor Documentation**6.125.2.1 SNReliefF()**

```
SNReliefF::SNReliefF ( 
    Dataset * ds,
    PLink * plinkPtr )
```

Definition at line 22 of file SNReliefF.cpp.

6.125.2.2 ~SNReliefF()

```
SNReliefF::~SNReliefF ( ) [virtual]
```

Definition at line 27 of file SNReliefF.cpp.

6.125.3 Member Function Documentation

6.125.3.1 ComputeAttributeScores()

```
bool SNReliefF::ComputeAttributeScores ( ) [override], [virtual]
```

Compute the [ReliefF](#) scores for the current set of attributes.

Implements [ReliefF](#) algorithm: Marko Robnik-Sikonja, Igor Kononenko: Theoretical and Empirical Analysis of [ReliefF](#) and [RReliefF](#). Machine Learning Journal, 53:23-69, 2003 <http://lkm.fri.uni-lj.si/rmarko/papers/robnik03-mlj.pdf> algorithm line 1

algorithm line 2

algorithm lines 4, 5 and 6

algorithm line 7

algorithm line 8

algorithm line 9

Reimplemented from [ReliefF](#).

Definition at line 30 of file SNReliefF.cpp.

6.125.3.2 ComputeInstanceStats()

```
bool SNReliefF::ComputeInstanceStats (
    DatasetInstance * dsi,
    std::vector< unsigned int > hitIndices,
    std::vector< unsigned int > missIndices,
    InstanceHitMissStats & hitMissStats ) [private]
```

Computes the nearest neighbor statistics for a particular instance.

Definition at line 106 of file SNReliefF.cpp.

6.125.3.3 PreComputeNeighborGeneStats()

```
bool SNReliefF::PreComputeNeighborGeneStats ( )
```

Precompute nearest neighbor gene statistics for all instances.

Definition at line 83 of file SNReliefF.cpp.

6.125.3.4 PrintInstanceAttributeStats()

```
void SNReliefF::PrintInstanceAttributeStats (  
    InstanceAttributeStats stats ) [private]
```

Prints all attribute stats to stdout.

Definition at line 174 of file SNReliefF.cpp.

6.125.3.5 PrintNeighborStats()

```
void SNReliefF::PrintNeighborStats ( )
```

Print the neighbor statistics data structure.

Definition at line 156 of file SNReliefF.cpp.

6.125.4 Member Data Documentation

6.125.4.1 neighborStats

```
NeighborStats SNReliefF::neighborStats [private]
```

nearest neighbor attribute averages and standard deviations

Definition at line 70 of file SNReliefF.h.

The documentation for this class was generated from the following files:

- src/[SNReliefF.h](#)
- src/[SNReliefF.cpp](#)

6.126 SortedResult Class Reference

Collaboration diagram for SortedResult:

SortedResult
+ chisq
+ p
+ l
+ operator<()

Public Member Functions

- bool `operator<` (const `SortedResult` &*s2*) const

Public Attributes

- double `chisq`
- double `p`
- int `l`

6.126.1 Detailed Description

Definition at line 29 of file setscreen.cpp.

6.126.2 Member Function Documentation

6.126.2.1 `operator<()`

```
bool SortedResult::operator< (
    const SortedResult & s2 ) const [inline]
```

Definition at line 37 of file setscreen.cpp.

6.126.3 Member Data Documentation

6.126.3.1 `chisq`

```
double SortedResult::chisq
```

Definition at line 33 of file setscreen.cpp.

6.126.3.2 `l`

```
int SortedResult::l
```

Definition at line 35 of file setscreen.cpp.

6.126.3.3 p

```
double SortedResult::p
```

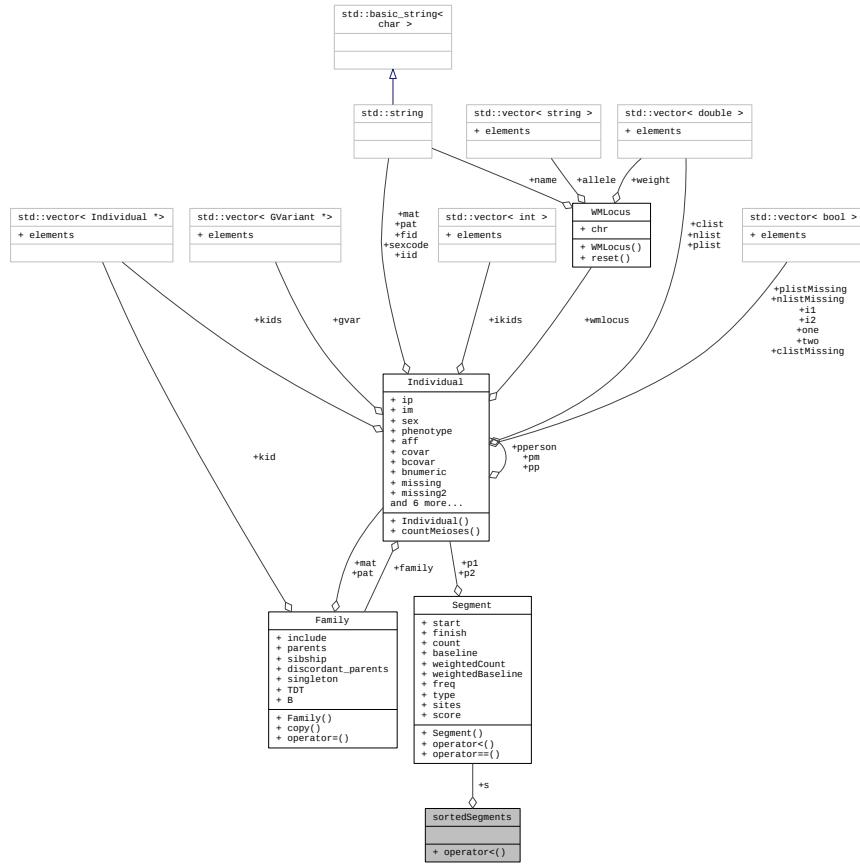
Definition at line 34 of file setscreen.cpp.

The documentation for this class was generated from the following file:

- [src/setscreen.cpp](#)

6.127 sortedSegments Class Reference

Collaboration diagram for sortedSegments:



Public Member Functions

- `bool operator< (const sortedSegments &b) const`

Public Attributes

- [Segment s](#)

6.127.1 Detailed Description

Definition at line 35 of file cnv.cpp.

6.127.2 Member Function Documentation

6.127.2.1 operator<()

```
bool sortedSegments::operator< (
    const sortedSegments & b ) const [inline]
```

Definition at line 39 of file cnv.cpp.

6.127.3 Member Data Documentation

6.127.3.1 s

```
Segment sortedSegments::s
```

Definition at line 38 of file cnv.cpp.

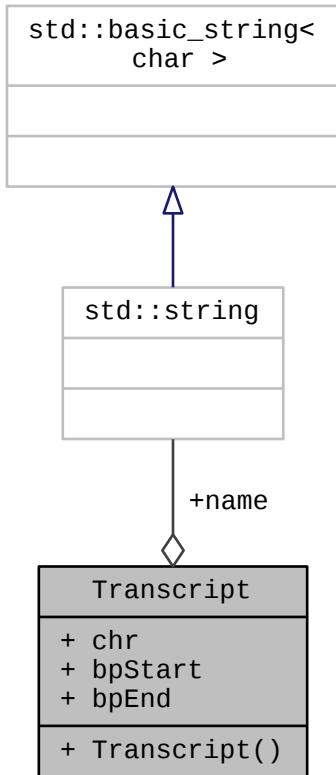
The documentation for this class was generated from the following file:

- [src/cnv.cpp](#)

6.128 Transcript Class Reference

```
#include <plink.h>
```

Collaboration diagram for Transcript:



Public Member Functions

- `Transcript()`

Public Attributes

- int `chr`
- string `name`
- int `bpStart`
- int `bpEnd`

6.128.1 Detailed Description

Definition at line 435 of file plink.h.

6.128.2 Constructor & Destructor Documentation

6.128.2.1 Transcript()

```
Transcript::Transcript () [inline]
```

Definition at line 438 of file plink.h.

6.128.3 Member Data Documentation

6.128.3.1 bpEnd

```
int Transcript::bpEnd
```

Definition at line 448 of file plink.h.

6.128.3.2 bpStart

```
int Transcript::bpStart
```

Definition at line 447 of file plink.h.

6.128.3.3 chr

```
int Transcript::chr
```

Definition at line 445 of file plink.h.

6.128.3.4 name

```
string Transcript::name
```

Definition at line 446 of file plink.h.

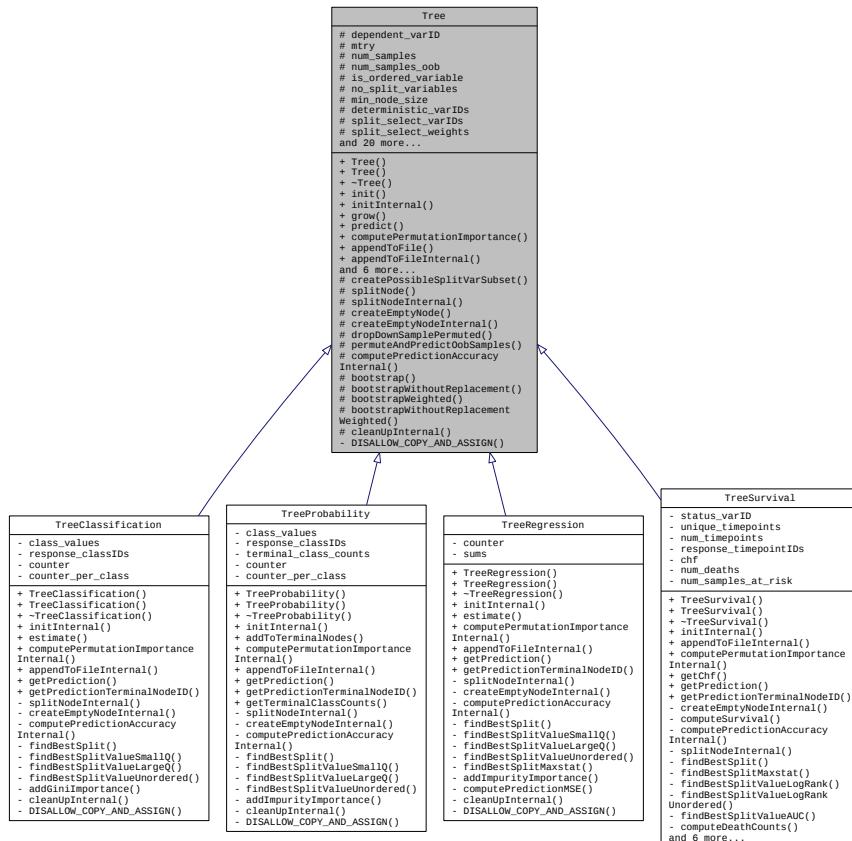
The documentation for this class was generated from the following file:

- [src/plink.h](#)

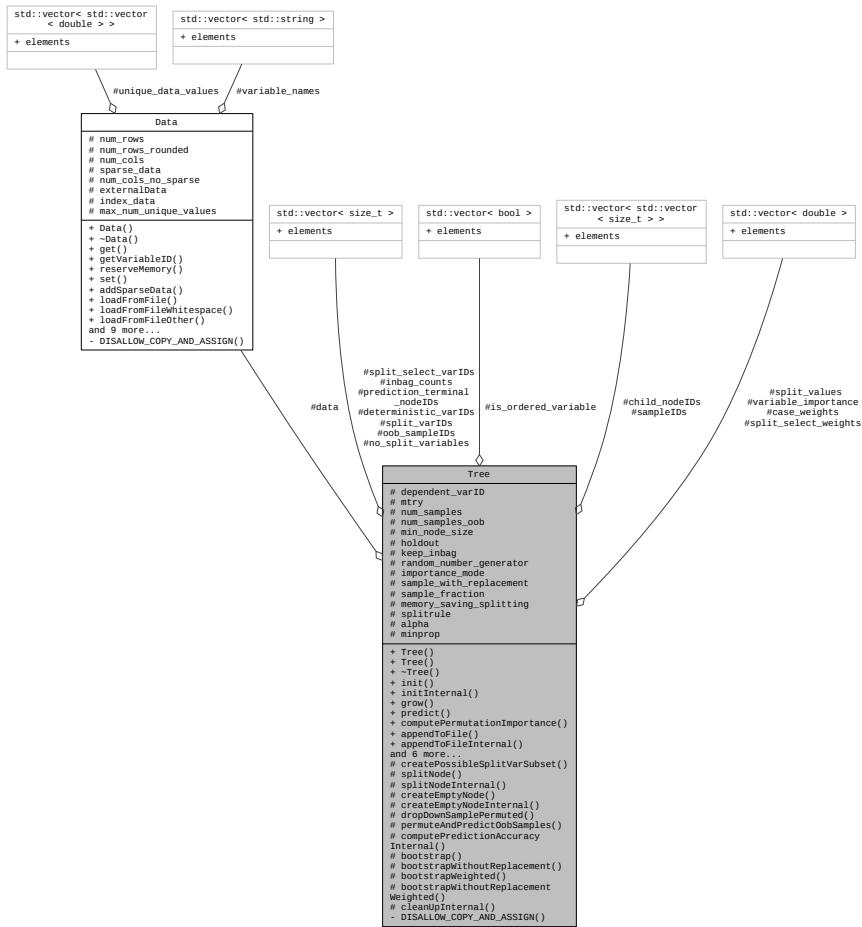
6.129 Tree Class Reference

```
#include <Tree.h>
```

Inheritance diagram for Tree:



Collaboration diagram for Tree:



Public Member Functions

- `Tree ()`
 - `Tree (std::vector< std::vector< size_t >> &child_nodeIDs, std::vector< size_t > &split_varIDs, std::vector< double > &split_values, std::vector< bool > *is_ordered_variable)`
 - `virtual ~Tree ()`
 - `void init (Data *data, uint mtry, size_t dependent_varID, size_t num_samples, uint seed, std::vector< size_t > *deterministic_varIDs, std::vector< size_t > *split_select_varIDs, std::vector< double > *split_select_weights, ImportanceMode importance_mode, uint min_node_size, std::vector< size_t > *no_split_variables, bool sample_with_replacement, std::vector< bool > *is_unordered, bool memory_saving_splitting, SplitRule splitrule, std::vector< double > *case_weights, bool keep_inbag, double sample_fraction, double alpha, double minprop, bool holdout)`
 - `virtual void initInternal ()=0`
 - `void grow (std::vector< double > *variable_importance)`
 - `void predict (const Data *prediction_data, bool oob_prediction)`
 - `void computePermutationImportance (std::vector< double > *forest_importance, std::vector< double > *forest_variance)`

- void [appendToFile](#) (std::ofstream &file)
- virtual void [appendToFileInternal](#) (std::ofstream &file)=0
- const std::vector< std::vector< size_t > > & [getChildNodeIDs](#) () const
- const std::vector< double > & [getSplitValues](#) () const
- const std::vector< size_t > & [getSplitVarIDs](#) () const
- const std::vector< size_t > & [getOobSampleIDs](#) () const
- size_t [getNumSamplesOob](#) () const
- const std::vector< size_t > & [getInbagCounts](#) () const

Protected Member Functions

- void [createPossibleSplitVarSubset](#) (std::vector< size_t > &result)
- bool [splitNode](#) (size_t nodeID)
- virtual bool [splitNodeInternal](#) (size_t nodeID, std::vector< size_t > &possible_split_varIDs)=0
- void [createEmptyNode](#) ()
- virtual void [createEmptyNodeInternal](#) ()=0
- size_t [dropDownSamplePermuted](#) (size_t permuted_varID, size_t sampleID, size_t permuted_sampleID)
- void [permuteAndPredictOobSamples](#) (size_t permuted_varID, std::vector< size_t > &permutations)
- virtual double [computePredictionAccuracyInternal](#) ()=0
- void [bootstrap](#) ()
- void [bootstrapWithoutReplacement](#) ()
- void [bootstrapWeighted](#) ()
- void [bootstrapWithoutReplacementWeighted](#) ()
- virtual void [cleanUpInternal](#) ()=0

Protected Attributes

- size_t [dependent_varID](#)
- uint [mtry](#)
- size_t [num_samples](#)
- size_t [num_samples_oob](#)
- std::vector< bool > * [is_ordered_variable](#)
- std::vector< size_t > * [no_split_variables](#)
- uint [min_node_size](#)
- std::vector< size_t > * [deterministic_varIDs](#)
- std::vector< size_t > * [split_select_varIDs](#)
- std::vector< double > * [split_select_weights](#)
- std::vector< double > * [case_weights](#)
- std::vector< size_t > [split_varIDs](#)
- std::vector< double > [split_values](#)
- std::vector< std::vector< size_t > > [child_nodeIDs](#)
- std::vector< std::vector< size_t > > [sampleIDs](#)
- std::vector< size_t > [oob_sampleIDs](#)
- bool [holdout](#)
- bool [keep_inbag](#)
- std::vector< size_t > [inbag_counts](#)
- std::mt19937_64 [random_number_generator](#)
- Data * [data](#)
- std::vector< double > * [variable_importance](#)

- `ImportanceMode importance_mode`
- `std::vector< size_t > prediction_terminal_nodeIDs`
- `bool sample_with_replacement`
- `double sample_fraction`
- `bool memory_saving_splitting`
- `SplitRule splitrule`
- `double alpha`
- `double minprop`

Private Member Functions

- `DISALLOW_COPY_AND_ASSIGN (Tree)`

6.129.1 Detailed Description

Definition at line 39 of file Tree.h.

6.129.2 Constructor & Destructor Documentation

6.129.2.1 Tree() [1/2]

```
Tree::Tree ( )
```

Definition at line 34 of file Tree.cpp.

6.129.2.2 Tree() [2/2]

```
Tree::Tree (   
    std::vector< std::vector< size_t >> & child_nodeIDs,  
    std::vector< size_t > & split_varIDs,  
    std::vector< double > & split_values,  
    std::vector< bool > * is_ordered_variable )
```

Definition at line 42 of file Tree.cpp.

6.129.2.3 ~Tree()

```
Tree::~Tree ( ) [virtual]
```

Definition at line 52 of file Tree.cpp.

6.129.3 Member Function Documentation

6.129.3.1 appendToFile()

```
void Tree::appendToFile (
    std::ofstream & file )
```

Definition at line 228 of file Tree.cpp.

6.129.3.2 appendToFileInternal()

```
virtual void Tree::appendToFileInternal (
    std::ofstream & file ) [pure virtual]
```

Implemented in [TreeProbability](#), [TreeClassification](#), [TreeRegression](#), and [TreeSurvival](#).

6.129.3.3 bootstrap()

```
void Tree::bootstrap ( ) [protected]
```

Definition at line 373 of file Tree.cpp.

6.129.3.4 bootstrapWeighted()

```
void Tree::bootstrapWeighted ( ) [protected]
```

Definition at line 407 of file Tree.cpp.

6.129.3.5 bootstrapWithoutReplacement()

```
void Tree::bootstrapWithoutReplacement ( ) [protected]
```

Definition at line 449 of file Tree.cpp.

6.129.3.6 bootstrapWithoutReplacementWeighted()

```
void Tree::bootstrapWithoutReplacementWeighted ( ) [protected]
```

Definition at line 466 of file Tree.cpp.

6.129.3.7 cleanUpInternal()

```
virtual void Tree::cleanUpInternal ( ) [protected], [pure virtual]
```

Implemented in [TreeSurvival](#), [TreeProbability](#), [TreeRegression](#), and [TreeClassification](#).

6.129.3.8 computePermutationImportance()

```
void Tree::computePermutationImportance (
    std::vector< double > * forest_importance,
    std::vector< double > * forest_variance )
```

Definition at line 189 of file Tree.cpp.

6.129.3.9 computePredictionAccuracyInternal()

```
virtual double Tree::computePredictionAccuracyInternal ( ) [protected], [pure virtual]
```

Implemented in [TreeProbability](#), [TreeSurvival](#), [TreeClassification](#), and [TreeRegression](#).

6.129.3.10 createEmptyNode()

```
void Tree::createEmptyNode ( )  [protected]
```

Definition at line 309 of file Tree.cpp.

6.129.3.11 createEmptyNodeInternal()

```
virtual void Tree::createEmptyNodeInternal ( )  [protected], [pure virtual]
```

Implemented in [TreeProbability](#), [TreeSurvival](#), [TreeClassification](#), and [TreeRegression](#).

6.129.3.12 createPossibleSplitVarSubset()

```
void Tree::createPossibleSplitVarSubset (
    std::vector< size_t > & result )  [protected]
```

Definition at line 239 of file Tree.cpp.

6.129.3.13 DISALLOW_COPY_AND_ASSIGN()

```
Tree::DISALLOW_COPY_AND_ASSIGN (
    Tree   )  [private]
```

6.129.3.14 dropDownSamplePermuted()

```
size_t Tree::dropDownSamplePermuted (
    size_t permuted_varID,
    size_t sampleID,
    size_t permuted_sampleID )  [protected]
```

Definition at line 319 of file Tree.cpp.

6.129.3.15 getChildNodeIDs()

```
const std::vector<std::vector<size_t>>& Tree::getChildNodeIDs() const [inline]
```

Definition at line 67 of file Tree.h.

6.129.3.16 getInbagCounts()

```
const std::vector<size_t>& Tree::getInbagCounts() const [inline]
```

Definition at line 84 of file Tree.h.

6.129.3.17 getNumSamplesOob()

```
size_t Tree::getNumSamplesOob() const [inline]
```

Definition at line 80 of file Tree.h.

6.129.3.18 getOobSampleIDs()

```
const std::vector<size_t>& Tree::getOobSampleIDs() const [inline]
```

Definition at line 77 of file Tree.h.

6.129.3.19 getSplitValues()

```
const std::vector<double>& Tree::getSplitValues() const [inline]
```

Definition at line 70 of file Tree.h.

6.129.3.20 getSplitVarIDs()

```
const std::vector<size_t>& Tree::getSplitVarIDs() const [inline]
```

Definition at line 73 of file Tree.h.

6.129.3.21 grow()

```
void Tree::grow (
    std::vector< double > * variable_importance )
```

Definition at line 95 of file [Tree.cpp](#).

6.129.3.22 init()

```
void Tree::init (
    Data * data,
    uint mtry,
    size_t dependent_varID,
    size_t num_samples,
    uint seed,
    std::vector< size_t > * deterministic_varIDs,
    std::vector< size_t > * split_select_varIDs,
    std::vector< double > * split_select_weights,
    ImportanceMode importance_mode,
    uint min_node_size,
    std::vector< size_t > * no_split_variables,
    bool sample_with_replacement,
    std::vector< bool > * is_unordered,
    bool memory_saving_splitting,
    SplitRule splitrule,
    std::vector< double > * case_weights,
    bool keep_inbag,
    double sample_fraction,
    double alpha,
    double minprop,
    bool holdout )
```

Definition at line 55 of file [Tree.cpp](#).

6.129.3.23 initInternal()

```
virtual void Tree::initInternal ( ) [pure virtual]
```

Implemented in [TreeProbability](#), [TreeClassification](#), [TreeSurvival](#), and [TreeRegression](#).

6.129.3.24 permuteAndPredictOobSamples()

```
void Tree::permuteAndPredictOobSamples (
    size_t permuted_varID,
    std::vector< size_t > & permutations ) [protected]
```

Definition at line 360 of file Tree.cpp.

6.129.3.25 predict()

```
void Tree::predict (
    const Data * prediction_data,
    bool oob_prediction )
```

Definition at line 132 of file Tree.cpp.

6.129.3.26 splitNode()

```
bool Tree::splitNode (
    size_t nodeID ) [protected]
```

Definition at line 254 of file Tree.cpp.

6.129.3.27 splitNodeInternal()

```
virtual bool Tree::splitNodeInternal (
    size_t nodeID,
    std::vector< size_t > & possible_split_varIDs ) [protected], [pure virtual]
```

Implemented in [TreeSurvival](#), [TreeProbability](#), [TreeClassification](#), and [TreeRegression](#).

6.129.4 Member Data Documentation

6.129.4.1 alpha

```
double Tree::alpha [protected]
```

Definition at line 179 of file Tree.h.

6.129.4.2 case_weights

```
std::vector<double>* Tree::case_weights [protected]
```

Definition at line 135 of file Tree.h.

6.129.4.3 child_nodeIDs

```
std::vector<std::vector<size_t> > Tree::child_nodeIDs [protected]
```

Definition at line 145 of file Tree.h.

6.129.4.4 data

```
Data* Tree::data [protected]
```

Definition at line 164 of file Tree.h.

6.129.4.5 dependent_varID

```
size_t Tree::dependent_varID [protected]
```

Definition at line 110 of file Tree.h.

6.129.4.6 deterministic_varIDs

```
std::vector<size_t>* Tree::deterministic_varIDs [protected]
```

Definition at line 130 of file Tree.h.

6.129.4.7 holdout

```
bool Tree::holdout [protected]
```

Definition at line 154 of file Tree.h.

6.129.4.8 importance_mode

`ImportanceMode Tree::importance_mode [protected]`

Definition at line 168 of file Tree.h.

6.129.4.9 inbag_counts

`std::vector<size_t> Tree::inbag_counts [protected]`

Definition at line 158 of file Tree.h.

6.129.4.10 is_ordered_variable

`std::vector<bool>* Tree::is_ordered_variable [protected]`

Definition at line 120 of file Tree.h.

6.129.4.11 keep_inbag

`bool Tree::keep_inbag [protected]`

Definition at line 157 of file Tree.h.

6.129.4.12 memory_saving_splitting

`bool Tree::memory_saving_splitting [protected]`

Definition at line 177 of file Tree.h.

6.129.4.13 min_node_size

`uint Tree::min_node_size [protected]`

Definition at line 126 of file Tree.h.

6.129.4.14 minprop

```
double Tree::minprop [protected]
```

Definition at line 180 of file Tree.h.

6.129.4.15 mtry

```
uint Tree::mtry [protected]
```

Definition at line 111 of file Tree.h.

6.129.4.16 no_split_variables

```
std::vector<size_t>* Tree::no_split_variables [protected]
```

Definition at line 123 of file Tree.h.

6.129.4.17 num_samples

```
size_t Tree::num_samples [protected]
```

Definition at line 114 of file Tree.h.

6.129.4.18 num_samples_oob

```
size_t Tree::num_samples_oob [protected]
```

Definition at line 117 of file Tree.h.

6.129.4.19 oob_sampleIDs

```
std::vector<size_t> Tree::oob_sampleIDs [protected]
```

Definition at line 151 of file Tree.h.

6.129.4.20 prediction_terminal_nodeIDs

```
std::vector<size_t> Tree::prediction_terminal_nodeIDs [protected]
```

Definition at line 172 of file Tree.h.

6.129.4.21 random_number_generator

```
std::mt19937_64 Tree::random_number_generator [protected]
```

Definition at line 161 of file Tree.h.

6.129.4.22 sample_fraction

```
double Tree::sample_fraction [protected]
```

Definition at line 175 of file Tree.h.

6.129.4.23 sample_with_replacement

```
bool Tree::sample_with_replacement [protected]
```

Definition at line 174 of file Tree.h.

6.129.4.24 sampleIDs

```
std::vector<std::vector<size_t>> Tree::sampleIDs [protected]
```

Definition at line 148 of file Tree.h.

6.129.4.25 split_select_varIDs

```
std::vector<size_t>* Tree::split_select_varIDs [protected]
```

Definition at line 131 of file Tree.h.

6.129.4.26 split_select_weights

```
std::vector<double>* Tree::split_select_weights [protected]
```

Definition at line 132 of file Tree.h.

6.129.4.27 split_values

```
std::vector<double> Tree::split_values [protected]
```

Definition at line 142 of file Tree.h.

6.129.4.28 split_varIDs

```
std::vector<size_t> Tree::split_varIDs [protected]
```

Definition at line 138 of file Tree.h.

6.129.4.29 splitrule

```
SplitRule Tree::splitrule [protected]
```

Definition at line 178 of file Tree.h.

6.129.4.30 variable_importance

```
std::vector<double>* Tree::variable_importance [protected]
```

Definition at line 167 of file Tree.h.

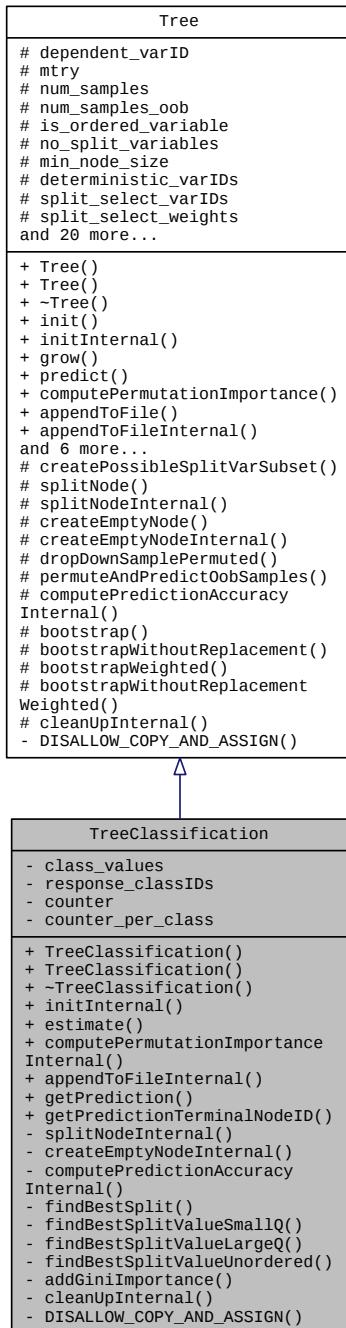
The documentation for this class was generated from the following files:

- src/[Tree.h](#)
- src/[Tree.cpp](#)

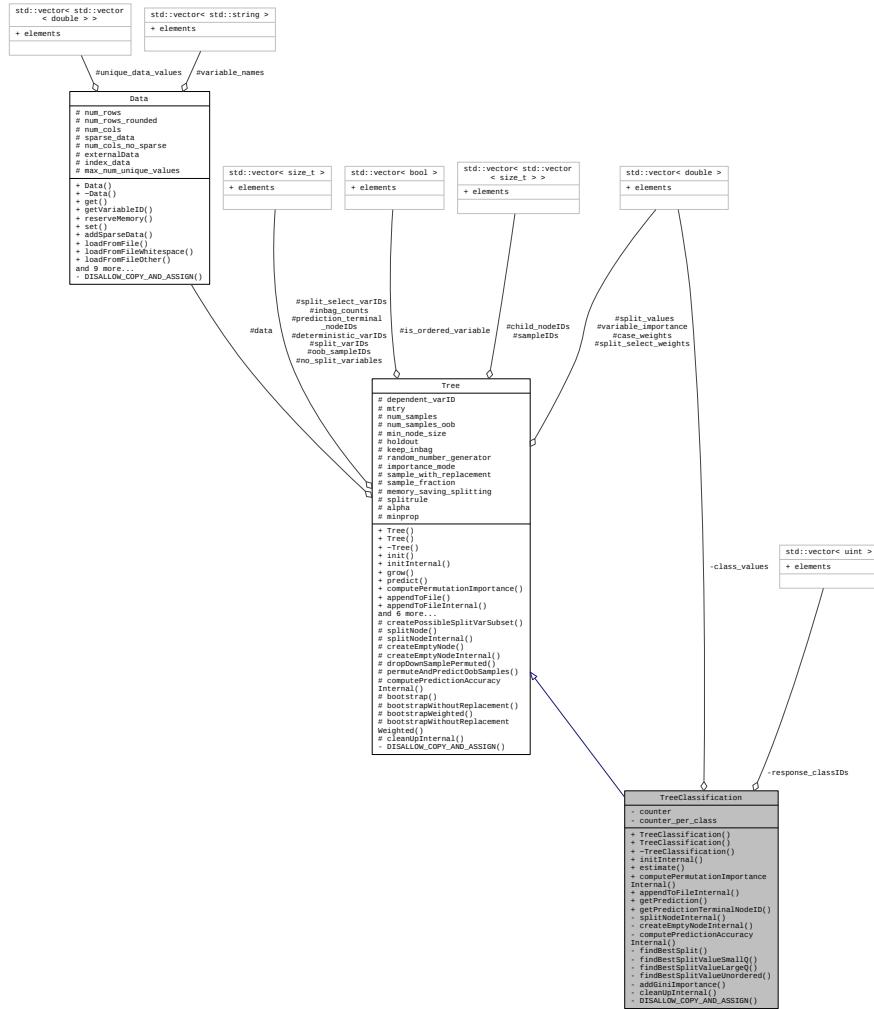
6.130 TreeClassification Class Reference

```
#include <TreeClassification.h>
```

Inheritance diagram for TreeClassification:



Collaboration diagram for TreeClassification:



Public Member Functions

- `TreeClassification` (`std::vector< double > *class_values`, `std::vector< uint > *response_classIDs`)
 - `TreeClassification` (`std::vector< std::vector< size_t >> &child_nodeIDs`, `std::vector< size_t > &split_varIDs`,
`std::vector< double > &split_values`, `std::vector< double > *class_values`, `std::vector< uint > *response_classIDs`, `std::vector< bool > *is_ordered_variable`)
 - virtual `~TreeClassification` ()
 - void `initInternal` () override
 - double `estimate` (`size_t nodeID`)
 - void `computePermutationImportanceInternal` (`std::vector< std::vector< size_t >> *permutations`)
 - void `appendToFileInternal` (`std::ofstream &file`) override
 - double `getPrediction` (`size_t sampleID`) const
 - `size_t getPredictionTerminalNodeID` (`size_t sampleID`) const

Private Member Functions

- bool `splitNodeInternal` (size_t nodeID, std::vector< size_t > &possible_split_varIDs) override
- void `createEmptyNodeInternal` () override
- double `computePredictionAccuracyInternal` () override
- bool `findBestSplit` (size_t nodeID, std::vector< size_t > &possible_split_varIDs)
- void `findBestSplitValueSmallQ` (size_t nodeID, size_t varID, size_t num_classes, size_t *class_counts, size_t num_samples_node, double &best_value, size_t &best_varID, double &best_decrease)
- void `findBestSplitValueLargeQ` (size_t nodeID, size_t varID, size_t num_classes, size_t *class_counts, size_t num_samples_node, double &best_value, size_t &best_varID, double &best_decrease)
- void `findBestSplitValueUnordered` (size_t nodeID, size_t varID, size_t num_classes, size_t *class_counts, size_t num_samples_node, double &best_value, size_t &best_varID, double &best_decrease)
- void `addGiniImportance` (size_t nodeID, size_t varID, double decrease)
- void `cleanUpInternal` () override
- `DISALLOW_COPY_AND_ASSIGN` (`TreeClassification`)

Private Attributes

- std::vector< double > * `class_values`
- std::vector< uint > * `response_classIDs`
- size_t * `counter`
- size_t * `counter_per_class`

Additional Inherited Members

6.130.1 Detailed Description

Definition at line 35 of file `TreeClassification.h`.

6.130.2 Constructor & Destructor Documentation

6.130.2.1 `TreeClassification()` [1/2]

```
TreeClassification::TreeClassification (
    std::vector< double > * class_values,
    std::vector< uint > * response_classIDs )
```

Definition at line 40 of file `TreeClassification.cpp`.

6.130.2.2 TreeClassification() [2/2]

```
TreeClassification::TreeClassification (
    std::vector< std::vector< size_t >> & child_nodeIDs,
    std::vector< size_t > & split_varIDs,
    std::vector< double > & split_values,
    std::vector< double > * class_values,
    std::vector< uint > * response_classIDs,
    std::vector< bool > * is_ordered_variable )
```

Definition at line 44 of file TreeClassification.cpp.

6.130.2.3 ~TreeClassification()

```
TreeClassification::~TreeClassification ( ) [virtual]
```

Definition at line 51 of file TreeClassification.cpp.

6.130.3 Member Function Documentation

6.130.3.1 addGiniImportance()

```
void TreeClassification::addGiniImportance (
    size_t nodeID,
    size_t varID,
    double decrease ) [private]
```

Definition at line 411 of file TreeClassification.cpp.

6.130.3.2 appendToFileInternal()

```
void TreeClassification::appendToFileInternal (
    std::ofstream & file ) [override], [virtual]
```

Implements [Tree](#).

Definition at line 77 of file TreeClassification.cpp.

6.130.3.3 cleanUpInternal()

```
void TreeClassification::cleanUpInternal () [inline], [override], [private], [virtual]
```

Implements [Tree](#).

Definition at line 78 of file `TreeClassification.h`.

6.130.3.4 computePermutationImportanceInternal()

```
void TreeClassification::computePermutationImportanceInternal (
    std::vector< std::vector< size_t >> * permutations )
```

6.130.3.5 computePredictionAccuracyInternal()

```
double TreeClassification::computePredictionAccuracyInternal () [override], [private], [virtual]
```

Implements [Tree](#).

Definition at line 119 of file `TreeClassification.cpp`.

6.130.3.6 createEmptyNodeInternal()

```
void TreeClassification::createEmptyNodeInternal () [override], [private], [virtual]
```

Implements [Tree](#).

Definition at line 115 of file `TreeClassification.cpp`.

6.130.3.7 DISALLOW_COPY_AND_ASSIGN()

```
TreeClassification::DISALLOW_COPY_AND_ASSIGN (
    TreeClassification ) [private]
```

6.130.3.8 estimate()

```
double TreeClassification::estimate (
    size_t nodeID )
```

Definition at line 65 of file TreeClassification.cpp.

6.130.3.9 findBestSplit()

```
bool TreeClassification::findBestSplit (
    size_t nodeID,
    std::vector< size_t > & possible_split_varIDs ) [private]
```

Definition at line 134 of file TreeClassification.cpp.

6.130.3.10 findBestSplitValueLargeQ()

```
void TreeClassification::findBestSplitValueLargeQ (
    size_t nodeID,
    size_t varID,
    size_t num_classes,
    size_t * class_counts,
    size_t num_samples_node,
    double & best_value,
    size_t & best_varID,
    double & best_decrease ) [private]
```

Definition at line 276 of file TreeClassification.cpp.

6.130.3.11 findBestSplitValueSmallQ()

```
void TreeClassification::findBestSplitValueSmallQ (
    size_t nodeID,
    size_t varID,
    size_t num_classes,
    size_t * class_counts,
    size_t num_samples_node,
    double & best_value,
    size_t & best_varID,
    double & best_decrease ) [private]
```

Definition at line 194 of file TreeClassification.cpp.

6.130.3.12 findBestSplitValueUnordered()

```
void TreeClassification::findBestSplitValueUnordered (
    size_t nodeID,
    size_t varID,
    size_t num_classes,
    size_t * class_counts,
    size_t num_samples_node,
    double & best_value,
    size_t & best_varID,
    double & best_decrease ) [private]
```

Definition at line 337 of file TreeClassification.cpp.

6.130.3.13 getPrediction()

```
double TreeClassification::getPrediction (
    size_t sampleID ) const [inline]
```

Definition at line 52 of file TreeClassification.h.

6.130.3.14 getPredictionTerminalNodeID()

```
size_t TreeClassification::getPredictionTerminalNodeID (
    size_t sampleID ) const [inline]
```

Definition at line 57 of file TreeClassification.h.

6.130.3.15 initInternal()

```
void TreeClassification::initInternal ( ) [override], [virtual]
```

Implements [Tree](#).

Definition at line 55 of file TreeClassification.cpp.

6.130.3.16 splitNodeInternal()

```
bool TreeClassification::splitNodeInternal (
    size_t nodeID,
    std::vector< size_t > & possible_split_varIDs ) [override], [private], [virtual]
```

Implements [Tree](#).

Definition at line 81 of file TreeClassification.cpp.

6.130.4 Member Data Documentation

6.130.4.1 class_values

```
std::vector<double>* TreeClassification::class_values [private]
```

Definition at line 88 of file TreeClassification.h.

6.130.4.2 counter

```
size_t* TreeClassification::counter [private]
```

Definition at line 91 of file TreeClassification.h.

6.130.4.3 counter_per_class

```
size_t* TreeClassification::counter_per_class [private]
```

Definition at line 92 of file TreeClassification.h.

6.130.4.4 response_classIDs

```
std::vector<uint>* TreeClassification::response_classIDs [private]
```

Definition at line 89 of file TreeClassification.h.

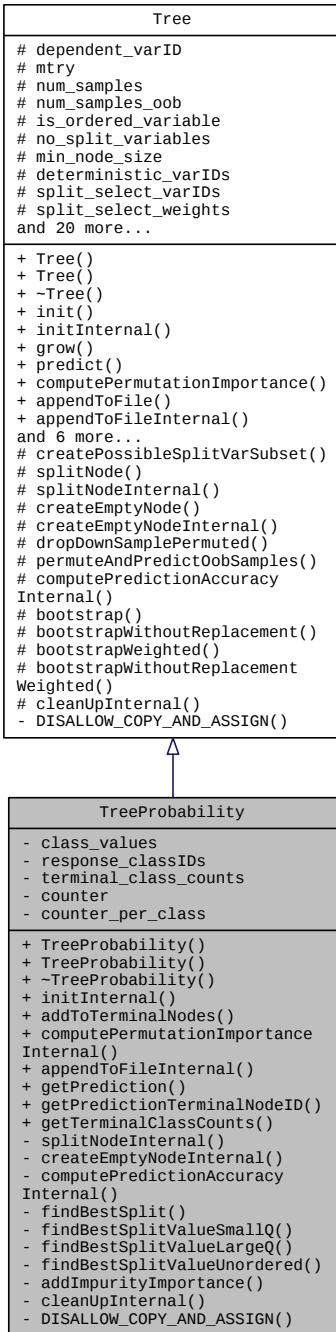
The documentation for this class was generated from the following files:

- [src/TreeClassification.h](#)
- [src/TreeClassification.cpp](#)

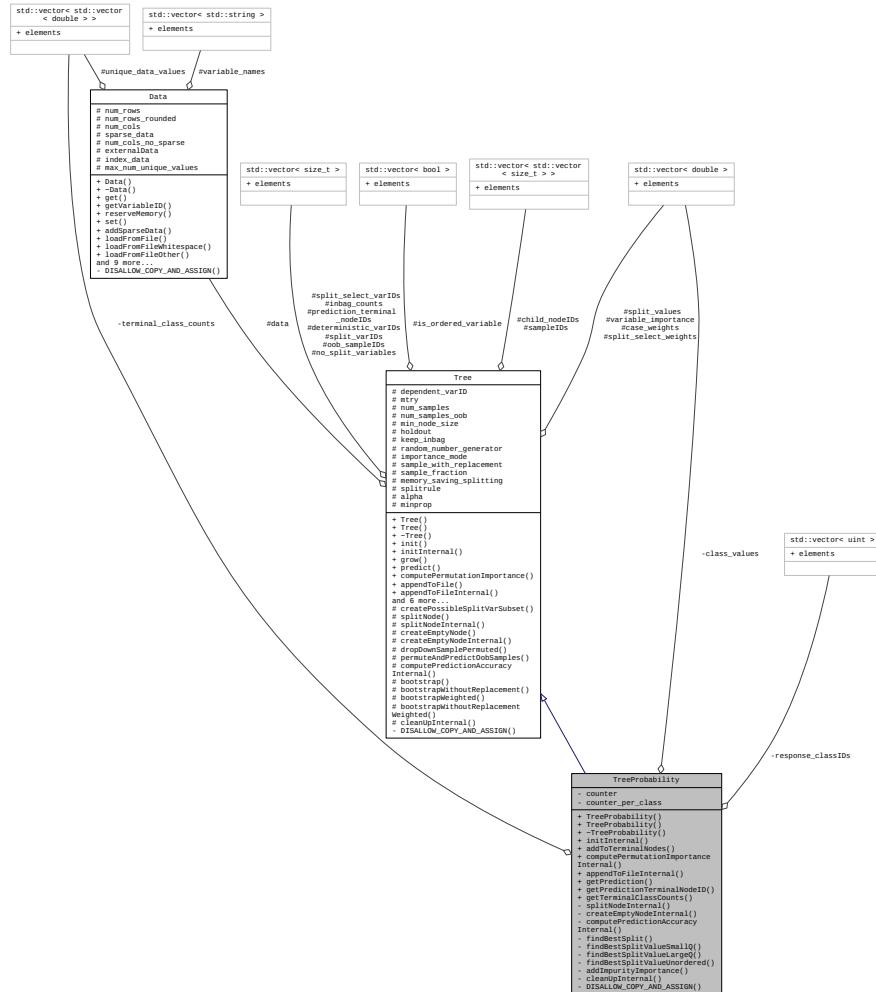
6.131 TreeProbability Class Reference

```
#include <TreeProbability.h>
```

Inheritance diagram for TreeProbability:



Collaboration diagram for TreeProbability:



Public Member Functions

- `TreeProbability` (`std::vector< double > *class_values`, `std::vector< uint > *response_classIDs`)
 - `TreeProbability` (`std::vector< std::vector< size_t >> &child_nodeIDs`, `std::vector< size_t > &split_varIDs`, `std::vector< double > &split_values`, `std::vector< double > *class_values`, `std::vector< uint > *response_classIDs`, `std::vector< std::vector< double >> &terminal_class_counts`, `std::vector< bool > *is_ordered_variable`)
 - virtual `~TreeProbability` ()
 - void `initInternal` ()
 - void `addToTerminalNodes` (`size_t nodeID`)
 - void `computePermutationImportanceInternal` (`std::vector< std::vector< size_t >> *permutations`)
 - void `appendToFileInternal` (`std::ofstream &file`)
 - const `std::vector< double > &getPrediction` (`size_t sampleID`) const
 - `size_t getPredictionTerminalNodeID` (`size_t sampleID`) const
 - const `std::vector< std::vector< double >> &getTerminalClassCounts` () const

Private Member Functions

- bool `splitNodeInternal` (size_t nodeID, std::vector< size_t > &possible_split_varIDs)
- void `createEmptyNodeInternal` ()
- double `computePredictionAccuracyInternal` ()
- bool `findBestSplit` (size_t nodeID, std::vector< size_t > &possible_split_varIDs)
- void `findBestSplitValueSmallQ` (size_t nodeID, size_t varID, size_t num_classes, size_t *class_counts, size_t num_samples_node, double &best_value, size_t &best_varID, double &best_decrease)
- void `findBestSplitValueLargeQ` (size_t nodeID, size_t varID, size_t num_classes, size_t *class_counts, size_t num_samples_node, double &best_value, size_t &best_varID, double &best_decrease)
- void `findBestSplitValueUnordered` (size_t nodeID, size_t varID, size_t num_classes, size_t *class_counts, size_t num_samples_node, double &best_value, size_t &best_varID, double &best_decrease)
- void `addImpurityImportance` (size_t nodeID, size_t varID, double decrease)
- void `cleanUpInternal` ()
- `DISALLOW_COPY_AND_ASSIGN` (`TreeProbability`)

Private Attributes

- std::vector< double > * `class_values`
- std::vector< uint > * `response_classIDs`
- std::vector< std::vector< double > > `terminal_class_counts`
- size_t * `counter`
- size_t * `counter_per_class`

Additional Inherited Members

6.131.1 Detailed Description

Definition at line 37 of file `TreeProbability.h`.

6.131.2 Constructor & Destructor Documentation

6.131.2.1 `TreeProbability()` [1/2]

```
TreeProbability::TreeProbability (
    std::vector< double > * class_values,
    std::vector< uint > * response_classIDs )
```

Definition at line 33 of file `TreeProbability.cpp`.

6.131.2.2 TreeProbability() [2/2]

```
TreeProbability::TreeProbability (
    std::vector< std::vector< size_t >> & child_nodeIDs,
    std::vector< size_t > & split_varIDs,
    std::vector< double > & split_values,
    std::vector< double > * class_values,
    std::vector< uint > * response_classIDs,
    std::vector< std::vector< double >> & terminal_class_counts,
    std::vector< bool > * is_ordered_variable )
```

Definition at line 37 of file TreeProbability.cpp.

6.131.2.3 ~TreeProbability()

```
TreeProbability::~TreeProbability ( ) [virtual]
```

Definition at line 44 of file TreeProbability.cpp.

6.131.3 Member Function Documentation**6.131.3.1 addImpurityImportance()**

```
void TreeProbability::addImpurityImportance (
    size_t nodeID,
    size_t varID,
    double decrease ) [private]
```

Definition at line 417 of file TreeProbability.cpp.

6.131.3.2 addToTerminalNodes()

```
void TreeProbability::addToTerminalNodes (
    size_t nodeID )
```

Definition at line 58 of file TreeProbability.cpp.

6.131.3.3 appendToFileInternal()

```
void TreeProbability::appendToFileInternal (
    std::ofstream & file ) [virtual]
```

Implements [Tree](#).

Definition at line 76 of file TreeProbability.cpp.

6.131.3.4 cleanUpInternal()

```
void TreeProbability::cleanUpInternal () [inline], [private], [virtual]
```

Implements [Tree](#).

Definition at line 84 of file TreeProbability.h.

6.131.3.5 computePermutationImportanceInternal()

```
void TreeProbability::computePermutationImportanceInternal (
    std::vector< std::vector< size_t >> * permutations )
```

6.131.3.6 computePredictionAccuracyInternal()

```
double TreeProbability::computePredictionAccuracyInternal () [private], [virtual]
```

Implements [Tree](#).

Definition at line 130 of file TreeProbability.cpp.

6.131.3.7 createEmptyNodeInternal()

```
void TreeProbability::createEmptyNodeInternal () [private], [virtual]
```

Implements [Tree](#).

Definition at line 126 of file TreeProbability.cpp.

6.131.3.8 DISALLOW_COPY_AND_ASSIGN()

```
TreeProbability::DISALLOW_COPY_AND_ASSIGN (
    TreeProbability ) [private]
```

6.131.3.9 findBestSplit()

```
bool TreeProbability::findBestSplit (
    size_t nodeID,
    std::vector< size_t > & possible_split_varIDs ) [private]
```

Definition at line 144 of file TreeProbability.cpp.

6.131.3.10 findBestSplitValueLargeQ()

```
void TreeProbability::findBestSplitValueLargeQ (
    size_t nodeID,
    size_t varID,
    size_t num_classes,
    size_t * class_counts,
    size_t num_samples_node,
    double & best_value,
    size_t & best_varID,
    double & best_decrease ) [private]
```

Definition at line 286 of file TreeProbability.cpp.

6.131.3.11 findBestSplitValueSmallQ()

```
void TreeProbability::findBestSplitValueSmallQ (
    size_t nodeID,
    size_t varID,
    size_t num_classes,
    size_t * class_counts,
    size_t num_samples_node,
    double & best_value,
    size_t & best_varID,
    double & best_decrease ) [private]
```

Definition at line 204 of file TreeProbability.cpp.

6.131.3.12 findBestSplitValueUnordered()

```
void TreeProbability::findBestSplitValueUnordered (
    size_t nodeID,
    size_t varID,
    size_t num_classes,
    size_t * class_counts,
    size_t num_samples_node,
    double & best_value,
    size_t & best_varID,
    double & best_decrease ) [private]
```

Definition at line 345 of file TreeProbability.cpp.

6.131.3.13 getPrediction()

```
const std::vector<double>& TreeProbability::getPrediction (
    size_t sampleID ) const [inline]
```

Definition at line 54 of file TreeProbability.h.

6.131.3.14 getPredictionTerminalNodeID()

```
size_t TreeProbability::getPredictionTerminalNodeID (
    size_t sampleID ) const [inline]
```

Definition at line 59 of file TreeProbability.h.

6.131.3.15 getTerminalClassCounts()

```
const std::vector<std::vector<double> >& TreeProbability::getTerminalClassCounts ( ) const [inline]
```

Definition at line 63 of file TreeProbability.h.

6.131.3.16 initInternal()

```
void TreeProbability::initInternal ( ) [virtual]
```

Implements [Tree](#).

Definition at line 48 of file TreeProbability.cpp.

6.131.3.17 splitNodeInternal()

```
bool TreeProbability::splitNodeInternal (
    size_t nodeID,
    std::vector< size_t > & possible_split_varIDs ) [private], [virtual]
```

Implements [Tree](#).

Definition at line 92 of file TreeProbability.cpp.

6.131.4 Member Data Documentation

6.131.4.1 class_values

```
std::vector<double>* TreeProbability::class_values [private]
```

Definition at line 94 of file TreeProbability.h.

6.131.4.2 counter

```
size_t* TreeProbability::counter [private]
```

Definition at line 100 of file TreeProbability.h.

6.131.4.3 counter_per_class

```
size_t* TreeProbability::counter_per_class [private]
```

Definition at line 101 of file TreeProbability.h.

6.131.4.4 response_classIDs

```
std::vector<uint>* TreeProbability::response_classIDs [private]
```

Definition at line 95 of file TreeProbability.h.

6.131.4.5 terminal_class_counts

```
std::vector<std::vector<double> > TreeProbability::terminal_class_counts [private]
```

Definition at line 98 of file TreeProbability.h.

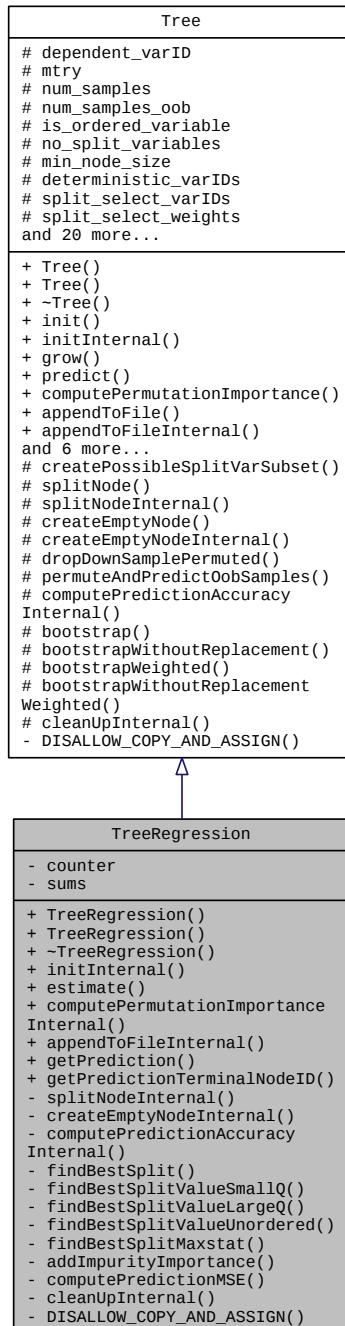
The documentation for this class was generated from the following files:

- src/[TreeProbability.h](#)
- src/[TreeProbability.cpp](#)

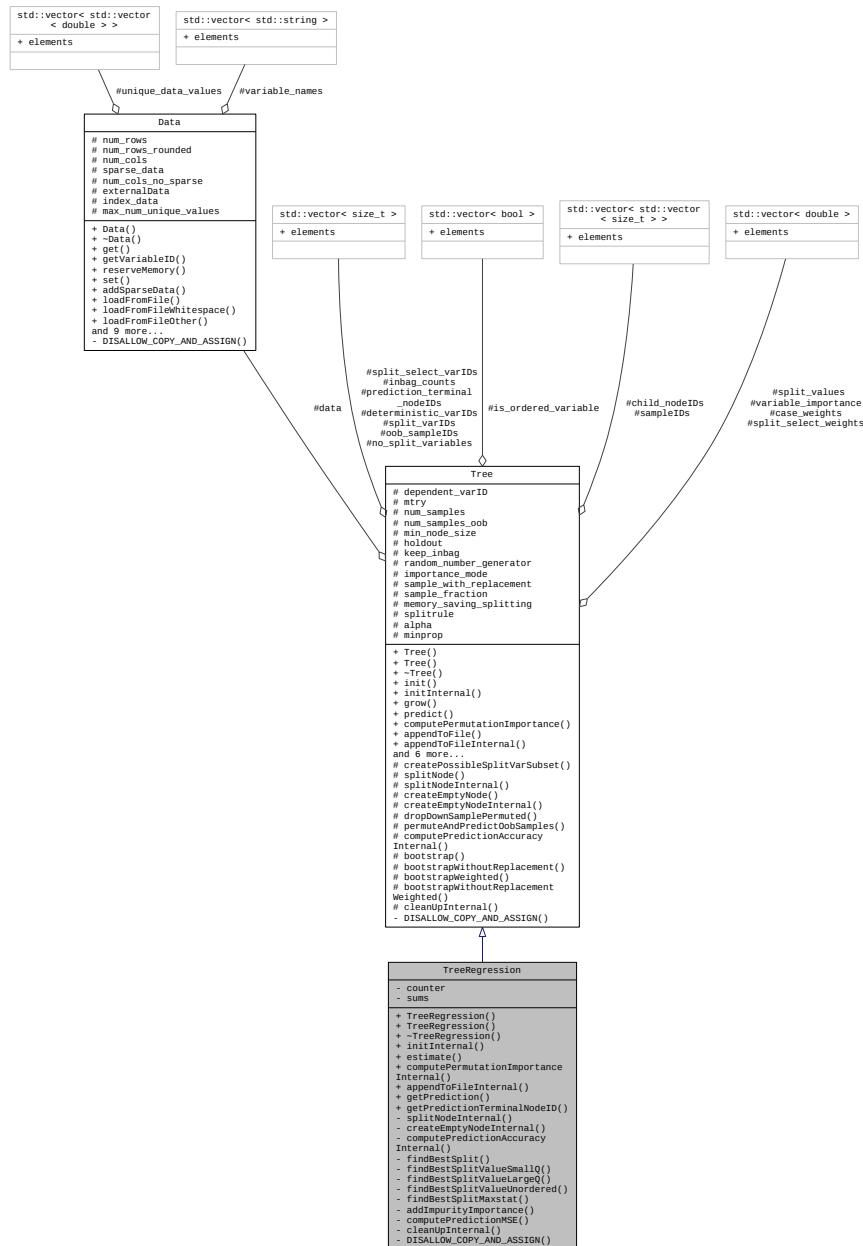
6.132 TreeRegression Class Reference

```
#include <TreeRegression.h>
```

Inheritance diagram for TreeRegression:



Collaboration diagram for TreeRegression:



Public Member Functions

- [TreeRegression \(\)](#)
- [TreeRegression \(std::vector<std::vector<size_t>>& child_nodeIDs, std::vector<size_t>& split_varIDs, std::vector<double>& split_values, std::vector<bool>* is_ordered_variable\)](#)
- [virtual ~TreeRegression \(\)](#)
- [void initInternal \(\)](#)

- double [estimate](#) (size_t nodeID)
- void [computePermutationImportanceInternal](#) (std::vector< std::vector< size_t >> *permutations)
- void [appendToFileInternal](#) (std::ofstream &file)
- double [getPrediction](#) (size_t sampleID) const
- size_t [getPredictionTerminalNodeID](#) (size_t sampleID) const

Private Member Functions

- bool [splitNodeInternal](#) (size_t nodeID, std::vector< size_t > &possible_split_varIDs)
- void [createEmptyNodeInternal](#) ()
- double [computePredictionAccuracyInternal](#) ()
- bool [findBestSplit](#) (size_t nodeID, std::vector< size_t > &possible_split_varIDs)
- void [findBestSplitValueSmallQ](#) (size_t nodeID, size_t varID, double sum_node, size_t num_samples_node, double &best_value, size_t &best_varID, double &best_decrease)
- void [findBestSplitValueLargeQ](#) (size_t nodeID, size_t varID, double sum_node, size_t num_samples_node, double &best_value, size_t &best_varID, double &best_decrease)
- void [findBestSplitValueUnordered](#) (size_t nodeID, size_t varID, double sum_node, size_t num_samples_node, double &best_value, size_t &best_varID, double &best_decrease)
- bool [findBestSplitMaxstat](#) (size_t nodeID, std::vector< size_t > &possible_split_varIDs)
- void [addImpurityImportance](#) (size_t nodeID, size_t varID, double decrease)
- double [computePredictionMSE](#) ()
- void [cleanUpInternal](#) ()
- [DISALLOW_COPY_AND_ASSIGN](#) (TreeRegression)

Private Attributes

- size_t * [counter](#)
- double * [sums](#)

Additional Inherited Members

6.132.1 Detailed Description

Definition at line 35 of file TreeRegression.h.

6.132.2 Constructor & Destructor Documentation

6.132.2.1 TreeRegression() [1/2]

TreeRegression::TreeRegression ()

Definition at line 40 of file TreeRegression.cpp.

6.132.2.2 TreeRegression() [2/2]

```
TreeRegression::TreeRegression (
    std::vector< std::vector< size_t >> & child_nodeIDs,
    std::vector< size_t > & split_varIDs,
    std::vector< double > & split_values,
    std::vector< bool > * is_ordered_variable )
```

Definition at line 44 of file TreeRegression.cpp.

6.132.2.3 ~TreeRegression()

```
TreeRegression::~TreeRegression ( ) [virtual]
```

Definition at line 49 of file TreeRegression.cpp.

6.132.3 Member Function Documentation

6.132.3.1 addImpurityImportance()

```
void TreeRegression::addImpurityImportance (
    size_t nodeID,
    size_t varID,
    double decrease ) [private]
```

Definition at line 459 of file TreeRegression.cpp.

6.132.3.2 appendToFileInternal()

```
void TreeRegression::appendToFileInternal (
    std::ofstream & file ) [virtual]
```

Implements [Tree](#).

Definition at line 73 of file TreeRegression.cpp.

6.132.3.3 cleanUpInternal()

```
void TreeRegression::cleanUpInternal ( ) [inline], [private], [virtual]
```

Implements [Tree](#).

Definition at line 80 of file TreeRegression.h.

6.132.3.4 computePermutationImportanceInternal()

```
void TreeRegression::computePermutationImportanceInternal (
    std::vector< std::vector< size_t >> * permutations )
```

6.132.3.5 computePredictionAccuracyInternal()

```
double TreeRegression::computePredictionAccuracyInternal ( ) [private], [virtual]
```

Implements [Tree](#).

Definition at line 121 of file TreeRegression.cpp.

6.132.3.6 computePredictionMSE()

```
double TreeRegression::computePredictionMSE ( ) [private]
```

6.132.3.7 createEmptyNodeInternal()

```
void TreeRegression::createEmptyNodeInternal ( ) [private], [virtual]
```

Implements [Tree](#).

Definition at line 117 of file TreeRegression.cpp.

6.132.3.8 `DISALLOW_COPY_AND_ASSIGN()`

```
TreeRegression::DISALLOW_COPY_AND_ASSIGN (
    TreeRegression ) [private]
```

6.132.3.9 `estimate()`

```
double TreeRegression::estimate (
    size_t nodeID )
```

Definition at line 62 of file `TreeRegression.cpp`.

6.132.3.10 `findBestSplit()`

```
bool TreeRegression::findBestSplit (
    size_t nodeID,
    std::vector< size_t > & possible_split_varIDs ) [private]
```

Definition at line 136 of file `TreeRegression.cpp`.

6.132.3.11 `findBestSplitMaxstat()`

```
bool TreeRegression::findBestSplitMaxstat (
    size_t nodeID,
    std::vector< size_t > & possible_split_varIDs ) [private]
```

Definition at line 369 of file `TreeRegression.cpp`.

6.132.3.12 `findBestSplitValueLargeQ()`

```
void TreeRegression::findBestSplitValueLargeQ (
    size_t nodeID,
    size_t varID,
    double sum_node,
    size_t num_samples_node,
    double & best_value,
    size_t & best_varID,
    double & best_decrease ) [private]
```

Definition at line 260 of file `TreeRegression.cpp`.

6.132.3.13 findBestSplitValueSmallQ()

```
void TreeRegression::findBestSplitValueSmallQ (
    size_t nodeID,
    size_t varID,
    double sum_node,
    size_t num_samples_node,
    double & best_value,
    size_t & best_varID,
    double & best_decrease ) [private]
```

Definition at line 188 of file TreeRegression.cpp.

6.132.3.14 findBestSplitValueUnordered()

```
void TreeRegression::findBestSplitValueUnordered (
    size_t nodeID,
    size_t varID,
    double sum_node,
    size_t num_samples_node,
    double & best_value,
    size_t & best_varID,
    double & best_decrease ) [private]
```

Definition at line 307 of file TreeRegression.cpp.

6.132.3.15 getPrediction()

```
double TreeRegression::getPrediction (
    size_t sampleID ) const [inline]
```

Definition at line 51 of file TreeRegression.h.

6.132.3.16 getPredictionTerminalNodeID()

```
size_t TreeRegression::getPredictionTerminalNodeID (
    size_t sampleID ) const [inline]
```

Definition at line 56 of file TreeRegression.h.

6.132.3.17 initInternal()

```
void TreeRegression::initInternal ( ) [virtual]
```

Implements [Tree](#).

Definition at line 53 of file [TreeRegression.cpp](#).

6.132.3.18 splitNodeInternal()

```
bool TreeRegression::splitNodeInternal (
    size_t nodeID,
    std::vector< size_t > & possible_split_varIDs ) [private], [virtual]
```

Implements [Tree](#).

Definition at line 77 of file [TreeRegression.cpp](#).

6.132.4 Member Data Documentation

6.132.4.1 counter

```
size_t* TreeRegression::counter [private]
```

Definition at line 89 of file [TreeRegression.h](#).

6.132.4.2 sums

```
double* TreeRegression::sums [private]
```

Definition at line 90 of file [TreeRegression.h](#).

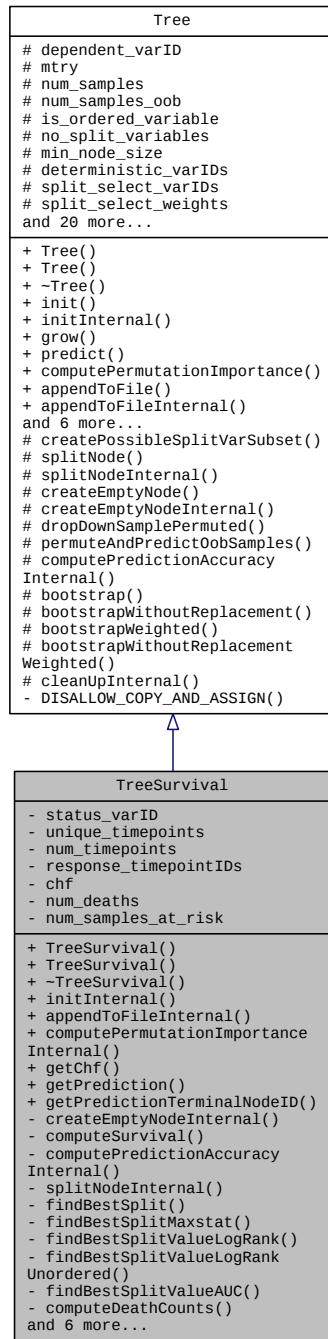
The documentation for this class was generated from the following files:

- [src/TreeRegression.h](#)
- [src/TreeRegression.cpp](#)

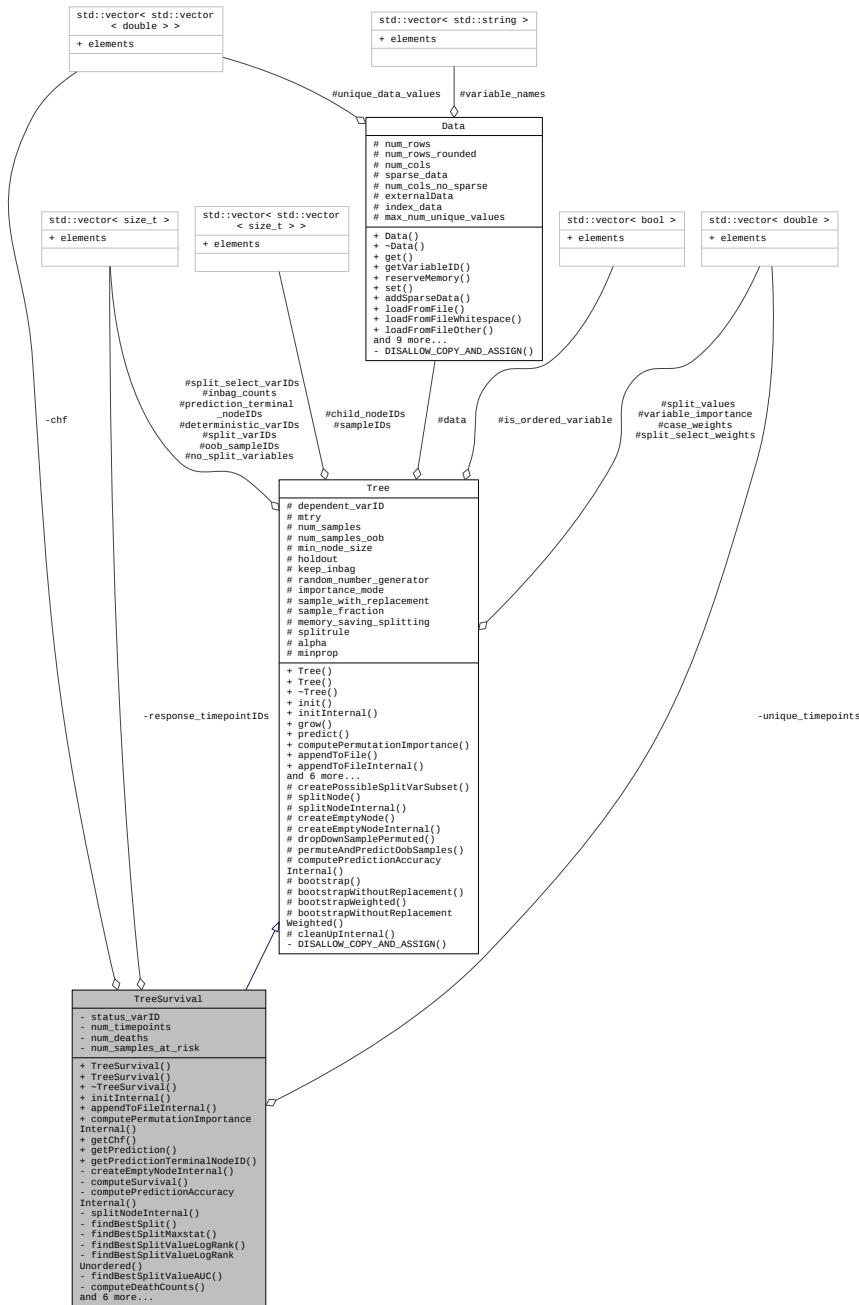
6.133 TreeSurvival Class Reference

```
#include <TreeSurvival.h>
```

Inheritance diagram for TreeSurvival:



Collaboration diagram for TreeSurvival:



Public Member Functions

- `TreeSurvival (std::vector< double > *unique_timepoints, size_t status_varID, std::vector< size_t > *response_timepointIDs)`
- `TreeSurvival (std::vector< std::vector< size_t > > &child_nodeIDs, std::vector< size_t > &split_varIDs, std::vector< double > &split_values, std::vector< std::vector< double > >&chf, std::vector< double > *unique_timepoints, std::vector< size_t > *response_timepointIDs, std::vector< bool > *is_ordered_variable)`

- virtual `~TreeSurvival ()`
- void `initInternal ()` override
- void `appendToFileInternal (std::ofstream &file)` override
- void `computePermutationImportanceInternal (std::vector< std::vector< size_t >> *permutations)`
- const std::vector< std::vector< double > > & `getChf ()` const
- const std::vector< double > & `getPrediction (size_t sampleID)` const
- size_t `getPredictionTerminalNodeID (size_t sampleID)` const

Private Member Functions

- void `createEmptyNodeInternal ()` override
- void `computeSurvival (size_t nodeID)`
- double `computePredictionAccuracyInternal ()` override
- bool `splitNodeInternal (size_t nodeID, std::vector< size_t > &possible_split_varIDs)` override
- bool `findBestSplit (size_t nodeID, std::vector< size_t > &possible_split_varIDs)`
- bool `findBestSplitMaxstat (size_t nodeID, std::vector< size_t > &possible_split_varIDs)`
- void `findBestSplitValueLogRank (size_t nodeID, size_t varID, std::vector< double > &possible_split_values, double &best_value, size_t &best_varID, double &best_logrank)`
- void `findBestSplitValueLogRankUnordered (size_t nodeID, size_t varID, std::vector< double > &factor_levels, double &best_value, size_t &best_varID, double &best_logrank)`
- void `findBestSplitValueAUC (size_t nodeID, size_t varID, double &best_value, size_t &best_varID, double &best_auc)`
- void `computeDeathCounts (size_t nodeID)`
- void `computeChildDeathCounts (size_t nodeID, size_t varID, std::vector< double > &possible_split_values, size_t *num_samples_right_child, size_t *num_samples_at_risk_right_child, size_t *num_deaths_right_child)`
- void `computeAucSplit (double time_k, double time_l, double status_k, double status_l, double value_k, double value_l, size_t num_splits, std::vector< double > &possible_split_values, double *num_count, double *num_total)`
- void `findBestSplitValueLogRank (size_t nodeID, size_t varID, double &best_value, size_t &best_varID, double &best_logrank)`
- void `findBestSplitValueLogRankUnordered (size_t nodeID, size_t varID, double &best_value, size_t &best_varID, double &best_logrank)`
- void `cleanUpInternal ()` override
- `DISALLOW_COPY_AND_ASSIGN (TreeSurvival)`

Private Attributes

- size_t `status_varID`
- std::vector< double > * `unique_timepoints`
- size_t `num_timepoints`
- std::vector< size_t > * `response_timepointIDs`
- std::vector< std::vector< double > > `chf`
- size_t * `num_deaths`
- size_t * `num_samples_at_risk`

Additional Inherited Members

6.133.1 Detailed Description

Definition at line 35 of file TreeSurvival.h.

6.133.2 Constructor & Destructor Documentation

6.133.2.1 TreeSurvival() [1/2]

```
TreeSurvival::TreeSurvival (
    std::vector< double > * unique_timepoints,
    size_t status_varID,
    std::vector< size_t > * response_timepointIDs )
```

Definition at line 40 of file TreeSurvival.cpp.

6.133.2.2 TreeSurvival() [2/2]

```
TreeSurvival::TreeSurvival (
    std::vector< std::vector< size_t >> & child_nodeIDs,
    std::vector< size_t > & split_varIDs,
    std::vector< double > & split_values,
    std::vector< std::vector< double >> chf,
    std::vector< double > * unique_timepoints,
    std::vector< size_t > * response_timepointIDs,
    std::vector< bool > * is_ordered_variable )
```

Definition at line 47 of file TreeSurvival.cpp.

6.133.2.3 ~TreeSurvival()

```
TreeSurvival::~TreeSurvival ( ) [virtual]
```

Definition at line 56 of file TreeSurvival.cpp.

6.133.3 Member Function Documentation

6.133.3.1 appendToFileInternal()

```
void TreeSurvival::appendToFileInternal (
    std::ofstream & file ) [override], [virtual]
```

Implements [Tree](#).

Definition at line 65 of file TreeSurvival.cpp.

6.133.3.2 cleanUpInternal()

```
void TreeSurvival::cleanUpInternal ( ) [inline], [override], [private], [virtual]
```

Implements [Tree](#).

Definition at line 94 of file TreeSurvival.h.

6.133.3.3 computeAucSplit()

```
void TreeSurvival::computeAucSplit (
    double time_k,
    double time_l,
    double status_k,
    double status_l,
    double value_k,
    double value_l,
    size_t num_splits,
    std::vector< double > & possible_split_values,
    double * num_count,
    double * num_total ) [private]
```

Definition at line 575 of file TreeSurvival.cpp.

6.133.3.4 computeChildDeathCounts()

```
void TreeSurvival::computeChildDeathCounts (
    size_t nodeID,
    size_t varID,
    std::vector< double > & possible_split_values,
    size_t * num_samples_right_child,
    size_t * num_samples_at_risk_right_child,
    size_t * num_deaths_right_child ) [private]
```

Definition at line 299 of file TreeSurvival.cpp.

6.133.3.5 computeDeathCounts()

```
void TreeSurvival::computeDeathCounts (
    size_t nodeID ) [private]
```

Definition at line 272 of file TreeSurvival.cpp.

6.133.3.6 computePermutationImportanceInternal()

```
void TreeSurvival::computePermutationImportanceInternal (
    std::vector< std::vector< size_t >> * permutations )
```

6.133.3.7 computePredictionAccuracyInternal()

```
double TreeSurvival::computePredictionAccuracyInternal ( ) [override], [private], [virtual]
```

Implements [Tree](#).

Definition at line 97 of file TreeSurvival.cpp.

6.133.3.8 computeSurvival()

```
void TreeSurvival::computeSurvival (
    size_t nodeID ) [private]
```

Definition at line 84 of file TreeSurvival.cpp.

6.133.3.9 createEmptyNodeInternal()

```
void TreeSurvival::createEmptyNodeInternal ( ) [override], [private], [virtual]
```

Implements [Tree](#).

Definition at line 80 of file TreeSurvival.cpp.

6.133.3.10 DISALLOW_COPY_AND_ASSIGN()

```
TreeSurvival::DISALLOW_COPY_AND_ASSIGN (
    TreeSurvival ) [private]
```

6.133.3.11 findBestSplit()

```
bool TreeSurvival::findBestSplit (
    size_t nodeID,
    std::vector< size_t > & possible_split_varIDs ) [private]
```

Definition at line 119 of file TreeSurvival.cpp.

6.133.3.12 findBestSplitMaxstat()

```
bool TreeSurvival::findBestSplitMaxstat (
    size_t nodeID,
    std::vector< size_t > & possible_split_varIDs ) [private]
```

Definition at line 160 of file TreeSurvival.cpp.

6.133.3.13 findBestSplitValueAUC()

```
void TreeSurvival::findBestSplitValueAUC (
    size_t nodeID,
    size_t varID,
    double & best_value,
    size_t & best_varID,
    double & best_auc ) [private]
```

Definition at line 498 of file TreeSurvival.cpp.

6.133.3.14 findBestSplitValueLogRank() [1/2]

```
void TreeSurvival::findBestSplitValueLogRank (
    size_t nodeID,
    size_t varID,
    std::vector< double > & possible_split_values,
    double & best_value,
    size_t & best_varID,
    double & best_logrank ) [private]
```

6.133.3.15 findBestSplitValueLogRank() [2/2]

```
void TreeSurvival::findBestSplitValueLogRank (
    size_t nodeID,
    size_t varID,
    double & best_value,
    size_t & best_varID,
    double & best_logrank ) [private]
```

Definition at line 324 of file TreeSurvival.cpp.

6.133.3.16 findBestSplitValueLogRankUnordered() [1/2]

```
void TreeSurvival::findBestSplitValueLogRankUnordered (
    size_t nodeID,
    size_t varID,
    std::vector< double > & factor_levels,
    double & best_value,
    size_t & best_varID,
    double & best_logrank ) [private]
```

6.133.3.17 findBestSplitValueLogRankUnordered() [2/2]

```
void TreeSurvival::findBestSplitValueLogRankUnordered (
    size_t nodeID,
    size_t varID,
    double & best_value,
    size_t & best_varID,
    double & best_logrank ) [private]
```

Definition at line 398 of file TreeSurvival.cpp.

6.133.3.18 getChf()

```
const std::vector<std::vector<double> >& TreeSurvival::getChf ( ) const [inline]
```

Definition at line 51 of file TreeSurvival.h.

6.133.3.19 getPrediction()

```
const std::vector<double>& TreeSurvival::getPrediction (
    size_t sampleID ) const [inline]
```

Definition at line 55 of file TreeSurvival.h.

6.133.3.20 getPredictionTerminalNodeID()

```
size_t TreeSurvival::getPredictionTerminalNodeID (
    size_t sampleID ) const [inline]
```

Definition at line 60 of file TreeSurvival.h.

6.133.3.21 initInternal()

```
void TreeSurvival::initInternal ( ) [override], [virtual]
```

Implements [Tree](#).

Definition at line 59 of file TreeSurvival.cpp.

6.133.3.22 splitNodeInternal()

```
bool TreeSurvival::splitNodeInternal (
    size_t nodeID,
    std::vector< size_t > & possible_split_varIDs ) [override], [private], [virtual]
```

Implements [Tree](#).

Definition at line 110 of file TreeSurvival.cpp.

6.133.4 Member Data Documentation

6.133.4.1 chf

```
std::vector<std::vector<double> > TreeSurvival::chf [private]
```

Definition at line 107 of file TreeSurvival.h.

6.133.4.2 num_deaths

```
size_t* TreeSurvival::num_deaths [private]
```

Definition at line 110 of file TreeSurvival.h.

6.133.4.3 num_samples_at_risk

```
size_t* TreeSurvival::num_samples_at_risk [private]
```

Definition at line 111 of file TreeSurvival.h.

6.133.4.4 num_timepoints

```
size_t TreeSurvival::num_timepoints [private]
```

Definition at line 103 of file TreeSurvival.h.

6.133.4.5 response_timepointIDs

```
std::vector<size_t*>* TreeSurvival::response_timepointIDs [private]
```

Definition at line 104 of file TreeSurvival.h.

6.133.4.6 status_varID

```
size_t TreeSurvival::status_varID [private]
```

Definition at line 99 of file TreeSurvival.h.

6.133.4.7 unique_timepoints

```
std::vector<double>* TreeSurvival::unique_timepoints [private]
```

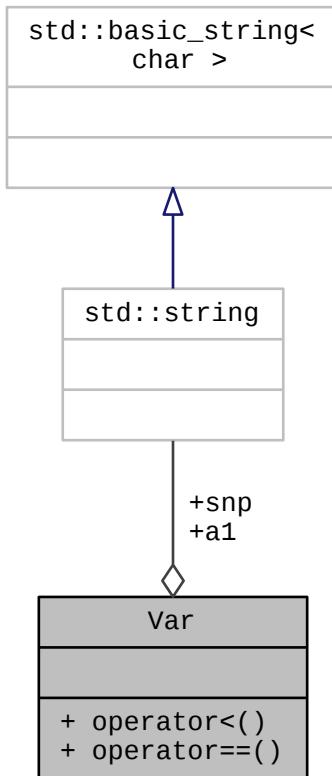
Definition at line 102 of file TreeSurvival.h.

The documentation for this class was generated from the following files:

- src/[TreeSurvival.h](#)
- src/[TreeSurvival.cpp](#)

6.134 Var Class Reference

Collaboration diagram for Var:



Public Member Functions

- bool `operator<` (const `Var &b`) const
- bool `operator==` (const `Var &b`) const

Public Attributes

- string `snp`
- string `a1`

6.134.1 Detailed Description

Definition at line 34 of file dosage.cpp.

6.134.2 Member Function Documentation

6.134.2.1 operator<()

```
bool Var::operator< (
    const Var & b ) const [inline]
```

Definition at line 38 of file dosage.cpp.

6.134.2.2 operator==()

```
bool Var::operator== (
    const Var & b ) const [inline]
```

Definition at line 42 of file dosage.cpp.

6.134.3 Member Data Documentation

6.134.3.1 a1

```
string Var::a1
```

Definition at line 37 of file dosage.cpp.

6.134.3.2 snp

```
string Var::snp
```

Definition at line 36 of file dosage.cpp.

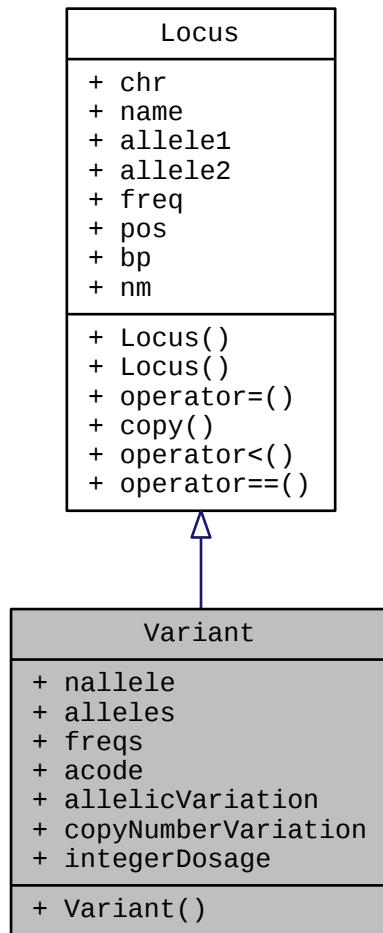
The documentation for this class was generated from the following file:

- [src/dosage.cpp](#)

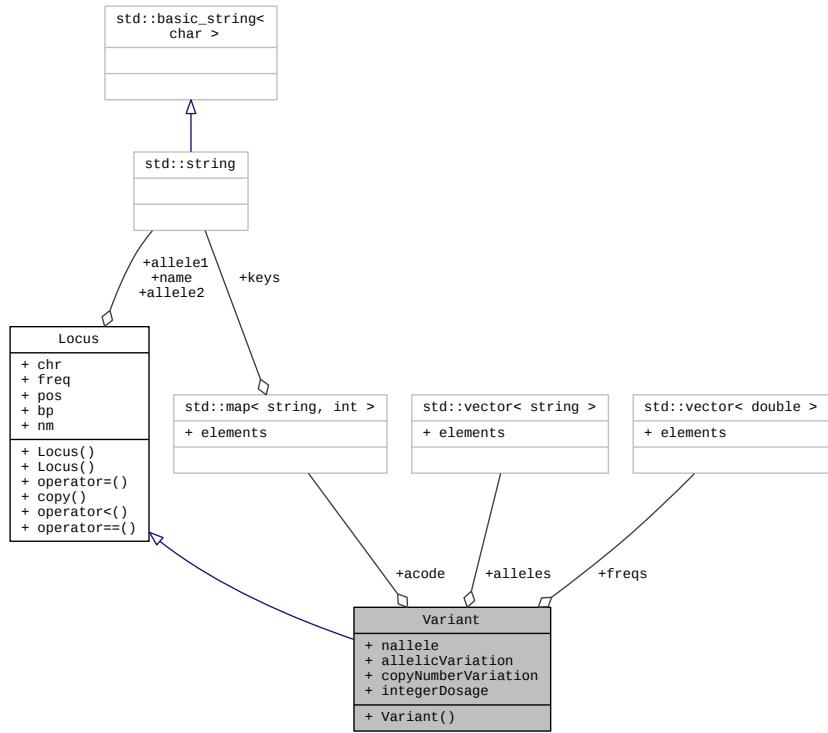
6.135 Variant Class Reference

```
#include <gvar.h>
```

Inheritance diagram for Variant:



Collaboration diagram for Variant:



Public Member Functions

- `Variant ()`

Public Attributes

- int `nallele`
- `vector<string>` `alleles`
- `vector_t` `freqs`
- `map<string, int>` `acode`
- bool `allelicVariation`
- bool `copyNumberVariation`
- bool `integerDosage`

6.135.1 Detailed Description

Definition at line 23 of file gvar.h.

6.135.2 Constructor & Destructor Documentation

6.135.2.1 Variant()

```
Variant::Variant ( ) [inline]
```

Definition at line 35 of file gvar.h.

6.135.3 Member Data Documentation

6.135.3.1 acode

```
map<string,int> Variant::acode
```

Definition at line 30 of file gvar.h.

6.135.3.2 alleles

```
vector<string> Variant::alleles
```

Definition at line 28 of file gvar.h.

6.135.3.3 allelicVariation

```
bool Variant::allelicVariation
```

Definition at line 32 of file gvar.h.

6.135.3.4 copyNumberVariation

```
bool Variant::copyNumberVariation
```

Definition at line 33 of file gvar.h.

6.135.3.5 freqs

```
vector_t Variant::freqs
```

Definition at line 29 of file gvar.h.

6.135.3.6 integerDosage

```
bool Variant::integerDosage
```

Definition at line 34 of file gvar.h.

6.135.3.7 nallele

```
int Variant::nallele
```

Definition at line 27 of file gvar.h.

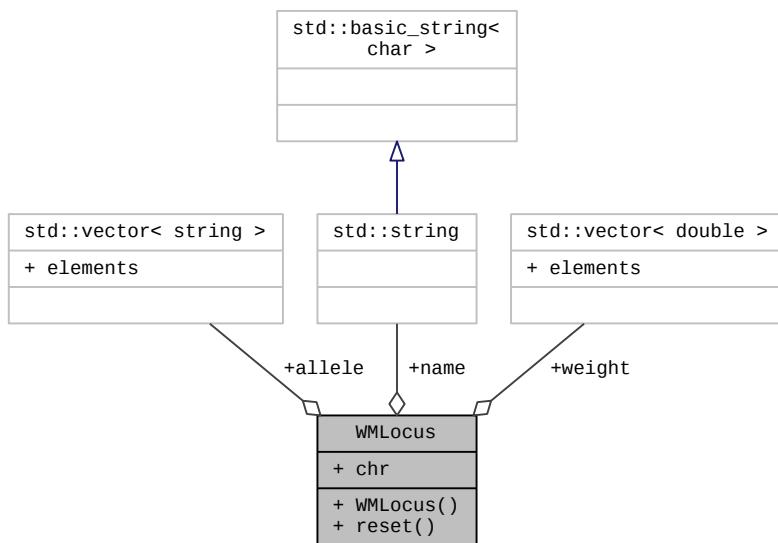
The documentation for this class was generated from the following file:

- [src/gvar.h](#)

6.136 WMLocus Class Reference

```
#include <plink.h>
```

Collaboration diagram for WMLocus:



Public Member Functions

- [WMLocus \(\)](#)
- [void reset \(\)](#)

Public Attributes

- [int chr](#)
- [string name](#)
- [vector< string > allele](#)
- [vector< double > weight](#)

6.136.1 Detailed Description

Definition at line 199 of file plink.h.

6.136.2 Constructor & Destructor Documentation

6.136.2.1 WMLocus()

```
WMLocus::WMLocus ( ) [inline]
```

Definition at line 202 of file plink.h.

6.136.3 Member Function Documentation

6.136.3.1 reset()

```
void WMLocus::reset ( ) [inline]
```

Definition at line 207 of file plink.h.

6.136.4 Member Data Documentation

6.136.4.1 allele

```
vector<string> WMLocus::allele
```

Definition at line 215 of file plink.h.

6.136.4.2 chr

```
int WMLocus::chr
```

Definition at line 212 of file plink.h.

6.136.4.3 name

```
string WMLocus::name
```

Definition at line 213 of file plink.h.

6.136.4.4 weight

```
vector<double> WMLocus::weight
```

Definition at line 216 of file plink.h.

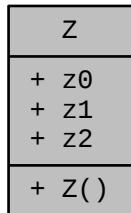
The documentation for this class was generated from the following file:

- [src/plink.h](#)

6.137 Z Class Reference

```
#include <plink.h>
```

Collaboration diagram for Z:



Public Member Functions

- [Z \(\)](#)

Public Attributes

- double [z0](#)
- double [z1](#)
- double [z2](#)

6.137.1 Detailed Description

Definition at line 491 of file plink.h.

6.137.2 Constructor & Destructor Documentation

6.137.2.1 Z()

```
Z::Z ( ) [inline]
```

Definition at line 494 of file plink.h.

6.137.3 Member Data Documentation

6.137.3.1 z0

```
double Z::z0
```

Definition at line 498 of file plink.h.

6.137.3.2 z1

```
double Z::z1
```

Definition at line 499 of file plink.h.

6.137.3.3 z2

```
double z::z2
```

Definition at line 500 of file plink.h.

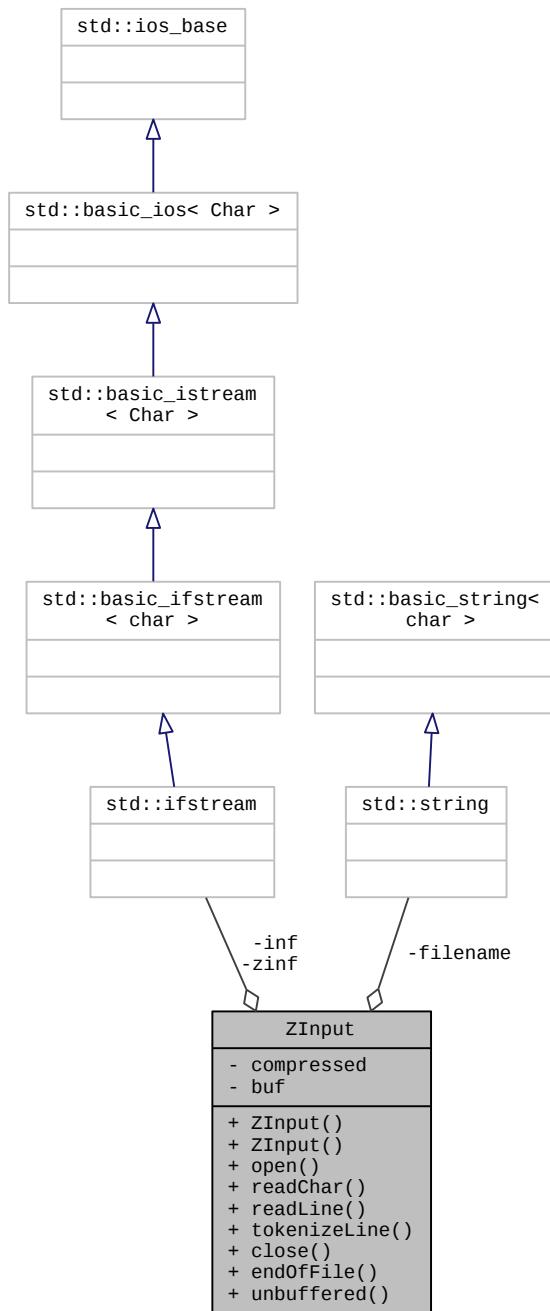
The documentation for this class was generated from the following file:

- src/[plink.h](#)

6.138 ZInput Class Reference

```
#include <zed.h>
```

Collaboration diagram for ZInput:



Public Member Functions

- `ZInput (string, bool)`

- `ZInput()`
- `void open(string, bool)`
- `char readChar()`
- `string readLine()`
- `vector<string> tokenizeLine()`
- `void close()`
- `bool endOfFile()`
- `void unbuffered()`

Private Attributes

- `string filename`
- `bool compressed`
- `char buf[MAX_LINE_LENGTH]`
- `ifstream zinf`
- `ifstream inf`

6.138.1 Detailed Description

Definition at line 19 of file zed.h.

6.138.2 Constructor & Destructor Documentation

6.138.2.1 `ZInput()` [1/2]

```
ZInput::ZInput (
    string f,
    bool cmode )
```

Definition at line 24 of file zed.cpp.

6.138.2.2 `ZInput()` [2/2]

```
ZInput::ZInput ( )
```

Definition at line 29 of file zed.cpp.

6.138.3 Member Function Documentation

6.138.3.1 close()

```
void ZInput::close ( )
```

Definition at line 105 of file zed.cpp.

6.138.3.2 endOfFile()

```
bool ZInput::endOfFile ( )
```

Definition at line 113 of file zed.cpp.

6.138.3.3 open()

```
void ZInput::open (
    string f,
    bool cmode )
```

Definition at line 35 of file zed.cpp.

6.138.3.4 readChar()

```
char ZInput::readChar ( )
```

Definition at line 80 of file zed.cpp.

6.138.3.5 readLine()

```
string ZInput::readLine ( )
```

Definition at line 61 of file zed.cpp.

6.138.3.6 tokenizeLine()

```
vector< string > ZInput::tokenizeLine ( )
```

Definition at line 94 of file zed.cpp.

6.138.3.7 unbuffered()

```
void ZInput::unbuffered ( )
```

Definition at line 124 of file zed.cpp.

6.138.4 Member Data Documentation

6.138.4.1 buf

```
char ZInput::buf[MAX_LINE_LENGTH] [private]
```

Definition at line 23 of file zed.h.

6.138.4.2 compressed

```
bool ZInput::compressed [private]
```

Definition at line 22 of file zed.h.

6.138.4.3 filename

```
string ZInput::filename [private]
```

Definition at line 21 of file zed.h.

6.138.4.4 inf

```
ifstream ZInput::inf [private]
```

Definition at line 31 of file zed.h.

6.138.4.5 zinf

```
ifstream ZInput::zinf [private]
```

Definition at line 28 of file zed.h.

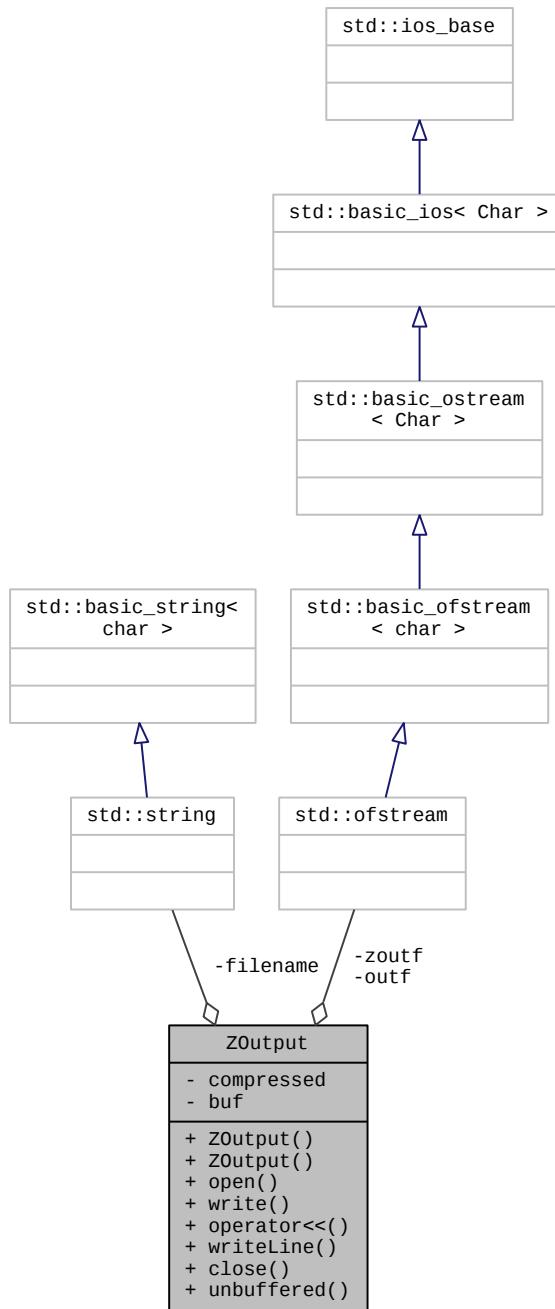
The documentation for this class was generated from the following files:

- src/[zed.h](#)
- src/[zed.cpp](#)

6.139 ZOutput Class Reference

```
#include <zed.h>
```

Collaboration diagram for ZOutput:



Public Member Functions

- `ZOutput (string, bool)`

- `ZOutput()`
- `void open(string, bool)`
- `void write(string)`
- `ZOutput & operator<<(const string &s)`
- `void writeLine(string)`
- `void close()`
- `void unbuffered()`

Private Attributes

- `ofstream zoutf`
- `ofstream outf`
- `string filename`
- `bool compressed`
- `char buf[MAX_LINE_LENGTH]`

6.139.1 Detailed Description

Definition at line 45 of file zed.h.

6.139.2 Constructor & Destructor Documentation

6.139.2.1 `ZOutput()` [1/2]

```
ZOutput::ZOutput (
    string f,
    bool cmode )
```

Definition at line 161 of file zed.cpp.

6.139.2.2 `ZOutput()` [2/2]

```
ZOutput::ZOutput ( )
```

Definition at line 166 of file zed.cpp.

6.139.3 Member Function Documentation

6.139.3.1 close()

```
void ZOutput::close ( )
```

Definition at line 187 of file zed.cpp.

6.139.3.2 open()

```
void ZOutput::open (
    string f,
    bool cmode )
```

Definition at line 130 of file zed.cpp.

6.139.3.3 operator<<()

```
ZOutput& ZOutput::operator<< (
    const string & s ) [inline]
```

Definition at line 64 of file zed.h.

6.139.3.4 unbuffered()

```
void ZOutput::unbuffered ( )
```

Definition at line 195 of file zed.cpp.

6.139.3.5 write()

```
void ZOutput::write (
    string s )
```

Definition at line 171 of file zed.cpp.

6.139.3.6 writeLine()

```
void ZOutput::writeLine (
    string s )
```

Definition at line 179 of file zed.cpp.

6.139.4 Member Data Documentation

6.139.4.1 buf

```
char ZOutput::buf[MAX_LINE_LENGTH] [private]
```

Definition at line 58 of file zed.h.

6.139.4.2 compressed

```
bool ZOutput::compressed [private]
```

Definition at line 57 of file zed.h.

6.139.4.3 filename

```
string ZOutput::filename [private]
```

Definition at line 56 of file zed.h.

6.139.4.4 outf

```
ofstream ZOutput::outf [private]
```

Definition at line 54 of file zed.h.

6.139.4.5 zoutf

```
ofstream ZOutput::zoutf [private]
```

Definition at line 51 of file zed.h.

The documentation for this class was generated from the following files:

- src/[zed.h](#)
- src/[zed.cpp](#)

6.140 ZZ Class Reference

```
#include <plink.h>
```

Collaboration diagram for ZZ:

ZZ
+ z00 + z10 + z20 + z01 + z11 + z21 + z02 + z12 + z22 + ZZ()

Public Member Functions

- [ZZ \(\)](#)

Public Attributes

- double [z00](#)
- double [z10](#)
- double [z20](#)
- double [z01](#)
- double [z11](#)
- double [z21](#)
- double [z02](#)
- double [z12](#)
- double [z22](#)

6.140.1 Detailed Description

Definition at line 558 of file plink.h.

6.140.2 Constructor & Destructor Documentation

6.140.2.1 ZZ()

```
ZZ::ZZ () [inline]
```

Definition at line 561 of file plink.h.

6.140.3 Member Data Documentation

6.140.3.1 z00

```
double ZZ::z00
```

Definition at line 566 of file plink.h.

6.140.3.2 z01

```
double ZZ::z01
```

Definition at line 569 of file plink.h.

6.140.3.3 z02

```
double ZZ::z02
```

Definition at line 572 of file plink.h.

6.140.3.4 z10

```
double ZZ::z10
```

Definition at line 567 of file plink.h.

6.140.3.5 z11

```
double ZZ::z11
```

Definition at line 570 of file plink.h.

6.140.3.6 z12

```
double ZZ::z12
```

Definition at line 573 of file plink.h.

6.140.3.7 z20

```
double ZZ::z20
```

Definition at line 568 of file plink.h.

6.140.3.8 z21

```
double ZZ::z21
```

Definition at line 571 of file plink.h.

6.140.3.9 z22

```
double ZZ::z22
```

Definition at line 574 of file plink.h.

The documentation for this class was generated from the following file:

- [src/plink.h](#)

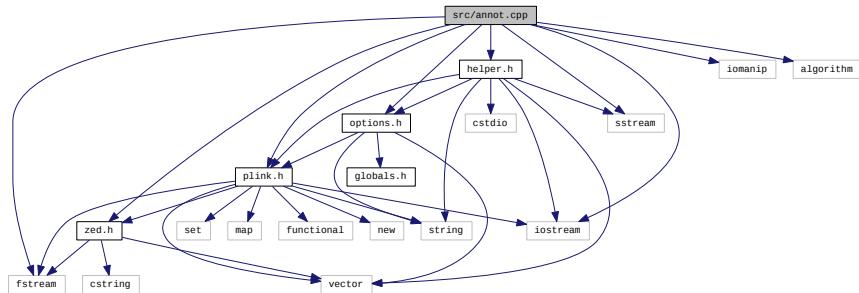
Chapter 7

File Documentation

7.1 src/annot.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <algorithm>
#include "options.h"
#include "plink.h"
#include "helper.h"
#include "zed.h"
```

Include dependency graph for annot.cpp:



Functions

- `map< string, set< Range > > filterRanges (map< string, set< Range > > &ranges, string filename)`

Variables

- `Plink * PP`

7.1.1 Function Documentation

7.1.1.1 filterRanges()

```
map<string, set<Range>> filterRanges (
    map< string, set< Range >> & ranges,
    string filename )
```

Definition at line 3132 of file helper.cpp.

7.1.2 Variable Documentation

7.1.2.1 PP

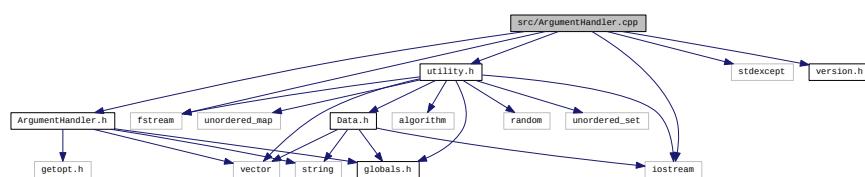
[Plink*](#) PP

Definition at line 85 of file inbix.cpp.

7.2 src/ArgumentHandler.cpp File Reference

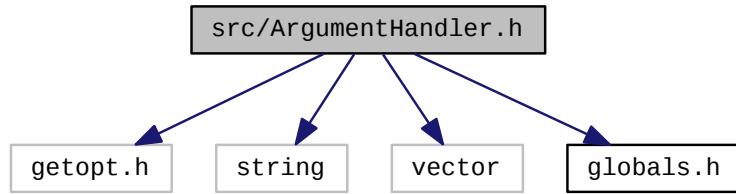
```
#include <iostream>
#include <fstream>
#include <stdexcept>
#include "ArgumentHandler.h"
#include "version.h"
#include "utility.h"
```

Include dependency graph for ArgumentHandler.cpp:

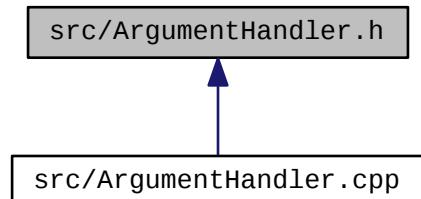


7.3 src/ArgumentHandler.h File Reference

```
#include <getopt.h>
#include <string>
#include <vector>
#include "globals.h"
Include dependency graph for ArgumentHandler.h:
```



This graph shows which files directly or indirectly include this file:



Classes

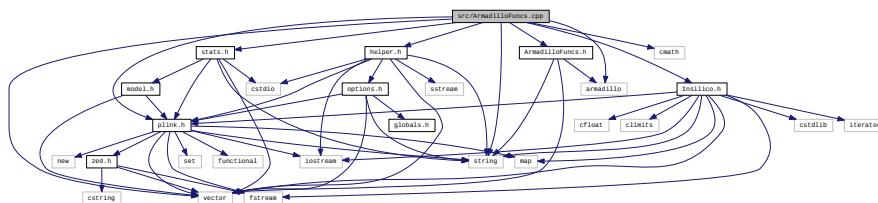
- class [ArgumentHandler](#)

7.4 src/ArmadilloFuncs.cpp File Reference

```
#include <string>
#include <vector>
#include <cmath>
```

```
#include <armadillo>
#include "plink.h"
#include "helper.h"
#include "stats.h"
#include "ArmadilloFuncs.h"
#include "Insilico.h"

Include dependency graph for ArmadilloFuncs.cpp:
```



Functions

- bool [armaDcgain](#) (sp_mat &results, mat &pvals, bool computeDiagonal)
- bool [armaComputeCovariance](#) (mat X, mat &covMatrix, mat &corMatrix)
- bool [armaComputeSparseCovariance](#) (mat X, sp_mat &covMatrix, sp_mat &corMatrix)
- bool [armaReadMatrix](#) (string mFilename, mat &m, vector< string > &variableNames)
- bool [armaWriteMatrix](#) (mat &m, string mFilename, vector< string > variableNames)
- bool [armaWriteSparseMatrix](#) (sp_mat &m, string mFilename, vector< string > variableNames)
- bool [armaGetPlinkNumericToMatrixAll](#) (mat &X)
- bool [armaGetPlinkNumericToMatrixCaseControl](#) (mat &X, mat &Y)

7.4.1 Function Documentation

7.4.1.1 [armaComputeCovariance\(\)](#)

```
bool armaComputeCovariance (
    mat X,
    mat & covMatrix,
    mat & corMatrix )
```

Definition at line 217 of file `ArmadilloFuncs.cpp`.

7.4.1.2 armaComputeSparseCovariance()

```
bool armaComputeSparseCovariance (
    mat X,
    sp_mat & covMatrix,
    sp_mat & corMatrix )
```

Definition at line 256 of file ArmadilloFuncs.cpp.

7.4.1.3 armaDcgain()

```
bool armaDcgain (
    sp_mat & results,
    mat & pvals,
    bool computeDiagonal )
```

Definition at line 27 of file ArmadilloFuncs.cpp.

7.4.1.4 armaGetPlinkNumericToMatrixAll()

```
bool armaGetPlinkNumericToMatrixAll (
    mat & X )
```

Definition at line 437 of file ArmadilloFuncs.cpp.

7.4.1.5 armaGetPlinkNumericToMatrixCaseControl()

```
bool armaGetPlinkNumericToMatrixCaseControl (
    mat & X,
    mat & Y )
```

Definition at line 451 of file ArmadilloFuncs.cpp.

7.4.1.6 armaReadMatrix()

```
bool armaReadMatrix (
    string mFilename,
    mat & m,
    vector< string > & variableNames )
```

Definition at line 281 of file ArmadilloFuncs.cpp.

7.4.1.7 armaWriteMatrix()

```
bool armaWriteMatrix (
    mat & m,
    string mFilename,
    vector< string > variableNames )
```

Definition at line 357 of file ArmadilloFuncs.cpp.

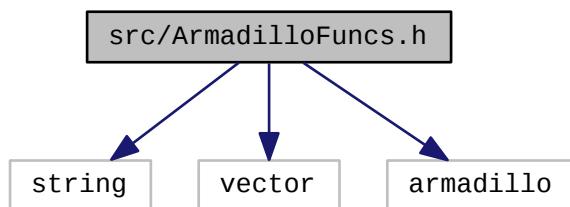
7.4.1.8 armaWriteSparseMatrix()

```
bool armaWriteSparseMatrix (
    sp_mat & m,
    string mFilename,
    vector< string > variableNames )
```

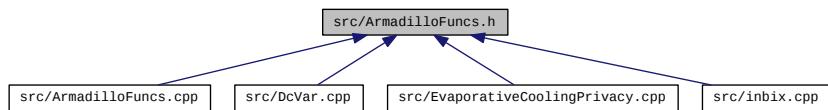
Definition at line 397 of file ArmadilloFuncs.cpp.

7.5 src/ArmadilloFuncs.h File Reference

```
#include <string>
#include <vector>
#include <armadillo>
Include dependency graph for ArmadilloFuncs.h:
```



This graph shows which files directly or indirectly include this file:



Functions

- bool `armaComputeCovariance` (arma::mat X, arma::mat &covMatrix, arma::mat &corMatrix)
- bool `armaComputeSparseCovariance` (arma::mat X, arma::sp_mat &covMatrix, arma::sp_mat &corMatrix)
- bool `armaReadMatrix` (std::string mFilename, arma::mat &m, std::vector< std::string > &variableNames)
- bool `armaWriteMatrix` (arma::mat &m, std::string mFilename, std::vector< std::string > variableNames)
- bool `armaWriteSparseMatrix` (arma::sp_mat &m, std::string mFilename, std::vector< std::string > variableNames)
- bool `armaGetPlinkNumericToMatrixAll` (arma::mat &X)
- bool `armaGetPlinkNumericToMatrixCaseControl` (arma::mat &X, arma::mat &Y)
- bool `armaDcgain` (arma::sp_mat &results, arma::mat &pvals, bool computeDiagonal=false)

7.5.1 Function Documentation

7.5.1.1 `armaComputeCovariance()`

```
bool armaComputeCovariance (
    arma::mat X,
    arma::mat & covMatrix,
    arma::mat & corMatrix )
```

7.5.1.2 `armaComputeSparseCovariance()`

```
bool armaComputeSparseCovariance (
    arma::mat X,
    arma::sp_mat & covMatrix,
    arma::sp_mat & corMatrix )
```

7.5.1.3 `armaDcgain()`

```
bool armaDcgain (
    arma::sp_mat & results,
    arma::mat & pvals,
    bool computeDiagonal = false )
```

7.5.1.4 armaGetPlinkNumericToMatrixAll()

```
bool armaGetPlinkNumericToMatrixAll (
    arma::mat & X )
```

7.5.1.5 armaGetPlinkNumericToMatrixCaseControl()

```
bool armaGetPlinkNumericToMatrixCaseControl (
    arma::mat & X,
    arma::mat & Y )
```

7.5.1.6 armaReadMatrix()

```
bool armaReadMatrix (
    std::string mFilename,
    arma::mat & m,
    std::vector< std::string > & variableNames )
```

7.5.1.7 armaWriteMatrix()

```
bool armaWriteMatrix (
    arma::mat & m,
    std::string mFilename,
    std::vector< std::string > variableNames )
```

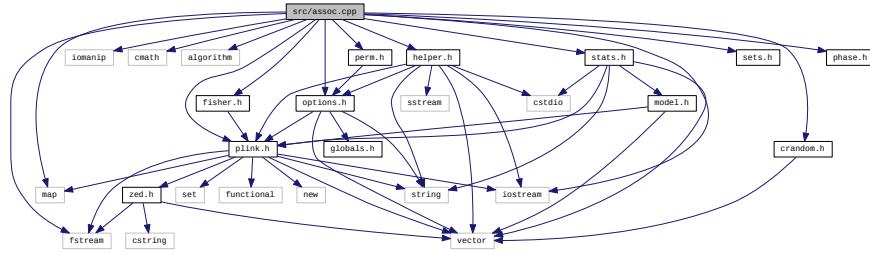
7.5.1.8 armaWriteSparseMatrix()

```
bool armaWriteSparseMatrix (
    arma::sp_mat & m,
    std::string mFilename,
    std::vector< std::string > variableNames )
```

7.6 src/assoc.cpp File Reference

```
#include <iostream>
#include <fstream>
#include <iomanip>
#include <cmath>
#include <algorithm>
#include <map>
#include "plink.h"
#include "fisher.h"
#include "stats.h"
#include "helper.h"
#include "options.h"
#include "crandom.h"
#include "sets.h"
#include "perm.h"
#include "phase.h"
```

Include dependency graph for assoc.cpp:



Variables

- ofstream LOG

7.6.1 Variable Documentation

7.6.1.1 LOG

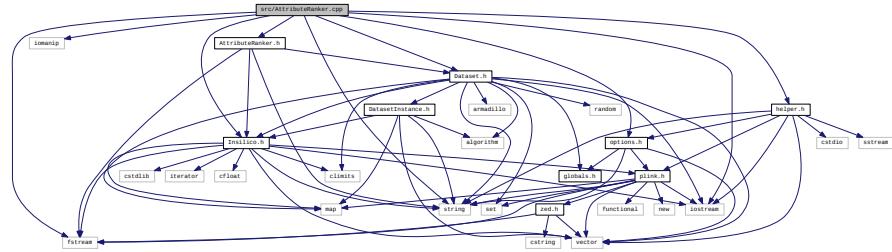
ofstream LOG

Definition at line 81 of file inbix.cpp.

7.7 src/AttributeRanker.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <string>
#include "AttributeRanker.h"
#include "Dataset.h"
#include "Insilico.h"
#include "options.h"
#include "helper.h"
```

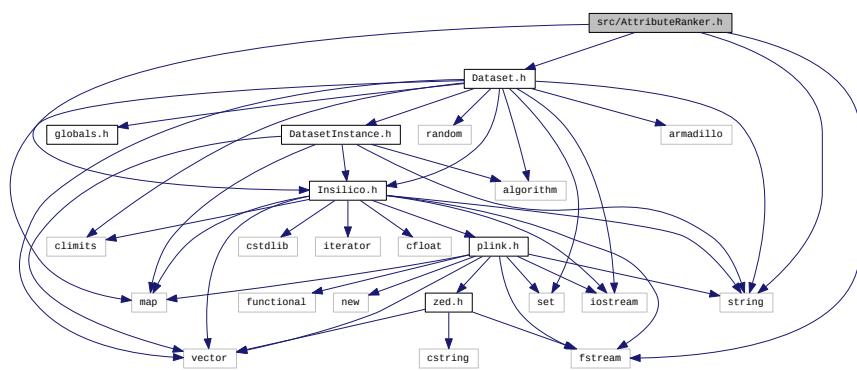
Include dependency graph for AttributeRanker.cpp:



7.8 src/AttributeRanker.h File Reference

```
#include <string>
#include <fstream>
#include "Dataset.h"
#include "Insilico.h"
```

Include dependency graph for AttributeRanker.h:



This graph shows which files directly or indirectly include this file:



Classes

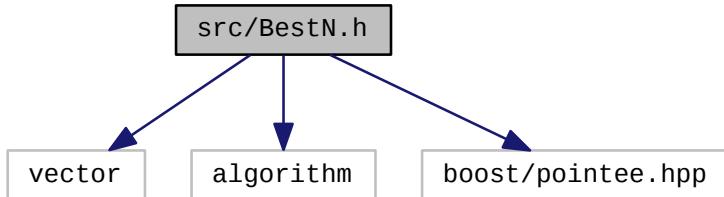
- class [AttributeRanker](#)

Abstract base class for attribute rankers.

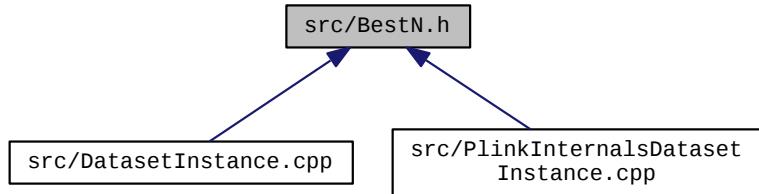
7.9 src/BestN.h File Reference

Find the best n keeping original order for ties - stable sort.

```
#include <vector>
#include <algorithm>
#include <boost/pointee.hpp>
Include dependency graph for BestN.h:
```



This graph shows which files directly or indirectly include this file:



Namespaces

- `insilico`

Functions

- template<typename InputIt , typename OutputIt , typename Comp >
void `insilico::best_n` (InputIt begin, InputIt end, OutputIt out, size_t n, Comp comp)
Get the best n values with ties keeping same original order.

7.9.1 Detailed Description

Find the best n keeping original order for ties - stable sort.

Author

Nate Barney

Version

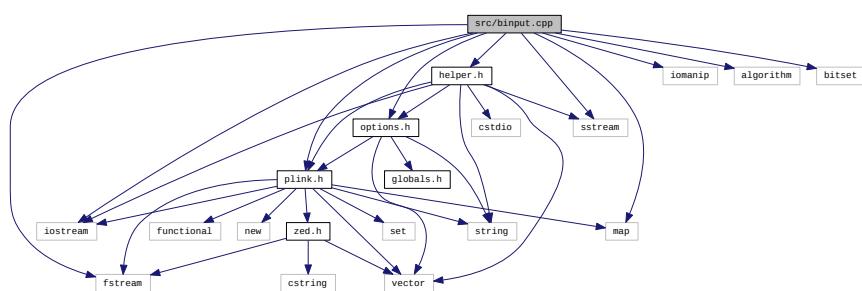
1.0

Contact: bill.c.white@gmail.com Created on: 4/7/04

7.10 src/binput.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <map>
#include <algorithm>
#include <set>
#include "plink.h"
#include "options.h"
#include "helper.h"
```

Include dependency graph for binput.cpp:



Variables

- ofstream LOG

7.10.1 Variable Documentation

7.10.1.1 LOG

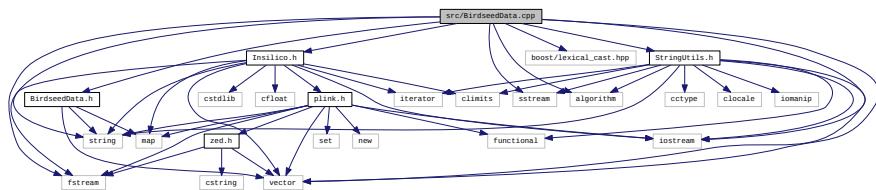
ofstream LOG

Definition at line 81 of file inbix.cpp.

7.11 src/BirdseedData.cpp File Reference

```
#include <iostream>
#include <fstream>
#include <string>
#include <sstream>
#include <vector>
#include <algorithm>
#include <boost/lexical_cast.hpp>
#include "BirdseedData.h"
#include "Insilico.h"
#include "StringUtils.h"
Include dependency graph for BirdseedData.cpp:
```

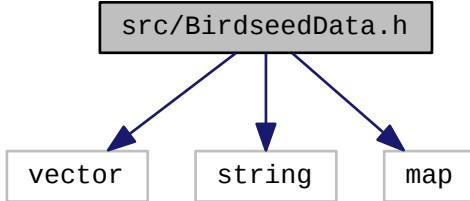
Include dependency graph for BirdseedData.cpp:



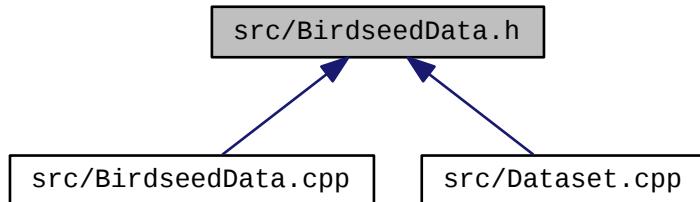
7.12 src/BirdseedData.h File Reference

```
#include <vector>
#include <string>
```

```
#include <map>
Include dependency graph for BirdseedData.h:
```



This graph shows which files directly or indirectly include this file:



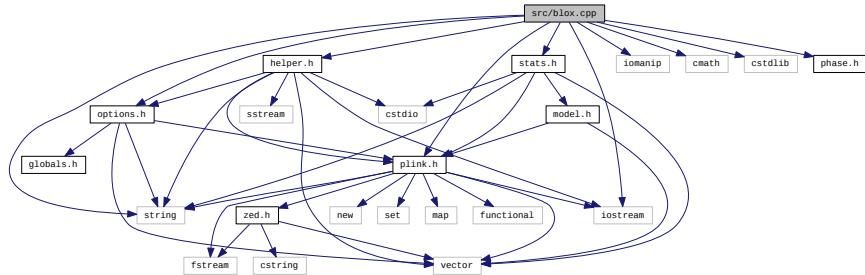
Classes

- class [BirdseedData](#)
Read Broad's Birdsuite Birdseed-called SNP data.

7.13 src/blox.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <string>
#include <cmath>
#include <cstdlib>
#include "options.h"
#include "helper.h"
```

```
#include "plink.h"
#include "phase.h"
#include "stats.h"
Include dependency graph for blox.cpp:
```



Classes

- class [LDPair](#)
- struct [Pair_cmp](#)
- class [DPrime](#)
- class [PairwiseLinkage](#)

Variables

- [Plink * PP](#)

7.13.1 Variable Documentation

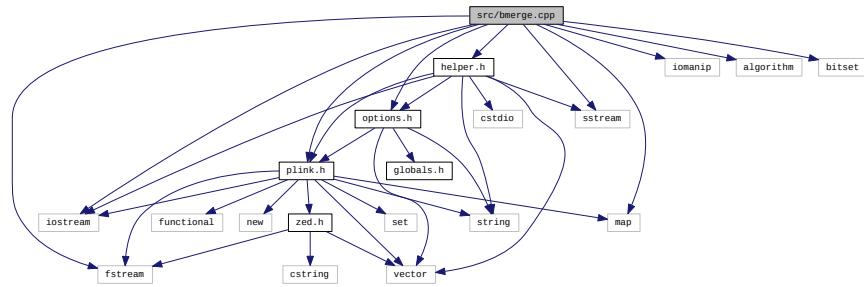
7.13.1.1 PP

[Plink* PP](#)

Definition at line 85 of file `inbix.cpp`.

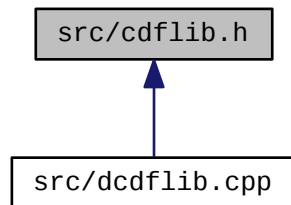
7.14 src/bmerge.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <map>
#include <algorithm>
#include <set>
#include "plink.h"
#include "options.h"
#include "helper.h"
Include dependency graph for bmerge.cpp:
```



7.15 src/cdflib.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- double `aldiv` (double *, double *)

- double **alngam** (double *)
- double **alnrel** (double *)
- double **apser** (double *, double *, double *, double *)
- double **basym** (double *, double *, double *, double *)
- double **bcorr** (double *, double *)
- double **betain** (double *, double *)
- double **bfrac** (double *, double *, double *, double *, double *, double *)
- void **bgrat** (double *, double *, double *, double *, double *, int *)
- double **bpser** (double *, double *, double *, double *)
- void **bratio** (double *, double *, double *, double *, double *, int *)
- double **brcmp1** (int *, double *, double *, double *, double *)
- double **brcomp** (double *, double *, double *, double *)
- double **bup** (double *, double *, double *, int *, double *)
- void **cdfbet** (int *, double *, double *, double *, double *, double *, int *, double *)
- void **cdfbin** (int *, double *, double *, double *, double *, double *, int *, double *)
- void **cdfchi** (int *, double *, double *, double *, double *, int *, double *)
- void **cdfchn** (int *, double *, double *, double *, double *, double *, int *, double *)
- void **cdff** (int *, double *, double *, double *, double *, int *, double *)
- void **cdfffnc** (int *, double *, double *, double *, double *, double *, double *, int *s, double *)
- void **cdfgam** (int *, double *, double *, double *, double *, double *, int *, double *)
- void **cdfnbn** (int *, double *, double *, double *, double *, double *, double *, int *, double *)
- void **cdfnor** (int *, double *, double *, double *, double *, double *, int *, double *)
- void **cdfpoi** (int *, double *, double *, double *, double *, int *, double *)
- void **cdft** (int *, double *, double *, double *, int *, double *)
- void **cumbet** (double *, double *, double *, double *, double *)
- void **cumbin** (double *, double *, double *, double *, double *)
- void **cumchi** (double *, double *, double *)
- void **cumchn** (double *, double *, double *, double *)
- void **cumf** (double *, double *, double *, double *)
- void **cumfnc** (double *, double *, double *, double *, double *)
- void **cumgam** (double *, double *, double *, double *)
- void **cumnbn** (double *, double *, double *, double *, double *, double *)
- void **cumnor** (double *, double *)
- void **cumpoi** (double *, double *, double *, double *)
- void **cumt** (double *, double *, double *, double *)
- double **dbetrm** (double *, double *)
- double **devlpl** (double [], int *, double *)
- double **dexpm1** (double *)
- double **dinvnr** (double *p, double *q)
- static void **E0000** (int, int *, double *, double *, unsigned long *, unsigned long *, double *, double *, double *, double *)
- void **dinrv** (int *, double *, double *, unsigned long *, unsigned long *)
- void **dstinv** (double *, double *, double *, double *, double *, double *)
- double **dlanor** (double *)
- double **dln1mx** (double *)
- double **dln1px** (double *)
- double **dlnbet** (double *, double *)
- double **dngam** (double *)
- double **dstrem** (double *)
- double **dt1** (double *, double *, double *)

- static void `E0001` (int, int *, double *, double *, double *, unsigned long *, unsigned long *, double *, double *, double *)
- void `dzror` (int *, double *, double *, double *, double *, unsigned long *, unsigned long *)
- void `dstzr` (double *zxlo, double *zxhi, double *zabstl, double *zreitl)
- double `erf1` (double *)
- double `erfc1` (int *, double *)
- double `esum` (int *, double *)
- double `exparg` (int *)
- double `fpser` (double *, double *, double *, double *)
- double `gam1` (double *)
- void `gaminv` (double *, double *, double *, double *, double *, int *)
- double `gamln` (double *)
- double `gamln1` (double *)
- double `Xgamm` (double *)
- void `grat1` (double *, double *, double *, double *, double *, double *)
- void `gratio` (double *, double *, double *, double *, int *)
- double `gsumln` (double *, double *)
- double `psi` (double *)
- double `rcomp` (double *, double *)
- double `rexp` (double *)
- double `rlog` (double *)
- double `rlog1` (double *)
- double `spmpar` (int *)
- double `stvaln` (double *)
- double `fifdint` (double)
- double `fifdmax1` (double, double)
- double `fifdmin1` (double, double)
- double `fifdsign` (double, double)
- long `fifidint` (double)
- long `fifmod` (long, long)
- void `ftnstop` (char *)
- int `ipmpar` (int *)

7.15.1 Function Documentation

7.15.1.1 algdiv()

```
double algdiv (
    double * ,
    double * )
```

Definition at line 33 of file dcdflib.cpp.

7.15.1.2 alngam()

```
double alngam (
    double * )
```

Definition at line 86 of file dcdflib.cpp.

7.15.1.3 alnrel()

```
double alnrel (
    double * )
```

Definition at line 198 of file dcdflib.cpp.

7.15.1.4 apser()

```
double apser (
    double * ,
    double * ,
    double * ,
    double * )
```

Definition at line 227 of file dcdflib.cpp.

7.15.1.5 basym()

```
double basym (
    double * ,
    double * ,
    double * ,
    double * )
```

Definition at line 262 of file dcdflib.cpp.

7.15.1.6 bcorr()

```
double bcorr (
    double * ,
    double * )
```

Definition at line 370 of file dcdflib.cpp.

7.15.1.7 **betain()**

```
double betain (
    double * ,
    double * )
```

Definition at line 419 of file dcdflib.cpp.

7.15.1.8 **bfrac()**

```
double bfrac (
    double * ,
    double * )
```

Definition at line 525 of file dcdflib.cpp.

7.15.1.9 **bgrat()**

```
void bgrat (
    double * ,
    int * i )
```

Definition at line 592 of file dcdflib.cpp.

7.15.1.10 **bpser()**

```
double bpser (
    double * ,
    double * ,
    double * ,
    double * )
```

Definition at line 674 of file dcdflib.cpp.

7.15.1.11 bratio()

```
void bratio (
    double * ,
    int * )
```

Definition at line 772 of file dcdflib.cpp.

7.15.1.12 brcmp1()

```
double brcmp1 (
    int * ,
    double * ,
    double * ,
    double * ,
    double * )
```

Definition at line 1009 of file dcdflib.cpp.

7.15.1.13 brcomp()

```
double brcomp (
    double * ,
    double * ,
    double * ,
    double * )
```

Definition at line 1147 of file dcdflib.cpp.

7.15.1.14 bup()

```
double bup (
    double * ,
    double * ,
    double * ,
    double * ,
    int * ,
    double * )
```

Definition at line 1285 of file dcdflib.cpp.

7.15.1.15 cdfbet()

```
void cdfbet (
    int * ,
    double * ,
    int * ,
    double * )
```

Definition at line 1363 of file dcdflib.cpp.

7.15.1.16 cdfbin()

```
void cdfbin (
    int * ,
    double * ,
    int * ,
    double * )
```

Definition at line 1734 of file dcdflib.cpp.

7.15.1.17 cdfchi()

```
void cdfchi (
    int * ,
    double * ,
    double * ,
    double * ,
    double * ,
    int * ,
    double * )
```

Definition at line 2100 of file dcdflib.cpp.

7.15.1.18 cdfchn()

```
void cdfchn (
    int * ,
    double * ,
    int * ,
    double * )
```

Definition at line 2382 of file dcdflib.cpp.

7.15.1.19 cdff()

```
void cdff (
    int * ,
    double * ,
    int * ,
    double * )
```

Definition at line 2652 of file dcdflib.cpp.

7.15.1.20 cdffnc()

```
void cdffnc (
    int * ,
    double * ,
    int * s,
    double * )
```

Definition at line 2965 of file dcdflib.cpp.

7.15.1.21 cdfgam()

```
void cdfgam (
    int * ,
    double * ,
    int * ,
    double * )
```

Definition at line 3285 of file dcdflib.cpp.

7.15.1.22 cdfnbn()

```
void cdfnbn (
    int * ,
    double * ,
    int * ,
    double * )
```

Definition at line 3578 of file dcdflib.cpp.

7.15.1.23 cdfnor()

```
void cdfnor (
    int * ,
    double * ,
    int * ,
    double * )
```

Definition at line 3946 of file dcdflib.cpp.

7.15.1.24 cdfpoi()

```
void cdfpoi (
    int * ,
    double * ,
    double * ,
    double * ,
    double * ,
    int * ,
    double * )
```

Definition at line 4147 of file dcdflib.cpp.

7.15.1.25 cdft()

```
void cdft (
    int * ,
    double * ,
    double * ,
    double * ,
    double * ,
    int * ,
    double * )
```

Definition at line 4401 of file dcdflib.cpp.

7.15.1.26 cumbet()

```
void cumbet (
    double * ,
    double * )
```

Definition at line 4650 of file dcdflib.cpp.

7.15.1.27 cumbin()

```
void cumbin (
    double * ,
    double * )
```

Definition at line 4726 of file dcdflib.cpp.

7.15.1.28 cumchi()

```
void cumchi (
    double * ,
    double * ,
    double * ,
    double * )
```

Definition at line 4792 of file dcdflib.cpp.

7.15.1.29 cumchn()

```
void cumchn (
    double * ,
    double * ,
    double * ,
    double * ,
    double * )
```

Definition at line 4842 of file dcdflib.cpp.

7.15.1.30 cumf()

```
void cumf (
    double * ,
    double * ,
    double * ,
    double * ,
    double * )
```

Definition at line 5055 of file dcdflib.cpp.

7.15.1.31 cumfnc()

```
void cumfnc (
    double * ,
    double * )
```

Definition at line 5138 of file dcdflib.cpp.

7.15.1.32 cumgam()

```
void cumgam (
    double * ,
    double * ,
    double * ,
    double * )
```

Definition at line 5307 of file dcdflib.cpp.

7.15.1.33 cumnbn()

```
void cumnbn (
    double * ,
    double * )
```

Definition at line 5366 of file dcdflib.cpp.

7.15.1.34 cumnor()

```
void cumnor (
    double * ,
    double * ,
    double * )
```

Definition at line 5431 of file dcdflib.cpp.

7.15.1.35 cumpoi()

```
void cumpoi (
    double * ,
    double * ,
    double * ,
    double * )
```

Definition at line 5640 of file dcdflib.cpp.

7.15.1.36 cumt()

```
void cumt (
    double * ,
    double * ,
    double * ,
    double * )
```

Definition at line 5691 of file dcdflib.cpp.

7.15.1.37 dbetrm()

```
double dbetrm (
    double * ,
    double * )
```

Definition at line 5753 of file dcdflib.cpp.

7.15.1.38 devlpl()

```
double devlpl (
    double [ ],
    int * ,
    double * )
```

Definition at line 5803 of file dcdflib.cpp.

7.15.1.39 dexpm1()

```
double dexpm1 (
    double * )
```

Definition at line 5844 of file dcdflib.cpp.

7.15.1.40 dinvnr()

```
double dinvnr (
    double * p,
    double * q )
```

Definition at line 5894 of file dcdflib.cpp.

7.15.1.41 dinvr()

```
void dinvr (
    int * ,
    double * ,
    double * ,
    unsigned long * ,
    unsigned long * )
```

Definition at line 6207 of file dcdflib.cpp.

7.15.1.42 dlanor()

```
double dlanor (
    double * )
```

Definition at line 6343 of file dcdflib.cpp.

7.15.1.43 dln1mx()

```
double dln1mx (
    double * )
```

Definition at line 6402 of file dcdflib.cpp.

7.15.1.44 dln1px()

```
double dln1px (
    double * )
```

Definition at line 6444 of file dcdflib.cpp.

7.15.1.45 dlnbet()

```
double dlnbet (
    double * ,
    double * )
```

Definition at line 6504 of file dcdflib.cpp.

7.15.1.46 dlngam()

```
double dlngam (
    double * )
```

Definition at line 6641 of file dcdflib.cpp.

7.15.1.47 dstinv()

```
void dstinv (
    double * ,
    double * )
```

Definition at line 6272 of file dcdflib.cpp.

7.15.1.48 dstrem()

```
double dstrem (
    double * )
```

Definition at line 6722 of file dcdflib.cpp.

7.15.1.49 dstzr()

```
void dstzr (
    double * zxlo,
    double * zxhi,
    double * zabstl,
    double * zreqlt )
```

Definition at line 7087 of file dcdflib.cpp.

7.15.1.50 dt1()

```
double dt1 (
    double * ,
    double * ,
    double * )
```

Definition at line 6786 of file dcdflib.cpp.

7.15.1.51 dzror()

```
void dzror (
    int * ,
    double * ,
    double * ,
    double * ,
    double * ,
    unsigned long * ,
    unsigned long * )
```

Definition at line 7019 of file dcdflib.cpp.

7.15.1.52 E0000()

```
static void E0000 (
    int ,
    int * ,
    double * ,
    double * ,
    unsigned long * ,
    unsigned long * ,
    double * ) [static]
```

7.15.1.53 E0001()

```
static void E0001 (
    int ,
    int * ,
    double * ,
    double * ,
    double * ,
    double * ,
    unsigned long * ,
    unsigned long * ,
    double * ,
    double * ,
    double * ,
    double * ) [static]
```

7.15.1.54 erf1()

```
double erf1 (
    double * )
```

Definition at line 7134 of file dcdflib.cpp.

7.15.1.55 erfc1()

```
double erfc1 (
    int * ,
    double * )
```

Definition at line 7202 of file dcdflib.cpp.

7.15.1.56 esum()

```
double esum (
    int * ,
    double * )
```

Definition at line 7308 of file dcdflib.cpp.

7.15.1.57 exparg()

```
double exparg (
    int * )
```

Definition at line 7337 of file dcdflib.cpp.

7.15.1.58 fifdint()

```
double fifdint (
    double )
```

Definition at line 9132 of file dcdflib.cpp.

7.15.1.59 fifdmax1()

```
double fifdmax1 (
    double ,
    double )
```

Definition at line 9141 of file dcdflib.cpp.

7.15.1.60 fifdmin1()

```
double fifdmin1 (
    double ,
    double )
```

Definition at line 9152 of file dcdflib.cpp.

7.15.1.61 fifdsign()

```
double fifdsign (
    double ,
    double )
```

Definition at line 9163 of file dcdflib.cpp.

7.15.1.62 fifidint()

```
long fifidint (
    double   )
```

Definition at line 9176 of file dcdflib.cpp.

7.15.1.63 fifmod()

```
long fifmod (
    long ,
    long   )
```

Definition at line 9186 of file dcdflib.cpp.

7.15.1.64 fpser()

```
double fpser (
    double * ,
    double * ,
    double * ,
    double *   )
```

Definition at line 7383 of file dcdflib.cpp.

7.15.1.65 ftnstop()

```
void ftnstop (
    char *   )
```

Definition at line 9196 of file dcdflib.cpp.

7.15.1.66 gam1()

```
double gam1 (
    double *   )
```

Definition at line 7427 of file dcdflib.cpp.

7.15.1.67 gaminv()

```
void gaminv (
    double * ,
    int * )
```

Definition at line 7487 of file dcdflib.cpp.

7.15.1.68 gamln()

```
double gamln (
    double * )
```

Definition at line 7849 of file dcdflib.cpp.

7.15.1.69 gamln1()

```
double gamln1 (
    double * )
```

Definition at line 7902 of file dcdflib.cpp.

7.15.1.70 grat1()

```
void grat1 (
    double * ,
    double * )
```

Definition at line 8099 of file dcdflib.cpp.

7.15.1.71 gratio()

```
void gratio (
    double * ,
    double * ,
    double * ,
    double * ,
    int * )
```

Definition at line 8202 of file dcdflib.cpp.

7.15.1.72 gsumln()

```
double gsumln (
    double * ,
    double * )
```

Definition at line 8619 of file dcdflib.cpp.

7.15.1.73 ipmpar()

```
int ipmpar (
    int * )
```

Definition at line 71 of file ipmpar.h.

7.15.1.74 psi()

```
double psi (
    double * )
```

Definition at line 8646 of file dcdflib.cpp.

7.15.1.75 rcomp()

```
double rcomp (
    double * ,
    double * )
```

Definition at line 8839 of file dcdflib.cpp.

7.15.1.76 rexp()

```
double rexp (
    double * )
```

Definition at line 8872 of file dcdflib.cpp.

7.15.1.77 rlog()

```
double rlog (
    double * )
```

Definition at line 8902 of file dcdflib.cpp.

7.15.1.78 rlog1()

```
double rlog1 (
    double * )
```

Definition at line 8952 of file dcdflib.cpp.

7.15.1.79 spmpar()

```
double spmpar (
    int * )
```

Definition at line 9002 of file dcdflib.cpp.

7.15.1.80 stvaln()

```
double stvaln (
    double * )
```

Definition at line 9068 of file dcdflib.cpp.

7.15.1.81 Xgamm()

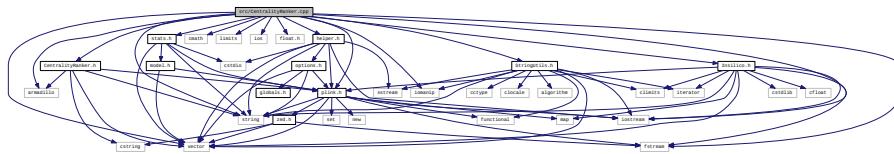
```
double Xgamm (
    double *  )
```

Definition at line 7950 of file dcdflib.cpp.

7.16 src/CentralityRanker.cpp File Reference

```
#include <iostream>
#include <fstream>
#include <iomanip>
#include <string>
#include <vector>
#include <cmath>
#include <limits>
#include <ios>
#include <float.h>
#include <armadillo>
#include "plink.h"
#include "stats.h"
#include "helper.h"
#include "StringUtils.h"
#include "CentralityRanker.h"
#include "Insilico.h"
```

Include dependency graph for CentralityRanker.cpp:

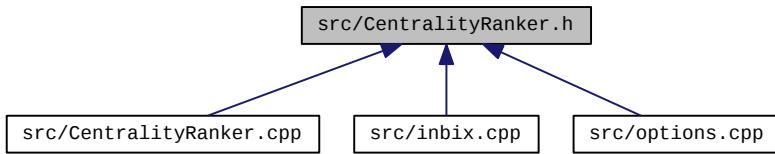
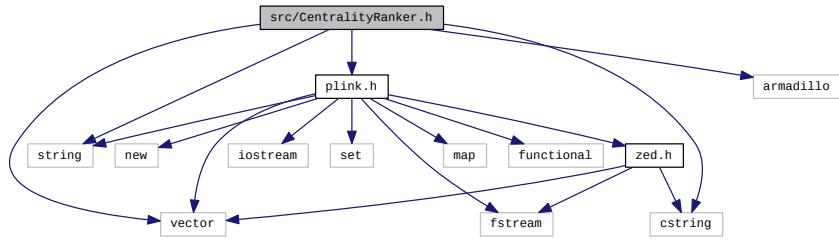


7.17 src/CentralityRanker.h File Reference

```
#include <string>
#include <vector>
#include <cstring>
#include <armadillo>
```

```
#include "plink.h"
```

Include dependency graph for CentralityRanker.h:



Classes

- class [CentralityRanker](#)

Enumerations

- enum [SolverMethod](#) { [POWER_METHOD](#), [GAUSS_ELIMINATION](#) }

7.17.1 Enumeration Type Documentation

7.17.1.1 SolverMethod

```
enum SolverMethod
```

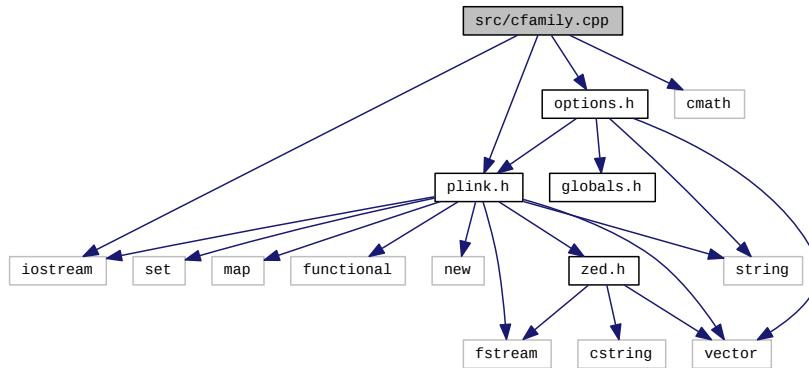
Enumerator

POWER_METHOD	
GAUSS_ELIMINATION	

Definition at line 16 of file CentralityRanker.h.

7.18 src/cfamily.cpp File Reference

```
#include <iostream>
#include "plink.h"
#include "options.h"
#include <cmath>
Include dependency graph for cfamily.cpp:
```



Functions

- bool `isAncestorOf (Individual *indx, Individual *f)`
- int `mCount (Individual *indx, Individual *f)`
- void `listAllAncestors (Individual *a, set< Individual *> &anclist, int d)`
- double `genrel (Individual *a, Individual *b)`

7.18.1 Function Documentation

7.18.1.1 genrel()

```
double genrel (
    Individual * a,
    Individual * b )
```

Definition at line 49 of file cfamily.cpp.

7.18.1.2 isAncestorOf()

```
bool isAncestorOf (
    Individual * indx,
    Individual * f )
```

Definition at line 248 of file cfamily.cpp.

7.18.1.3 listAllAncestors()

```
void listAllAncestors (
    Individual * a,
    set< Individual *> & anclist,
    int d )
```

Definition at line 24 of file cfamily.cpp.

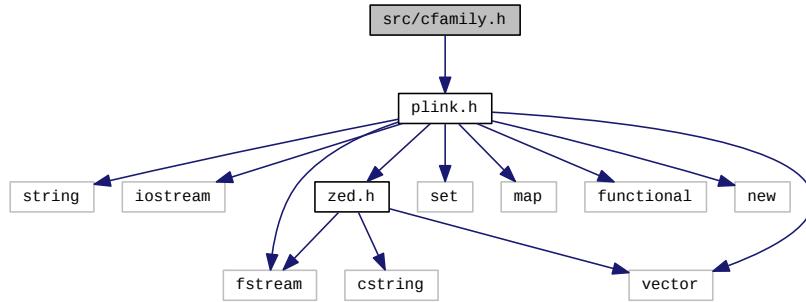
7.18.1.4 mCount()

```
int mCount (
    Individual * indx,
    Individual * f )
```

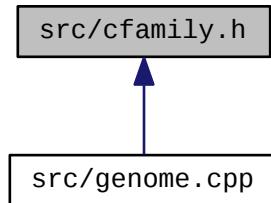
Definition at line 185 of file cfamily.cpp.

7.19 src/cfamily.h File Reference

```
#include "plink.h"
Include dependency graph for cfamily.h:
```



This graph shows which files directly or indirectly include this file:



Functions

- double `genrel (Individual *a, Individual *b)`

7.19.1 Function Documentation

7.19.1.1 genrel()

```
double genrel (
```

Individual * a,

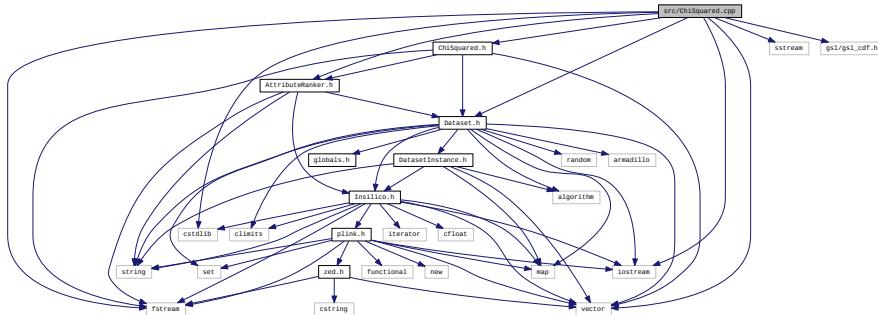
Individual * b)

Definition at line 49 of file cfamily.cpp.

7.20 src/ChiSquared.cpp File Reference

```
#include <cstdlib>
#include <iostream>
#include <fstream>
#include <sstream>
#include <vector>
#include "gsl/gsl_cdf.h"
#include "AttributeRanker.h"
#include "ChiSquared.h"
#include "Dataset.h"
Include dependency graph for ChiSquared
```

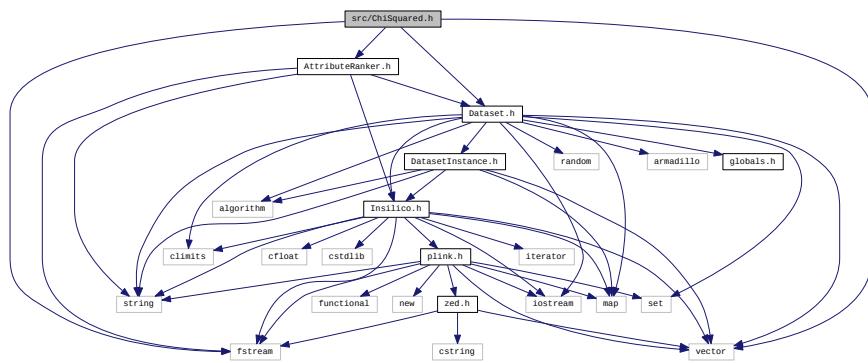
Include dependency graph for ChiSquared.cpp:



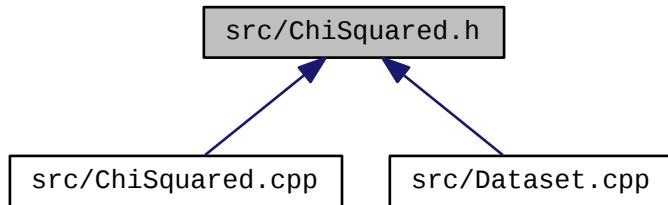
7.21 src/ChiSquared.h File Reference

```
#include <vector>
#include <fstream>
#include "AttributeRanker.h"
```

```
#include "Dataset.h"
Include dependency graph for ChiSquared.h:
```



This graph shows which files directly or indirectly include this file:



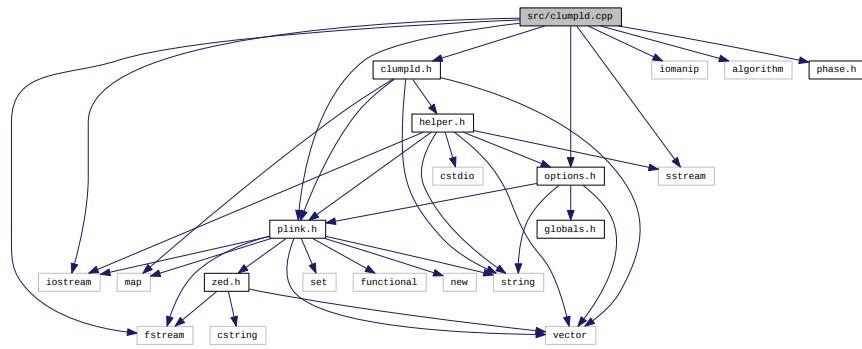
Classes

- class [ChiSquared](#)
Chi-squared attribute ranking algorithm.

7.22 src/clumpld.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <algorithm>
#include "clumpld.h"
#include "phase.h"
```

```
#include "options.h"
#include "plink.h"
Include dependency graph for clumpld.cpp:
```



Functions

- string `returnFullRangeList (Range &r1, map< string, set< Range > > &ranges, bool verbose)`

7.22.1 Function Documentation

7.22.1.1 `returnFullRangeList()`

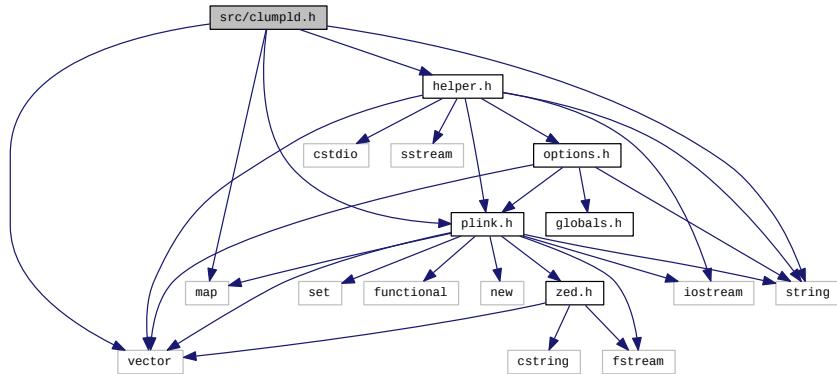
```
string returnFullRangeList (
    Range & r1,
    map< string, set< Range > > & ranges,
    bool verbose )
```

Definition at line 1207 of file `clumpld.cpp`.

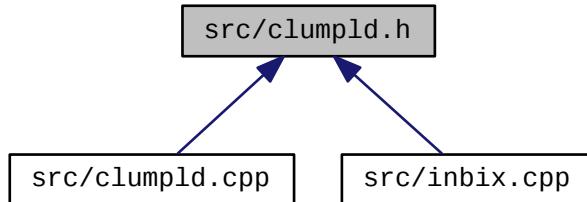
7.23 src/clumpld.h File Reference

```
#include <string>
#include <vector>
#include <map>
#include "plink.h"
```

```
#include "helper.h"
Include dependency graph for clumpld.h:
```



This graph shows which files directly or indirectly include this file:



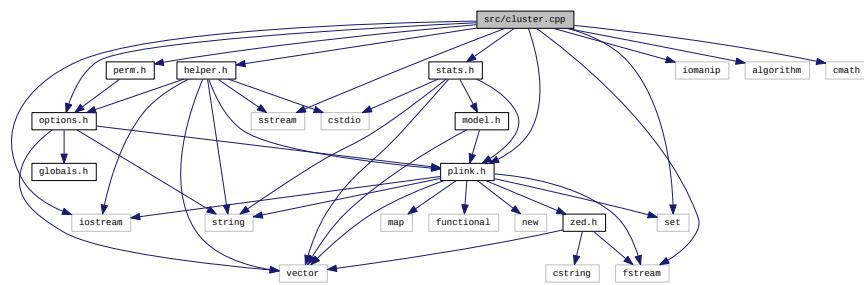
Classes

- class [ResultTrio](#)
- class [ClumpPair](#)
- class [ClumpResults](#)
- class [clump_LD](#)

7.24 src/cluster.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
```

```
#include <sstream>
#include <set>
#include <algorithm>
#include <cmath>
#include "plink.h"
#include "helper.h"
#include "options.h"
#include "perm.h"
#include "stats.h"
Include dependency graph for cluster.cpp:
```



Classes

- class [Neighbour](#)

Functions

- double `cldist` (`vector< vector< double > >` &, `vector< int >` &, `vector< int >` &)
- double `groupAvgLink` (`vector< vector< double > >` &, `vector< int >` &, `vector< int >` &)
- bool `homogeneous_clusters` (`Plink` &, `vector< int >` &, `vector< int >` &)
- bool `spec_clusters` (`Plink` &, `vector< int >` &, `vector< int >` &)
- bool `pairable_cluster` (`vector< vector< bool > >` &, `vector< int >` &, `vector< int >` &)
- bool `selcon_inds` (`Plink` &, `vector< int >` &, `vector< int >` &, `set< int >` &)

Variables

- `ofstream LOG`

7.24.1 Function Documentation

7.24.1.1 `cldist()`

```
double cldist (
    vector< vector< double > > & d,
    vector< int > & a,
    vector< int > & b )
```

Definition at line 1097 of file cluster.cpp.

7.24.1.2 `groupAvgLink()`

```
double groupAvgLink (
    vector< vector< double > > & d,
    vector< int > & a,
    vector< int > & b )
```

Definition at line 1117 of file cluster.cpp.

7.24.1.3 `homogeneous_clusters()`

```
bool homogeneous_clusters (
    PLink & P,
    vector< int > & a,
    vector< int > & b )
```

Definition at line 1137 of file cluster.cpp.

7.24.1.4 `pairable_cluster()`

```
bool pairable_cluster (
    vector< vector< bool > > & pairable,
    vector< int > & a,
    vector< int > & b )
```

Definition at line 1172 of file cluster.cpp.

7.24.1.5 selcon_inds()

```
bool selcon_inds (
    Plink & P,
    vector< int > & a,
    vector< int > & b,
    set< int > & inc )
```

Definition at line 1179 of file cluster.cpp.

7.24.1.6 spec_clusters()

```
bool spec_clusters (
    Plink & P,
    vector< int > & a,
    vector< int > & b )
```

Definition at line 1151 of file cluster.cpp.

7.24.2 Variable Documentation

7.24.2.1 LOG

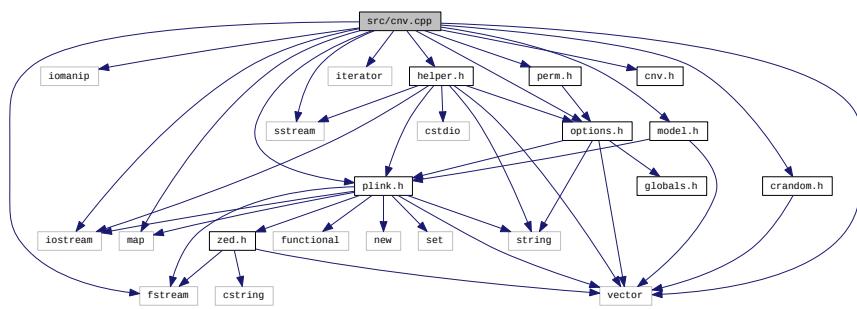
```
ofstream LOG
```

Definition at line 81 of file inbix.cpp.

7.25 src/cnv.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <vector>
#include <map>
#include <iterator>
#include "plink.h"
#include "helper.h"
#include "options.h"
#include "perm.h"
#include "cnv.h"
#include "crandom.h"
```

```
#include "model.h"
Include dependency graph for cnv.cpp:
```



Classes

- class [sortedSegments](#)

Functions

- double [probOverlap \(Segment *s, set< Range > &isection\)](#)
- bool [intersects \(set< Range > &isection, set< Range > &iintersects, int chr, int p1, int p2\)](#)
- int [count_intersects \(set< Range > &isection, int chr, int p1, int p2\)](#)
- double [weighted_count_intersects \(set< Range > &isection, int chr, int p1, int p2\)](#)
- vector< int > [segmentCountCaseControls \(Plink *P, bool countCases\)](#)
- set< Segment > [allSegmentsIntersecting \(Range &r\)](#)

Variables

- [Plink * PP](#)
- const double [EPS_OVERLAP = 1e-6](#)

7.25.1 Function Documentation

7.25.1.1 allSegmentsIntersecting()

```
set<Segment> allSegmentsIntersecting (
    Range & r )
```

Definition at line 1836 of file cnv.cpp.

7.25.1.2 count_intersects()

```
int count_intersects (
    set< Range > & isection,
    int chr,
    int p1,
    int p2 )
```

Definition at line 1664 of file cnv.cpp.

7.25.1.3 intersects()

```
bool intersects (
    set< Range > & isection,
    set< Range > & iintersects,
    int chr,
    int p1,
    int p2 )
```

Definition at line 1543 of file cnv.cpp.

7.25.1.4 probOverlap()

```
double probOverlap (
    Segment * s,
    set< Range > & isection )
```

Definition at line 52 of file cnv.cpp.

7.25.1.5 segmentCountCaseControls()

```
vector<int> segmentCountCaseControls (
    PLink * P,
    bool countCases )
```

Definition at line 1815 of file cnv.cpp.

7.25.1.6 weighted_count_intersects()

```
double weighted_count_intersects (
    set< Range > & isection,
    int chr,
    int p1,
    int p2 )
```

Definition at line 1740 of file cnv.cpp.

7.25.2 Variable Documentation

7.25.2.1 EPS_OVERLAP

```
const double EPS_OVERLAP = 1e-6
```

Definition at line 32 of file cnv.cpp.

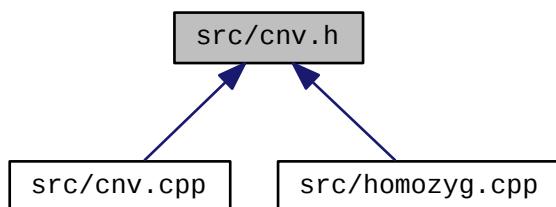
7.25.2.2 PP

```
Plink* PP
```

Definition at line 85 of file inbix.cpp.

7.26 src/cnv.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [CNVIndivReport](#)

Functions

- bool [intersects](#) (set< Range > &, set< Range > &, int, int, int)
- int [count_intersects](#) (set< Range > &, int, int, int)
- double [weighted_count_intersects](#) (set< Range > &, int, int, int)
- vector< int > [segmentCountCaseControls](#) (Plink *, bool)
- set< Segment > [allSegmentsIntersecting](#) (Range &)

7.26.1 Function Documentation

7.26.1.1 allSegmentsIntersecting()

```
set<Segment> allSegmentsIntersecting (
    Range & )
```

Definition at line 1836 of file cnv.cpp.

7.26.1.2 count_intersects()

```
int count_intersects (
    set< Range > & ,
    int ,
    int ,
    int )
```

Definition at line 1664 of file cnv.cpp.

7.26.1.3 intersects()

```
bool intersects (
    set< Range > & ,
    set< Range > & ,
    int ,
    int ,
    int )
```

Definition at line 1543 of file cnv.cpp.

7.26.1.4 segmentCountCaseControls()

```
vector<int> segmentCountCaseControls (
    Plink * ,
    bool   )
```

Definition at line 1815 of file cnv.cpp.

7.26.1.5 weighted_count_intersects()

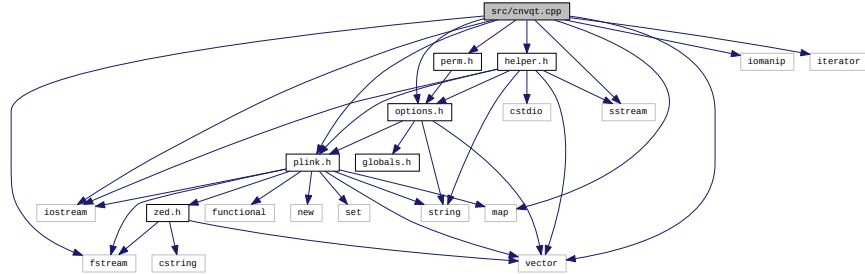
```
double weighted_count_intersects (
    set< Range > & ,
    int ,
    int ,
    int )
```

Definition at line 1740 of file cnv.cpp.

7.27 src/cnvqt.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <vector>
#include <map>
#include <iterator>
#include "plink.h"
#include "helper.h"
#include "options.h"
#include "perm.h"
```

Include dependency graph for cnvqt.cpp:



Variables

- `Plink * PP`

7.27.1 Variable Documentation

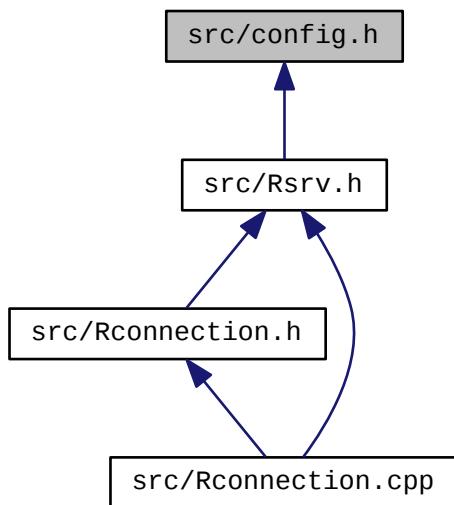
7.27.1.1 PP

`Plink* PP`

Definition at line 85 of file `inbx.cpp`.

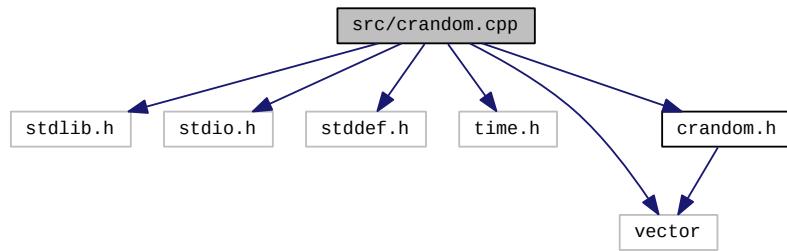
7.28 src/config.h File Reference

This graph shows which files directly or indirectly include this file:



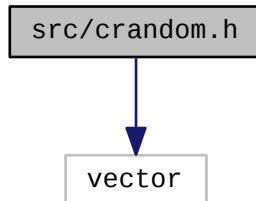
7.29 src/crandom.cpp File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <stddef.h>
#include <time.h>
#include <vector>
#include "crandom.h"
Include dependency graph for crandom.cpp:
```

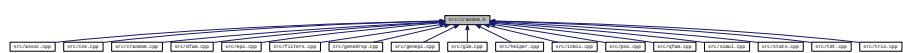


7.30 src/crandom.h File Reference

```
#include <vector>
Include dependency graph for crandom.h:
```



This graph shows which files directly or indirectly include this file:

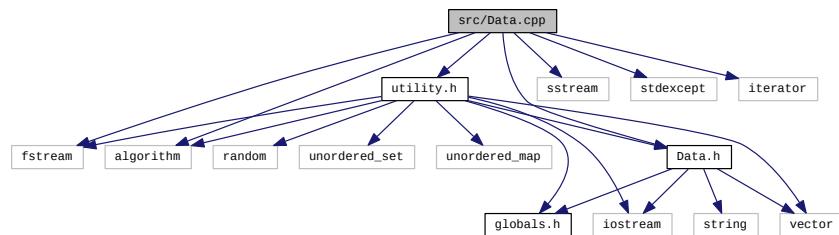


Classes

- class [CRandom](#)

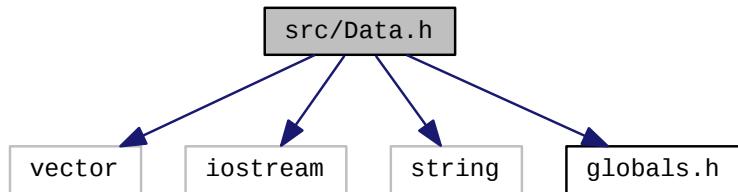
7.31 src/Data.cpp File Reference

```
#include <fstream>
#include <sstream>
#include <stdexcept>
#include <algorithm>
#include <iterator>
#include "Data.h"
#include "utility.h"
Include dependency graph for Data.cpp:
```



7.32 src/Data.h File Reference

```
#include <vector>
#include <iostream>
#include <string>
#include "globals.h"
Include dependency graph for Data.h:
```



This graph shows which files directly or indirectly include this file:

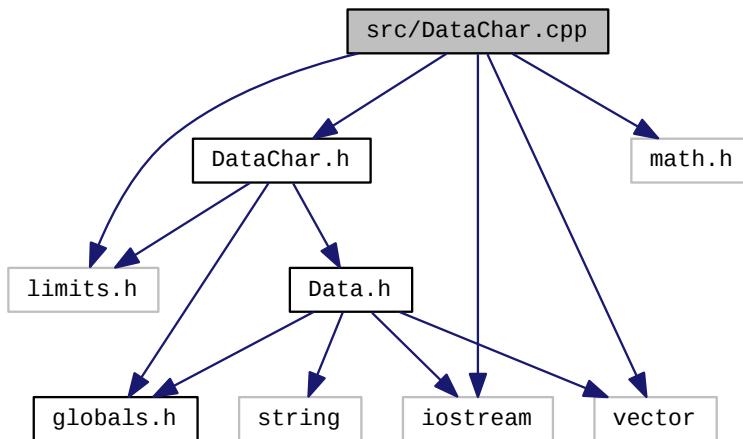


Classes

- class [Data](#)

7.33 src/DataChar.cpp File Reference

```
#include <limits.h>
#include <math.h>
#include <iostream>
#include <vector>
#include "DataChar.h"
Include dependency graph for DataChar.cpp:
```

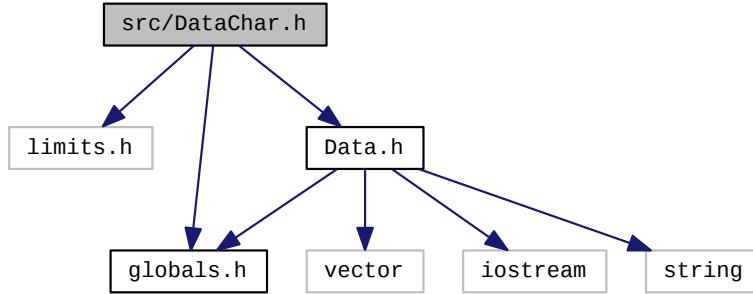


7.34 src/DataChar.h File Reference

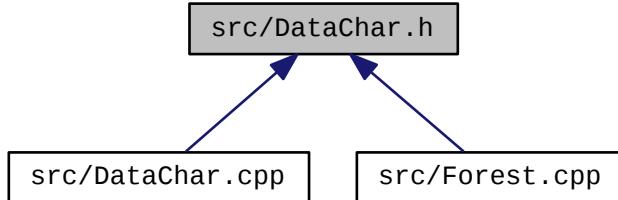
```
#include <limits.h>
#include "globals.h"
```

```
#include "Data.h"
```

Include dependency graph for DataChar.h:



This graph shows which files directly or indirectly include this file:



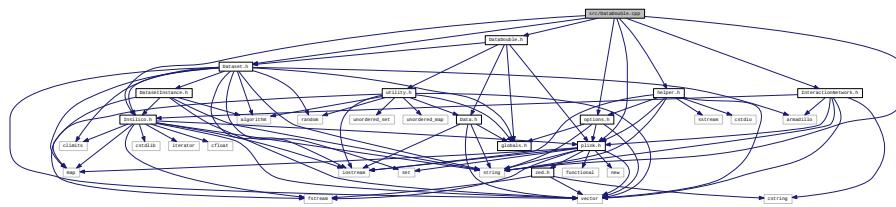
Classes

- class [DataChar](#)

7.35 src/DataDouble.cpp File Reference

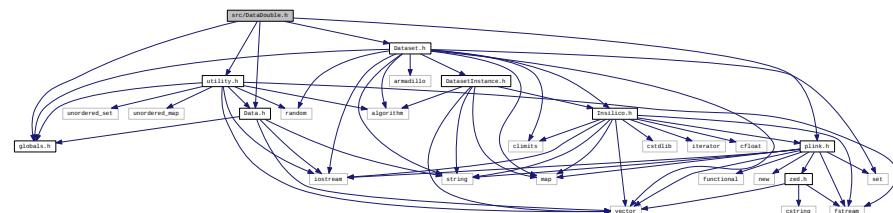
```
#include <string>
#include "DataDouble.h"
#include "Insilico.h"
#include "Dataset.h"
#include "InteractionNetwork.h"
#include "plink.h"
```

```
#include "options.h"
#include "helper.h"
Include dependency graph for DataDouble.cpp:
```

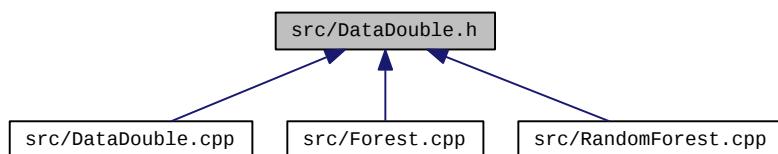


7.36 src/DataDouble.h File Reference

```
#include "globals.h"
#include "utility.h"
#include "Data.h"
#include "Dataset.h"
#include "plink.h"
Include dependency graph for DataDouble.h:
```



This graph shows which files directly or indirectly include this file:

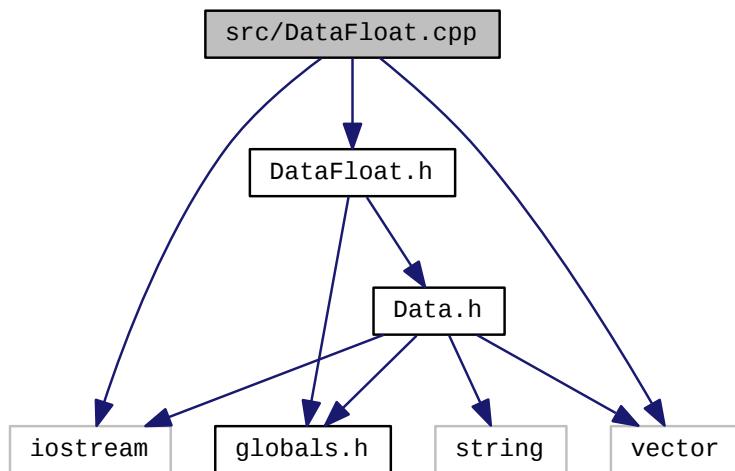


Classes

- class [DataDouble](#)

7.37 src/DataFloat.cpp File Reference

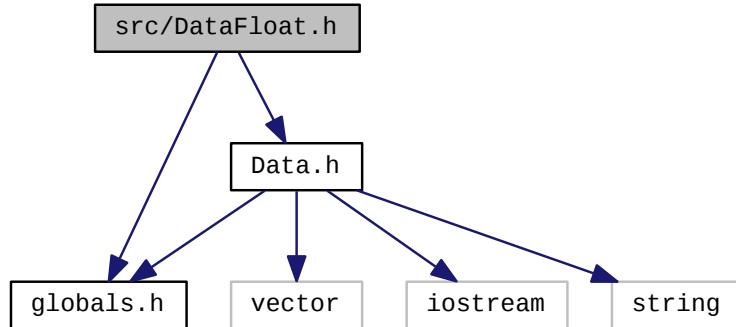
```
#include <iostream>
#include <vector>
#include "DataFloat.h"
Include dependency graph for DataFloat.cpp:
```



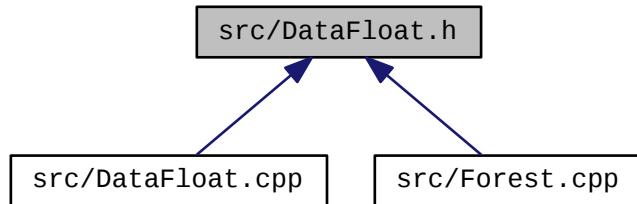
7.38 src/DataFloat.h File Reference

```
#include "globals.h"
#include "Data.h"
```

Include dependency graph for DataFloat.h:



This graph shows which files directly or indirectly include this file:



Classes

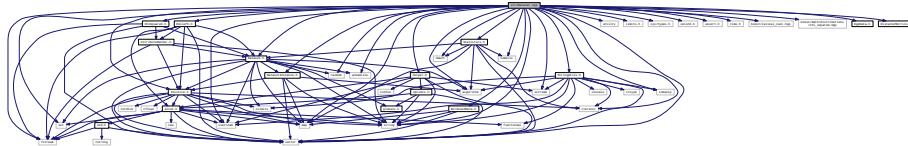
- class [DataFloat](#)

7.39 src/Dataset.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <string>
#include <vector>
#include <set>
```

```
#include <map>
#include <iterator>
#include <cmath>
#include <algorithm>
#include <numeric>
#include <sstream>
#include <utility>
#include <random>
#include <limits.h>
#include <sys/types.h>
#include <unistd.h>
#include <assert.h>
#include <time.h>
#include <boost/lexical_cast.hpp>
#include <boost/math/distributions/chi_squared.hpp>
#include <armadillo>
#include "ChiSquared.h"
#include "Dataset.h"
#include "DatasetInstance.h"
#include "StringUtil.h"
#include "Statistics.h"
#include "Insilico.h"
#include "DgeData.h"
#include "BirdseedData.h"
#include "DistanceMetrics.h"
#include "ReliefF.h"
#include "helper.h"
```

Include dependency graph for Dataset.cpp:



Variables

- [Plink * PP](#)

7.39.1 Variable Documentation

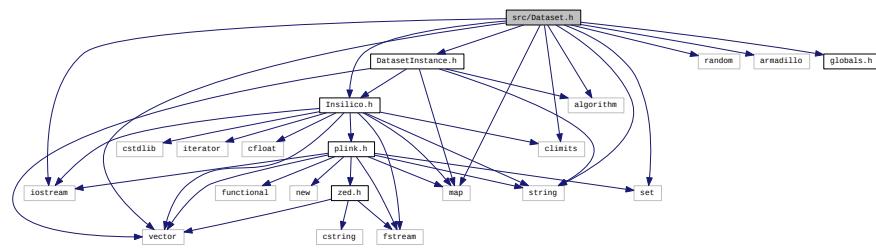
7.39.1.1 PP

[Plink* PP](#)

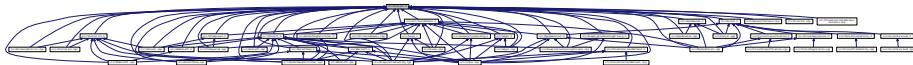
Definition at line 85 of file [inbix.cpp](#).

7.40 src/Dataset.h File Reference

```
#include <iostream>
#include <string>
#include <vector>
#include <map>
#include <set>
#include <algorithm>
#include <climits>
#include <random>
#include <armadillo>
#include "DatasetInstance.h"
#include "Insilico.h"
#include "globals.h"
Include dependency graph for Dataset.h:
```



This graph shows which files directly or indirectly include this file:



Classes

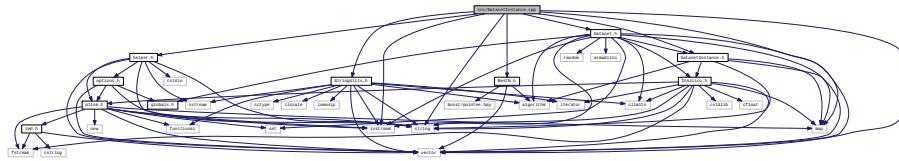
- class `Dataset`

Base class for collections of instances containing attributes and class.

7.41 src/DatasetInstance.cpp File Reference

```
#include <iostream>
#include <string>
#include <vector>
#include <map>
#include "Dataset.h"
#include "DatasetInstance.h"
#include "StringUtils.h"
```

```
#include "BestN.h"
#include "helper.h"
Include dependency graph for DataSetInstance.cpp:
```



Classes

- class [deref_less_bcw](#)

TypeDefs

- typedef [DistancePair T](#)
functor for T comparison

7.41.1 TypeDef Documentation

7.41.1.1 T

```
typedef DistancePair T
```

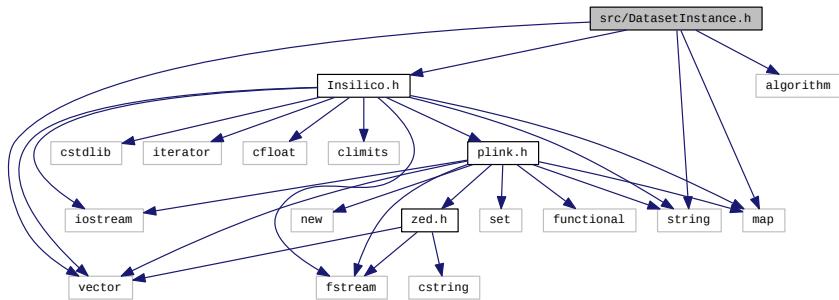
functor for T comparison

Definition at line 23 of file DataSetInstance.cpp.

7.42 src/DataSetInstance.h File Reference

```
#include <string>
#include <vector>
#include <map>
#include <algorithm>
```

```
#include "Insilico.h"
Include dependency graph for DatasetInstance.h:
```



This graph shows which files directly or indirectly include this file:

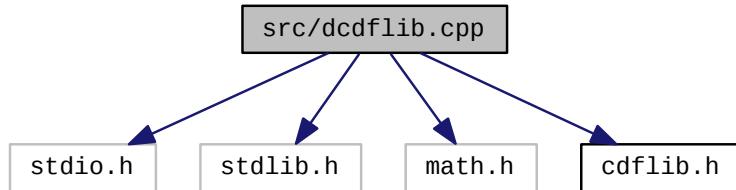


Classes

- class [DatasetInstance](#)
Class to hold dataset instances (rows of attributes).

7.43 src/dcdflib.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "cdflib.h"
Include dependency graph for dcdflib.cpp:
```



Macros

- #define **hln2pi** 0.91893853320467274178e0
- #define **tol** (1.0e-8)
- #define **atol** (1.0e-50)
- #define **zero** (1.0e-300)
- #define **inf** 1.0e300
- #define **one** 1.0e0
- #define **atol** (1.0e-50)
- #define **tol** (1.0e-8)
- #define **zero** (1.0e-300)
- #define **inf** 1.0e300
- #define **one** 1.0e0
- #define **tol** (1.0e-8)
- #define **atol** (1.0e-50)
- #define **zero** (1.0e-300)
- #define **inf** 1.0e300
- #define **tent4** 1.0e4
- #define **tol** (1.0e-8)
- #define **atol** (1.0e-50)
- #define **zero** (1.0e-300)
- #define **one** (1.0e0-1.0e-16)
- #define **inf** 1.0e300
- #define **tol** (1.0e-8)
- #define **atol** (1.0e-50)
- #define **zero** (1.0e-300)
- #define **inf** 1.0e300
- #define **tent4** 1.0e4
- #define **tol** (1.0e-8)
- #define **atol** (1.0e-50)
- #define **zero** (1.0e-300)
- #define **one** (1.0e0-1.0e-16)
- #define **inf** 1.0e300
- #define **tol** (1.0e-8)
- #define **atol** (1.0e-50)
- #define **zero** (1.0e-300)
- #define **inf** 1.0e300
- #define **tol** (1.0e-8)
- #define **atol** (1.0e-50)
- #define **inf** 1.0e300
- #define **one** 1.0e0
- #define **tol** (1.0e-8)
- #define **atol** (1.0e-50)
- #define **inf** 1.0e300
- #define **tol** (1.0e-8)
- #define **atol** (1.0e-50)
- #define **zero** (1.0e-300)
- #define **inf** 1.0e300
- #define **maxdf** 1.0e10
- #define **dg**(i) (*df+2.0e0*(double)(i))
- #define **qsmall**(xx) (int)(sum < 1.0e-20 || (xx) < eps*sum)

- #define **qtired**(i) (int)((i) > ntired)
- #define **half** 0.5e0
- #define **done** 1.0e0
- #define **qsmall**(x) (int)(sum < 1.0e-20 || (x) < eps*sum)
- #define **half** 0.5e0
- #define **done** 1.0e0
- #define **maxit** 100
- #define **eps** (1.0e-13)
- #define **r2pi** 0.3989422804014326e0
- #define **nhalf** (-0.5e0)
- #define **dennor**(x) (**r2pi***exp(**nhalf***(x)*(x)))
- #define **qxmon**(zx, zy, zz) (int)((zx) <= (zy) && (zy) <= (zz))
- #define **dlsqpi** 0.91893853320467274177e0
- #define **hln2pi** 0.91893853320467274178e0
- #define **ncoef** 10
- #define **ftol**(zx) (0.5e0*fifdmax1(abstol,reltol*fabs((zx))))

Functions

- double **algdiv** (double ***a**, double ***b**)
- double **alngam** (double ***x**)
- double **alnrel** (double ***a**)
- double **apser** (double ***a**, double ***b**, double ***x**, double ***eps**)
- double **basym** (double ***a**, double ***b**, double *lambda, double ***eps**)
- double **bcorr** (double ***a0**, double ***b0**)
- double **betaln** (double ***a0**, double ***b0**)
- double **bfrac** (double ***a**, double ***b**, double ***x**, double ***y**, double *lambda, double ***eps**)
- void **bgrat** (double ***a**, double ***b**, double ***x**, double ***y**, double ***w**, double ***eps**, int ***ierr**)
- double **bpser** (double ***a**, double ***b**, double ***x**, double ***eps**)
- void **bratio** (double ***a**, double ***b**, double ***x**, double ***y**, double ***w**, double ***w1**, int ***ierr**)
- double **brcmp1** (int ***mu**, double ***a**, double ***b**, double ***x**, double ***y**)
- double **brcmp** (double ***a**, double ***b**, double ***x**, double ***y**)
- double **bup** (double ***a**, double ***b**, double ***x**, double ***y**, int ***n**, double ***eps**)
- void **cdfbet** (int *which, double ***p**, double ***q**, double ***x**, double ***y**, double ***a**, double ***b**, int *status, double *bound)
- void **cdfbin** (int *which, double ***p**, double ***q**, double ***s**, double ***xn**, double ***pr**, double *ompr, int *status, double *bound)
- void **cdfchi** (int *which, double ***p**, double ***q**, double ***x**, double *df, int *status, double *bound)
- void **cdfchn** (int *which, double ***p**, double ***q**, double ***x**, double *df, double *pnocn, int *status, double *bound)
- void **cdff** (int *which, double ***p**, double ***q**, double *f, double *dfn, double *dfd, int *status, double *bound)
- void **cdfffnc** (int *which, double ***p**, double ***q**, double *f, double *dfn, double *dfd, double *phonc, int *status, double *bound)
- void **cdfgam** (int *which, double ***p**, double ***q**, double ***x**, double *shape, double *scale, int *status, double *bound)
- void **cdfnbn** (int *which, double ***p**, double ***q**, double ***s**, double ***xn**, double ***pr**, double *ompr, int *status, double *bound)
- void **cdfnor** (int *which, double ***p**, double ***q**, double ***x**, double *mean, double *sd, int *status, double *bound)
- void **cdfpoi** (int *which, double ***p**, double ***q**, double ***s**, double *xlam, int *status, double *bound)
- void **cdft** (int *which, double ***p**, double ***q**, double *t, double *df, int *status, double *bound)
- void **cumbet** (double ***x**, double ***y**, double ***a**, double ***b**, double ***cum**, double *ccum)

- void **cumbin** (double *s, double *xn, double *pr, double *ompr, double *cum, double *ccum)
- void **cumchi** (double *x, double *df, double *cum, double *ccum)
- void **cumchn** (double *x, double *df, double *pnong, double *cum, double *ccum)
- void **cumf** (double *f, double *dfn, double *dfd, double *cum, double *ccum)
- void **cumfnc** (double *f, double *dfn, double *dfd, double *pnong, double *cum, double *ccum)
- void **cumgam** (double *x, double *a, double *cum, double *ccum)
- void **cumnbn** (double *s, double *xn, double *pr, double *ompr, double *cum, double *ccum)
- void **cumnor** (double *arg, double *result, double *ccum)
- void **cumpoi** (double *s, double *xlam, double *cum, double *ccum)
- void **cumt** (double *t, double *df, double *cum, double *ccum)
- double **dbetrm** (double *a, double *b)
- double **devlpl** (double a[], int *n, double *x)
- double **dexpm1** (double *x)
- double **dinvnr** (double *p, double *q)
- static void **E0000** (int IENTRY, int *status, double *x, double *fx, unsigned long *qleft, unsigned long *qhi, double *zabsst, double *zabsto, double *zbig, double *zrelst, double *zrelo, double *zsmall, double *zstpmu)
- void **dinrv** (int *status, double *x, double *fx, unsigned long *qleft, unsigned long *qhi)
- void **dstinv** (double *zsmall, double *zbig, double *zabsst, double *zrelst, double *zstpmu, double *zabsto, double *zrelo)
- double **dlnor** (double *x)
- double **dln1mx** (double *x)
- double **dln1px** (double *a)
- double **dlnbet** (double *a0, double *b0)
- double **dlngam** (double *a)
- double **dstrem** (double *z)
- double **dt1** (double *p, double *q, double *df)
- static void **E0001** (int IENTRY, int *status, double *x, double *fx, double *xlo, double *xhi, unsigned long *qleft, unsigned long *qhi, double *zabstl, double *zretl, double *zxhi, double *zxlo)
- void **dzror** (int *status, double *x, double *fx, double *xlo, double *xhi, unsigned long *qleft, unsigned long *qhi)
- void **dstzr** (double *zxlo, double *zxhi, double *zabstl, double *zretl)
- double **erf1** (double *x)
- double **erfc1** (int *ind, double *x)
- double **esum** (int *mu, double *x)
- double **exparg** (int *l)
- double **fpser** (double *a, double *b, double *x, double *eps)
- double **gam1** (double *a)
- void **gaminv** (double *a, double *x, double *x0, double *p, double *q, int *ierr)
- double **gamln** (double *a)
- double **gamln1** (double *a)
- double **Xgamm** (double *a)
- void **grat1** (double *a, double *x, double *r, double *p, double *q, double *eps)
- void **gratio** (double *a, double *x, double *ans, double *qans, int *ind)
- double **gsumln** (double *a, double *b)
- double **psi** (double *xx)
- double **rcomp** (double *a, double *xx)
- double **rexp** (double *x)
- double **rlog** (double *x)
- double **rlog1** (double *x)
- double **spmpar** (int *i)
- double **stvaln** (double *p)
- double **fifdint** (double a)

- double **fifdmax1** (double **a**, double **b**)
- double **fifdmin1** (double **a**, double **b**)
- double **fifdsign** (double mag, double sign)
- long **fifidint** (double **a**)
- long **fifmod** (long **a**, long **b**)
- void **ftnstop** (char *msg)

7.43.1 Macro Definition Documentation

7.43.1.1 **atol** [1/10]

```
#define atol (1.0e-50)
```

7.43.1.2 **atol** [2/10]

```
#define atol (1.0e-50)
```

7.43.1.3 **atol** [3/10]

```
#define atol (1.0e-50)
```

7.43.1.4 **atol** [4/10]

```
#define atol (1.0e-50)
```

7.43.1.5 **atol** [5/10]

```
#define atol (1.0e-50)
```

7.43.1.6 atol [6/10]

```
#define atol (1.0e-50)
```

7.43.1.7 atol [7/10]

```
#define atol (1.0e-50)
```

7.43.1.8 atol [8/10]

```
#define atol (1.0e-50)
```

7.43.1.9 atol [9/10]

```
#define atol (1.0e-50)
```

7.43.1.10 atol [10/10]

```
#define atol (1.0e-50)
```

7.43.1.11 dennor

```
#define dennor( x ) (r2pi*exp(nhalf*(x)*(x)))
```

7.43.1.12 dg

```
#define dg( i ) (*df+2.0e0*(double)(i))
```

7.43.1.13 dlsqpi

```
#define dlsqpi 0.91893853320467274177e0
```

7.43.1.14 done [1/2]

```
#define done 1.0e0
```

7.43.1.15 done [2/2]

```
#define done 1.0e0
```

7.43.1.16 eps

```
#define eps (1.0e-13)
```

7.43.1.17 ftol

```
#define ftol( zx ) (0.5e0*fifdmax1(abstol,reltol*fabs((zx))))
```

7.43.1.18 half [1/2]

```
#define half 0.5e0
```

7.43.1.19 half [2/2]

```
#define half 0.5e0
```

7.43.1.20 hln2pi [1/2]

```
#define hln2pi 0.91893853320467274178e0
```

7.43.1.21 hln2pi [2/2]

```
#define hln2pi 0.91893853320467274178e0
```

7.43.1.22 inf [1/10]

```
#define inf 1.0e300
```

7.43.1.23 inf [2/10]

```
#define inf 1.0e300
```

7.43.1.24 inf [3/10]

```
#define inf 1.0e300
```

7.43.1.25 inf [4/10]

```
#define inf 1.0e300
```

7.43.1.26 inf [5/10]

```
#define inf 1.0e300
```

7.43.1.27 inf [6/10]

```
#define inf 1.0e300
```

7.43.1.28 inf [7/10]

```
#define inf 1.0e300
```

7.43.1.29 inf [8/10]

```
#define inf 1.0e300
```

7.43.1.30 inf [9/10]

```
#define inf 1.0e300
```

7.43.1.31 inf [10/10]

```
#define inf 1.0e300
```

7.43.1.32 maxdf

```
#define maxdf 1.0e10
```

7.43.1.33 maxit

```
#define maxit 100
```

7.43.1.34 ncoef

```
#define ncoef 10
```

7.43.1.35 nhalf

```
#define nhalf (-0.5e0)
```

7.43.1.36 one [1/5]

```
#define one 1.0e0
```

7.43.1.37 one [2/5]

```
#define one 1.0e0
```

7.43.1.38 one [3/5]

```
#define one (1.0e0-1.0e-16)
```

7.43.1.39 one [4/5]

```
#define one (1.0e0-1.0e-16)
```

7.43.1.40 one [5/5]

```
#define one 1.0e0
```

7.43.1.41 qsmall [1/2]

```
#define qsmall(
    xx ) (int)(sum < 1.0e-20 || (xx) < eps*sum)
```

7.43.1.42 qsmall [2/2]

```
#define qsmall(
    x ) (int)(sum < 1.0e-20 || (x) < eps*sum)
```

7.43.1.43 qtired

```
#define qtired(
    i ) (int)((i) > ntired)
```

7.43.1.44 qxmon

```
#define qxmon(
    zx,
    zy,
    zz ) (int)((zx) <= (zy) && (zy) <= (zz))
```

7.43.1.45 r2pi

```
#define r2pi 0.3989422804014326e0
```

7.43.1.46 tent4 [1/2]

```
#define tent4 1.0e4
```

7.43.1.47 tent4 [2/2]

```
#define tent4 1.0e4
```

7.43.1.48 tol [1/10]

```
#define tol (1.0e-8)
```

7.43.1.49 tol [2/10]

```
#define tol (1.0e-8)
```

7.43.1.50 tol [3/10]

```
#define tol (1.0e-8)
```

7.43.1.51 tol [4/10]

```
#define tol (1.0e-8)
```

7.43.1.52 tol [5/10]

```
#define tol (1.0e-8)
```

7.43.1.53 tol [6/10]

```
#define tol (1.0e-8)
```

7.43.1.54 tol [7/10]

```
#define tol (1.0e-8)
```

7.43.1.55 tol [8/10]

```
#define tol (1.0e-8)
```

7.43.1.56 tol [9/10]

```
#define tol (1.0e-8)
```

7.43.1.57 tol [10/10]

```
#define tol (1.0e-8)
```

7.43.1.58 zero [1/8]

```
#define zero (1.0e-300)
```

7.43.1.59 zero [2/8]

```
#define zero (1.0e-300)
```

7.43.1.60 zero [3/8]

```
#define zero (1.0e-300)
```

7.43.1.61 zero [4/8]

```
#define zero (1.0e-300)
```

7.43.1.62 zero [5/8]

```
#define zero (1.0e-300)
```

7.43.1.63 zero [6/8]

```
#define zero (1.0e-300)
```

7.43.1.64 zero [7/8]

```
#define zero (1.0e-300)
```

7.43.1.65 zero [8/8]

```
#define zero (1.0e-300)
```

7.43.2 Function Documentation

7.43.2.1 algdiv()

```
double algdiv (
    double * a,
    double * b )
```

Definition at line 33 of file dcdflib.cpp.

7.43.2.2 alngam()

```
double alngam (
    double * x )
```

Definition at line 86 of file dcdflib.cpp.

7.43.2.3 alnrel()

```
double alnrel (
    double * a )
```

Definition at line 198 of file dcdflib.cpp.

7.43.2.4 apser()

```
double apser (
    double * a,
    double * b,
    double * x,
    double * eps )
```

Definition at line 227 of file dcdflib.cpp.

7.43.2.5 basym()

```
double basym (
    double * a,
    double * b,
    double * lambda,
    double * eps )
```

Definition at line 262 of file dcdflib.cpp.

7.43.2.6 bcorr()

```
double bcorr (
    double * a0,
    double * b0 )
```

Definition at line 370 of file dcdflib.cpp.

7.43.2.7 betaln()

```
double betaln (
    double * a0,
    double * b0 )
```

Definition at line 419 of file dcdflib.cpp.

7.43.2.8 bfrac()

```
double bfrac (
    double * a,
    double * b,
    double * x,
    double * y,
    double * lambda,
    double * eps )
```

Definition at line 525 of file dcdflib.cpp.

7.43.2.9 bgrat()

```
void bgrat (
    double * a,
    double * b,
    double * x,
    double * y,
    double * w,
    double * eps,
    int * ierr )
```

Definition at line 592 of file dcdflib.cpp.

7.43.2.10 bpser()

```
double bpser (
    double * a,
    double * b,
    double * x,
    double * eps )
```

Definition at line 674 of file dcdflib.cpp.

7.43.2.11 bratio()

```
void bratio (
    double * a,
    double * b,
    double * x,
    double * y,
    double * w,
    double * wl,
    int * ierr )
```

Definition at line 772 of file dcdflib.cpp.

7.43.2.12 brcmp1()

```
double brcmp1 (
    int * mu,
    double * a,
    double * b,
    double * x,
    double * y )
```

Definition at line 1009 of file dcdflib.cpp.

7.43.2.13 brcomp()

```
double brcomp (
    double * a,
    double * b,
    double * x,
    double * y )
```

Definition at line 1147 of file dcdflib.cpp.

7.43.2.14 bup()

```
double bup (
    double * a,
    double * b,
    double * x,
    double * y,
    int * n,
    double * eps )
```

Definition at line 1285 of file dcdflib.cpp.

7.43.2.15 cdfbet()

```
void cdfbet (
    int * which,
    double * p,
    double * q,
    double * x,
    double * y,
    double * a,
    double * b,
    int * status,
    double * bound )
```

Definition at line 1363 of file dcdflib.cpp.

7.43.2.16 cdfbin()

```
void cdfbin (
    int * which,
    double * p,
    double * q,
    double * s,
    double * xn,
    double * pr,
    double * ompr,
    int * status,
    double * bound )
```

Definition at line 1734 of file dcdflib.cpp.

7.43.2.17 cdfchi()

```
void cdfchi (
    int * which,
    double * p,
    double * q,
    double * x,
    double * df,
    int * status,
    double * bound )
```

Definition at line 2100 of file dcdflib.cpp.

7.43.2.18 cdfchn()

```
void cdfchn (
    int * which,
    double * p,
    double * q,
    double * x,
    double * df,
    double * pnonc,
    int * status,
    double * bound )
```

Definition at line 2382 of file dcdflib.cpp.

7.43.2.19 cdff()

```
void cdff (
    int * which,
    double * p,
    double * q,
    double * f,
    double * dfn,
    double * dfd,
    int * status,
    double * bound )
```

Definition at line 2652 of file dcdflib.cpp.

7.43.2.20 cdffnc()

```
void cdffnc (
    int * which,
    double * p,
    double * q,
    double * f,
    double * dfn,
    double * dfd,
    double * phonc,
    int * status,
    double * bound )
```

Definition at line 2965 of file dcdflib.cpp.

7.43.2.21 cdfgam()

```
void cdfgam (
    int * which,
    double * p,
    double * q,
    double * x,
    double * shape,
    double * scale,
    int * status,
    double * bound )
```

Definition at line 3285 of file dcdflib.cpp.

7.43.2.22 cdfnbn()

```
void cdfnbn (
    int * which,
    double * p,
    double * q,
    double * s,
    double * xn,
    double * pr,
    double * ompr,
    int * status,
    double * bound )
```

Definition at line 3578 of file dcdflib.cpp.

7.43.2.23 cdfnor()

```
void cdfnor (
    int * which,
    double * p,
    double * q,
    double * x,
    double * mean,
    double * sd,
    int * status,
    double * bound )
```

Definition at line 3946 of file dcdflib.cpp.

7.43.2.24 cdfpoi()

```
void cdfpoi (
    int * which,
    double * p,
    double * q,
    double * s,
    double * xlam,
    int * status,
    double * bound )
```

Definition at line 4147 of file dcdflib.cpp.

7.43.2.25 cdft()

```
void cdft (
    int * which,
    double * p,
    double * q,
    double * t,
    double * df,
    int * status,
    double * bound )
```

Definition at line 4401 of file dcdflib.cpp.

7.43.2.26 cumbet()

```
void cumbet (
    double * x,
    double * y,
    double * a,
    double * b,
    double * cum,
    double * ccum )
```

Definition at line 4650 of file dcdflib.cpp.

7.43.2.27 cumbin()

```
void cumbin (
    double * s,
    double * xn,
    double * pr,
    double * ompr,
    double * cum,
    double * ccum )
```

Definition at line 4726 of file dcdflib.cpp.

7.43.2.28 cumchi()

```
void cumchi (
    double * x,
    double * df,
    double * cum,
    double * ccum )
```

Definition at line 4792 of file dcdflib.cpp.

7.43.2.29 cumchn()

```
void cumchn (
    double * x,
    double * df,
    double * pnonc,
    double * cum,
    double * ccum )
```

Definition at line 4842 of file dcdflib.cpp.

7.43.2.30 cumf()

```
void cumf (
    double * f,
    double * dfn,
    double * dfd,
    double * cum,
    double * ccum )
```

Definition at line 5055 of file dcdflib.cpp.

7.43.2.31 cumfnc()

```
void cumfnc (
    double * f,
    double * dfn,
    double * dfd,
    double * pnonc,
    double * cum,
    double * ccum )
```

Definition at line 5138 of file dcdflib.cpp.

7.43.2.32 cumgam()

```
void cumgam (
    double * x,
    double * a,
    double * cum,
    double * ccum )
```

Definition at line 5307 of file dcdflib.cpp.

7.43.2.33 cumnbn()

```
void cumnbn (
    double * s,
    double * xn,
    double * pr,
    double * ompr,
    double * cum,
    double * ccum )
```

Definition at line 5366 of file dcdflib.cpp.

7.43.2.34 cumnor()

```
void cumnor (
    double * arg,
    double * result,
    double * ccum )
```

Definition at line 5431 of file dcdflib.cpp.

7.43.2.35 cumpoi()

```
void cumpoi (
    double * s,
    double * xlam,
    double * cum,
    double * ccum )
```

Definition at line 5640 of file dcdflib.cpp.

7.43.2.36 cumt()

```
void cumt (
    double * t,
    double * df,
    double * cum,
    double * ccum )
```

Definition at line 5691 of file dcdflib.cpp.

7.43.2.37 dbetrm()

```
double dbetrm (
    double * a,
    double * b )
```

Definition at line 5753 of file dcdflib.cpp.

7.43.2.38 devlpl()

```
double devlpl (
    double a[],
    int * n,
    double * x )
```

Definition at line 5803 of file dcdflib.cpp.

7.43.2.39 dexpm1()

```
double dexpm1 (
    double * x )
```

Definition at line 5844 of file dcdflib.cpp.

7.43.2.40 dinvnr()

```
double dinvnr (
    double * p,
    double * q )
```

Definition at line 5894 of file dcdflib.cpp.

7.43.2.41 dinvr()

```
void dinvr (
    int * status,
    double * x,
    double * fx,
    unsigned long * qleft,
    unsigned long * qhi )
```

Definition at line 6207 of file dcdflib.cpp.

7.43.2.42 dlanor()

```
double dlanor (
    double * x )
```

Definition at line 6343 of file dcdflib.cpp.

7.43.2.43 dln1mx()

```
double dln1mx (
    double * x )
```

Definition at line 6402 of file dcdflib.cpp.

7.43.2.44 dln1px()

```
double dln1px (
    double * a )
```

Definition at line 6444 of file dcdflib.cpp.

7.43.2.45 dlnbet()

```
double dlnbet (
    double * a0,
    double * b0 )
```

Definition at line 6504 of file dcdflib.cpp.

7.43.2.46 dlngam()

```
double dlngam (
    double * a )
```

Definition at line 6641 of file dcdflib.cpp.

7.43.2.47 dstinv()

```
void dstinv (
    double * zsmall,
    double * zbig,
    double * zabsst,
    double * zrelst,
    double * zstpmu,
    double * zabsto,
    double * zrelo )
```

Definition at line 6272 of file dcdflib.cpp.

7.43.2.48 dstrem()

```
double dstrem (
    double * z )
```

Definition at line 6722 of file dcdflib.cpp.

7.43.2.49 dstzr()

```
void dstzr (
    double * zxlo,
    double * zxhi,
    double * zabstl,
    double * zreql )
```

Definition at line 7087 of file dcdflib.cpp.

7.43.2.50 dt1()

```
double dt1 (
    double * p,
    double * q,
    double * df )
```

Definition at line 6786 of file dcdflib.cpp.

7.43.2.51 dzror()

```
void dzror (
    int * status,
    double * x,
    double * fx,
    double * xlo,
    double * xhi,
    unsigned long * qleft,
    unsigned long * qhi )
```

Definition at line 7019 of file dcdflib.cpp.

7.43.2.52 E0000()

```
static void E0000 (
    int IENTRY,
    int * status,
    double * x,
    double * fx,
    unsigned long * qleft,
    unsigned long * qhi,
    double * zabsst,
    double * zabsto,
    double * zbig,
    double * zrelst,
    double * zrelto,
    double * zsmall,
    double * zstpmu ) [static]
```

Definition at line 5991 of file dcdflib.cpp.

7.43.2.53 E0001()

```
static void E0001 (
    int IENTRY,
    int * status,
    double * x,
    double * fx,
    double * xlo,
    double * xhi,
    unsigned long * qleft,
    unsigned long * qhi,
    double * zabstl,
    double * zreltl,
    double * zxhi,
    double * zxlo ) [static]
```

Definition at line 6854 of file dcdflib.cpp.

7.43.2.54 erf1()

```
double erf1 (
    double * x )
```

Definition at line 7134 of file dcdflib.cpp.

7.43.2.55 erfc1()

```
double erfc1 (
    int * ind,
    double * x )
```

Definition at line 7202 of file dcdflib.cpp.

7.43.2.56 esum()

```
double esum (
    int * mu,
    double * x )
```

Definition at line 7308 of file dcdflib.cpp.

7.43.2.57 exparg()

```
double exparg (
    int * l )
```

Definition at line 7337 of file dcdflib.cpp.

7.43.2.58 fifdint()

```
double fifdint (
    double a )
```

Definition at line 9132 of file dcdflib.cpp.

7.43.2.59 fifdmax1()

```
double fifdmax1 (
    double a,
    double b )
```

Definition at line 9141 of file dcdflib.cpp.

7.43.2.60 fifdmin1()

```
double fifdmin1 (
    double a,
    double b )
```

Definition at line 9152 of file dcdflib.cpp.

7.43.2.61 fifdsign()

```
double fifdsign (
    double mag,
    double sign )
```

Definition at line 9163 of file dcdflib.cpp.

7.43.2.62 fifidint()

```
long fifidint (
    double a )
```

Definition at line 9176 of file dcdflib.cpp.

7.43.2.63 fifmod()

```
long fifmod (
    long a,
    long b )
```

Definition at line 9186 of file dcdflib.cpp.

7.43.2.64 fpser()

```
double fpser (
    double * a,
    double * b,
    double * x,
    double * eps )
```

Definition at line 7383 of file dcdflib.cpp.

7.43.2.65 ftnstop()

```
void ftnstop (
    char * msg )
```

Definition at line 9196 of file dcdflib.cpp.

7.43.2.66 gam1()

```
double gam1 (
    double * a )
```

Definition at line 7427 of file dcdflib.cpp.

7.43.2.67 gaminv()

```
void gaminv (
    double * a,
    double * x,
    double * x0,
    double * p,
    double * q,
    int * ierr )
```

Definition at line 7487 of file dcdflib.cpp.

7.43.2.68 gamln()

```
double gamln (
    double * a )
```

Definition at line 7849 of file dcdflib.cpp.

7.43.2.69 gamln1()

```
double gamln1 (
    double * a )
```

Definition at line 7902 of file dcdflib.cpp.

7.43.2.70 grat1()

```
void grat1 (
    double * a,
    double * x,
    double * r,
    double * p,
    double * q,
    double * eps )
```

Definition at line 8099 of file dcdflib.cpp.

7.43.2.71 gratio()

```
void gratio (
    double * a,
    double * x,
    double * ans,
    double * qans,
    int * ind )
```

Definition at line 8202 of file dcdflib.cpp.

7.43.2.72 gsumln()

```
double gsumln (
    double * a,
    double * b )
```

Definition at line 8619 of file dcdflib.cpp.

7.43.2.73 psi()

```
double psi (
    double * xx )
```

Definition at line 8646 of file dcdflib.cpp.

7.43.2.74 rcomp()

```
double rcomp (
    double * a,
    double * x )
```

Definition at line 8839 of file dcdflib.cpp.

7.43.2.75 rexp()

```
double rexp (
    double * x )
```

Definition at line 8872 of file dcdflib.cpp.

7.43.2.76 rlog()

```
double rlog (
    double * x )
```

Definition at line 8902 of file dcdflib.cpp.

7.43.2.77 rlog1()

```
double rlog1 (
    double * x )
```

Definition at line 8952 of file dcdflib.cpp.

7.43.2.78 spmpar()

```
double spmpar (
    int * i )
```

Definition at line 9002 of file dcdflib.cpp.

7.43.2.79 stvaln()

```
double stvaln (
    double * p )
```

Definition at line 9068 of file dcdflib.cpp.

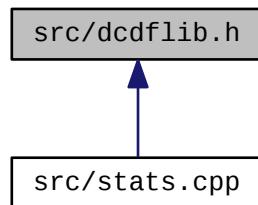
7.43.2.80 Xgamm()

```
double Xgamm (
    double * a )
```

Definition at line 7950 of file dcdflib.cpp.

7.44 src/dcdflib.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- void [cdfbet](#) (int *, double *, double *, double *, double *, int *, double *)
- void [cdfbin](#) (int *, double *, double *, double *, double *, double *, int *, double *)
- void [cdfchi](#) (int *, double *, double *, double *, double *, int *, double *)
- void [cdfchn](#) (int *, double *, double *, double *, double *, int *, double *)
- void [cdff](#) (int *, double *, double *, double *, double *, int *, double *)
- void [cdffnc](#) (int *, double *, double *, double *, double *, double *, int *s, double *)
- void [cdfgam](#) (int *, double *, double *, double *, double *, double *, int *, double *)
- void [cdfnbn](#) (int *, double *, double *, double *, double *, double *, int *, double *)
- void [cdfnor](#) (int *, double *, double *, double *, double *, double *, int *, double *)
- void [cdfpoi](#) (int *, double *, double *, double *, double *, int *, double *)
- void [cdfit](#) (int *, double *, double *, double *, double *, int *, double *)

7.44.1 Function Documentation

7.44.1.1 [cdfbet\(\)](#)

```
void cdfbet (
```

int * ,
double * ,
double * ,
double * ,
double * ,
double * ,
int * ,
double *)

Definition at line 1363 of file `dcdflib.cpp`.

7.44.1.2 `cdfbin()`

```
void cdfbin (
    int * ,
    double * ,
    int * ,
    double * )
```

Definition at line 1734 of file dcdflib.cpp.

7.44.1.3 `cdfchi()`

```
void cdfchi (
    int * ,
    double * ,
    double * ,
    double * ,
    double * ,
    int * ,
    double * )
```

Definition at line 2100 of file dcdflib.cpp.

7.44.1.4 `cdfchn()`

```
void cdfchn (
    int * ,
    double * ,
    int * ,
    double * )
```

Definition at line 2382 of file dcdflib.cpp.

7.44.1.5 cdff()

```
void cdff (
    int * ,
    double * ,
    int * ,
    double * )
```

Definition at line 2652 of file dcdflib.cpp.

7.44.1.6 cdffnc()

```
void cdffnc (
    int * ,
    double * ,
    int * s,
    double * )
```

Definition at line 2965 of file dcdflib.cpp.

7.44.1.7 cdfgam()

```
void cdfgam (
    int * ,
    double * ,
    int * ,
    double * )
```

Definition at line 3285 of file dcdflib.cpp.

7.44.1.8 cdfnbn()

```
void cdfnbn (
    int * ,
    double * ,
    int * ,
    double * )
```

Definition at line 3578 of file dcdflib.cpp.

7.44.1.9 cdfnor()

```
void cdfnor (
    int * ,
    double * ,
    int * ,
    double * )
```

Definition at line 3946 of file dcdflib.cpp.

7.44.1.10 cdfpoi()

```
void cdfpoi (
    int * ,
    double * ,
    double * ,
    double * ,
    double * ,
    int * ,
    double * )
```

Definition at line 4147 of file dcdflib.cpp.

7.44.1.11 cdft()

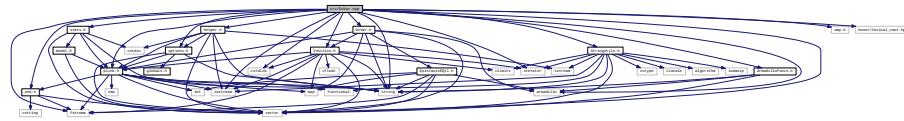
```
void cdft (
    int * ,
    double * ,
    double * ,
    double * ,
    double * ,
    int * ,
    double * )
```

Definition at line 4401 of file dcdflib.cpp.

7.45 src/DcVar.cpp File Reference

```
#include <cstdlib>
#include <iostream>
#include <iterator>
#include <fstream>
#include <string>
#include <vector>
#include <omp.h>
#include <armadillo>
#include <boost/lexical_cast.hpp>
#include "options.h"
#include "plink.h"
#include "helper.h"
#include "zed.h"
#include "stats.h"
#include "DcVar.h"
#include "Insilico.h"
#include "StringUtils.h"
#include "ArmadilloFuncs.h"
```

Include dependency graph for DcVar.cpp:



Functions

- bool [pvalComparatorAscending](#) (const [matrixElement](#) &l, const [matrixElement](#) &r)
- bool [snpinfoComparatorAscending](#) (const [SNP_INFO](#) &l, const [SNP_INFO](#) &r)
- bool [chipseqComparatorAscending](#) (const [CHIP_SEQ_INFO](#) &l, const [CHIP_SEQ_INFO](#) &r)

7.45.1 Function Documentation

7.45.1.1 chipseqComparatorAscending()

```
bool chipseqComparatorAscending (
    const CHIP_SEQ_INFO & l,
    const CHIP_SEQ_INFO & r )
```

Definition at line 46 of file DcVar.cpp.

7.45.1.2 pvalComparatorAscending()

```
bool pvalComparatorAscending (
    const matrixElement & l,
    const matrixElement & r )
```

Definition at line 38 of file DcVar.cpp.

7.45.1.3 snpinfoComparatorAscending()

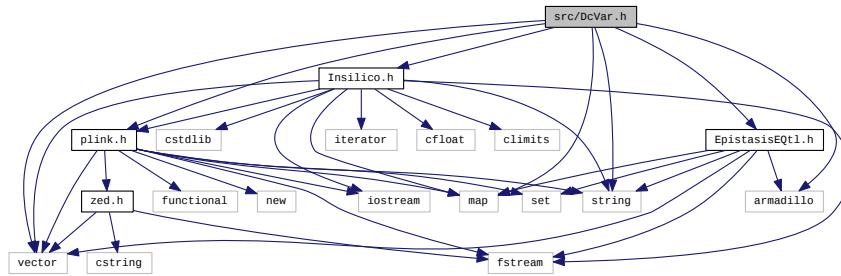
```
bool snpinfoComparatorAscending (
    const SNP_INFO & l,
    const SNP_INFO & r )
```

Definition at line 42 of file DcVar.cpp.

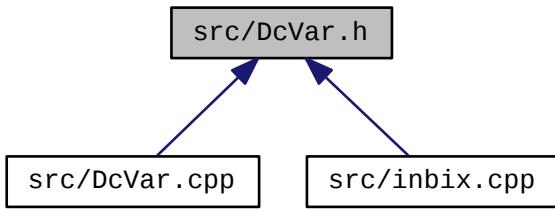
7.46 src/DcVar.h File Reference

```
#include <string>
#include <map>
#include <vector>
#include <armadillo>
#include "plink.h"
#include "Insilico.h"
```

```
#include "EpistasisEQtl.h"
Include dependency graph for DcVar.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- struct `SNP_INFO`
- struct `CHIP_SEQ_INFO`
- class `DcVar`

Typedefs

- typedef std::vector<`SNP_INFO`> `SNP_INFO_LIST`
- typedef std::vector<`SNP_INFO`>::const_iterator `SNP_INFO_LIST_IT`
- typedef std::vector<`CHIP_SEQ_INFO`> `CHIP_SEQ_INFO_LIST`

Enumerations

- enum `SNP_INPUT_TYPE` { `SNP_SRC_PLINK`, `SNP_SRC_FILE` }
- enum `SNP_EXTRACT_FIELD_IDX` {
`SNP_ID` =0, `SNP_CHROM` =1, `SNP_POS` =2, `SNP_REF_ALLELE` =3,
`SNP_DBNSNP_ALLELE` =4 }
- enum `CHIP_SEQ_EXTRACT_FIELD_IDX` { `CHIP_SEQ_CHROM` =0, `CHIP_SEQ_POS` =1, `CHIP_SEQ_EXPR` =12, `CHIP_SEQ_SNP` =15 }

Variables

- const string `CHECKPOINT_FILENAME` = "dcvar.chk"

7.46.1 Typedef Documentation

7.46.1.1 CHIP_SEQ_INFO_LIST

```
typedef std::vector<CHIP_SEQ_INFO> CHIP_SEQ_INFO_LIST
```

Definition at line 58 of file DcVar.h.

7.46.1.2 SNP_INFO_LIST

```
typedef std::vector<SNP_INFO> SNP_INFO_LIST
```

Definition at line 38 of file DcVar.h.

7.46.1.3 SNP_INFO_LIST_IT

```
typedef std::vector<SNP_INFO>::const_iterator SNP_INFO_LIST_IT
```

Definition at line 39 of file DcVar.h.

7.46.2 Enumeration Type Documentation

7.46.2.1 CHIP_SEQ_EXTRACT_FIELD_IDX

```
enum CHIP_SEQ_EXTRACT_FIELD_IDX
```

Enumerator

CHIP_SEQ_CHROM	
CHIP_SEQ_POS	
CHIP_SEQ_EXPR	
CHIP_SEQ_SNP	

Definition at line 49 of file DcVar.h.

7.46.2.2 SNP_EXTRACT_FIELD_IDX

```
enum SNP_EXTRACT_FIELD_IDX
```

Enumerator

SNP_ID	
SNP_CHROM	
SNP_POS	
SNP_REF_ALLELE	
SNP_DBSNP_ALLELE	

Definition at line 29 of file DcVar.h.

7.46.2.3 SNP_INPUT_TYPE

```
enum SNP_INPUT_TYPE
```

Enumerator

SNP_SRC_PLINK	
SNP_SRC_FILE	

Definition at line 23 of file DcVar.h.

7.46.3 Variable Documentation

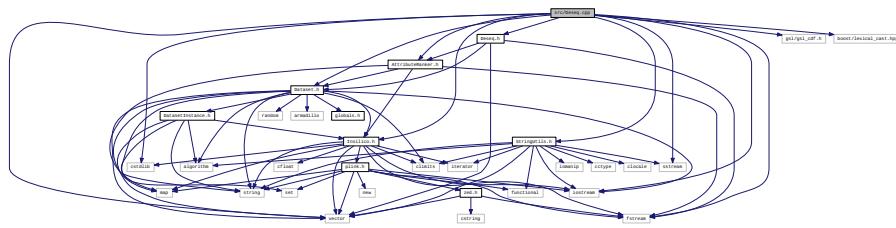
7.46.3.1 CHECKPOINT_FILENAME

```
const string CHECKPOINT_FILENAME = "dcvar.chk"
```

Definition at line 60 of file DcVar.h.

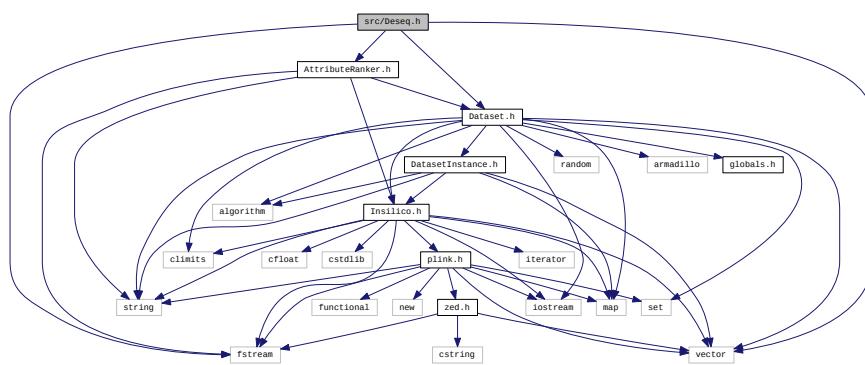
7.47 src/Deseq.cpp File Reference

```
#include <cstdlib>
#include <iostream>
#include <fstream>
#include <sstream>
#include <vector>
#include "gsl/gsl_cdf.h"
#include "boost/lexical_cast.hpp"
#include "AttributeRanker.h"
#include "Deseq.h"
#include "Dataset.h"
#include "Insilico.h"
#include "StringUtils.h"
Include dependency graph for Deseq.cpp:
```

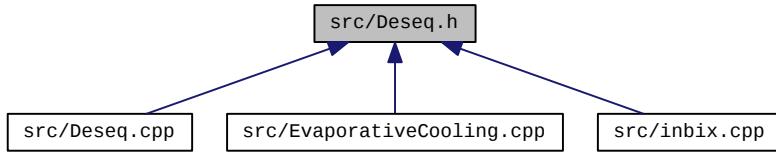


7.48 src/Deseq.h File Reference

```
#include <vector>
#include <fstream>
#include "AttributeRanker.h"
#include "Dataset.h"
Include dependency graph for Deseq.h:
```



This graph shows which files directly or indirectly include this file:



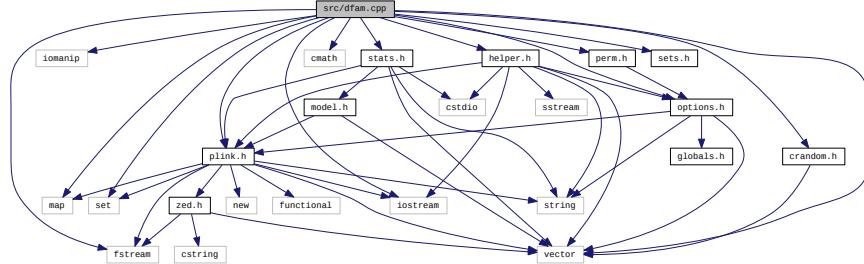
Classes

- class [Deseq](#)
DESeq attribute ranking algorithm.

7.49 src/dfam.cpp File Reference

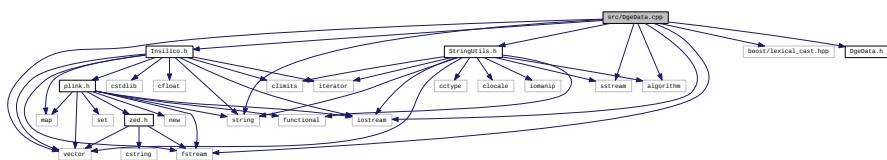
```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <map>
#include <vector>
#include <set>
#include <cmath>
#include "plink.h"
#include "options.h"
#include "helper.h"
#include "crandom.h"
#include "sets.h"
#include "perm.h"
#include "stats.h"
```

Include dependency graph for dfam.cpp:



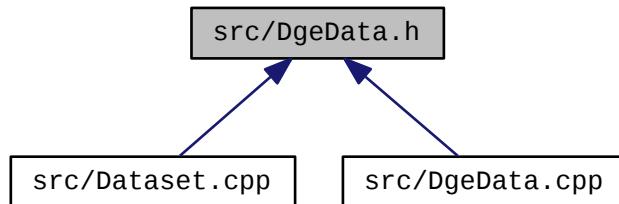
7.50 src/DgeData.cpp File Reference

```
#include <iostream>
#include <fstream>
#include <string>
#include <sstream>
#include <vector>
#include <algorithm>
#include <boost/lexical_cast.hpp>
#include "DgeData.h"
#include "Insilico.h"
#include "StringUtil.h"
Include dependency graph for DgeData.cpp:
```



7.51 src/DgeData.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

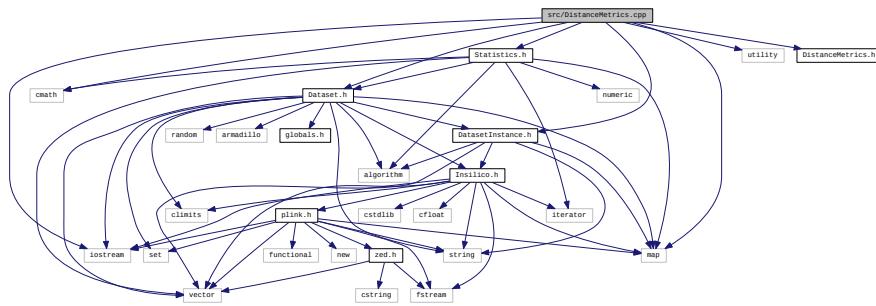
- class [DgeData](#)

Digital gene expression data.

7.52 src/DistanceMetrics.cpp File Reference

```
#include <cmath>
#include <iostream>
#include <map>
#include <utility>
#include "Dataset.h"
#include "DistanceMetrics.h"
#include "DatasetInstance.h"
#include "Statistics.h"

Include dependency graph for DistanceMetrics.cpp:
```



Functions

- pair< bool, double > **CheckMissing** (unsigned int attributeIndex, **DatasetInstance** *dsi1, **DatasetInstance** *dsi2)

Check for a missing discrete value and return value.
- pair< bool, double > **CheckMissingNumeric** (unsigned int numericIndex, **DatasetInstance** *dsi1, **DatasetInstance** *dsi2)

Check for a missing continuous value and return value.
- double **norm** (double x, double minX, double maxX)

Normalizes a given value of a numeric attribute.
- double **diffAMM** (unsigned int attributeIndex, **DatasetInstance** *dsi1, **DatasetInstance** *dsi2)

Allele mismatch metric.
- double **diffGMM** (unsigned int attributeIndex, **DatasetInstance** *dsi1, **DatasetInstance** *dsi2)

Genotype mismatch metric.
- double **diffNCA** (unsigned int attributeIndex, **DatasetInstance** *dsi1, **DatasetInstance** *dsi2)

Nucleotide count array (NCA) metric.
- double **diffTITV** (unsigned int attributeIndex, **DatasetInstance** *dsi1, **DatasetInstance** *dsi2)

Nucleotide count array + G/C sum + A/T sum metric, TITV.
- double **diffKM** (unsigned int attributeIndex, **DatasetInstance** *dsi1, **DatasetInstance** *dsi2)

Kimura distance - considers transition/transversion mutation types.
- double **diffManhattan** (unsigned int attributeIndex, **DatasetInstance** *dsi1, **DatasetInstance** *dsi2)

"Manhattan" distance between continuous attributes.
- double **diffEuclidean** (unsigned int attributeIndex, **DatasetInstance** *dsi1, **DatasetInstance** *dsi2)

Euclidean distance between continuous attributes.
- double **diffPredictedValueTau** (**DatasetInstance** *dsi1, **DatasetInstance** *dsi2)

Same as "Manhattan" distance but uses method calls versus public variables.

7.52.1 Function Documentation

7.52.1.1 CheckMissing()

```
pair<bool, double> CheckMissing (
    unsigned int attributeIndex,
    DatasetInstance * dsi1,
    DatasetInstance * dsi2 )
```

Check for a missing discrete value and return value.

Parameters

in	<i>attributeIndex</i>	index into the vector of attributes
in	<i>dsi1</i>	data set instance 1
in	<i>dsi2</i>	data set instance 2

Returns

pair: has missing value?, value for missing (true) or 0.0 (false)

Definition at line 21 of file DistanceMetrics.cpp.

7.52.1.2 CheckMissingNumeric()

```
pair<bool, double> CheckMissingNumeric (
    unsigned int numericIndex,
    DatasetInstance * dsi1,
    DatasetInstance * dsi2 )
```

Check for a missing continuous value and return value.

Parameters

in	<i>attributeIndex</i>	index into the vector of attributes
in	<i>dsi1</i>	data set instance 1
in	<i>dsi2</i>	data set instance 2

Returns

pair: has missing value?, value for missing (true) or 0.0 (false)

Definition at line 88 of file DistanceMetrics.cpp.

7.52.1.3 diffAMM()

```
double diffAMM (
    unsigned int attributeIndex,
    DatasetInstance * dsi1,
    DatasetInstance * dsi2 )
```

Allele mismatch metric.

Parameters

in	<i>attributeIndex</i>	index into the vector of attributes
in	<i>dsi1</i>	data set instance 1
in	<i>dsi2</i>	data set instance 2

Returns

diff(erence) between attribute values: 0.0, 0.5, 1.0

Definition at line 135 of file DistanceMetrics.cpp.

7.52.1.4 diffEuclidean()

```
double diffEuclidean (
    unsigned int attributeIndex,
    DatasetInstance * dsi1,
    DatasetInstance * dsi2 )
```

Euclidean distance between continuous attributes.

Parameters

in	<i>attributeIndex</i>	index into the vector of attributes
in	<i>dsi1</i>	data set instance 1
in	<i>dsi2</i>	data set instance 2

Returns

absolute value of difference divided by attribute's range

Definition at line 276 of file DistanceMetrics.cpp.

7.52.1.5 diffGMM()

```
double diffGMM (
    unsigned int attributeIndex,
    DatasetInstance * dsi1,
    DatasetInstance * dsi2 )
```

Genotype mismatch metric.

Parameters

in	<i>attributeIndex</i>	index into the vector of attributes
in	<i>dsi1</i>	data set instance 1
in	<i>dsi2</i>	data set instance 2

Returns

diff(erence) between attribute values: 0.0 (same) or 1.0 (not same)

Definition at line 150 of file DistanceMetrics.cpp.

7.52.1.6 diffKM()

```
double diffKM (
    unsigned int attributeIndex,
    DatasetInstance * dsi1,
    DatasetInstance * dsi2 )
```

Kimura distance - considers transition/transversion mutation types.

Parameters

in	<i>attributeIndex</i>	index into the vector of attributes
in	<i>dsi1</i>	data set instance 1
in	<i>dsi2</i>	data set instance 2

Returns

diff(erence) considering nucleotide mutation types

Definition at line 226 of file DistanceMetrics.cpp.

7.52.1.7 diffManhattan()

```
double diffManhattan (
    unsigned int attributeIndex,
    DatasetInstance * dsi1,
    DatasetInstance * dsi2 )
```

"Manhattan" distance between continuous attributes.

Parameters

in	<i>attributeIndex</i>	index into the vector of attributes
in	<i>dsi1</i>	data set instance 1
in	<i>dsi2</i>	data set instance 2

Returns

absolute value of difference divided by attribute's range

Definition at line 251 of file DistanceMetrics.cpp.

7.52.1.8 diffNCA()

```
double diffNCA (
    unsigned int attributeIndex,
    DatasetInstance * dsi1,
    DatasetInstance * dsi2 )
```

Nucleotide count array (NCA) metric.

Parameters

in	<i>attributeIndex</i>	index into the vector of attributes
in	<i>dsi1</i>	data set instance 1
in	<i>dsi2</i>	data set instance 2

Returns

diff(erence) considering nucleotide counts

Definition at line 164 of file DistanceMetrics.cpp.

7.52.1.9 diffPredictedValueTau()

```
double diffPredictedValueTau (
    DatasetInstance * dsi1,
    DatasetInstance * dsi2 )
```

Same as "Manhattan" distance but uses method calls versus public variables.

Parameters

in	<i>attributeIndex</i>	index into the vector of attributes
in	<i>dsi1</i>	data set instance 1
in	<i>dsi2</i>	data set instance 2

Returns

absolute value of difference divided by attribute's range

Definition at line 291 of file DistanceMetrics.cpp.

7.52.1.10 diffTiTV()

```
double diffTiTV (
    unsigned int attributeIndex,
    DatasetInstance * dsi1,
    DatasetInstance * dsi2 )
```

Nucleotide count array + G/C sum + A/T sum metric, TiTv.

considers transition/transversion mutation types

Parameters

in	<i>attributeIndex</i>	index into the vector of attributes
in	<i>dsi1</i>	data set instance 1
in	<i>dsi2</i>	data set instance 2

Returns

diff(erence) considering nucleotide mutation types

Definition at line 206 of file DistanceMetrics.cpp.

7.52.1.11 norm()

```
double norm (
    double x,
    double minX,
    double maxX )
```

Normalizes a given value of a numeric attribute.

Borrowed from Weka 8/18/11

Parameters

in	x	value
in	minX	minimum value for x
in	maxX	maximum value for x

Returns

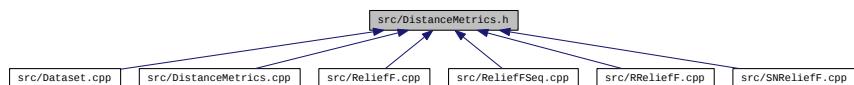
normalized value

Definition at line 127 of file DistanceMetrics.cpp.

7.53 src/DistanceMetrics.h File Reference

Distance metrics for [ReliefF](#).

This graph shows which files directly or indirectly include this file:



Functions

- std::pair< bool, double > [CheckMissing](#) (unsigned int attributeIndex, [DatasetInstance](#) *dsi1, [DatasetInstance](#) *dsi2)

Check for a missing discrete value and return value.

- std::pair< bool, double > [CheckMissingNumeric](#) (unsigned int numericIndex, [DatasetInstance](#) *dsi1, [DatasetInstance](#) *dsi2)

Check for a missing continuous value and return value.

- double [norm](#) (double x, double minX, double maxX)

Normalizes a given value of a numeric attribute.

- double `diffAMM` (unsigned int attributeIndex, `DatasetInstance` *dsi1, `DatasetInstance` *dsi2)
Allele mismatch metric.
- double `diffGMM` (unsigned int attributeIndex, `DatasetInstance` *dsi1, `DatasetInstance` *dsi2)
Genotype mismatch metric.
- double `diffNCA` (unsigned int attributeIndex, `DatasetInstance` *dsi1, `DatasetInstance` *dsi2)
Nucleotide count array (NCA) metric.
- double `diffTITV` (unsigned int attributeIndex, `DatasetInstance` *dsi1, `DatasetInstance` *dsi2)
Nucleotide count array + G/C sum + A/T sum metric, TiTv.
- double `diffKM` (unsigned int attributeIndex, `DatasetInstance` *dsi1, `DatasetInstance` *dsi2)
Kimura distance - considers transition/transversion mutation types.
- double `diffManhattan` (unsigned int attributeIndex, `DatasetInstance` *dsi1, `DatasetInstance` *dsi2)
"Manhattan" distance between continuous attributes.
- double `diffEuclidean` (unsigned int attributeIndex, `DatasetInstance` *dsi1, `DatasetInstance` *dsi2)
Euclidean distance between continuous attributes.
- double `diffPredictedValueTau` (`DatasetInstance` *dsi1, `DatasetInstance` *dsi2)
Same as "Manhattan" distance but uses method calls versus public variables.

7.53.1 Detailed Description

Distance metrics for `ReliefF`.

Author

: Bill White

Version

1.0

Contact: bill.c.white@gmail.com Created on 3/29/11

7.53.2 Function Documentation

7.53.2.1 CheckMissing()

```
std::pair<bool, double> CheckMissing (
    unsigned int attributeIndex,
    DatasetInstance * dsi1,
    DatasetInstance * dsi2 )
```

Check for a missing discrete value and return value.

Parameters

in	<i>attributeIndex</i>	index into the vector of attributes
in	<i>dsi1</i>	data set instance 1
in	<i>dsi2</i>	data set instance 2

Returns

pair: has missing value?, value for missing (true) or 0.0 (false)

Definition at line 21 of file DistanceMetrics.cpp.

7.53.2.2 CheckMissingNumeric()

```
std::pair<bool, double> CheckMissingNumeric (
    unsigned int numericIndex,
    DatasetInstance * dsi1,
    DatasetInstance * dsi2 )
```

Check for a missing continuous value and return value.

Parameters

in	<i>attributeIndex</i>	index into the vector of attributes
in	<i>dsi1</i>	data set instance 1
in	<i>dsi2</i>	data set instance 2

Returns

pair: has missing value?, value for missing (true) or 0.0 (false)

Definition at line 88 of file DistanceMetrics.cpp.

7.53.2.3 diffAMM()

```
double diffAMM (
    unsigned int attributeIndex,
    DatasetInstance * dsi1,
    DatasetInstance * dsi2 )
```

Allele mismatch metric.

Parameters

in	<i>attributeIndex</i>	index into the vector of attributes
in	<i>dsi1</i>	data set instance 1
in	<i>dsi2</i>	data set instance 2

Returns

diff(erence) between attribute values: 0.0, 0.5, 1.0

Definition at line 135 of file DistanceMetrics.cpp.

7.53.2.4 diffEuclidean()

```
double diffEuclidean (
    unsigned int attributeIndex,
    DatasetInstance * dsi1,
    DatasetInstance * dsi2 )
```

Euclidean distance between continuous attributes.

Parameters

in	<i>attributeIndex</i>	index into the vector of attributes
in	<i>dsi1</i>	data set instance 1
in	<i>dsi2</i>	data set instance 2

Returns

absolute value of difference divided by attribute's range

Definition at line 276 of file DistanceMetrics.cpp.

7.53.2.5 diffGMM()

```
double diffGMM (
    unsigned int attributeIndex,
    DatasetInstance * dsi1,
    DatasetInstance * dsi2 )
```

Genotype mismatch metric.

Parameters

in	<i>attributeIndex</i>	index into the vector of attributes
in	<i>dsi1</i>	data set instance 1
in	<i>dsi2</i>	data set instance 2

Returns

diff(erence) between attribute values: 0.0 (same) or 1.0 (not same)

Definition at line 150 of file DistanceMetrics.cpp.

7.53.2.6 diffKM()

```
double diffKM (
    unsigned int attributeIndex,
    DatasetInstance * dsi1,
    DatasetInstance * dsi2 )
```

Kimura distance - considers transition/transversion mutation types.

Parameters

in	<i>attributeIndex</i>	index into the vector of attributes
in	<i>dsi1</i>	data set instance 1
in	<i>dsi2</i>	data set instance 2

Returns

diff(erence) considering nucleotide mutation types

Definition at line 226 of file DistanceMetrics.cpp.

7.53.2.7 diffManhattan()

```
double diffManhattan (
    unsigned int attributeIndex,
    DatasetInstance * dsi1,
    DatasetInstance * dsi2 )
```

"Manhattan" distance between continuous attributes.

Parameters

in	<i>attributeIndex</i>	index into the vector of attributes
in	<i>dsi1</i>	data set instance 1
in	<i>dsi2</i>	data set instance 2

Returns

absolute value of difference divided by attribute's range

Definition at line 251 of file DistanceMetrics.cpp.

7.53.2.8 diffNCA()

```
double diffNCA (
    unsigned int attributeIndex,
    DatasetInstance * dsi1,
    DatasetInstance * dsi2 )
```

Nucleotide count array (NCA) metric.

Parameters

in	<i>attributeIndex</i>	index into the vector of attributes
in	<i>dsi1</i>	data set instance 1
in	<i>dsi2</i>	data set instance 2

Returns

diff(erence) considering nucleotide counts

Definition at line 164 of file DistanceMetrics.cpp.

7.53.2.9 diffPredictedValueTau()

```
double diffPredictedValueTau (
    DatasetInstance * dsi1,
    DatasetInstance * dsi2 )
```

Same as "Manhattan" distance but uses method calls versus public variables.

Parameters

in	<i>attributeIndex</i>	index into the vector of attributes
in	<i>dsi1</i>	data set instance 1
in	<i>dsi2</i>	data set instance 2

Returns

absolute value of difference divided by attribute's range

Definition at line 291 of file DistanceMetrics.cpp.

7.53.2.10 diffTiTV()

```
double diffTiTV (
    unsigned int attributeIndex,
    DatasetInstance * dsi1,
    DatasetInstance * dsi2 )
```

Nucleotide count array + G/C sum + A/T sum metric, TiTv.

considers transition/transversion mutation types

Parameters

in	<i>attributeIndex</i>	index into the vector of attributes
in	<i>dsi1</i>	data set instance 1
in	<i>dsi2</i>	data set instance 2

Returns

diff(erence) considering nucleotide mutation types

Definition at line 206 of file DistanceMetrics.cpp.

7.53.2.11 norm()

```
double norm (
    double x,
    double minX,
    double maxX )
```

Normalizes a given value of a numeric attribute.

Borrowed from Weka 8/18/11

Parameters

in	<i>x</i>	value
in	<i>minX</i>	minimum value for <i>x</i>
in	<i>maxX</i>	maximum value for <i>x</i>

Returns

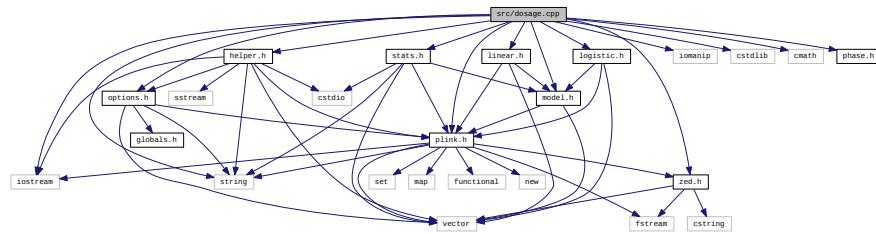
normalized value

Definition at line 127 of file DistanceMetrics.cpp.

7.54 src/dosage.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <string>
#include <cstdlib>
#include <cmath>
#include "options.h"
#include "helper.h"
#include "plink.h"
#include "phase.h"
#include "stats.h"
#include "zed.h"
#include "model.h"
#include "logistic.h"
#include "linear.h"
```

Include dependency graph for dosage.cpp:



Classes

- class [Var](#)

Functions

- void [setUpQScoring](#) (map< string, double > &, vector< double2 > &, vector< string > &)

Variables

- [Plink * PP](#)

7.54.1 Function Documentation

7.54.1.1 setUpQScoring()

```
void setUpQScoring (
    map< string, double > & qscore,
    vector< double2 > & qthresh,
    vector< string > & qlabel )
```

Definition at line 1335 of file dosage.cpp.

7.54.2 Variable Documentation

7.54.2.1 PP

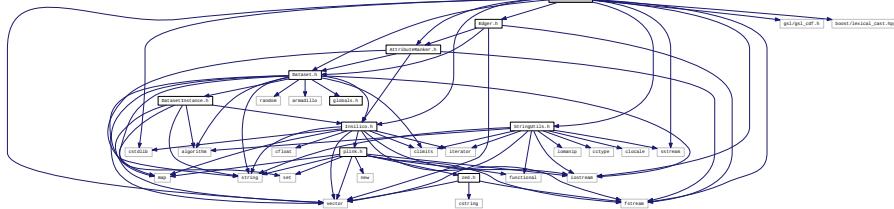
[Plink* PP](#)

Definition at line 85 of file inbix.cpp.

7.55 src/Edger.cpp File Reference

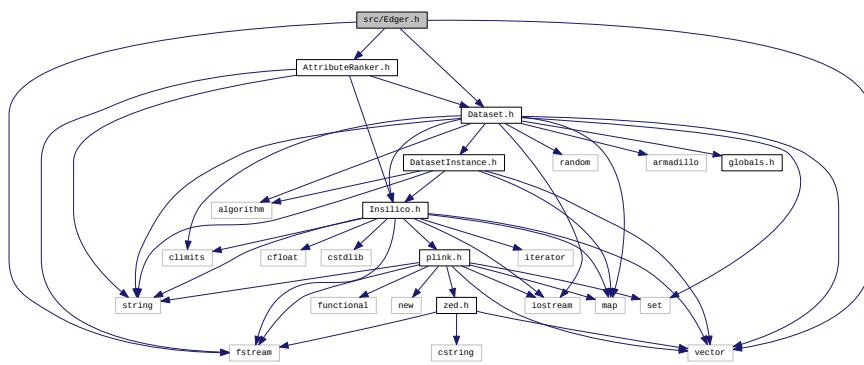
```
#include <cstdlib>
#include <iostream>
#include <fstream>
#include <sstream>
#include <vector>
#include "gsl/gsl_cdf.h"
#include "boost/lexical_cast.hpp"
#include "AttributeRanker.h"
#include "Edger.h"
#include "Dataset.h"
#include "Insilico.h"
#include "StringUtils.h"
```

Include dependency graph for Edger.cpp:

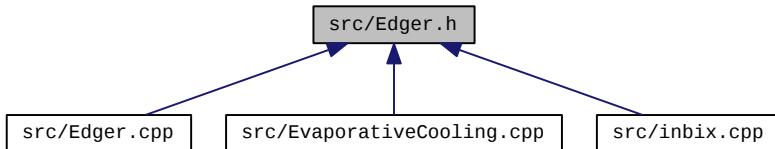


7.56 src/Edger.h File Reference

```
#include <vector>
#include <fstream>
#include "AttributeRanker.h"
#include "Dataset.h"
Include dependency graph for Edger.h:
```



This graph shows which files directly or indirectly include this file:



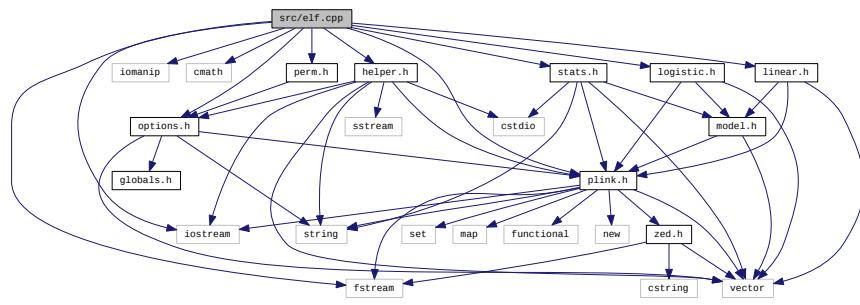
Classes

- class [Edger](#)
Edger attribute ranking algorithm.

7.57 src/elf.cpp File Reference

```
#include <iostream>
#include <fstream>
#include <iomanip>
#include <cmath>
```

```
#include "plink.h"
#include "helper.h"
#include "options.h"
#include "perm.h"
#include "stats.h"
#include "linear.h"
#include "logistic.h"
Include dependency graph for elf.cpp:
```



Classes

- class [RCount](#)

Functions

- void [displayScoresPerson](#) (ofstream &O, [RCount](#) &rc)
- void [displayScoresRegion](#) (ofstream &O, [RCount](#) &rc)

Variables

- [Plink](#) * PP

7.57.1 Function Documentation

7.57.1.1 [displayScoresPerson\(\)](#)

```
void displayScoresPerson (
    ofstream & O,
    RCount & rc )
```

Definition at line 470 of file elf.cpp.

7.57.1.2 `displayScoresRegion()`

```
void displayScoresRegion (
    ofstream & O,
    RCount & rc )
```

Definition at line 486 of file elf.cpp.

7.57.2 Variable Documentation

7.57.2.1 `PP`

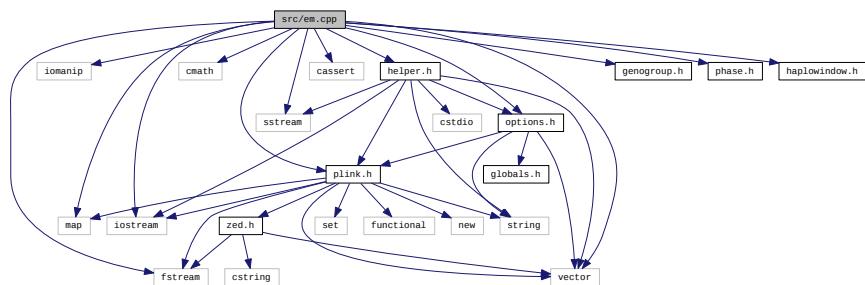
`Plink*` `PP`

Definition at line 85 of file inbix.cpp.

7.58 src/em.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <cmath>
#include <vector>
#include <map>
#include <cassert>
#include "plink.h"
#include "options.h"
#include "helper.h"
#include "genogroup.h"
#include "phase.h"
#include "haplowindow.h"
```

Include dependency graph for em.cpp:



Variables

- ofstream LOG

7.58.1 Variable Documentation

7.58.1.1 LOG

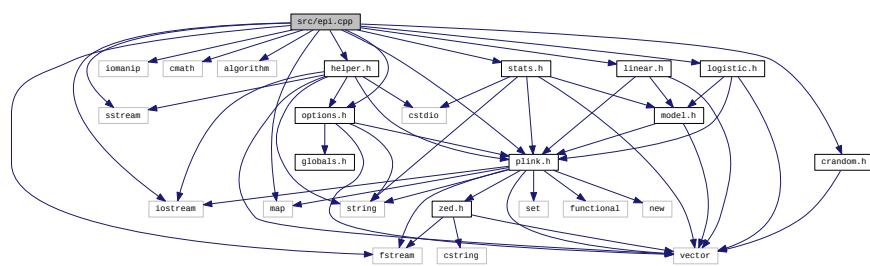
ofstream LOG

Definition at line 81 of file inbix.cpp.

7.59 src/epi.cpp File Reference

```
#include <iostream>
#include <fstream>
#include <sstream>
#include <iomanip>
#include <cmath>
#include <algorithm>
#include <map>
#include "plink.h"
#include "options.h"
#include "helper.h"
#include "crandom.h"
#include "linear.h"
#include "logistic.h"
#include "stats.h"
```

Include dependency graph for epi.cpp:



Variables

- ofstream LOG

7.59.1 Variable Documentation

7.59.1.1 LOG

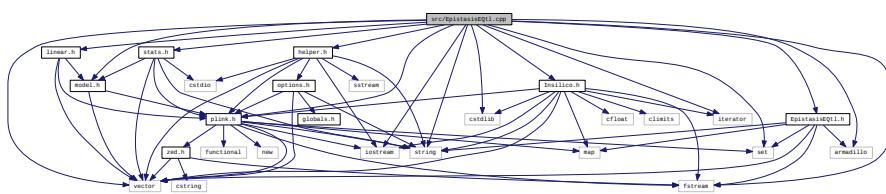
ofstream LOG

Definition at line 81 of file inbix.cpp.

7.60 src/EpistasisEQtl.cpp File Reference

```
#include <cstdlib>
#include <iostream>
#include <iterator>
#include <string>
#include <vector>
#include <set>
#include <fstream>
#include <armadillo>
#include "plink.h"
#include "model.h"
#include "linear.h"
#include "stats.h"
#include "helper.h"
#include "EpistasisEQtl.h"
#include "Insilico.h"
```

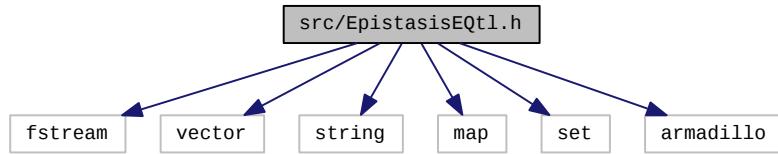
Include dependency graph for EpistasisEQtl.cpp:



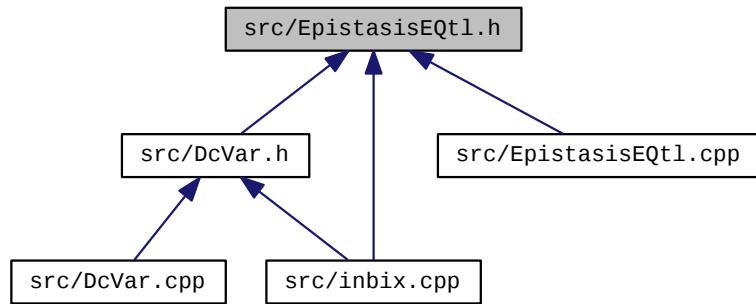
7.61 src/EpistasisEQtl.h File Reference

```
#include <fstream>
#include <vector>
#include <string>
#include <map>
#include <set>
```

```
#include <armadillo>
Include dependency graph for EpistasisEQtl.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [EpistasisEQtl](#)

Typedefs

- `typedef std::map< std::string, std::vector< int > > CoordinateTable`
- `typedef std::map< std::string, std::vector< int > >::const_iterator CoordinateTableCIt`
- `typedef std::map< std::string, std::vector< int > > TranscriptFactorTable`
- `typedef std::map< std::string, std::vector< int > >::const_iterator TranscriptFactorTableCIt`

Enumerations

- enum `COORD_FIELDS { COORD_CHROM, COORD_BP_START, COORD_BP_END }`

7.61.1 Typedef Documentation

7.61.1.1 CoordinateTable

```
typedef std::map<std::string, std::vector<int> > CoordinateTable
```

Definition at line 19 of file EpistasisEQtl.h.

7.61.1.2 CoordinateTableCIt

```
typedef std::map<std::string, std::vector<int> >::const_iterator CoordinateTableCIt
```

Definition at line 20 of file EpistasisEQtl.h.

7.61.1.3 TranscriptFactorTable

```
typedef std::map<std::string, std::vector<int> > TranscriptFactorTable
```

Definition at line 22 of file EpistasisEQtl.h.

7.61.1.4 TranscriptFactorTableCIt

```
typedef std::map<std::string, std::vector<int> >::const_iterator TranscriptFactorTableCIt
```

Definition at line 23 of file EpistasisEQtl.h.

7.61.2 Enumeration Type Documentation

7.61.2.1 COORD_FIELDS

```
enum COORD_FIELDS
```

Enumerator

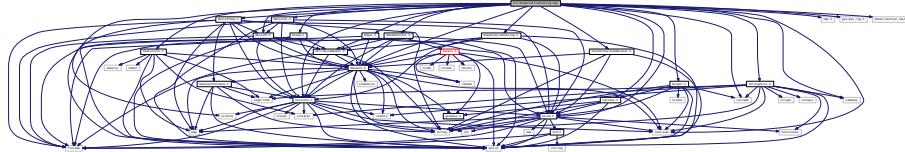
COORD_CHROM	
COORD_BP_START	
COORD_BP_END	

Definition at line 25 of file EpistasisEQtl.h.

7.62 src/EvaporativeCooling.cpp File Reference

```
#include <cstdlib>
#include <iostream>
#include <iomanip>
#include <sstream>
#include <omp.h>
#include <gsl/gsl_rng.h>
#include <boost/lexical_cast.hpp>
#include "plink.h"
#include "options.h"
#include "Insilico.h"
#include "Dataset.h"
#include "Statistics.h"
#include "StringUtils.h"
#include "EvaporativeCooling.h"
#include "Deseq.h"
#include "Edger.h"
#include "RandomForest.h"
#include "ReliefF.h"
#include "RReliefF.h"
#include "ReliefFSeq.h"
#include "PlinkInternalsDataset.h"
#include "helper.h"
```

Include dependency graph for EvaporativeCooling.cpp:

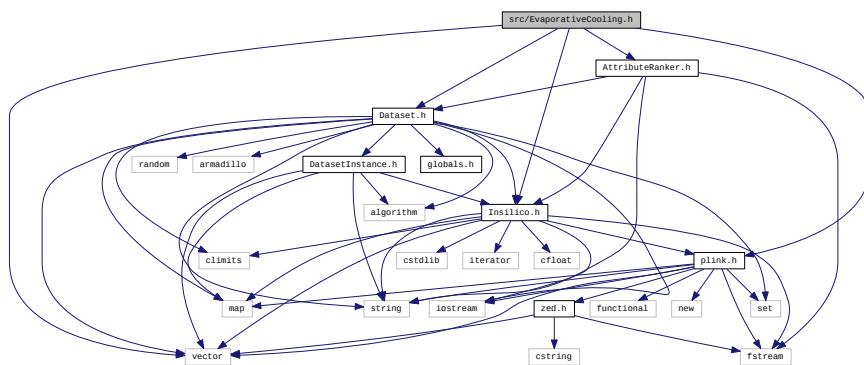


7.63 src/EvaporativeCooling.h File Reference

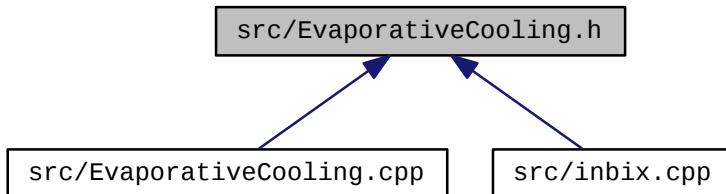
```
#include <vector>
#include "AttributeRanker.h"
#include "Dataset.h"
#include "Insilico.h"
```

```
#include "plink.h"
```

Include dependency graph for EvaporativeCooling.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [EvaporativeCooling](#)

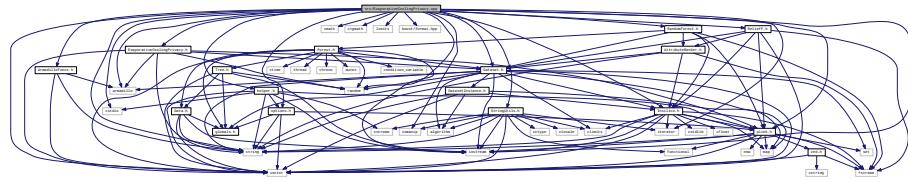
Evaporative Cooling attribute ranking algorithm.

7.64 src/EvaporativeCoolingPrivacy.cpp File Reference

```
#include <cstdio>
#include <iostream>
#include <iomanip>
#include <string>
#include <iterator>
#include <vector>
#include <armadillo>
```

```
#include <random>
#include <cmath>
#include <ctgmath>
#include <limits>
#include <boost/format.hpp>
#include "plink.h"
#include "options.h"
#include "helper.h"
#include "Insilico.h"
#include "ArmadilloFuncs.h"
#include "Dataset.h"
#include "EvaporativeCoolingPrivacy.h"
#include "ReliefF.h"
#include "RandomForest.h"
#include "StringUtils.h"
```

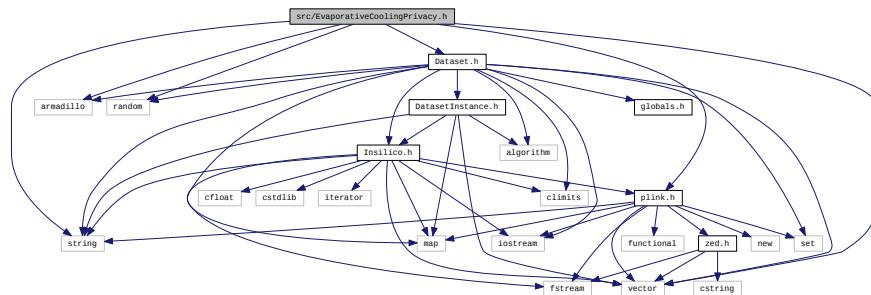
Include dependency graph for EvaporativeCoolingPrivacy.cpp:



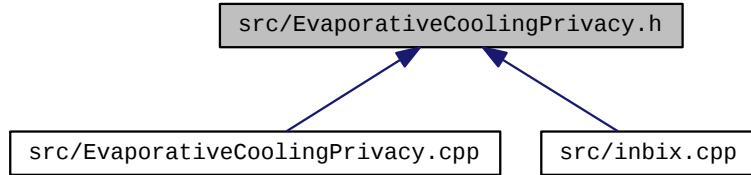
7.65 src/EvaporativeCoolingPrivacy.h File Reference

```
#include <string>
#include <vector>
#include <armadillo>
#include <random>
#include "plink.h"
#include "Dataset.h"
```

Include dependency graph for EvaporativeCoolingPrivacy.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [EvaporativeCoolingPrivacy](#)

TypeDefs

- typedef std::pair< std::vector< std::string >, std::vector< std::string > > [ResultsLists](#)

Enumerations

- enum [DATASET_TYPE](#) { [TRAIN](#), [HOLDOUT](#), [TEST](#) }

7.65.1 TypeDef Documentation

7.65.1.1 ResultsLists

```
typedef std::pair<std::vector<std::string>, std::vector<std::string> > ResultsLists
```

Definition at line 28 of file [EvaporativeCoolingPrivacy.h](#).

7.65.2 Enumeration Type Documentation

7.65.2.1 DATASET_TYPE

```
enum DATASET\_TYPE
```

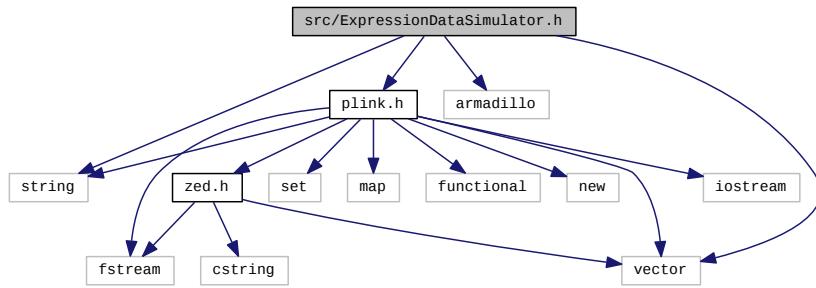
Enumerator

TRAIN	
HOLDOUT	
TEST	

Definition at line 22 of file EvaporativeCoolingPrivacy.h.

7.66 src/ExpressionDataSimulator.h File Reference

```
#include <string>
#include <vector>
#include <armadillo>
#include "plink.h"
Include dependency graph for ExpressionDataSimulator.h:
```



Classes

- class [ExpressionDataSimulator](#)

Enumerations

- enum [SplitType](#) { `TRAIN_SPLIT`, `HOLDOUT_SPLIT`, `TEST_SPLIT`, `NO_SPLIT` }

7.66.1 Enumeration Type Documentation

7.66.1.1 SplitType

```
enum SplitType
```

Train, holdout and test splits.

Enumerator

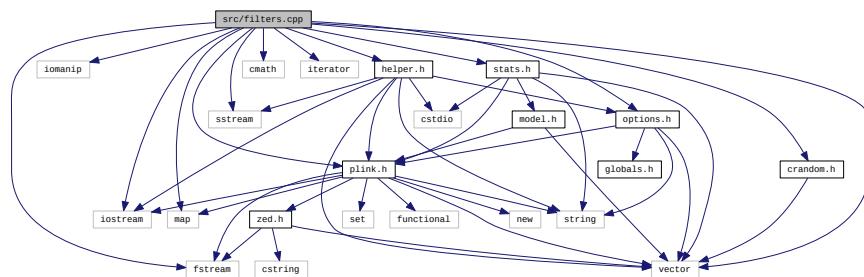
TRAIN_SPLIT	training data split type
HOLDOUT_SPLIT	holdout data split type
TEST_SPLIT	test data split type
NO_SPLIT	default no type

Definition at line 21 of file ExpressionDataSimulator.h.

7.67 src/filters.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <vector>
#include <map>
#include <cmath>
#include <iterator>
#include "plink.h"
#include "options.h"
#include "helper.h"
#include "stats.h"
#include "crandom.h"
```

Include dependency graph for filters.cpp:



Macros

- #define MISSING(i, l) (SNP[l]->one[i] && (! SNP[l]->two[i]))

7.67.1 Macro Definition Documentation

7.67.1.1 MISSING

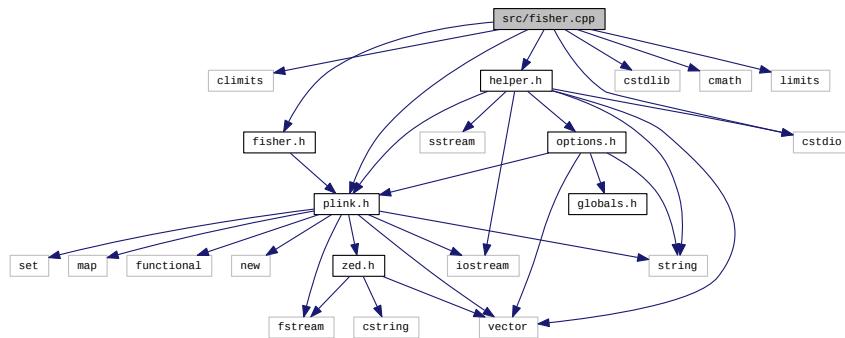
```
#define MISSING(
    i,
    l ) ( SNP[1]->one[i] && ( ! SNP[1]->two[i] ) )
```

Definition at line 30 of file filters.cpp.

7.68 src/fisher.cpp File Reference

```
#include <climits>
#include "fisher.h"
#include "plink.h"
#include "helper.h"
#include <cstdlib>
#include <cstdio>
#include <cmath>
#include <limits>
```

Include dependency graph for fisher.cpp:



Macros

- `#define SINT_MAX INT_MAX`
- `#define max(a, b) ((a) < (b) ? (b) : (a))`
- `#define min(a, b) ((a) > (b) ? (b) : (a))`
- `#define i_real 4`
- `#define i_int 2`
- `#define dwrk (equiv)`
- `#define iwrk ((int *)equiv)`
- `#define rwrk ((float *)equiv)`

Functions

- double `fisher (table_t t)`
- static void `f2xact (int *nrow, int *ncol, double *table, int *ldtbl, double *expect, double *percnt, double *emin, double *prt, double *pre, double *fact, int *ico, int *iro, int *kyy, int *idif, int *irn, int *key, int *ldkey, int *ipoin, double *stp, int *ldstp, int *ifrq, double *dlp, double *dsp, double *tm, int *key2, int *iwk, double *rwk)`
- static void `f3xact (int *nrow, int *irow, int *ncol, int *icol, double *dlp, int *mm, double *fact, int *ico, int *iro, int *it, int *lb, int *nr, int *nt, int *nu, int *itc, int *ist, double *stv, double *alen, const double *tol)`
- static void `f4xact (int *nrow, int *irow, int *ncol, int *icol, double *dsp, double *fact, int *icstk, int *ncstk, int *lstk, int *mstk, int *nstk, int *nrstk, int *irstk, double *ystk, const double *tol)`
- static void `f5xact (double *pastp, const double *tol, int *kval, int *key, int *ldkey, int *ipoin, double *stp, int *ldstp, int *ifrq, int *npoin, int *nr, int *nl, int *ifreq, int *itop, int *ipsh)`
- static void `f6xact (int *nrow, int *irow, int *iflag, int *kyy, int *key, int *ldkey, int *last, int *ipn)`
- static void `f7xact (int *nrow, int *imax, int *idif, int *k, int *ks, int *iflag)`
- static void `f8xact (int *irow, int *is, int *i1, int *izero, int *knew)`
- static double `f9xact (int *n, int *mm, int *ir, double *fact)`
- static void `f10act (int *nrow, int *irow, int *ncol, int *icol, double *val, int *xmin, double *fact, int *nd, int *ne, int *m)`
- static void `f11act (int *irow, int *i1, int *i2, int *knew)`
- static void `prterr (int icode, char *mes)`
- static int `iwork (int iwkmax, int *iwpkt, int number, int itype)`
- static void `isort (int *n, int *ix)`
- static double `gammads (double *y, double *p, int *ifault)`
- static double `alogam (double *x, int *ifault)`
- void `fexact (int *nrow, int *ncol, double *table, int *ldtbl, double *expect, double *percnt, double *emin, double *prt, double *pre, int *workspace)`

7.68.1 Macro Definition Documentation

7.68.1.1 dwrk

```
#define dwrk (equiv)
```

7.68.1.2 i_int

```
#define i_int 2
```

7.68.1.3 i_real

```
#define i_real 4
```

7.68.1.4 iwrk

```
#define iwrk ((int *)equiv)
```

7.68.1.5 max

```
#define max(
    a,
    b ) ((a) < (b) ? (b) : (a))
```

Definition at line 75 of file fisher.cpp.

7.68.1.6 min

```
#define min(
    a,
    b ) ((a) > (b) ? (b) : (a))
```

Definition at line 76 of file fisher.cpp.

7.68.1.7 rwrk

```
#define rwrk ((float *)equiv)
```

7.68.1.8 SINT_MAX

```
#define SINT_MAX INT_MAX
```

Definition at line 71 of file fisher.cpp.

7.68.2 Function Documentation

7.68.2.1 alogam()

```
double alogam (
    double * x,
    int * ifault ) [static]
```

Definition at line 2174 of file fisher.cpp.

7.68.2.2 f10act()

```
void f10act (
    int * nrow,
    int * irow,
    int * ncol,
    int * icol,
    double * val,
    int * xmin,
    double * fact,
    int * nd,
    int * ne,
    int * m ) [static]
```

Definition at line 1883 of file fisher.cpp.

7.68.2.3 f11act()

```
void f11act (
    int * irow,
    int * il,
    int * i2,
    int * knew ) [static]
```

Definition at line 1951 of file fisher.cpp.

7.68.2.4 f2xact()

```
void f2xact (
    int * nrow,
    int * ncol,
    double * table,
    int * ldtabl,
    double * expect,
    double * percnt,
    double * emin,
    double * prt,
    double * pre,
    double * fact,
    int * ico,
    int * iro,
    int * kyy,
    int * idif,
    int * irn,
    int * key,
    int * ldkey,
    int * ipoin,
    double * stp,
    int * ldstp,
    int * ifrq,
    double * dlp,
    double * dsp,
    double * tm,
    int * key2,
    int * iwk,
    double * rwk ) [static]
```

Definition at line 388 of file fisher.cpp.

7.68.2.5 f3xact()

```
void f3xact (
    int * nrow,
    int * irow,
    int * ncol,
    int * icol,
    double * dlp,
    int * mm,
    double * fact,
    int * ico,
    int * iro,
    int * it,
    int * lb,
    int * nr,
    int * nt,
    int * nu,
```

```
    int * itc,
    int * ist,
    double * stv,
    double * alen,
    const double * tol ) [static]
```

Definition at line 919 of file fisher.cpp.

7.68.2.6 f4xact()

```
void f4xact (
    int * nrow,
    int * irow,
    int * ncol,
    int * icol,
    double * dsp,
    double * fact,
    int * icstk,
    int * ncstk,
    int * lstk,
    int * mstk,
    int * nstk,
    int * nrstk,
    int * irstk,
    double * ystk,
    const double * tol ) [static]
```

Definition at line 1268 of file fisher.cpp.

7.68.2.7 f5xact()

```
void f5xact (
    double * pastp,
    const double * tol,
    int * kval,
    int * key,
    int * ldkey,
    int * ipoin,
    double * stp,
    int * ldstp,
    int * ifrq,
    int * npoin,
    int * nr,
    int * nl,
    int * ifreq,
    int * itop,
    int * ipsh ) [static]
```

Definition at line 1493 of file fisher.cpp.

7.68.2.8 f6xact()

```
void f6xact (
    int * nrow,
    int * irow,
    int * iflag,
    int * kyy,
    int * key,
    int * ldkey,
    int * last,
    int * ipn ) [static]
```

Definition at line 1652 of file fisher.cpp.

7.68.2.9 f7xact()

```
void f7xact (
    int * nrow,
    int * imax,
    int * idif,
    int * k,
    int * ks,
    int * iflag ) [static]
```

Definition at line 1703 of file fisher.cpp.

7.68.2.10 f8xact()

```
void f8xact (
    int * irow,
    int * is,
    int * il,
    int * izero,
    int * knew ) [static]
```

Definition at line 1806 of file fisher.cpp.

7.68.2.11 f9xact()

```
double f9xact (
    int * n,
    int * mm,
    int * ir,
    double * fact ) [static]
```

Definition at line 1850 of file fisher.cpp.

7.68.2.12 fexact()

```
void fexact (
    int * nrow,
    int * ncol,
    double * table,
    int * idtbl,
    double * expect,
    double * percnt,
    double * emin,
    double * prt,
    double * pre,
    int * workspace )
```

Definition at line 122 of file fisher.cpp.

7.68.2.13 fisher()

```
double fisher (
    table_t t )
```

Definition at line 19 of file fisher.cpp.

7.68.2.14 gammnds()

```
double gammnds (
    double * y,
    double * p,
    int * ifault ) [static]
```

Definition at line 2091 of file fisher.cpp.

7.68.2.15 isort()

```
void isort (
    int * n,
    int * ix ) [static]
```

Definition at line 2008 of file fisher.cpp.

7.68.2.16 iwork()

```
int iwork (
    int iwkmax,
    int * iwkpt,
    int number,
    int itype ) [static]
```

Definition at line 1987 of file fisher.cpp.

7.68.2.17 prterr()

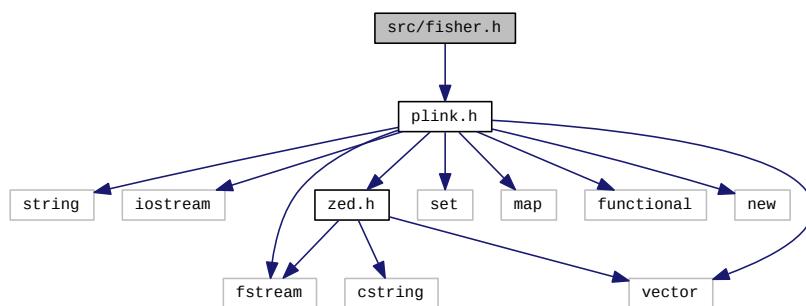
```
void prterr (
    int icode,
    char * mes ) [static]
```

Definition at line 1965 of file fisher.cpp.

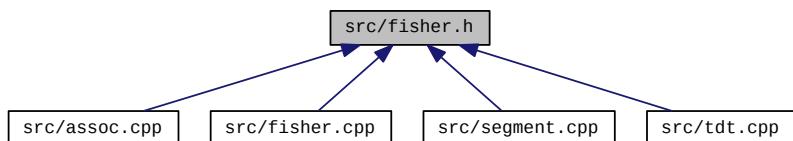
7.69 src/fisher.h File Reference

```
#include "plink.h"
```

Include dependency graph for fisher.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define R_EXT_MEMORY_H_
- #define R_EXT_BOOLEAN_H_
- #define R_EXT_CONSTANTS_H_
- #define M_PI 3.141592653589793238462643383279502884197169399375
- #define PI M_PI
- #define SINGLE_EPS FLT_EPSILON
- #define SINGLE_BASE FLT_RADIX
- #define SINGLE_XMIN FLT_MIN
- #define SINGLE_XMAX FLT_MAX
- #define DOUBLE_DIGITS DBL_MANT_DIG
- #define DOUBLE_EPS DBL_EPSILON
- #define DOUBLE_XMAX DBL_MAX
- #define DOUBLE_XMIN DBL_MIN

Enumerations

- enum Rboolean { FALSE = 0, TRUE }

Functions

- char * vmaxget (void)
- void vmaxset (char *)
- void R_gc (void)
- char * R_alloc (long, int)
- char * S_alloc (long, int)
- char * S_realloc (char *, long, long, int)
- void fexact (int *nrow, int *ncol, double *table, int *ldtbl, double *expect, double *percnt, double *emin, double *prt, double *pre, int *workspace)
- double fisher (table_t t)

7.69.1 Macro Definition Documentation

7.69.1.1 DOUBLE_DIGITS

```
#define DOUBLE_DIGITS DBL_MANT_DIG
```

Definition at line 94 of file fisher.h.

7.69.1.2 DOUBLE_EPS

```
#define DOUBLE_EPS DBL_EPSILON
```

Definition at line 95 of file fisher.h.

7.69.1.3 DOUBLE_XMAX

```
#define DOUBLE_XMAX DBL_MAX
```

Definition at line 96 of file fisher.h.

7.69.1.4 DOUBLE_XMIN

```
#define DOUBLE_XMIN DBL_MIN
```

Definition at line 97 of file fisher.h.

7.69.1.5 M_PI

```
#define M_PI 3.141592653589793238462643383279502884197169399375
```

Definition at line 86 of file fisher.h.

7.69.1.6 PI

```
#define PI M_PI
```

Definition at line 89 of file fisher.h.

7.69.1.7 R_EXT_BOOLEAN_H_

```
#define R_EXT_BOOLEAN_H_
```

Definition at line 66 of file fisher.h.

7.69.1.8 R_EXT_CONSTANTS_H_

```
#define R_EXT_CONSTANTS_H_
```

Definition at line 83 of file fisher.h.

7.69.1.9 R_EXT_MEMORY_H_

```
#define R_EXT_MEMORY_H_
```

Definition at line 43 of file fisher.h.

7.69.1.10 SINGLE_BASE

```
#define SINGLE_BASE FLT_RADIX
```

Definition at line 91 of file fisher.h.

7.69.1.11 SINGLE_EPS

```
#define SINGLE_EPS FLT_EPSILON
```

Definition at line 90 of file fisher.h.

7.69.1.12 SINGLE_XMAX

```
#define SINGLE_XMAX FLT_MAX
```

Definition at line 93 of file fisher.h.

7.69.1.13 SINGLE_XMIN

```
#define SINGLE_XMIN FLT_MIN
```

Definition at line 92 of file fisher.h.

7.69.2 Enumeration Type Documentation

7.69.2.1 Rboolean

```
enum Rboolean
```

Enumerator

FALSE	
TRUE	

Definition at line 74 of file fisher.h.

7.69.3 Function Documentation

7.69.3.1 fexact()

```
void fexact (
    int * nrow,
    int * ncol,
    double * table,
    int * ldtabl,
    double * expect,
    double * percnt,
    double * emin,
    double * prt,
    double * pre,
    int * workspace )
```

Definition at line 122 of file fisher.cpp.

7.69.3.2 fisher()

```
double fisher (
    table_t t )
```

Definition at line 19 of file fisher.cpp.

7.69.3.3 R_alloc()

```
char* R_alloc (
    long ,
    int  )
```

7.69.3.4 R_gc()

```
void R_gc (
    void )
```

7.69.3.5 S_alloc()

```
char* S_alloc (
    long ,
    int )
```

7.69.3.6 S_realloc()

```
char* S_realloc (
    char * ,
    long ,
    long ,
    int )
```

7.69.3.7 vmaxget()

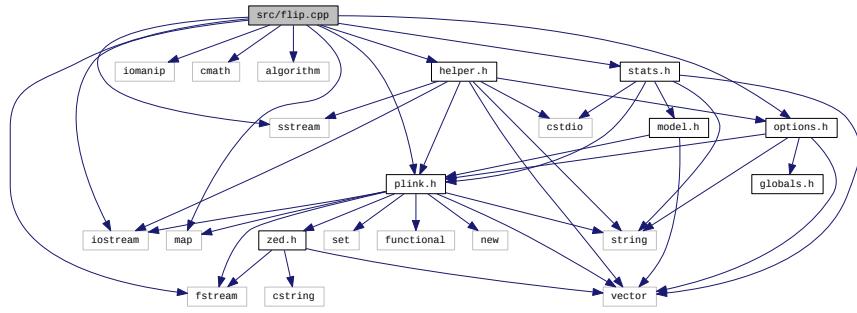
```
char* vmaxget (
    void )
```

7.69.3.8 vmaxset()

```
void vmaxset (
    char * )
```

7.70 src/flip.cpp File Reference

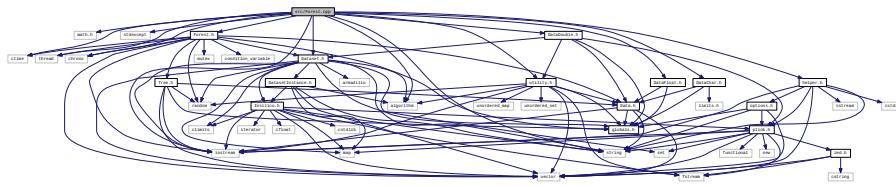
```
#include <iostream>
#include <fstream>
#include <sstream>
#include <iomanip>
#include <cmath>
#include <algorithm>
#include <map>
#include "plink.h"
#include "stats.h"
#include "helper.h"
#include "options.h"
Include dependency graph for flip.cpp:
```



7.71 src/Forest.cpp File Reference

```
#include <math.h>
#include <algorithm>
#include <stdexcept>
#include <string>
#include <ctime>
#include <thread>
#include <chrono>
#include <iostream>
#include "utility.h"
#include "Forest.h"
#include "DataChar.h"
#include "DataDouble.h"
#include "DataFloat.h"
#include "helper.h"
#include "Dataset.h"
```

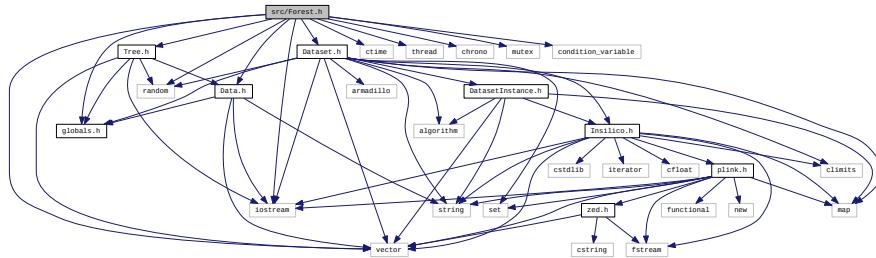
Include dependency graph for Forest.cpp:



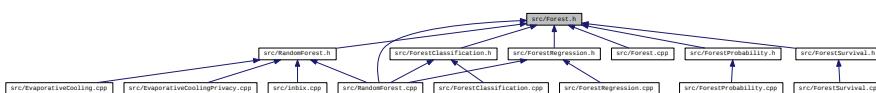
7.72 src/Forest.h File Reference

```
#include <vector>
#include <iostream>
#include <random>
#include <ctime>
#include <thread>
#include <chrono>
#include <mutex>
#include <condition_variable>
#include "globals.h"
#include "Tree.h"
#include "Data.h"
#include "Dataset.h"
Include dependency graph for Forest.h:
```

Include dependency graph for Forest.h:



This graph shows which files directly or indirectly include this file:



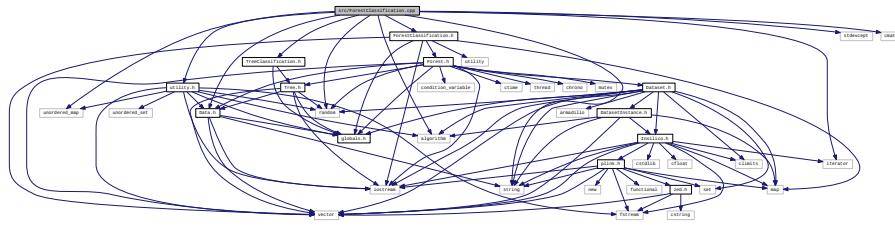
Classes

- class Forest

7.73 src/ForestClassification.cpp File Reference

```
#include <unordered_map>
#include <algorithm>
#include <iterator>
#include <random>
#include <stdexcept>
#include <cmath>
#include <string>
#include "utility.h"
#include "ForestClassification.h"
#include "TreeClassification.h"
#include "Data.h"
```

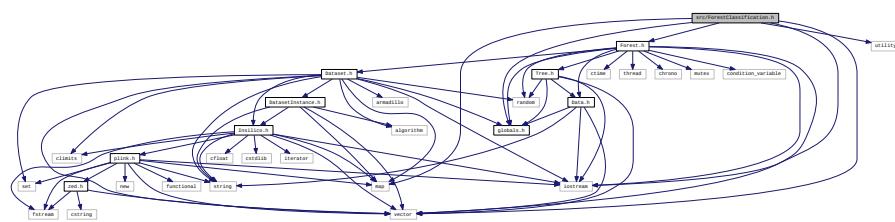
Include dependency graph for ForestClassification.cpp:



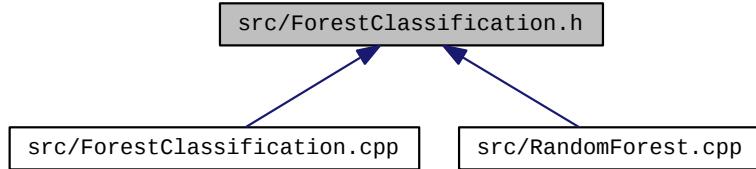
7.74 src/ForestClassification.h File Reference

```
#include <iostream>
#include <map>
#include <utility>
#include <vector>
#include "globals.h"
#include "Forest.h"
```

Include dependency graph for ForestClassification.h:



This graph shows which files directly or indirectly include this file:

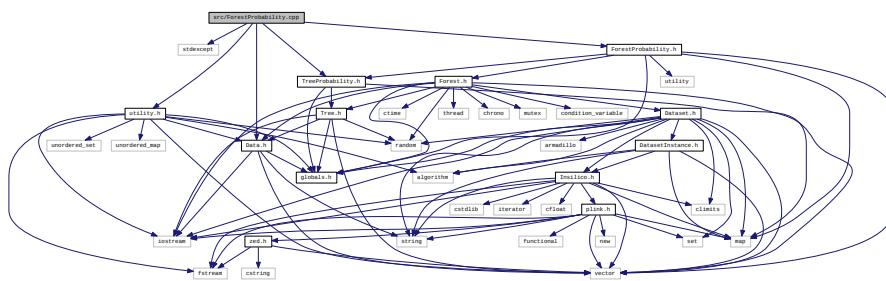


Classes

- class [ForestClassification](#)

7.75 src/ForestProbability.cpp File Reference

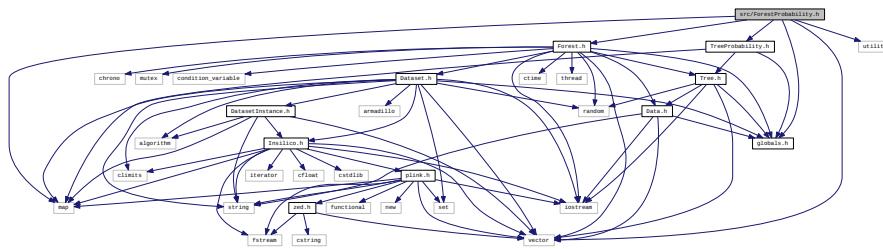
```
#include <stdexcept>
#include "utility.h"
#include "ForestProbability.h"
#include "TreeProbability.h"
#include "Data.h"
Include dependency graph for ForestProbability.cpp:
```



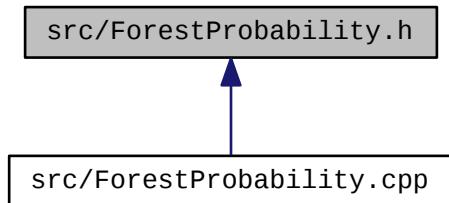
7.76 src/ForestProbability.h File Reference

```
#include <map>
#include <utility>
#include <vector>
#include "globals.h"
#include "Forest.h"
```

```
#include "TreeProbability.h"
Include dependency graph for ForestProbability.h:
```



This graph shows which files directly or indirectly include this file:



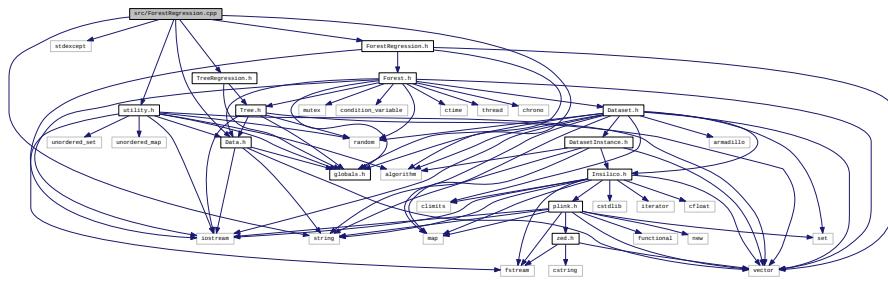
Classes

- class [ForestProbability](#)

7.77 src/ForestRegression.cpp File Reference

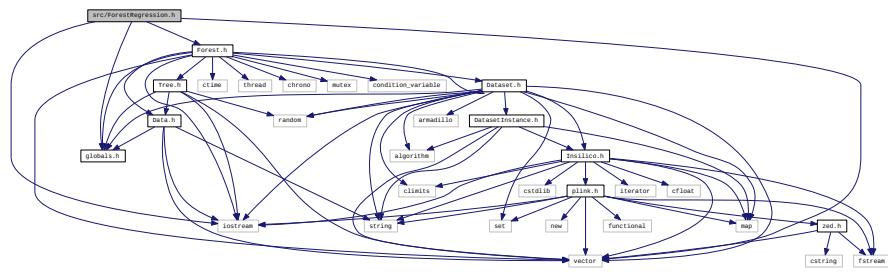
```
#include <algorithm>
#include <stdexcept>
#include <string>
#include "utility.h"
#include "ForestRegression.h"
#include "TreeRegression.h"
```

```
#include "Data.h"
Include dependency graph for ForestRegression.cpp:
```

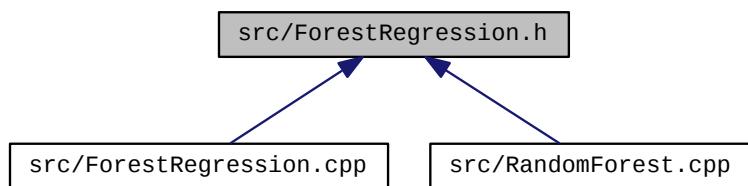


7.78 src/ForestRegression.h File Reference

```
#include <iostream>
#include <vector>
#include "globals.h"
#include "Forest.h"
Include dependency graph for ForestRegression.h:
```



This graph shows which files directly or indirectly include this file:



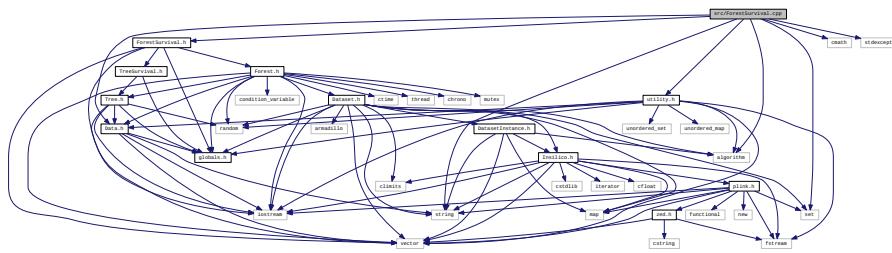
Classes

- class [ForestRegression](#)

7.79 src/ForestSurvival.cpp File Reference

```
#include <set>
#include <algorithm>
#include <cmath>
#include <stdexcept>
#include <string>
#include "utility.h"
#include "ForestSurvival.h"
#include "Data.h"

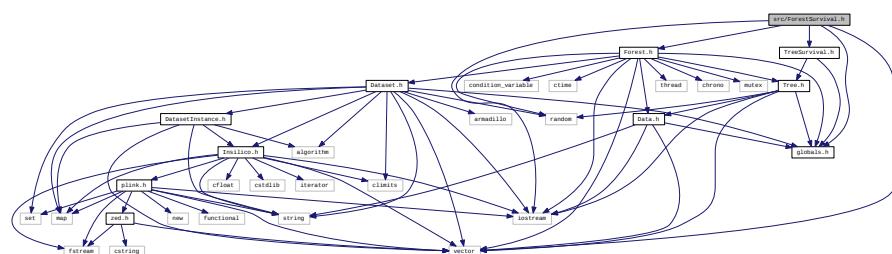
Include dependency graph for ForestSurvival.cpp:
```



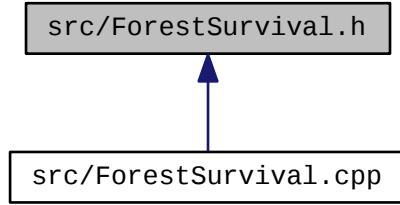
7.80 src/ForestSurvival.h File Reference

```
#include <iostream>
#include <vector>
#include "globals.h"
#include "Forest.h"
#include "TreeSurvival.h"

Include dependency graph for ForestSurvival.h:
```



This graph shows which files directly or indirectly include this file:



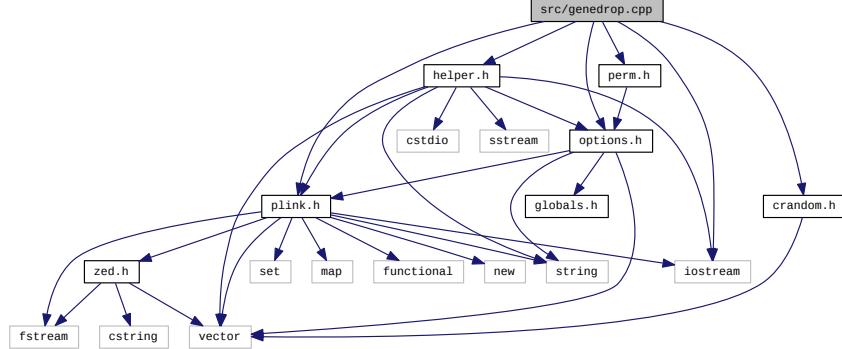
Classes

- class [ForestSurvival](#)

7.81 src/genedrop.cpp File Reference

```
#include <iostream>
#include "plink.h"
#include "perm.h"
#include "options.h"
#include "helper.h"
#include "crandom.h"
```

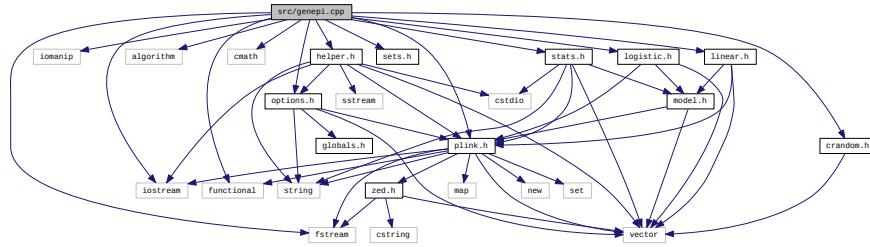
Include dependency graph for genedrop.cpp:



7.82 src/genepi.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <algorithm>
#include <functional>
#include <cmath>
#include "plink.h"
#include "options.h"
#include "sets.h"
#include "helper.h"
#include "stats.h"
#include "crandom.h"
#include "linear.h"
#include "logistic.h"
```

Include dependency graph for genepi.cpp:



Typedefs

- `typedef vector< long double > vector_tld`

Functions

- `double factln (int n)`
- `double bico (int n, int k)`
- `double betaln (double z, double w)`
- `double C (int s, double m, double n)`
- `double I (const double t, const double m, const double n)`
- `long double pillai (int N, int p, int q, double lroot)`
- `long double bartlett (int N, int p, int q, vector_t eigen)`
- `void transposeMatrix (matrix_t &M)`
- `int calcGENEPIMeanVariance (vector< CSNP *> &, int, int, bool, Plink *, vector< double > &, vector< vector< double > &, vector< Individual *> &, vector< int > &, vector< int > &)`
- `void CCA_logit (bool perm, vector< vector< int > &blperm, Set &S, Plink &P)`
- `void CCA_caseonly (bool perm, vector< vector< int > &blperm, Set &S, Plink &P)`

7.82.1 Typedef Documentation

7.82.1.1 `vector_tld`

```
typedef vector<long double> vector_tld
```

Definition at line 33 of file genepi.cpp.

7.82.2 Function Documentation

7.82.2.1 `bartlett()`

```
long double bartlett (
    int N,
    int p,
    int q,
    vector_t eigen )
```

Definition at line 174 of file genepi.cpp.

7.82.2.2 `betaln()`

```
double betaln (
    double z,
    double w )
```

Definition at line 57 of file genepi.cpp.

7.82.2.3 `bico()`

```
double bico (
    int n,
    int k )
```

Definition at line 50 of file genepi.cpp.

7.82.2.4 C()

```
double C (
    int s,
    double m,
    double n )
```

Definition at line 65 of file genepi.cpp.

7.82.2.5 calcGENEPIMeanVariance()

```
int calcGENEPIMeanVariance (
    vector< CSNP *> & pSNP,
    int n1,
    int n2,
    bool perm,
    Plink * P,
    vector< double > & mean,
    vector< vector< double > > & variance,
    vector< Individual *> & sample,
    vector< int > & gp1,
    vector< int > & gp2 )
```

Definition at line 1108 of file genepi.cpp.

7.82.2.6 CCA_caseonly()

```
void CCA_caseonly (
    bool perm,
    vector< vector< int > > & blperm,
    Set & S,
    Plink & P )
```

Definition at line 902 of file genepi.cpp.

7.82.2.7 CCA_logit()

```
void CCA_logit (
    bool perm,
    vector< vector< int > > & blperm,
    Set & S,
    Plink & P )
```

Definition at line 417 of file genepi.cpp.

7.82.2.8 factln()

```
double factln (
    int n )
```

Definition at line 40 of file genepi.cpp.

7.82.2.9 I()

```
double I (
    const double t,
    const double m,
    const double n )
```

Definition at line 82 of file genepi.cpp.

7.82.2.10 pillai()

```
long double pillai (
    int N,
    int p,
    int q,
    double lroot )
```

Definition at line 90 of file genepi.cpp.

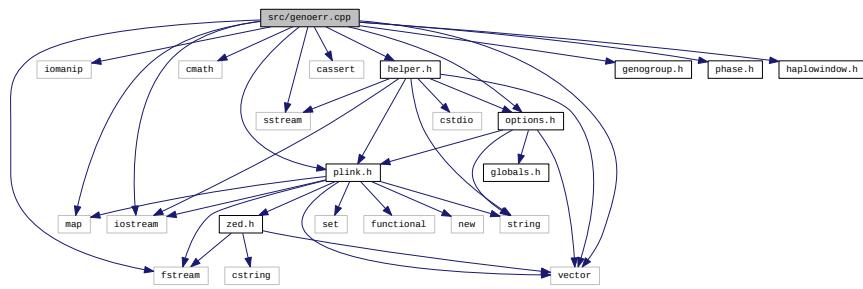
7.82.2.11 transposeMatrix()

```
void transposeMatrix (
    matrix_t & M )
```

Definition at line 190 of file genepi.cpp.

7.83 src/genoerr.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <cmath>
#include <vector>
#include <map>
#include <cassert>
#include "plink.h"
#include "options.h"
#include "helper.h"
#include "genogroup.h"
#include "phase.h"
#include "haplindow.h"
Include dependency graph for genoerr.cpp:
```



Variables

- ofstream LOG

7.83.1 Variable Documentation

7.83.1.1 LOG

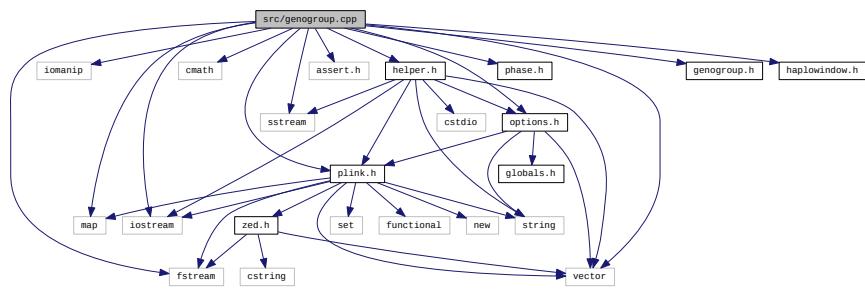
ofstream LOG

Definition at line 81 of file inbix.cpp.

7.84 src/genogroup.cpp File Reference

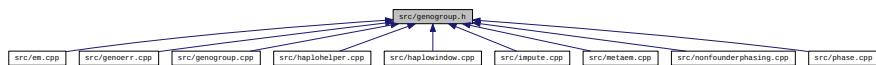
```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <cmath>
#include <vector>
#include <map>
#include <assert.h>
#include "plink.h"
#include "options.h"
#include "phase.h"
#include "helper.h"
#include "genogroup.h"
#include "haplindow.h"

Include dependency graph for genogroup.cpp:
```



7.85 src/genogroup.h File Reference

This graph shows which files directly or indirectly include this file:



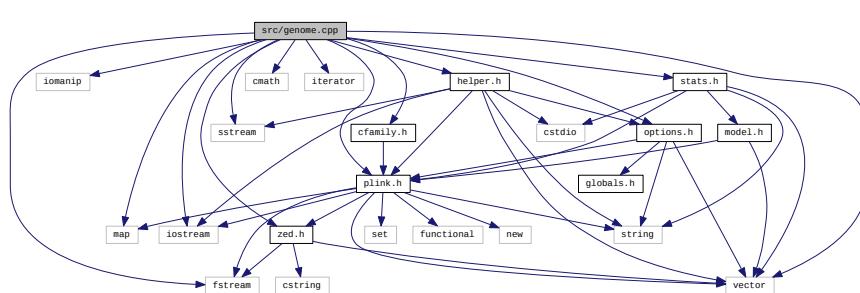
Classes

- class [MultiLocusGenotype](#)
- class [std::less< MultiLocusGenotype * >](#)

7.86 src/genome.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <vector>
#include <map>
#include <cmath>
#include <iterator>
#include "plink.h"
#include "options.h"
#include "helper.h"
#include "stats.h"
#include "cfamily.h"
#include "zed.h"
```

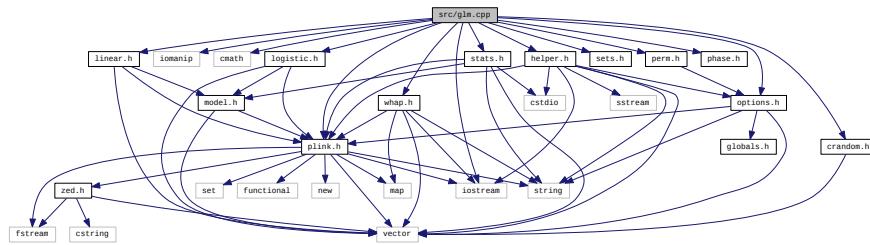
Include dependency graph for genome.cpp:



7.87 src/glm.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <cmath>
#include "linear.h"
#include "logistic.h"
#include "helper.h"
#include "plink.h"
#include "options.h"
#include "crandom.h"
#include "sets.h"
#include "perm.h"
#include "phase.h"
#include "whap.h"
#include "stats.h"
```

Include dependency graph for glm.cpp:



7.88 src/globals.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- `#define DISALLOW_COPY_AND_ASSIGN(TypeName)`
 - `#define M_PI 3.14159265358979323846`

TypeDefs

- `typedef unsigned int uint`

Enumerations

- enum TreeType { TREE_CLASSIFICATION = 1, TREE_REGRESSION = 3, TREE_SURVIVAL = 5, TREE_PROBABILITY = 9 }
 - enum MemoryMode { MEM_DOUBLE = 0, MEM_FLOAT = 1, MEM_CHAR = 2 }
 - enum ImportanceMode {
IMP_NONE = 0, IMP_GINI = 1, IMP_PERM_BREIMAN = 2, IMP_PERM_LIAW = 4,
IMP_PERM_RAW = 3 }
 - enum SplitRule { LOGRANK = 1, AUC = 2, AUC_IGNORE_TIES = 3, MAXSTAT = 4 }
 - enum PredictionType { RESPONSE = 1, TERMINALNODES = 2 }

Variables

- const `uint MAX_MEM_MODE = 2`
- static const `int mask [4] = {192,48,12,3}`
- static const `int offset [4] = {6,4,2,0}`
- const `uint MAX_IMP_MODE = 4`
- const `uint DEFAULT_NUM_TREE = 500`
- const `uint DEFAULT_NUM_THREADS = 0`
- const `ImportanceMode DEFAULT_IMPORTANCE_MODE = IMP_NONE`
- const `uint DEFAULT_MIN_NODE_SIZE_CLASSIFICATION = 1`
- const `uint DEFAULT_MIN_NODE_SIZE_REGRESSION = 5`
- const `uint DEFAULT_MIN_NODE_SIZE_SURVIVAL = 3`
- const `uint DEFAULT_MIN_NODE_SIZE_PROBABILITY = 10`
- const `SplitRule DEFAULT_SPLITRULE = LOGRANK`
- const `double DEFAULT_ALPHA = 0.5`
- const `double DEFAULT_MINPROP = 0.1`
- const `PredictionType DEFAULT_PREDICTIONTYPE = RESPONSE`
- const `double STATUS_INTERVAL = 30.0`
- const `double Q_THRESHOLD = 0.02`

7.88.1 Macro Definition Documentation

7.88.1.1 DISALLOW_COPY_AND_ASSIGN

```
#define DISALLOW_COPY_AND_ASSIGN(  
    TypeName )
```

Value:

```
TypeName(const TypeName&);           \  
void operator=(const TypeName&)
```

Definition at line 32 of file globals.h.

7.88.1.2 M_PI

```
#define M_PI 3.14159265358979323846
```

Definition at line 38 of file globals.h.

7.88.2 Typedef Documentation

7.88.2.1 uint

```
typedef unsigned int uint
```

Definition at line 50 of file globals.h.

7.88.3 Enumeration Type Documentation

7.88.3.1 ImportanceMode

```
enum ImportanceMode
```

Enumerator

IMP_NONE	
IMP_GINI	
IMP_PERM_BREIMAN	
IMP_PERM_LIAW	
IMP_PERM_RAW	

Definition at line 73 of file globals.h.

7.88.3.2 MemoryMode

```
enum MemoryMode
```

Enumerator

MEM_DOUBLE	
MEM_FLOAT	
MEM_CHAR	

Definition at line 61 of file globals.h.

7.88.3.3 PredictionType

```
enum PredictionType
```

Enumerator

RESPONSE	
TERMINALNODES	

Definition at line 91 of file globals.h.

7.88.3.4 SplitRule

```
enum SplitRule
```

Enumerator

LOGRANK	
AUC	
AUC_IGNORE_TIES	
MAXSTAT	

Definition at line 83 of file globals.h.

7.88.3.5 TreeType

```
enum TreeType
```

Enumerator

TREE_CLASSIFICATION	
TREE_REGRESSION	
TREE_SURVIVAL	
TREE_PROBABILITY	

Definition at line 53 of file globals.h.

7.88.4 Variable Documentation

7.88.4.1 DEFAULT_ALPHA

```
const double DEFAULT_ALPHA = 0.5
```

Definition at line 107 of file globals.h.

7.88.4.2 DEFAULT_IMPORTANCE_MODE

```
const ImportanceMode DEFAULT_IMPORTANCE_MODE = IMP_NONE
```

Definition at line 99 of file globals.h.

7.88.4.3 DEFAULT_MIN_NODE_SIZE_CLASSIFICATION

```
const uint DEFAULT_MIN_NODE_SIZE_CLASSIFICATION = 1
```

Definition at line 101 of file globals.h.

7.88.4.4 DEFAULT_MIN_NODE_SIZE_PROBABILITY

```
const uint DEFAULT_MIN_NODE_SIZE_PROBABILITY = 10
```

Definition at line 104 of file globals.h.

7.88.4.5 DEFAULT_MIN_NODE_SIZE_REGRESSION

```
const uint DEFAULT_MIN_NODE_SIZE_REGRESSION = 5
```

Definition at line 102 of file globals.h.

7.88.4.6 DEFAULT_MIN_NODE_SIZE_SURVIVAL

```
const uint DEFAULT_MIN_NODE_SIZE_SURVIVAL = 3
```

Definition at line 103 of file globals.h.

7.88.4.7 DEFAULT_MINPROP

```
const double DEFAULT_MINPROP = 0.1
```

Definition at line 108 of file globals.h.

7.88.4.8 DEFAULT_NUM_THREADS

```
const uint DEFAULT_NUM_THREADS = 0
```

Definition at line 98 of file globals.h.

7.88.4.9 DEFAULT_NUM_TREE

```
const uint DEFAULT_NUM_TREE = 500
```

Definition at line 97 of file globals.h.

7.88.4.10 DEFAULT_PREDICTIONTYPE

```
const PredictionType DEFAULT_PREDICTIONTYPE = RESPONSE
```

Definition at line 110 of file globals.h.

7.88.4.11 DEFAULT_SPLITRULE

```
const SplitRule DEFAULT_SPLITRULE = LOGRANK
```

Definition at line 106 of file globals.h.

7.88.4.12 mask

```
const int mask[4] = {192, 48, 12, 3} [static]
```

Definition at line 69 of file globals.h.

7.88.4.13 MAX_IMP_MODE

```
const uint MAX_IMP_MODE = 4
```

Definition at line 80 of file globals.h.

7.88.4.14 MAX_MEM_MODE

```
const uint MAX_MEM_MODE = 2
```

Definition at line 66 of file globals.h.

7.88.4.15 offset

```
const int offset[4] = {6, 4, 2, 0} [static]
```

Definition at line 70 of file globals.h.

7.88.4.16 Q_THRESHOLD

```
const double Q_THRESHOLD = 0.02
```

Definition at line 116 of file globals.h.

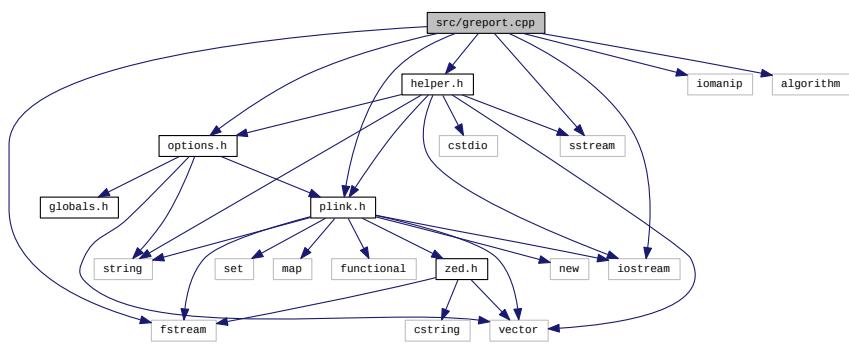
7.88.4.17 STATUS_INTERVAL

```
const double STATUS_INTERVAL = 30.0
```

Definition at line 113 of file globals.h.

7.89 src/greport.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <algorithm>
#include "options.h"
#include "plink.h"
#include "helper.h"
Include dependency graph for greport.cpp:
```



Variables

- [Plink * PP](#)

7.89.1 Variable Documentation

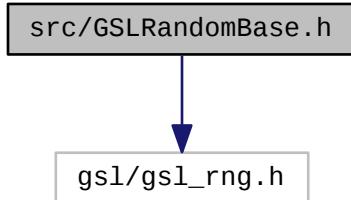
7.89.1.1 PP

[Plink* PP](#)

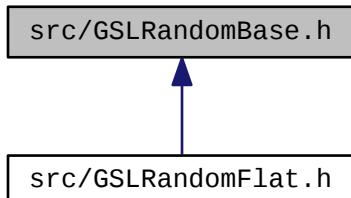
Definition at line 85 of file inbix.cpp.

7.90 src/GSLRandomBase.h File Reference

```
#include "gsl/gsl_rng.h"
Include dependency graph for GSLRandomBase.h:
```



This graph shows which files directly or indirectly include this file:



Classes

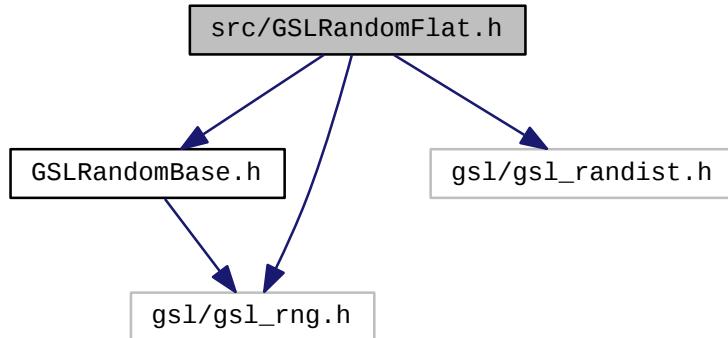
- class [GSLRandomBase](#)

A base class for GNU Scientific Library (GSL) random number functions.

7.91 src/GSLRandomFlat.h File Reference

```
#include "GSLRandomBase.h"
#include "gsl/gsl_rng.h"
```

```
#include "gsl/gsl_randist.h"
Include dependency graph for GSLRandomFlat.h:
```



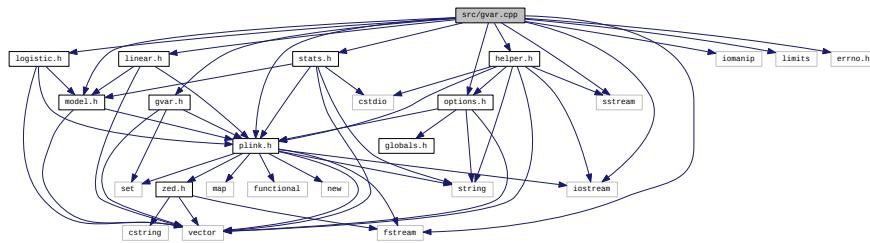
Classes

- class [GSLRandomFlat](#)
Random numbers in a flat, or uniform distribution.

7.92 src/gvar.cpp File Reference

```
#include "gvar.h"
#include "helper.h"
#include "options.h"
#include "plink.h"
#include "model.h"
#include "logistic.h"
#include "linear.h"
#include "stats.h"
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <limits>
#include <errno.h>
```

Include dependency graph for gvar.cpp:



Classes

- class [fullGenotype](#)

Functions

- [Model * analyseModel \(Plink *, Variant *, int, bool, bool\)](#)
- [double compareModels \(Model *, Model *\)](#)

7.92.1 Function Documentation

7.92.1.1 analyseModel()

```
Model * analyseModel (
    Plink * P,
    Variant * v,
    int g,
    bool allelic,
    bool cnv )
```

Definition at line 1009 of file gvar.cpp.

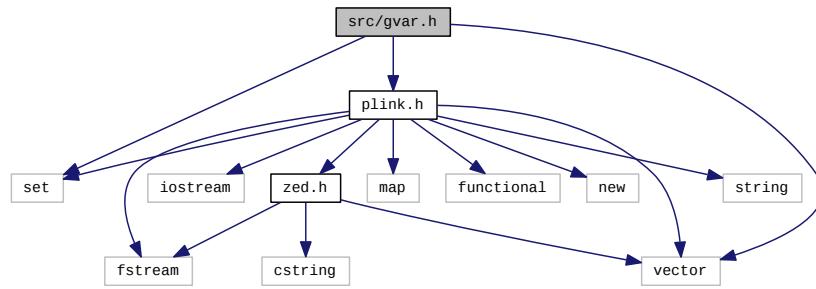
7.92.1.2 compareModels()

```
double compareModels (
    Model * alternate,
    Model * null )
```

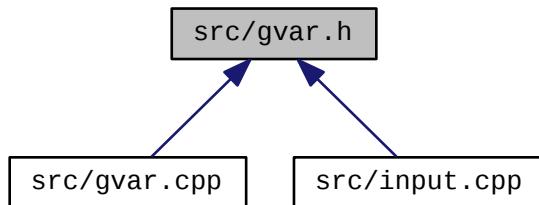
Definition at line 1103 of file gvar.cpp.

7.93 src/gvar.h File Reference

```
#include <vector>
#include <set>
#include "plink.h"
Include dependency graph for gvar.h:
```



This graph shows which files directly or indirectly include this file:



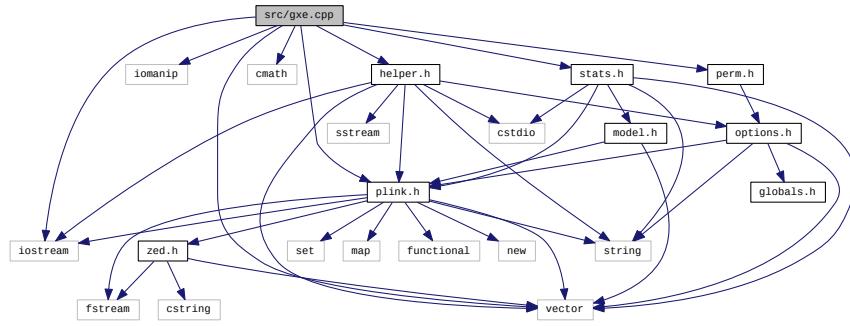
Classes

- class [Variant](#)
- class [GVariant](#)

7.94 src/gxe.cpp File Reference

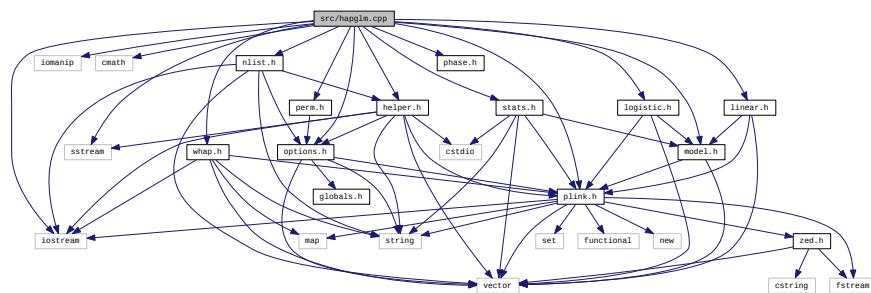
```
#include <iostream>
#include <iomanip>
#include <vector>
```

```
#include <cmath>
#include "plink.h"
#include "helper.h"
#include "stats.h"
#include "perm.h"
Include dependency graph for gxe.cpp:
```



7.95 src/hapglm.cpp File Reference

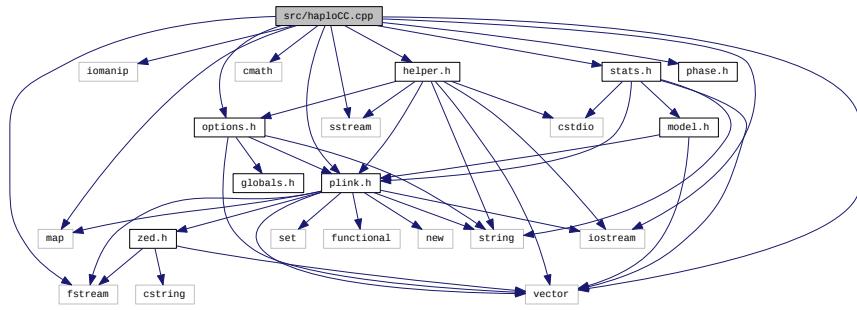
```
#include <iostream>
#include <iomanip>
#include <cmath>
#include <sstream>
#include "whap.h"
#include "helper.h"
#include "plink.h"
#include "options.h"
#include "perm.h"
#include "nlist.h"
#include "phase.h"
#include "model.h"
#include "linear.h"
#include "logistic.h"
#include "stats.h"
Include dependency graph for hapglm.cpp:
```



7.96 src/haploCC.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <cmath>
#include <vector>
#include <map>
#include "plink.h"
#include "options.h"
#include "phase.h"
#include "helper.h"
#include "stats.h"
```

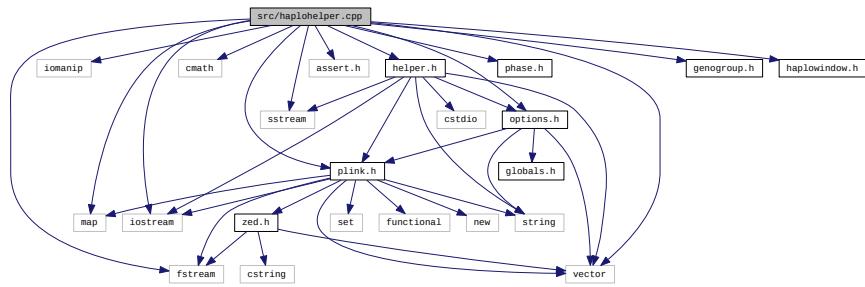
Include dependency graph for haploCC.cpp:



7.97 src/haplohelper.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <cmath>
#include <vector>
#include <map>
#include <assert.h>
#include "plink.h"
#include "options.h"
#include "phase.h"
#include "helper.h"
#include "genogroup.h"
#include "haplowindow.h"
```

Include dependency graph for haplohelper.cpp:



Variables

- `ofstream LOG`

7.97.1 Variable Documentation

7.97.1.1 LOG

`ofstream LOG`

Definition at line 81 of file `inbix.cpp`.

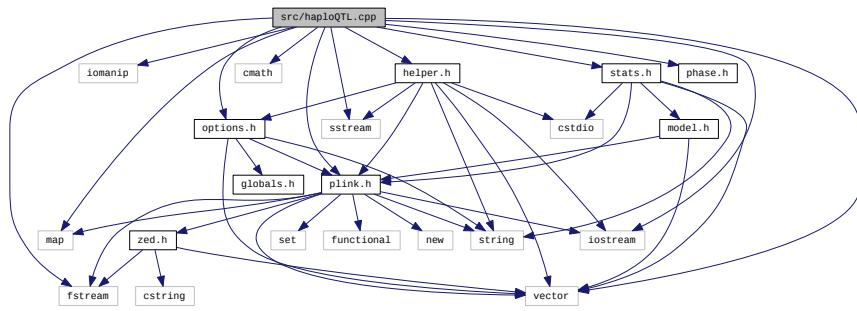
7.98 src/haploQTL.cpp File Reference

```

#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <cmath>
#include <vector>
#include <map>
#include "plink.h"
#include "options.h"
#include "phase.h"
#include "helper.h"
  
```

```
#include "stats.h"
```

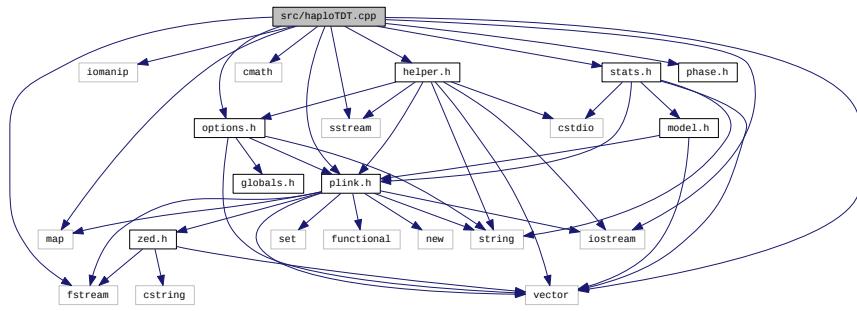
Include dependency graph for haploQTL.cpp:



7.99 src/haploTDT.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <cmath>
#include <vector>
#include <map>
#include "plink.h"
#include "options.h"
#include "phase.h"
#include "helper.h"
#include "stats.h"
```

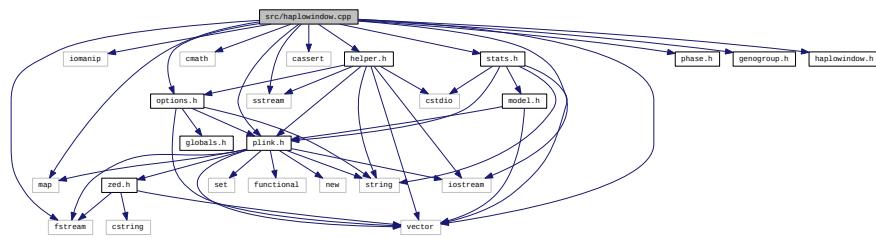
Include dependency graph for haploTDT.cpp:



7.100 src/haplowindow.cpp File Reference

```
#include <iostream>
#include <iomanip>
```

```
#include <fstream>
#include <sstream>
#include <cmath>
#include <vector>
#include <map>
#include <cassert>
#include "plink.h"
#include "options.h"
#include "helper.h"
#include "stats.h"
#include "phase.h"
#include "genogroup.h"
#include "haplindow.h"
Include dependency graph for haplindow.cpp:
```



Functions

- void `verboseDisplayWindows2 (HaploPhase *haplo, int i, bool use_ref=true)`

7.100.1 Function Documentation

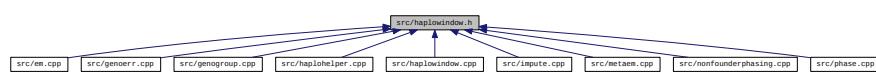
7.100.1.1 verboseDisplayWindows2()

```
void verboseDisplayWindows2 (
    HaploPhase * haplo,
    int i,
    bool use_ref = true )
```

Definition at line 35 of file `haplindow.cpp`.

7.101 src/haplindow.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

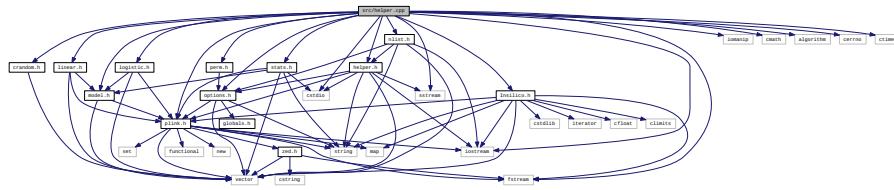
- class HaploWindow

7.102 src/helper.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <cmath>
#include <algorithm>
#include <cerrno>
#include <cstdio>
#include <ctime>
#include "helper.h"
#include "random.h"
#include "options.h"
#include "plink.h"
#include "perm.h"
#include "stats.h"
#include "nlist.h"
#include "model.h"
#include "logistic.h"
#include "linear.h"
#include "Insilico.h"
```

Include dependency graph for

Include dependency graph for helper.cpp:



Macros

- `#define FPMIN 1.0e-30`
 - `#define MISSING1(i, l) (P.SNP[l]->one[i] && (! P.SNP[l]->two[i]))`
 - `#define MISSING2(i, l) (P.sample[i]->one[l] && (! P.sample[i]->two[l]))`

Functions

- `vector< bool > nvec_bool ()`
 - `bool is_rare (Locus *loc)`
 - `string display (vector< string > &s)`

- string `displayLine` (vector< string > &s)
- void `display` (matrix_t &m)
- void `display` (vector_t &m)
- void `display` (vector< int > &m)
- vector< string > `parse2str` (string s)
- vector< int > `parse2int` (string s)
- int `getInt` (string s, string a)
- long unsigned int `getLongUnsignedInt` (string s, string a)
- double `getDouble` (string s, string a)
- void `checkDuplicates` (Plink &P)
- void `error` (string msg)
- void `shutdown` ()
- void `affCoding` (Plink &P)
- void `summaryBasics` (Plink &P)
- double `genotypingRate` (Plink &P, int l)
- bool `identicalSNPs` (Plink *P, int l1, int l2)
- vector< string > `listPossibleHaplotypes` (Plink &P, vector< int > S)
- bool `readString` (FILE *fp, string &s)
- void `removeMissingPhenotypes` (Plink &P)
- void `geno2matrix` (vector< int > &snps, matrix_t &g, boolmatrix_t &m, bool dom)
- string `genotypeToFile` (Plink &P, int i, int l)
- string `genotype` (Plink &P, int i, int l)
- string `genotype` (Plink &P, Individual *person, int l)
- void `permute` (vector< long int > &a)
- void `permute` (vector< int > &a)
- int `getChromosomeCode` (string chr)
- string `chromosomeName` (int c)
- int `getMarkerChromosome` (Plink &P, string m)
- int `getMarkerNumber` (Plink &P, string m)
- bool `seeChromosome` (Plink &P, int c)
- vector< int > `getChromosomeMarkerRange` (Plink &P, int c)
- vector< int > `getWindowRange` (Plink &P, int s)
- vector< int > `getChromosomeRange` (Plink &P)
- std::string `int2str` (int n)
- std::string `longint2str` (long int n)
- std::string `dbl2str` (double n, int prc)
- std::string `dbl2str_fixed` (double n, int prc)
- std::string `sw` (std::string s, int n)
- std::string `sw` (double d, int n)
- std::string `sw` (double d, int f, int n)
- std::string `sw` (int i, int n)
- void `NoMem` ()
- std::string `itoa` (int value, int base)
- void `checkFileExists` (string f)
- void `checkFileExists` (vector< string > f)
- bool `doesFileExist` (string f)
- bool `compressed` (string s)
- vector< string > `tokenizeLine` (ifstream &F1)
- vector< string > `tokenizeLine` (string sline)
- vector< string > `tokenizeLine` (ifstream &F1, string d)
- vector< bool > `vif_prune` (vector< vector< double > > m, double threshold, vector< int > &varcode)

- `vector< vector< double > > calcSetCovarianceMatrix (vector< int > &nSNP)`
- `string leftWindowEdge (Plink &P, int chr, int bp)`
- `string rightWindowEdge (Plink &P, int chr, int bp)`
- `void defineHorseChromosomes ()`
- `void defineSheepChromosomes ()`
- `void defineRiceChromosomes ()`
- `void defineDogChromosomes ()`
- `void defineMouseChromosomes ()`
- `void defineCowChromosomes ()`
- `void defineHumanChromosomes ()`
- `void sizeMatrix (matrix_t &m, int r, int c)`
- `void sizefMatrix (fmatrix_t &m, int r, int c)`
- `void sizeTable (table_t &m, int r, int c)`
- `double SNPHWE (int obs_hets, int obs_hom1, int obs_hom2)`
- `vector< string > commaParse (string s)`
- `string searchAndReplace (string str, string searchString, string replaceString)`
- `void makePersonMap (Plink &P, map< string, Individual *> &uid)`
- `void makeLocusMap (Plink &P, map< string, int > &mlocus)`
- `void smoother (Plink &P, vector_t &input, int n, vector_t &output1, vector_t &output2, vector< int > &count)`
- `map< string, set< Range > > readRange (string filename)`
- `double modelComparisonPValue (Model *alternate, Model *null)`
- `set< Range * > rangeIntersect (Range &r1, map< string, set< Range > > &ranges)`
- `set< Range * > mapRanges2SNP (int l, map< string, set< Range > > &ranges)`
- `int2 mapSNPs2Range (Plink &P, const Range *range)`
- `void makeScaffold (Plink &P)`
- `void mapRangesToSNPs (string filename, map< string, set< Range > > &ranges, map< int, set< Range *> > &snp2range)`
- `string relType (Individual *a, Individual *b)`
- `vector< vector< int > > two_locus_table (int l1, int l2)`
- `map< string, set< Range > > filterRanges (map< string, set< Range > > &ranges, string filename)`
- `bool matrixGetNumericAll (matrix_t &X)`
- `bool matrixGetNumericCaseControl (matrix_t &X, matrix_t &Y)`

Variables

- `ofstream LOG`
- `Plink * PP`

7.102.1 Macro Definition Documentation

7.102.1.1 FPMIN

```
#define FPMIN 1.0e-30
```

Definition at line 34 of file helper.cpp.

7.102.1.2 MISSING1

```
#define MISSING1(
    i,
    l ) ( P.SNP[l]->one[i] && ( ! P.SNP[l]->two[i] ) )
```

Definition at line 556 of file helper.cpp.

7.102.1.3 MISSING2

```
#define MISSING2(
    i,
    l ) ( P.sample[i]->one[l] && ( ! P.sample[i]->two[l] ) )
```

Definition at line 557 of file helper.cpp.

7.102.2 Function Documentation

7.102.2.1 affCoding()

```
void affCoding (
    PLink & P )
```

Definition at line 494 of file helper.cpp.

7.102.2.2 calcSetCovarianceMatrix()

```
vector<vector<double> > calcSetCovarianceMatrix (
    vector< int > & nSNP )
```

Definition at line 1245 of file helper.cpp.

7.102.2.3 checkDuplicates()

```
void checkDuplicates (
    PLink & P )
```

Definition at line 412 of file helper.cpp.

7.102.2.4 checkFileExists() [1/2]

```
void checkFileExists (
    string f )
```

Definition at line 1004 of file helper.cpp.

7.102.2.5 checkFileExists() [2/2]

```
void checkFileExists (
    vector< string > f )
```

Definition at line 1017 of file helper.cpp.

7.102.2.6 chromosomeName()

```
string chromosomeName (
    int c )
```

Definition at line 808 of file helper.cpp.

7.102.2.7 commaParse()

```
vector<string> commaParse (
    string s )
```

Definition at line 2497 of file helper.cpp.

7.102.2.8 compressed()

```
bool compressed (
    string s )
```

Definition at line 1035 of file helper.cpp.

7.102.2.9 dbl2str()

```
std::string dbl2str (
    double n,
    int prc )
```

Definition at line 917 of file helper.cpp.

7.102.2.10 dbl2str_fixed()

```
std::string dbl2str_fixed (
    double n,
    int prc )
```

Definition at line 925 of file helper.cpp.

7.102.2.11 defineCowChromosomes()

```
void defineCowChromosomes ( )
```

Definition at line 1516 of file helper.cpp.

7.102.2.12 defineDogChromosomes()

```
void defineDogChromosomes ( )
```

Definition at line 1448 of file helper.cpp.

7.102.2.13 defineHorseChromosomes()

```
void defineHorseChromosomes ( )
```

Definition at line 1372 of file helper.cpp.

7.102.2.14 defineHumanChromosomes()

```
void defineHumanChromosomes ( )
```

Definition at line 1546 of file helper.cpp.

7.102.2.15 defineMouseChromosomes()

```
void defineMouseChromosomes ( )
```

Definition at line 1486 of file helper.cpp.

7.102.2.16 defineRiceChromosomes()

```
void defineRiceChromosomes ( )
```

Definition at line 1433 of file helper.cpp.

7.102.2.17 defineSheepChromosomes()

```
void defineSheepChromosomes ( )
```

Definition at line 1402 of file helper.cpp.

7.102.2.18 display() [1/4]

```
string display (
    vector< string > & s )
```

Definition at line 53 of file helper.cpp.

7.102.2.19 display() [2/4]

```
void display (
    matrix_t & m )
```

Definition at line 68 of file helper.cpp.

7.102.2.20 display() [3/4]

```
void display (
    vector_t & m )
```

Definition at line 80 of file helper.cpp.

7.102.2.21 display() [4/4]

```
void display (
    vector< int > & m )
```

Definition at line 89 of file helper.cpp.

7.102.2.22 displayLine()

```
string displayLine (
    vector< string > & s )
```

Definition at line 60 of file helper.cpp.

7.102.2.23 doesFileExist()

```
bool doesFileExist (
    string f )
```

Definition at line 1023 of file helper.cpp.

7.102.2.24 error()

```
void error (
    string msg )
```

Definition at line 452 of file helper.cpp.

7.102.2.25 filterRanges()

```
map<string, set<Range> > filterRanges (
    map< string, set< Range > > & ranges,
    string filename )
```

Definition at line 3132 of file helper.cpp.

7.102.2.26 geno2matrix()

```
void geno2matrix (
    vector< int > & snps,
    matrix\_t & g,
    boolmatrix\_t & m,
    bool dom )
```

Definition at line 671 of file helper.cpp.

7.102.2.27 genotype() [1/2]

```
string genotype (
    Plink & P,
    int i,
    int l )
```

Definition at line 725 of file helper.cpp.

7.102.2.28 genotype() [2/2]

```
string genotype (
    Plink & P,
    Individual * person,
    int l )
```

Definition at line 753 of file helper.cpp.

7.102.2.29 genotypeToFile()

```
string genotypeToFile (
    PLink & P,
    int i,
    int l )
```

Definition at line 701 of file helper.cpp.

7.102.2.30 genotypingRate()

```
double genotypingRate (
    PLink & P,
    int l )
```

Definition at line 559 of file helper.cpp.

7.102.2.31 getChromosomeCode()

```
int getChromosomeCode (
    string chr )
```

Definition at line 803 of file helper.cpp.

7.102.2.32 getChromosomeMarkerRange()

```
vector<int> getChromosomeMarkerRange (
    PLink & P,
    int c )
```

Definition at line 836 of file helper.cpp.

7.102.2.33 getChromosomeRange()

```
vector<int> getChromosomeRange (
    PLink & P )
```

Definition at line 893 of file helper.cpp.

7.102.2.34 getDouble()

```
double getDouble (
    string s,
    string a )
```

Definition at line 350 of file helper.cpp.

7.102.2.35 getInt()

```
int getInt (
    string s,
    string a )
```

Definition at line 332 of file helper.cpp.

7.102.2.36 getLongUnsignedInt()

```
long unsigned int getLongUnsignedInt (
    string s,
    string a )
```

Definition at line 341 of file helper.cpp.

7.102.2.37 getMarkerChromosome()

```
int getMarkerChromosome (
    Plink & P,
    string m )
```

Definition at line 814 of file helper.cpp.

7.102.2.38 getMarkerNumber()

```
int getMarkerNumber (
    Plink & P,
    string m )
```

Definition at line 820 of file helper.cpp.

7.102.2.39 getWindowRange()

```
vector<int> getWindowRange (
    PLink & P,
    int s )
```

Definition at line 850 of file helper.cpp.

7.102.2.40 identicalSNPs()

```
bool identicalSNPs (
    PLink * P,
    int l1,
    int l2 )
```

Definition at line 582 of file helper.cpp.

7.102.2.41 int2str()

```
std::string int2str (
    int n )
```

Definition at line 905 of file helper.cpp.

7.102.2.42 is_rare()

```
bool is_rare (
    Locus * loc ) [inline]
```

Definition at line 44 of file helper.cpp.

7.102.2.43 itoa()

```
std::string itoa (
    int value,
    int base )
```

Definition at line 978 of file helper.cpp.

7.102.2.44 leftWindowEdge()

```
string leftWindowEdge (
    Plink & P,
    int chr,
    int bp )
```

Definition at line 1272 of file helper.cpp.

7.102.2.45 listPossibleHaplotypes()

```
vector<string> listPossibleHaplotypes (
    Plink & P,
    vector< int > S )
```

Definition at line 602 of file helper.cpp.

7.102.2.46 longint2str()

```
std::string longint2str (
    long int n )
```

Definition at line 911 of file helper.cpp.

7.102.2.47 makeLocusMap()

```
void makeLocusMap (
    Plink & P,
    map< string, int > & mlocus )
```

Definition at line 2519 of file helper.cpp.

7.102.2.48 makePersonMap()

```
void makePersonMap (
    Plink & P,
    map< string, Individual *> & uid )
```

Definition at line 2514 of file helper.cpp.

7.102.2.49 makeScaffold()

```
void makeScaffold (
    PLink & P )
```

Definition at line 2850 of file helper.cpp.

7.102.2.50 mapRanges2SNP()

```
set<Range*> mapRanges2SNP (
    int l,
    map< string, set< Range > > & ranges )
```

Definition at line 2746 of file helper.cpp.

7.102.2.51 mapRangesToSNPs()

```
void mapRangesToSNPs (
    string filename,
    map< string, set< Range > > & ranges,
    map< int, set< Range *> > & snp2range )
```

Definition at line 2885 of file helper.cpp.

7.102.2.52 mapSNPs2Range()

```
int2 mapSNPs2Range (
    PLink & P,
    const Range * range )
```

Definition at line 2755 of file helper.cpp.

7.102.2.53 matrixGetNumericAll()

```
bool matrixGetNumericAll (
    matrix_t & X )
```

Definition at line 3185 of file helper.cpp.

7.102.2.54 matrixGetNumericCaseControl()

```
bool matrixGetNumericCaseControl (
    matrix_t & X,
    matrix_t & Y )
```

Definition at line 3199 of file helper.cpp.

7.102.2.55 modelComparisonPValue()

```
double modelComparisonPValue (
    Model * alternate,
    Model * null )
```

Definition at line 2688 of file helper.cpp.

7.102.2.56 NoMem()

```
void NoMem ( )
```

Definition at line 967 of file helper.cpp.

7.102.2.57 nvec_bool()

```
vector<bool> nvec_bool ( )
```

Definition at line 39 of file helper.cpp.

7.102.2.58 parse2int()

```
vector<int> parse2int (
    string s )
```

Definition at line 287 of file helper.cpp.

7.102.2.59 parse2str()

```
vector<string> parse2str (
    string s )
```

Definition at line 272 of file helper.cpp.

7.102.2.60 permute() [1/2]

```
void permute (
    vector< long int > & a )
```

Definition at line 769 of file helper.cpp.

7.102.2.61 permute() [2/2]

```
void permute (
    vector< int > & a )
```

Definition at line 786 of file helper.cpp.

7.102.2.62 rangeIntersect()

```
set<Range*> rangeIntersect (
    Range & r1,
    map< string, set< Range > > & ranges )
```

Definition at line 2712 of file helper.cpp.

7.102.2.63 readRange()

```
map<string, set<Range> > readRange (
    string filename )
```

Definition at line 2587 of file helper.cpp.

7.102.2.64 readString()

```
bool readString (
    FILE * fp,
    string & s )
```

Definition at line 638 of file helper.cpp.

7.102.2.65 relType()

```
string relType (
    Individual * a,
    Individual * b )
```

Definition at line 2971 of file helper.cpp.

7.102.2.66 removeMissingPhenotypes()

```
void removeMissingPhenotypes (
    PLink & P )
```

Definition at line 659 of file helper.cpp.

7.102.2.67 rightWindowEdge()

```
string rightWindowEdge (
    PLink & P,
    int chr,
    int bp )
```

Definition at line 1293 of file helper.cpp.

7.102.2.68 searchAndReplace()

```
string searchAndReplace (
    string str,
    string searchString,
    string replaceString )
```

Definition at line 2503 of file helper.cpp.

7.102.2.69 seeChromosome()

```
bool seeChromosome (
    PLink & P,
    int c )
```

Definition at line 826 of file helper.cpp.

7.102.2.70 shutdown()

```
void shutdown ( )
```

Definition at line 468 of file helper.cpp.

7.102.2.71 sizefMatrix()

```
void sizefMatrix (
    fmatrix_t & m,
    int r,
    int c )
```

Definition at line 1605 of file helper.cpp.

7.102.2.72 sizeMatrix()

```
void sizeMatrix (
    matrix_t & m,
    int r,
    int c )
```

Definition at line 1598 of file helper.cpp.

7.102.2.73 sizeTable()

```
void sizeTable (
    table_t & m,
    int r,
    int c )
```

Definition at line 1612 of file helper.cpp.

7.102.2.74 smoother()

```
void smoother (
    Plink & P,
    vector_t & input,
    int n,
    vector_t & output1,
    vector_t & output2,
    vector< int > & count )
```

Definition at line 2524 of file helper.cpp.

7.102.2.75 SNPHWE()

```
double SNPHWE (
    int obs_hets,
    int obs_hom1,
    int obs_hom2 )
```

Definition at line 1627 of file helper.cpp.

7.102.2.76 summaryBasics()

```
void summaryBasics (
    Plink & P )
```

Definition at line 505 of file helper.cpp.

7.102.2.77 sw() [1/4]

```
std::string sw (
    std::string s,
    int n )
```

Definition at line 934 of file helper.cpp.

7.102.2.78 sw() [2/4]

```
std::string sw (
    double d,
    int n )
```

Definition at line 941 of file helper.cpp.

7.102.2.79 sw() [3/4]

```
std::string sw (
    double d,
    int f,
    int n )
```

Definition at line 949 of file helper.cpp.

7.102.2.80 sw() [4/4]

```
std::string sw (
    int i,
    int n )
```

Definition at line 959 of file helper.cpp.

7.102.2.81 tokenizeLine() [1/3]

```
vector<string> tokenizeLine (
    ifstream & F1 )
```

Definition at line 1043 of file helper.cpp.

7.102.2.82 tokenizeLine() [2/3]

```
vector<string> tokenizeLine (
    string sline )
```

Definition at line 1055 of file helper.cpp.

7.102.2.83 tokenizeLine() [3/3]

```
vector<string> tokenizeLine (
    ifstream & F1,
    string d )
```

Definition at line 1064 of file helper.cpp.

7.102.2.84 two_locus_table()

```
vector<vector<int> > two_locus_table (
    int l1,
    int l2 )
```

Definition at line 3042 of file helper.cpp.

7.102.2.85 vif_prune()

```
vector<bool> vif_prune (
    vector< vector< double > > m,
    double threshold,
    vector< int > & varcode )
```

Definition at line 1103 of file helper.cpp.

7.102.3 Variable Documentation

7.102.3.1 LOG

```
ofstream LOG
```

Definition at line 81 of file inbix.cpp.

7.102.3.2 PP

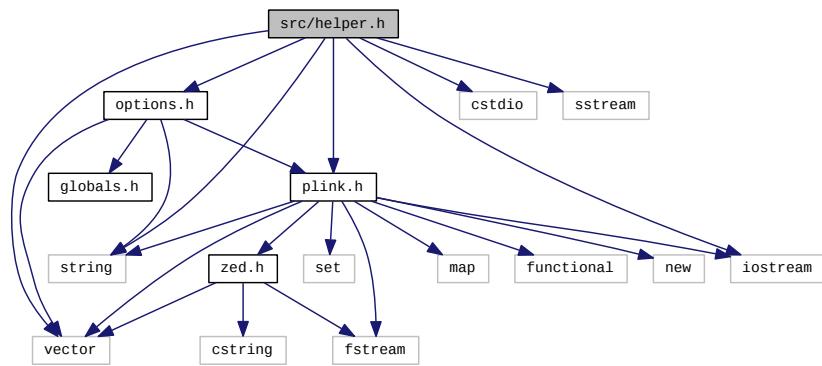
```
Plink* PP
```

Definition at line 85 of file inbix.cpp.

7.103 src/helper.h File Reference

```
#include <string>
#include <vector>
#include <cstdio>
#include <iostream>
#include <sstream>
#include "plink.h"
#include "options.h"
```

Include dependency graph for helper.h:



Classes

- class [CArgs](#)

TypeDefs

- `typedef vector< vector< double > > matrix_t`
- `typedef vector< double > vector_t`

Functions

- `template<class T>`
`const T SQR (const T a)`
- `template<class T>`
`const T MAX (const T &a, const T &b)`
- `template<class T>`
`const T MIN (const T &a, const T &b)`
- `template<class T>`
`const T SIGN (const T &a, const T &b)`
- `template<class T>`
`void SWAP (T &a, T &b)`

- void `sizeMatrix` (`matrix_t` &, int, int)
- void `sizefMatrix` (`fmatrix_t` &, int, int)
- void `sizeTable` (`table_t` &, int, int)
- void `NoMem` ()
- `vector< bool > nvec_bool` ()
- `vector< string > parse2str` (string)
- `vector< int > parse2int` (string)
- string `searchAndReplace` (string, string, string)
- `vector< string > commaParse` (string)
- template<class T >
 - bool `from_string` (T &t, const std::string &s, std::ios_base &(*f)(std::ios_base &))
- string `display` (`vector< string >` &)
- string `displayLine` (`vector< string >` &)
- void `error` (string)
- void `shutdown` ()
- void `checkDups` (`Plink` &)
- bool `readString` (FILE *, string &)
- void `summaryBasics` (`Plink` &)
- string `relType` (Individual *, Individual *)
- void `display` (`matrix_t` &)
- void `display` (`vector_t` &)
- void `display` (`vector< int >` &)
- double `genotypingRate` (`Plink` &, int)
- bool `identicalSNPs` (`Plink` *, int, int)
- `vector< string > listPossibleHaplotypes` (`Plink` &, `vector< int >`)
- void `geno2matrix` (`vector< int > &snps`, `matrix_t` &, `boolmatrix_t` &, bool)
- int `getInt` (string, string)
- long unsigned int `getLongUnsignedInt` (string, string)
- double `getDouble` (string, string)
- void `permute` (`vector< long int >` &)
- void `permute` (`vector< int >` &)
- `vector< double > FDR_BH` (`vector< double >` &)
- void `affCoding` (`Plink` &)
- void `removeMissingPhenotypes` (`Plink` &)
- string `genotype` (`Plink` &, int i, int l)
- string `genotype` (`Plink` &P, Individual *, int)
- string `genotypeToFile` (`Plink` &, int i, int l)
- int `getChromosomeCode` (string)
- string `chromosomeName` (int)
- int `getMarkerChromosome` (`Plink` &, string)
- int `getMarkerNumber` (`Plink` &, string)
- string `leftWindowEdge` (`Plink` &P, int bp, int chr)
- string `rightWindowEdge` (`Plink` &P, int bp, int chr)
- `vector< int > getChromosomeMarkerRange` (`Plink` &, int)
- bool `seeChromosome` (`Plink` &, int)
- `vector< int > getChromosomeRange` (`Plink` &)
- `vector< int > getWindowRange` (`Plink` &P, int)
- `vector< vector< int > > two_locus_table` (int, int)
- void `makePersonMap` (`Plink` &, map< string, Individual *> &)
- void `makeLocusMap` (`Plink` &, map< string, int > &)
- double `SNPHWE` (int obs_hets, int obs_hom1, int obs_hom2)

- string `int2str` (int)
- string `dbl2str` (double, int prc=-1)
- string `dbl2str_fixed` (double, int prc=-1)
- string `longint2str` (long int)
- std::string `sw` (std::string s, int n)
- std::string `sw` (double d, int n)
- std::string `sw` (double d, int f, int n)
- std::string `sw` (int i, int n)
- std::string `itoa` (int, int)
- void `checkFileExists` (string)
- void `checkFileExists` (vector< string >)
- bool `doesFileExist` (string)
- bool `compressed` (string)
- vector< string > `tokenizeLine` (ifstream &)
- vector< string > `tokenizeLine` (string)
- vector< string > `tokenizeLine` (ifstream &, string)
- void `defineDogChromosomes` ()
- void `defineMouseChromosomes` ()
- void `defineCowChromosomes` ()
- void `defineSheepChromosomes` ()
- void `defineHorseChromosomes` ()
- void `defineHumanChromosomes` ()
- void `defineRiceChromosomes` ()
- vector< bool > `vif_prune` (vector< vector< double > >, double threshold, vector< int > &)
- vector< vector< double > > `calcSetCovarianceMatrix` (vector< int > &nSNP)
- void `smoother` (Plink &P, vector_t &input, int n, vector_t &output1, vector_t &output2, vector< int > &count)
- map< string, set< Range > > `readRange` (string)
- double `modelComparisonPValue` (Model *alternate, Model *null)
- set< Range * > `rangelIntersect` (Range &r1, map< string, set< Range > > &ranges)
- set< Range * > `mapRanges2SNP` (int l, map< string, set< Range > > &ranges)
- int2 `mapSNPs2Range` (Plink &P, const Range *range)
- void `makeScaffold` (Plink &P)
- void `mapRangesToSNPs` (string, map< string, set< Range > > &ranges, map< int, set< Range *> > &snp2range)
- map< string, set< Range > > `filterRanges` (map< string, set< Range > > &ranges, string filename)
- bool `matrixGetNumericAll` (matrix_t &X)
- bool `matrixGetNumericCaseControl` (matrix_t &X, matrix_t &Y)

7.103.1 Typedef Documentation

7.103.1.1 matrix_t

```
typedef vector<vector<double>> matrix_t
```

Definition at line 116 of file helper.h.

7.103.1.2 vector_t

```
typedef vector<double> vector_t
```

Definition at line 117 of file helper.h.

7.103.2 Function Documentation

7.103.2.1 affCoding()

```
void affCoding (
    Plink & )
```

Definition at line 494 of file helper.cpp.

7.103.2.2 calcSetCovarianceMatrix()

```
vector<vector<double> > calcSetCovarianceMatrix (
    vector< int > & nSNP )
```

Definition at line 1245 of file helper.cpp.

7.103.2.3 checkDuplicates()

```
void checkDuplicates (
    Plink & )
```

Definition at line 412 of file helper.cpp.

7.103.2.4 checkFileExists() [1/2]

```
void checkFileExists (
    string )
```

Definition at line 1004 of file helper.cpp.

7.103.2.5 checkFileExists() [2/2]

```
void checkFileExists (
    vector< string > )
```

Definition at line 1017 of file helper.cpp.

7.103.2.6 chromosomeName()

```
string chromosomeName (
    int    )
```

Definition at line 808 of file helper.cpp.

7.103.2.7 commaParse()

```
vector<string> commaParse (
    string    )
```

Definition at line 2497 of file helper.cpp.

7.103.2.8 compressed()

```
bool compressed (
    string    )
```

Definition at line 1035 of file helper.cpp.

7.103.2.9 dbl2str()

```
string dbl2str (
    double ,
    int prc = -1 )
```

Definition at line 917 of file helper.cpp.

7.103.2.10 dbl2str_fixed()

```
string dbl2str_fixed (
    double ,
    int prc = -1 )
```

Definition at line 925 of file helper.cpp.

7.103.2.11 defineCowChromosomes()

```
void defineCowChromosomes ( )
```

Definition at line 1516 of file helper.cpp.

7.103.2.12 defineDogChromosomes()

```
void defineDogChromosomes ( )
```

Definition at line 1448 of file helper.cpp.

7.103.2.13 defineHorseChromosomes()

```
void defineHorseChromosomes ( )
```

Definition at line 1372 of file helper.cpp.

7.103.2.14 defineHumanChromosomes()

```
void defineHumanChromosomes ( )
```

Definition at line 1546 of file helper.cpp.

7.103.2.15 defineMouseChromosomes()

```
void defineMouseChromosomes ( )
```

Definition at line 1486 of file helper.cpp.

7.103.2.16 defineRiceChromosomes()

```
void defineRiceChromosomes ( )
```

Definition at line 1433 of file helper.cpp.

7.103.2.17 defineSheepChromosomes()

```
void defineSheepChromosomes ( )
```

Definition at line 1402 of file helper.cpp.

7.103.2.18 display() [1/4]

```
string display (
    vector< string > & )
```

Definition at line 53 of file helper.cpp.

7.103.2.19 display() [2/4]

```
void display (
    matrix_t & )
```

Definition at line 68 of file helper.cpp.

7.103.2.20 display() [3/4]

```
void display (
    vector_t & )
```

Definition at line 80 of file helper.cpp.

7.103.2.21 display() [4/4]

```
void display (
    vector< int > & )
```

Definition at line 89 of file helper.cpp.

7.103.2.22 displayLine()

```
string displayLine (
    vector< string > & )
```

Definition at line 60 of file helper.cpp.

7.103.2.23 doesFileExist()

```
bool doesFileExist (
    string )
```

Definition at line 1023 of file helper.cpp.

7.103.2.24 error()

```
void error (
    string )
```

Definition at line 452 of file helper.cpp.

7.103.2.25 FDR_BH()

```
vector<double> FDR_BH (
    vector< double > & )
```

7.103.2.26 filterRanges()

```
map<string, set<Range> > filterRanges (
    map< string, set< Range > > & ranges,
    string filename )
```

Definition at line 3132 of file helper.cpp.

7.103.2.27 from_string()

```
template<class T >
bool from_string (
    T & t,
    const std::string & s,
    std::ios_base &(*)(std::ios_base &) f )
```

Definition at line 97 of file helper.h.

7.103.2.28 geno2matrix()

```
void geno2matrix (
    vector< int > & snps,
    matrix_t & ,
    boolmatrix_t & ,
    bool )
```

Definition at line 671 of file helper.cpp.

7.103.2.29 genotype() [1/2]

```
string genotype (
    PLink & ,
    int i,
    int l )
```

Definition at line 725 of file helper.cpp.

7.103.2.30 genotype() [2/2]

```
string genotype (
    PLink & P,
    Individual * ,
    int )
```

Definition at line 753 of file helper.cpp.

7.103.2.31 genotypeToFile()

```
string genotypeToFile (
    PLink & ,
    int i,
    int l )
```

Definition at line 701 of file helper.cpp.

7.103.2.32 genotypingRate()

```
double genotypingRate (
    PLink & ,
    int )
```

Definition at line 559 of file helper.cpp.

7.103.2.33 getChromosomeCode()

```
int getChromosomeCode (
    string )
```

Definition at line 803 of file helper.cpp.

7.103.2.34 getChromosomeMarkerRange()

```
vector<int> getChromosomeMarkerRange (
    PLink & ,
    int )
```

Definition at line 836 of file helper.cpp.

7.103.2.35 getChromosomeRange()

```
vector<int> getChromosomeRange (
    Plink &
)
```

Definition at line 893 of file helper.cpp.

7.103.2.36 getDouble()

```
double getDouble (
    string ,
    string
)
```

Definition at line 350 of file helper.cpp.

7.103.2.37 getInt()

```
int getInt (
    string ,
    string
)
```

Definition at line 332 of file helper.cpp.

7.103.2.38 getLongUnsignedInt()

```
long unsigned int getLongUnsignedInt (
    string ,
    string
)
```

Definition at line 341 of file helper.cpp.

7.103.2.39 getMarkerChromosome()

```
int getMarkerChromosome (
    Plink & ,
    string
)
```

Definition at line 814 of file helper.cpp.

7.103.2.40 getMarkerNumber()

```
int getMarkerNumber (
    PLink & ,
    string )
```

Definition at line 820 of file helper.cpp.

7.103.2.41 getWindowRange()

```
vector<int> getWindowRange (
    PLink & P,
    int )
```

Definition at line 850 of file helper.cpp.

7.103.2.42 identicalSNPs()

```
bool identicalSNPs (
    PLink * ,
    int ,
    int )
```

Definition at line 582 of file helper.cpp.

7.103.2.43 int2str()

```
string int2str (
    int )
```

Definition at line 905 of file helper.cpp.

7.103.2.44 itoa()

```
std::string itoa (
    int ,
    int )
```

Definition at line 978 of file helper.cpp.

7.103.2.45 leftWindowEdge()

```
string leftWindowEdge (
    PLink & P,
    int bp,
    int chr )
```

Definition at line 1272 of file helper.cpp.

7.103.2.46 listPossibleHaplotypes()

```
vector<string> listPossibleHaplotypes (
    PLink & ,
    vector< int > )
```

Definition at line 602 of file helper.cpp.

7.103.2.47 longint2str()

```
string longint2str (
    long int )
```

Definition at line 911 of file helper.cpp.

7.103.2.48 makeLocusMap()

```
void makeLocusMap (
    PLink & ,
    map< string, int > & )
```

Definition at line 2519 of file helper.cpp.

7.103.2.49 makePersonMap()

```
void makePersonMap (
    PLink & ,
    map< string, Individual *> & )
```

Definition at line 2514 of file helper.cpp.

7.103.2.50 makeScaffold()

```
void makeScaffold (
    PLink & P )
```

Definition at line 2850 of file helper.cpp.

7.103.2.51 mapRanges2SNP()

```
set<Range*> mapRanges2SNP (
    int l,
    map< string, set< Range > > & ranges )
```

Definition at line 2746 of file helper.cpp.

7.103.2.52 mapRangesToSNPs()

```
void mapRangesToSNPs (
    string ,
    map< string, set< Range > > & ranges,
    map< int, set< Range *> > & snp2range )
```

Definition at line 2885 of file helper.cpp.

7.103.2.53 mapSNPs2Range()

```
int2 mapSNPs2Range (
    PLink & P,
    const Range * range )
```

Definition at line 2755 of file helper.cpp.

7.103.2.54 matrixGetNumericAll()

```
bool matrixGetNumericAll (
    matrix_t & X )
```

Definition at line 3185 of file helper.cpp.

7.103.2.55 matrixGetNumericCaseControl()

```
bool matrixGetNumericCaseControl (
    matrix_t & X,
    matrix_t & Y )
```

Definition at line 3199 of file helper.cpp.

7.103.2.56 MAX()

```
template<class T >
const T MAX (
    const T & a,
    const T & b ) [inline]
```

Definition at line 27 of file helper.h.

7.103.2.57 MIN()

```
template<class T >
const T MIN (
    const T & a,
    const T & b ) [inline]
```

Definition at line 31 of file helper.h.

7.103.2.58 modelComparisonPValue()

```
double modelComparisonPValue (
    Model * alternate,
    Model * null )
```

Definition at line 2688 of file helper.cpp.

7.103.2.59 NoMem()

```
void NoMem ( )
```

Definition at line 967 of file helper.cpp.

7.103.2.60 nvec_bool()

```
vector<bool> nvec_bool ( )
```

Definition at line 39 of file helper.cpp.

7.103.2.61 parse2int()

```
vector<int> parse2int (
    string   )
```

Definition at line 287 of file helper.cpp.

7.103.2.62 parse2str()

```
vector<string> parse2str (
    string   )
```

Definition at line 272 of file helper.cpp.

7.103.2.63 permute() [1/2]

```
void permute (
    vector< long int > &  )
```

Definition at line 769 of file helper.cpp.

7.103.2.64 permute() [2/2]

```
void permute (
    vector< int > &  )
```

Definition at line 786 of file helper.cpp.

7.103.2.65 rangelIntersect()

```
set<Range*> rangelIntersect (
    Range & r1,
    map< string, set< Range > > & ranges )
```

Definition at line 2712 of file helper.cpp.

7.103.2.66 readRange()

```
map<string, set<Range>> readRange (
    string )
```

Definition at line 2587 of file helper.cpp.

7.103.2.67 readString()

```
bool readString (
    FILE * ,
    string & )
```

Definition at line 638 of file helper.cpp.

7.103.2.68 relType()

```
string relType (
    Individual * ,
    Individual * )
```

Definition at line 2971 of file helper.cpp.

7.103.2.69 removeMissingPhenotypes()

```
void removeMissingPhenotypes (
    Plink & )
```

Definition at line 659 of file helper.cpp.

7.103.2.70 rightWindowEdge()

```
string rightWindowEdge (
    PLink & P,
    int bp,
    int chr )
```

Definition at line 1293 of file helper.cpp.

7.103.2.71 searchAndReplace()

```
string searchAndReplace (
    string ,
    string ,
    string )
```

Definition at line 2503 of file helper.cpp.

7.103.2.72 seeChromosome()

```
bool seeChromosome (
    PLink & ,
    int )
```

Definition at line 826 of file helper.cpp.

7.103.2.73 shutdown()

```
void shutdown ( )
```

Definition at line 468 of file helper.cpp.

7.103.2.74 SIGN()

```
template<class T >
const T SIGN (
    const T & a,
    const T & b ) [inline]
```

Definition at line 35 of file helper.h.

7.103.2.75 sizefMatrix()

```
void sizefMatrix (
    fmatrix_t & ,
    int ,
    int )
```

Definition at line 1605 of file helper.cpp.

7.103.2.76 sizeMatrix()

```
void sizeMatrix (
    matrix_t & ,
    int ,
    int )
```

Definition at line 1598 of file helper.cpp.

7.103.2.77 sizeTable()

```
void sizeTable (
    table_t & ,
    int ,
    int )
```

Definition at line 1612 of file helper.cpp.

7.103.2.78 smoother()

```
void smoother (
    Plink & P,
    vector_t & input,
    int n,
    vector_t & output1,
    vector_t & output2,
    vector< int > & count )
```

Definition at line 2524 of file helper.cpp.

7.103.2.79 SNPHWE()

```
double SNPHWE (
    int obs_hets,
    int obs_hom1,
    int obs_hom2 )
```

Definition at line 1627 of file helper.cpp.

7.103.2.80 SQR()

```
template<class T >
const T SQR (
    const T a ) [inline]
```

Definition at line 24 of file helper.h.

7.103.2.81 summaryBasics()

```
void summaryBasics (
    PLink & )
```

Definition at line 505 of file helper.cpp.

7.103.2.82 sw() [1/4]

```
std::string sw (
    std::string s,
    int n )
```

Definition at line 934 of file helper.cpp.

7.103.2.83 sw() [2/4]

```
std::string sw (
    double d,
    int n )
```

Definition at line 941 of file helper.cpp.

7.103.2.84 sw() [3/4]

```
std::string sw (
    double d,
    int f,
    int n )
```

Definition at line 949 of file helper.cpp.

7.103.2.85 sw() [4/4]

```
std::string sw (
    int i,
    int n )
```

Definition at line 959 of file helper.cpp.

7.103.2.86 SWAP()

```
template<class T >
void SWAP (
    T & a,
    T & b ) [inline]
```

Definition at line 39 of file helper.h.

7.103.2.87 tokenizeLine() [1/3]

```
vector<string> tokenizeLine (
    ifstream & )
```

Definition at line 1043 of file helper.cpp.

7.103.2.88 tokenizeLine() [2/3]

```
vector<string> tokenizeLine (
    string )
```

Definition at line 1055 of file helper.cpp.

7.103.2.89 tokenizeLine() [3/3]

```
vector<string> tokenizeLine (
    ifstream & ,
    string   )
```

Definition at line 1064 of file helper.cpp.

7.103.2.90 two_locus_table()

```
vector<vector<int> > two_locus_table (
    int ,
    int   )
```

Definition at line 3042 of file helper.cpp.

7.103.2.91 vif_prune()

```
vector<bool> vif_prune (
    vector< vector< double > > ,
    double threshold,
    vector< int > & )
```

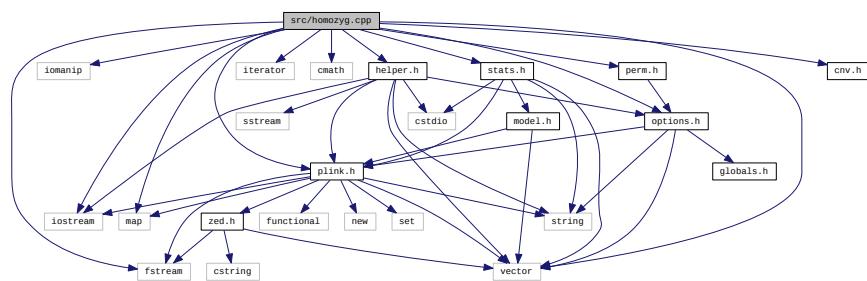
Definition at line 1103 of file helper.cpp.

7.104 src/homozyg.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <vector>
#include <map>
#include <iterator>
#include <cmath>
#include "plink.h"
#include "helper.h"
#include "options.h"
#include "perm.h"
#include "stats.h"
```

```
#include "cnv.h"
```

Include dependency graph for homozyg.cpp:



Classes

- class `std::less< Segment * >`
 - class `Pool`
 - class `std::less< Pool * >`
 - class `HWindow`

Functions

- `vector_t compareCNVs (CNVIndivReport &a, CNVIndivReport &b)`
 - `void summaryIndivSummaries (Plink *P, int kmask, map< indivPair, int > &segmentCount, map< indivPair, double > &segmentLength, CNVIndivReport &a, CNVIndivReport &u, vector_t &res)`
 - `bool segsMatch (Segment *s1, Segment *s2)`
 - `bool segsIBDMatchCON (Segment *s1, Segment *s2, int start, int finish)`
 - `bool segsIBDMatch (Segment *s1, Segment *s2)`
 - `bool segsMatchCON (Segment *s1, Segment *s2, int start, int finish)`
 - `void displayPoolVerbose (Plink &P, Pool *pool, ofstream &OUT)`
 - `bool segsOverlap (Segment *s1, Segment *s2)`

Variables

- Plink * PP

7.104.1 Function Documentation

7.104.1.1 compareCNVs()

```
vector_t compareCNVs (
    CNVIndivReport & a,
    CNVIndivReport & b )
```

Definition at line 100 of file homozyg.cpp.

7.104.1.2 displayPoolVerbose()

```
void displayPoolVerbose (
    Plink & P,
    Pool * pool,
    ofstream & OUT )
```

Definition at line 593 of file homozyg.cpp.

7.104.1.3 segsIBDMatch()

```
bool segsIBDMatch (
    Segment * s1,
    Segment * s2 )
```

Definition at line 484 of file homozyg.cpp.

7.104.1.4 segsIBDMatchCON()

```
bool segsIBDMatchCON (
    Segment * s1,
    Segment * s2,
    int start,
    int finish )
```

Definition at line 402 of file homozyg.cpp.

7.104.1.5 `segsMatch()`

```
bool segsMatch (
    Segment * s1,
    Segment * s2 )
```

Definition at line 374 of file homozyg.cpp.

7.104.1.6 `segsMatchCON()`

```
bool segsMatchCON (
    Segment * s1,
    Segment * s2,
    int start,
    int finish )
```

Definition at line 569 of file homozyg.cpp.

7.104.1.7 `segsOverlap()`

```
bool segsOverlap (
    Segment * s1,
    Segment * s2 )
```

Definition at line 2333 of file homozyg.cpp.

7.104.1.8 `summaryIndivSummaries()`

```
void summaryIndivSummaries (
    Plink * P,
    int kmask,
    map< indivPair, int > & segmentCount,
    map< indivPair, double > & segmentLength,
    CNVIndivReport & a,
    CNVIndivReport & u,
    vector_t & res )
```

Definition at line 148 of file homozyg.cpp.

7.104.2 Variable Documentation

7.104.2.1 PP

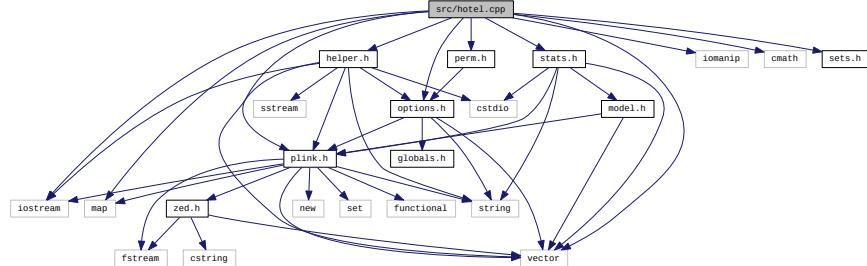
[Plink*](#) PP

Definition at line 85 of file inbix.cpp.

7.105 src/hotel.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <map>
#include <vector>
#include <cmath>
#include "plink.h"
#include "sets.h"
#include "options.h"
#include "helper.h"
#include "stats.h"
#include "perm.h"
```

Include dependency graph for hotel.cpp:



Functions

- void [calcHotelSetMeanVariance](#) (vector< CSNP *> &, vector< double > &, vector< double > &, vector< double > &, vector< Individual *> &, int, int)

7.105.1 Function Documentation

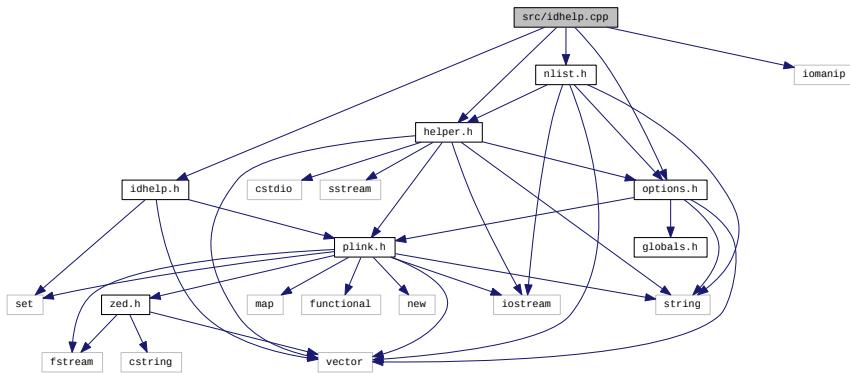
7.105.1.1 calcHotelSetMeanVariance()

```
void calcHotelSetMeanVariance (
    vector< CSNP *> & pSNP,
    vector< double > & mean1,
    vector< double > & mean2,
    vector< vector< double > > & pooled,
    vector< Individual *> & sample,
    int ncase,
    int ncontrol )
```

Definition at line 337 of file hotel.cpp.

7.106 src/idhelp.cpp File Reference

```
#include "idhelp.h"
#include "options.h"
#include "helper.h"
#include "nlist.h"
#include <iomanip>
Include dependency graph for idhelp.cpp:
```



Variables

- Plink * PP

7.106.1 Variable Documentation

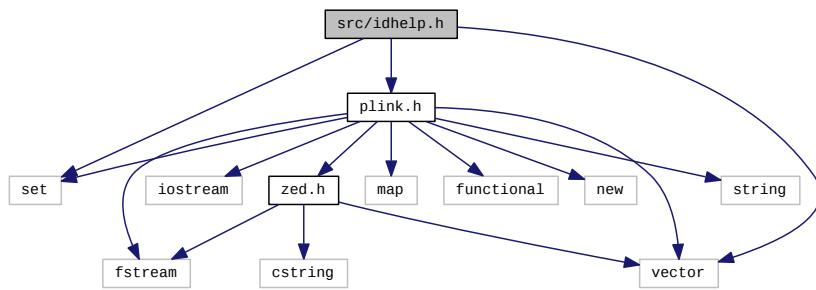
7.106.1.1 PP

```
Plink\* PP
```

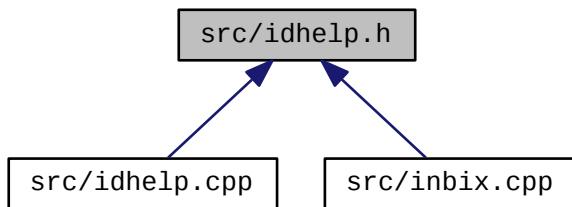
Definition at line 85 of file `inbix.cpp`.

7.107 src/idhelp.h File Reference

```
#include <vector>
#include <set>
#include "plink.h"
Include dependency graph for idhelp.h:
```



This graph shows which files directly or indirectly include this file:

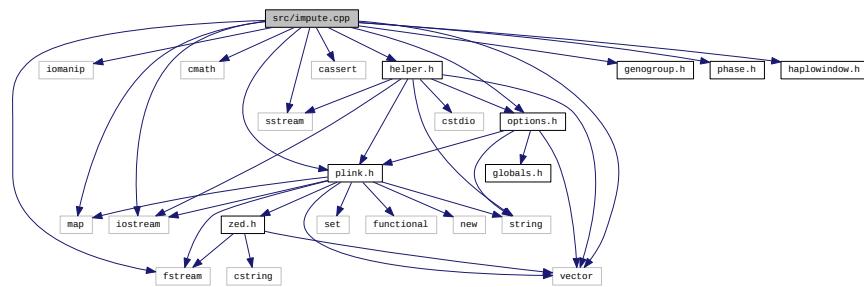


Classes

- class `IDField`
- class `IDFile`
- class `IDValue`
- class `IDGroup`
- class `IDHelper`

7.108 src/impute.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <cmath>
#include <vector>
#include <map>
#include <cassert>
#include "plink.h"
#include "options.h"
#include "helper.h"
#include "genogroup.h"
#include "phase.h"
#include "haplowindow.h"
Include dependency graph for impute.cpp:
```



Classes

- class [probabilisticGenotype](#)

Variables

- ofstream [LOG](#)

7.108.1 Variable Documentation

7.108.1.1 LOG

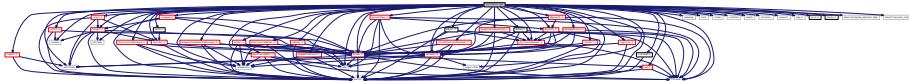
ofstream LOG

Definition at line 81 of file inbix.cpp.

7.109 src/inbix.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <vector>
#include <map>
#include <list>
#include <ctime>
#include <cstdio>
#include <cstdlib>
#include <algorithm>
#include <cmath>
#include <iterator>
#include <cassert>
#include <omp.h>
#include <armadillo>
#include "plink.h"
#include "options.h"
#include "helper.h"
#include "crandom.h"
#include "perm.h"
#include "sets.h"
#include "linear.h"
#include "logistic.h"
#include "phase.h"
#include "clumpld.h"
#include "nlist.h"
#include "stats.h"
#include "idhelp.h"
#include "zed.h"
#include <boost/program_options.hpp>
#include <boost/lexical_cast.hpp>
#include "regain.h"
#include "InteractionNetwork.h"
#include "CentralityRanker.h"
#include "ArmadilloFuncs.h"
#include "EpistasisEQtl.h"
#include "Dataset.h"
#include "PlinkInternalsDataset.h"
#include "AttributeRanker.h"
#include "ReliefF.h"
#include "RReliefF.h"
#include "ReliefFSeq.h"
#include "RandomForest.h"
#include "EvaporativeCooling.h"
#include "EvaporativeCoolingPrivacy.h"
#include "DcVar.h"
#include "Dseq.h"
#include "Edger.h"
```

Include dependency graph for inbix.cpp:



Functions

- int [main](#) (int argc, char *argv[])

Variables

- ofstream [LOG](#)
- string [PVERSION](#)
- string [PDATE](#)
- string [PREL](#)
- [Plink](#) * [PP](#)

7.109.1 Function Documentation

7.109.1.1 main()

```
int main (
    int argc,
    char * argv[ ] )
```

pointer to an interaction ranker algorithm object

Definition at line 88 of file inbix.cpp.

7.109.2 Variable Documentation

7.109.2.1 LOG

```
ofstream LOG
```

Definition at line 81 of file inbix.cpp.

7.109.2.2 PDATE

```
string PDATE
```

Definition at line 83 of file inbix.cpp.

7.109.2.3 PP

```
Plink* PP
```

Definition at line 85 of file inbix.cpp.

7.109.2.4 PREL

```
string PREL
```

Definition at line 84 of file inbix.cpp.

7.109.2.5 PVERSION

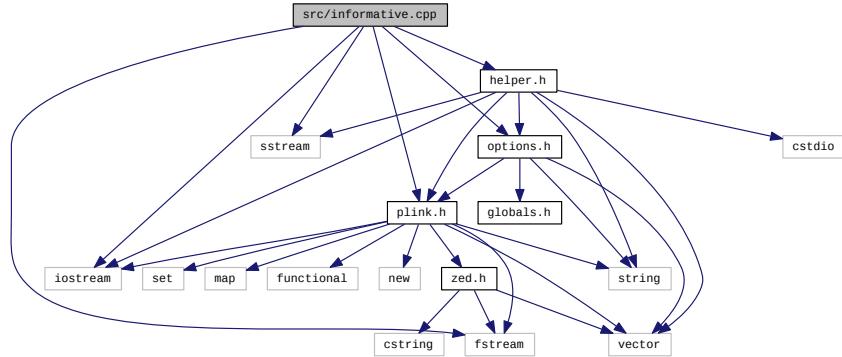
```
string PVERSION
```

Definition at line 82 of file inbix.cpp.

7.110 src/informative.cpp File Reference

```
#include <iostream>
#include <fstream>
#include <sstream>
#include "plink.h"
#include "options.h"
#include "helper.h"
```

Include dependency graph for informative.cpp:



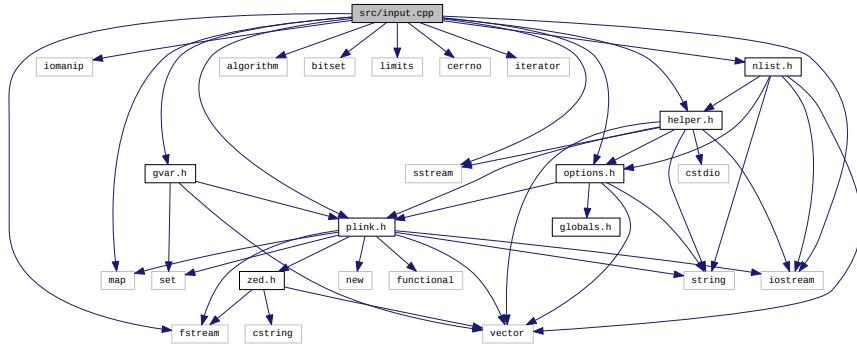
Classes

- class [Pair](#)

7.111 src/input.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <map>
#include <algorithm>
#include <set>
#include <limits>
#include <cerrno>
#include <iterator>
#include "plink.h"
#include "options.h"
#include "helper.h"
#include "nlist.h"
#include "gvar.h"
```

Include dependency graph for input.cpp:



Variables

- ofstream [LOG](#)

7.111.1 Variable Documentation

7.111.1.1 LOG

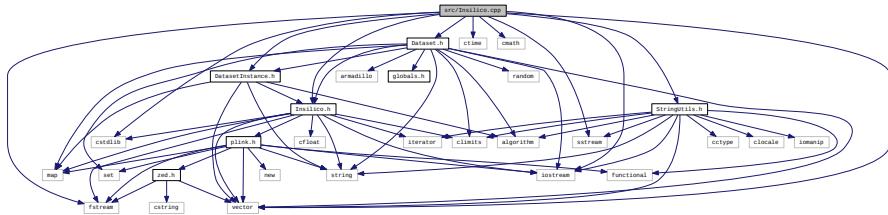
ofstream LOG

Definition at line 81 of file inbix.cpp.

7.112 src/Insilico.cpp File Reference

```
#include <cstdlib>
#include <iostream>
#include <fstream>
#include <vector>
#include <sstream>
#include <ctime>
#include <cmath>
#include "Dataset.h"
#include "DatasetInstance.h"
#include "Insilico.h"
#include "StringUtils.h"
```

Include dependency graph for Insilico.cpp:



Functions

- bool **scoresSortAsc** (const ScoreVarPair &p1, const ScoreVarPair &p2)
- bool **scoresSortDesc** (const ScoreVarPair &p1, const ScoreVarPair &p2)
- bool **scoresSortAscByName** (const ScoreVarPair &p1, const ScoreVarPair &p2)
- string **Timestamp** ()

Return a timestamp string for logging purposes.
- Dataset * **ChooseSnpsDatasetByType** (string snpsFilename, string snpsFileType)

Determines the data set type to instantiate based on the passed type string or data set filenames's extension.
- bool **LoadNumericIds** (string filename, vector< string > &retIds)

Loads the individual (instance) IDs from the numerics file.
- bool **LoadPhenoidls** (string filename, vector< string > &retIds)

Loads the individual (instance) IDs from the numerics file.
- bool **GetMatchingIds** (string numericsFilename, string altPhenotypeFilename, vector< string > numericsIds, vector< string > phenoids, vector< string > &matchingIds)

Return matching IDs from numeric and/or phenotype file IDs.
- ClassType **DetectClassType** (std::string filename, int classColumn, bool hasHeader)

- Detect the class type by reading the specified column from a whitespace- delimited text file.*
- bool [GetConfigValue](#) ([ConfigMap](#) &configMap, std::string key, std::string &value)
Get the parameter value from the configuration map key.
 - string [GetFileBasename](#) (string fileName)
Get the full filename without the extension.
 - string [GetFileExtension](#) (string fileName)
Get the filename extension.
 - double [ProtectedLog](#) (double x)
protected log function returns 0 for 0

7.112.1 Function Documentation

7.112.1.1 ChooseSnpsDatasetByType()

```
Dataset* ChooseSnpsDatasetByType (
    std::string snpsFilename,
    std::string snpsFileType = "")
```

Determines the data set type to instantiate based on the passed type string or data set filenames's extension.

Parameters

in	<i>snpsFilename</i>	SNP data set filename
in	<i>snpsFileType</i>	SNP data set file type (overrides detect by extension)

Returns

pointer to new data set or NULL if could not match a data set type

Definition at line 50 of file Insilico.cpp.

7.112.1.2 DetectClassType()

```
ClassType DetectClassType (
    std::string filename,
    int classColumn,
    bool hasHeader )
```

Detect the class type by reading the specified column from a whitespace- delimited text file.

Parameters

in	<i>filename</i>	whitespace-delimited text file name
in	<i>classColumn</i>	the column containing the class values
in	<i>hasHeader</i>	does the file have a header line?

ReturnsClassType defined in [Dataset.h](#)

Open the file for reading

Skip the header if it has one

Determine the phenotype type

Definition at line 269 of file Insilico.cpp.

7.112.1.3 GetConfigValue()

```
bool GetConfigValue (
    ConfigMap & configMap,
    std::string key,
    std::string & value )
```

Get the parameter value from the configuration map key.

Parameters

in	<i>configMap</i>	reference to a configuration map
in	<i>key</i>	parameter name
out	<i>parameter</i>	value

Returns

true if key found, false if not found

Definition at line 336 of file Insilico.cpp.

7.112.1.4 GetFileBasename()

```
string GetFileBasename (
    std::string fullFilename )
```

Get the full filename without the extension.

Parameters

in	<i>fullFilename</i>	complete filename
----	---------------------	-------------------

Returns

path/filename without extension

Definition at line 348 of file Insilico.cpp.

7.112.1.5 GetFileExtension()

```
string GetFileExtension (
    std::string fullFilename )
```

Get the filename extension.

Parameters

in	<i>fullFilename</i>	complete filename
----	---------------------	-------------------

Returns

filename extension

Definition at line 353 of file Insilico.cpp.

7.112.1.6 GetMatchingIds()

```
bool GetMatchingIds (
    std::string numericsFilename,
    std::string altPhenotypeFilename,
    std::vector< std::string > numericsIds,
    std::vector< std::string > phenoIds,
    std::vector< std::string > & matchingIds )
```

Return matching IDs from numeric and/or phenotype file IDs.

Parameters

in	<i>numericsFilename</i>	name of the PLINK covar format file
in	<i>altPhenotypeFilename</i>	name of the alternate pheno file PLINK
in	<i>numericsIds</i>	covar format file ids
in	<i>phenoIds</i>	alternate phenotype file ids
out	<i>matchingIds</i>	ids that match between numerics and phenotypes

Returns

success

Definition at line 203 of file Insilico.cpp.

7.112.1.7 LoadNumericIds()

```
bool LoadNumericIds (
    std::string filename,
    std::vector< std::string > & retIds )
```

Loads the individual (instance) IDs from the numerics file.

Returns the IDs through reference parameter retIds.

Parameters

in	<i>filename</i>	filename that contains numerics IDs
out	<i>vector</i>	of individual (instance) IDs (strings)

Returns

success

Definition at line 96 of file Insilico.cpp.

7.112.1.8 LoadPhenoIds()

```
bool LoadPhenoIds (
    std::string filename,
    std::vector< std::string > & retIds )
```

Loads the individual (instance) IDs from the numerics file.

Returns the IDs through reference parameter retIds.

Parameters

in	<i>filename</i>	filename that contains numerics IDs
out	<i>vector</i>	of individual (instance) IDs (strings)

Returns

success

Definition at line 151 of file Insilico.cpp.

7.112.1.9 ProtectedLog()

```
double ProtectedLog (
    double x )
```

protected log function returns 0 for 0

Definition at line 358 of file Insilico.cpp.

7.112.1.10 scoresSortAsc()

```
bool scoresSortAsc (
    const ScoreVarPair & p1,
    const ScoreVarPair & p2 )
```

Definition at line 26 of file Insilico.cpp.

7.112.1.11 scoresSortAscByName()

```
bool scoresSortAscByName (
    const ScoreVarPair & p1,
    const ScoreVarPair & p2 )
```

Definition at line 34 of file Insilico.cpp.

7.112.1.12 scoresSortDesc()

```
bool scoresSortDesc (
    const ScoreVarPair & p1,
    const ScoreVarPair & p2 )
```

Definition at line 30 of file Insilico.cpp.

7.112.1.13 Timestamp()

```
string Timestamp ( )
```

Return a timestamp string for logging purposes.

Returns

fixed-length, formatted timestamp as a string

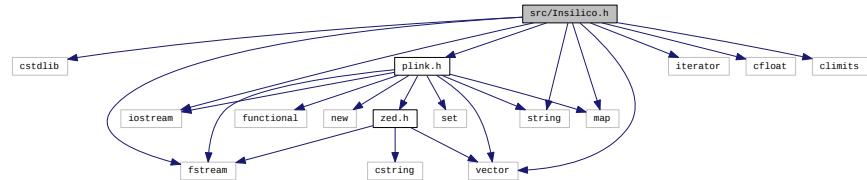
Definition at line 38 of file Insilico.cpp.

7.113 src/Insilico.h File Reference

Common functions for In silico Lab projects.

```
#include <cstdlib>
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <map>
#include <iterator>
#include <cfloat>
#include <climits>
#include "plink.h"
```

Include dependency graph for Insilico.h:



This graph shows which files directly or indirectly include this file:



Typedefs

- `typedef int AttributeLevel`
TYPEDEFS.
- `typedef double NumericLevel`
type of continuous attributes
- `typedef int ClassLevel`
type of instance class labels
- `typedef std::pair< double, std::string > ScoreVarPair`
- `typedef ScoreVarPair DistancePair`
distance pair type: distance, instance ID
- `typedef std::vector< DistancePair > DistancePairs`
vector of distance pairs represents distances to nearest neighbors
- `typedef DistancePairs::const_iterator DistancePairsIt`
distance pairs iterator
- `typedef std::map< std::string, std::string > ConfigMap`
Configuration map as an alternative to infix `parse.cpp`.
- `typedef std::vector< ScoreVarPair > AttributeScores`
attribute scores - sorted by score key
- `typedef AttributeScores::iterator AttributeScoresIt`
attribute scores iterator - sorted by score key
- `typedef AttributeScores::const_iterator AttributeScoresCIt`
attribute scores constant iterator - sorted by score key
- `typedef pair< double, pair< int, int > > matrixElement`

Enumerations

- `enum OutputDatasetType { TAB_DELIMITED_DATASET, CSV_DELIMITED_DATASET, ARFF_DATASET, PLINK_PED_DATASET, PLINK_BED_DATASET, PLINK_COVAR_DATASET, NO_OUTPUT_DATASET }`
ENUMS.
- `enum AnalysisType { SNP_ONLY_ANALYSIS, NUMERIC_ONLY_ANALYSIS, INTEGRATED_ANALYSIS, DIAGNOSTIC_ANALYSIS, REGRESSION_ANALYSIS, RNASEQ_ANALYSIS, DGE_ANALYSIS, BIRDSEED_ANALYSIS, DISTANCE_MATRIX_ANALYSIS, DATASET_CONVERSION, NO_ANALYSIS }`
- `enum ValueType { NUMERIC_VALUE, DISCRETE_VALUE, MISSING_VALUE, NO_VALUE }`
- `enum AttributeType { NUMERIC_TYPE, DISCRETE_TYPE, NO_TYPE }`
- `enum ClassType { CONTINUOUS_CLASS_TYPE, CASE_CONTROL_CLASS_TYPE, MULTI_CLASS_TYPE, NO_CLASS_TYPE }`
- `enum AttributeMutationType { TRANSITION_MUTATION, TRANSPERSION_MUTATION, UNKNOWN_MUTATION }`
- `enum RandomJungleTreeType { UNKNOWN_TREE_TYPE = 0, NOMINAL_NUMERIC_TREE, NOMINAL_NOMINAL_TREE, NUMERIC_NUMERIC_TREE, NUMERIC_NOMINAL_TREE, NOMINAL_NUMERIC_FLOATS }`
- `enum RandomJungleRunMode { UNKNOWN_RUN_MODE, LIBRARY_RUN_MODE, SYSTEM_CALL_RUN_MODE, LIBRARY_FILE_RUN_MODE }`
- `enum EcAlgorithmType { EC_ALG_ME_IT, EC_ALG_ME_ONLY, EC_ALG_IT_ONLY }`
- `enum EcMeAlgorithmType { EC_ME_ALG_RJ, EC_ME_ALG_DESEQ, EC_ME_ALG_EDGER }`
- `enum EcItAlgorithmType { EC_IT_ALG_RF, EC_IT_ALG_RFSEQ }`
- `enum CvSetType { CV_TRAIN, CV_HOLDOUT, CV_TEST, CV_NONE }`

Functions

- bool **scoresSortAsc** (const **ScoreVarPair** &p1, const **ScoreVarPair** &p2)
- bool **scoresSortAscByName** (const **ScoreVarPair** &p1, const **ScoreVarPair** &p2)
- bool **scoresSortDesc** (const **ScoreVarPair** &p1, const **ScoreVarPair** &p2)
- **RandomJungleTreeType DetermineRandomJungleTreeType** (**AttributeType** attributeType, **ClassType** classType)

Return random jungle tree type from the class and attribute types.
- std::string **Timestamp** ()

Return a timestamp string for logging purposes.
- **Dataset * ChooseSnpsDatasetByType** (std::string snpsFilename, std::string snpsFileType="")

Determines the data set type to instantiate based on the passed type string or data set filenames's extension.
- bool **LoadNumericIds** (std::string filename, std::vector< std::string > &retIds)

Loads the individual (instance) IDs from the numerics file.
- bool **LoadPhenIds** (std::string filename, std::vector< std::string > &retIds)

Loads the individual (instance) IDs from the numerics file.
- bool **GetMatchingIds** (std::string numericsFilename, std::string altPhenotypeFilename, std::vector< std::string > numericsIds, std::vector< std::string > phenIds, std::vector< std::string > &matchingIds)

Return matching IDs from numeric and/or phenotype file IDs.
- **ClassType DetectClassType** (std::string filename, int classColumn, bool hasHeader)

Detect the class type by reading the specified column from a whitespace- delimited text file.
- bool **GetConfigValue** (**ConfigMap** &configMap, std::string key, std::string &value)

Get the parameter value from the configuration map key.
- std::string **GetFileBasename** (std::string fullFilename)

Get the full filename without the extension.
- std::string **GetFileExtension** (std::string fullFilename)

Get the filename extension.
- template<class T >
 void **PrintVector** (std::vector< T > vec, std::string title="")

Print a vector of T values with optional title.
- double **ProtectedLog** (double x)

protected log function returns 0 for 0

Variables

- **PLink * PP**
- static const int **COMMAND_LINE_ERROR** = EXIT_FAILURE

CONSTANTS.
- static const int **DATASET_LOAD_ERROR** = EXIT_FAILURE
- static const int **INVALID_DISTANCE** = INT_MAX

return value for invalid distance
- static const int **INVALID_INDEX** = INT_MAX

return value for invalid index into attributes
- static const unsigned int **INVALID_INT_VALUE** = UINT_MAX

return value for invalid index into attributes
- static const **AttributeLevel** **INVALID_ATTRIBUTE_VALUE** = INT_MIN

invalid attribute value
- static const **NumericLevel** **INVALID_NUMERIC_VALUE** = INT_MIN

- static const **ClassLevel** INVALID_DISCRETE_CLASS_VALUE = INT_MIN
 - invalid attribute value*
 - stored value for missing discrete class*
- static const **NumericLevel** INVALID_NUMERIC_CLASS_VALUE = INT_MIN
 - stored value for missing numeric class*
- static const **AttributeLevel** MISSING_ATTRIBUTE_VALUE = -9
 - stored value for missing discrete attribute*
- static const **NumericLevel** MISSING_NUMERIC_VALUE = -9
 - stored value for missing numeric attribute*
- static const **ClassLevel** MISSING_DISCRETE_CLASS_VALUE = -9
 - stored value for missing discrete class*
- static const **NumericLevel** MISSING_NUMERIC_CLASS_VALUE = -9
 - stored value for missing numeric class*
- static const double DEFAULT_PVALUE = 1.0
 - default p-value threshold*
- static const double DEFAULT_PVALUE_THRESHOLD = 0.05
- static const double DEFAULT_FDR = 0.05
- static const double DEFAULT_ZVALUE = 0.0
- static const uint MIN_NUM_GENES = 4
- static const uint MIN_NUM_SUBJ_PER_GROUP = 4
- static std::map< std::string, std::string > datasetTypeToExt

7.113.1 Detailed Description

Common functions for In silico Lab projects.

Author

: Bill White

Version

1.0

Contact: bill.c.white@gmail.com Created on 10/13/11

7.113.2 Typedef Documentation

7.113.2.1 AttributeLevel

typedef int **AttributeLevel**

T Y P E D E F S.

type of discrete attribute values

Definition at line 36 of file Insilico.h.

7.113.2.2 AttributeScores

```
typedef std::vector<ScoreVarPair> AttributeScores
```

attribute scores - sorted by score key

Definition at line 56 of file Insilico.h.

7.113.2.3 AttributeScoresCIt

```
typedef AttributeScores::const_iterator AttributeScoresCIt
```

attribute scores constant iterator - sorted by score key

Definition at line 60 of file Insilico.h.

7.113.2.4 AttributeScoresIt

```
typedef AttributeScores::iterator AttributeScoresIt
```

attribute scores iterator - sorted by score key

Definition at line 58 of file Insilico.h.

7.113.2.5 ClassLevel

```
typedef int ClassLevel
```

type of instance class labels

Definition at line 40 of file Insilico.h.

7.113.2.6 ConfigMap

```
typedef std::map<std::string, std::string> ConfigMap
```

Configuration map as an alternative to inbix [parse.cpp](#).

Definition at line 53 of file Insilico.h.

7.113.2.7 DistancePair

```
typedef ScoreVarPair DistancePair
```

distance pair type: distance, instance ID

Definition at line 46 of file Insilico.h.

7.113.2.8 DistancePairs

```
typedef std::vector<DistancePair> DistancePairs
```

vector of distance pairs represents distances to nearest neighbors

Definition at line 48 of file Insilico.h.

7.113.2.9 DistancePairsIt

```
typedef DistancePairs::const_iterator DistancePairsIt
```

distance pairs iterator

Definition at line 50 of file Insilico.h.

7.113.2.10 matrixElement

```
typedef pair< double, pair<int, int> > matrixElement
```

Definition at line 68 of file Insilico.h.

7.113.2.11 NumericLevel

```
typedef double NumericLevel
```

type of continuous attributes

Definition at line 38 of file Insilico.h.

7.113.2.12 ScoreVarPair

```
typedef std::pair<double, std::string> ScoreVarPair
```

Definition at line 42 of file Insilico.h.

7.113.3 Enumeration Type Documentation

7.113.3.1 AnalysisType

```
enum AnalysisType
```

Type of analysis to perform.

Enumerator

SNP_ONLY_ANALYSIS	discrete analysis
NUMERIC_ONLY_ANALYSIS	continuous attributes
INTEGRATED_ANALYSIS	discrete and continuous analysis
DIAGNOSTIC_ANALYSIS	diagnostic mode - no ReliefF analysis
REGRESSION_ANALYSIS	regression ReliefF analysis
RNASEQ_ANALYSIS	rnaSeq count data ReliefF analysis
DGE_ANALYSIS	digital gene expression (DGE) analysis
BIRDSEED_ANALYSIS	Birdseed called SNPs analysis.
DISTANCE_MATRIX_ANALYSIS	distance matrix calculation
DATASET_CONVERSION	convert data set format types
NO_ANALYSIS	no analysis specified

Definition at line 129 of file Insilico.h.

7.113.3.2 AttributeMutationType

```
enum AttributeMutationType
```

Type of attribute mutation.

Enumerator

TRANSITION_MUTATION	transition within family
TRANSVERSION_MUTATION	transversion between families
UNKNOWN_MUTATION	unknown - no allele information

Definition at line 183 of file Insilico.h.

7.113.3.3 AttributeType

```
enum AttributeType
```

Type of attributes that are stored in data set instances.

Enumerator

NUMERIC_TYPE	continuous numeric type
DISCRETE_TYPE	discrete genotype type
NO_TYPE	default no type

Definition at line 160 of file Insilico.h.

7.113.3.4 ClassType

```
enum ClassType
```

Type of classes that are stored in data set instances.

Enumerator

CONTINUOUS_CLASS_TYPE	continuous numeric type
CASE_CONTROL_CLASS_TYPE	discrete case-control type
MULTI_CLASS_TYPE	multiclass type
NO_CLASS_TYPE	default no type

Definition at line 171 of file Insilico.h.

7.113.3.5 CvSetType

```
enum CvSetType
```

Cross validation set types.

Enumerator

CV_TRAIN	Trainging set.
CV_HOLDOUT	Hold out set.
CV_TEST	Testing set.
CV_NONE	

Definition at line 253 of file Insilico.h.

7.113.3.6 EcAlgorithmType

```
enum EcAlgorithmType
```

Type of algorithm steps to perform.

Enumerator

EC_ALG_ME_IT	main effects + interactions algorithms combined
EC_ALG_ME_ONLY	main effects algorithm only
EC_ALG_IT_ONLY	interactions algorithm only

Definition at line 220 of file Insilico.h.

7.113.3.7 EcItAlgorithmType

```
enum EcItAlgorithmType
```

Type of interactions algorithm to execute.

Enumerator

EC_IT_ALG_RF	ReliefF interactions algorithm.
EC_IT_ALG_RFSEQ	ReliefFSeq interactions algorithm.

Definition at line 242 of file Insilico.h.

7.113.3.8 EcMeAlgorithmType

```
enum EcMeAlgorithmType
```

Type of main effects algorithm to execute.

Enumerator

EC_ME_ALG_RJ	Random Jungle main effects algorithm.
EC_ME_ALG_DESEQ	DESeq main effects algorithm.
EC_ME_ALG_EDGER	edgeR main effects algorithm

Definition at line 231 of file Insilico.h.

7.113.3.9 OutputDatasetType

```
enum OutputDatasetType
```

E N U M S.

Type of data set to write filtered output.

Enumerator

TAB_DELIMITED_DATASET	tab-delimited .txt file
CSV_DELIMITED_DATASET	comma separated values .csv file
ARFF_DATASET	WEKA ARFF format .arff file.
PLINK_PED_DATASET	PLINK ped/map format.
PLINK_BED_DATASET	PLINK bed/bim/fam format.
PLINK_COVAR_DATASET	PLINK style covariate format format.
NO_OUTPUT_DATASET	no output data set specified

Definition at line 115 of file Insilico.h.

7.113.3.10 RandomJungleRunMode

```
enum RandomJungleRunMode
```

Run mode for random jungle.

Enumerator

UNKNOWN_RUN_MODE	unknown run mode
LIBRARY_RUN_MODE	Random Jungle through C++ library calls.
SYSTEM_CALL_RUN_MODE	call Random Jungle through C system() call
LIBRARY_FILE_RUN_MODE	C+ library calls with file I/O.

Definition at line 208 of file Insilico.h.

7.113.3.11 RandomJungleTreeType

```
enum RandomJungleTreeType
```

Type random jungle trees.

Enumerator

UNKNOWN_TREE_TYPE	place holder = 0
NOMINAL_NUMERIC_TREE	classification trees, numeric attributes (integers)
NOMINAL_NOMINAL_TREE	classification trees, discrete attributes (0/1/2)
NUMERIC_NUMERIC_TREE	regression trees, numeric attributes (doubles)
NUMERIC_NOMINAL_TREE	regression trees, discrete attributes (0/1/2)
NOMINAL_NUMERIC_FLOATS	classification trees, numeric attributes (doubles)

Definition at line 194 of file Insilico.h.

7.113.3.12 ValueType

```
enum ValueType
```

Return types for determining a value's type.

Enumerator

NUMERIC_VALUE	continuous numeric value
DISCRETE_VALUE	discrete genotype value
MISSING_VALUE	missing value
NO_VALUE	default no value type

Definition at line 148 of file Insilico.h.

7.113.4 Function Documentation

7.113.4.1 ChooseSnpsDatasetByType()

```
Dataset* ChooseSnpsDatasetByType (
    std::string snpsFilename,
    std::string snpsFileType = "")
```

Determines the data set type to instantiate based on the passed type string or data set filenames's extension.

Parameters

in	<i>snpsFilename</i>	SNP data set filename
in	<i>snpsFileType</i>	SNP data set file type (overrides detect by extension)

Returns

pointer to new data set or NULL if could not match a data set type

Definition at line 50 of file Insilico.cpp.

7.113.4.2 DetectClassType()

```
ClassType DetectClassType (
    std::string filename,
    int classColumn,
    bool hasHeader )
```

Detect the class type by reading the specified column from a whitespace- delimited text file.

Parameters

in	<i>filename</i>	whitespace-delimited text file name
in	<i>classColumn</i>	the column containing the class values
in	<i>hasHeader</i>	does the file have a header line?

Returns

ClassType defined in [Dataset.h](#)

Open the file for reading

Skip the header if it has one

Determine the phenotype type

Definition at line 269 of file Insilico.cpp.

7.113.4.3 DetermineRandomJungleTreeType()

```
RandomJungleTreeType DetermineRandomJungleTreeType (
    AttributeType attributeType,
    ClassType classType )
```

Return random jungle tree type from the class and attribute types.

Parameters

in	<i>attributeType</i>	attribute data type
in	<i>classType</i>	class data type

Returns

Random Jungle tree type

7.113.4.4 GetConfigValue()

```
bool GetConfigValue (
    ConfigMap & configMap,
    std::string key,
    std::string & value )
```

Get the parameter value from the configuration map key.

Parameters

in	<i>configMap</i>	reference to a configuration map
in	<i>key</i>	parameter name
out	<i>parameter</i>	value

Returns

true if key found, false if not found

Definition at line 336 of file Insilico.cpp.

7.113.4.5 GetFileBasename()

```
std::string GetFileBasename (
    std::string fullFilename )
```

Get the full filename without the extension.

Parameters

in	<i>fullFilename</i>	complete filename
----	---------------------	-------------------

Returns

path/filename without extension

Definition at line 348 of file Insilico.cpp.

7.113.4.6 GetFileExtension()

```
std::string GetFileExtension (
    std::string fullFilename )
```

Get the filename extension.

Parameters

in	<i>fullFilename</i>	complete filename
----	---------------------	-------------------

Returns

filename extension

Definition at line 353 of file Insilico.cpp.

7.113.4.7 GetMatchingIds()

```
bool GetMatchingIds (
    std::string numericsFilename,
    std::string altPhenotypeFilename,
    std::vector< std::string > numericsIds,
    std::vector< std::string > phenoIds,
    std::vector< std::string > & matchingIds )
```

Return matching IDs from numeric and/or phenotype file IDs.

Parameters

in	<i>numericsFilename</i>	name of the PLINK covar format file
in	<i>altPhenotypeFilename</i>	name of the alternate pheno file PLINK
in	<i>numericsIds</i>	covar format file ids
in	<i>phenoIds</i>	alternate phenotype file ids
out	<i>matchingIds</i>	ids that match between numerics and phenotypes

Returns

success

Definition at line 203 of file Insilico.cpp.

7.113.4.8 LoadNumericIds()

```
bool LoadNumericIds (
    std::string filename,
    std::vector< std::string > & retIds )
```

Loads the individual (instance) IDs from the numerics file.

Returns the IDs through reference parameter retIds.

Parameters

in	<i>filename</i>	filename that contains numerics IDs
out	<i>vector</i>	of individual (instance) IDs (strings)

Returns

success

Definition at line 96 of file Insilico.cpp.

7.113.4.9 LoadPhenoIds()

```
bool LoadPhenoIds (
    std::string filename,
    std::vector< std::string > & retIds )
```

Loads the individual (instance) IDs from the numerics file.

Returns the IDs through reference parameter retIds.

Parameters

in	<i>filename</i>	filename that contains numerics IDs
out	<i>vector</i>	of individual (instance) IDs (strings)

Returns

success

Definition at line 151 of file Insilico.cpp.

7.113.4.10 PrintVector()

```
template<class T >
void PrintVector (
    std::vector< T > vec,
    std::string title = "" )
```

Print a vector of T values with optional title.

Parameters

in	<i>vec</i>	vector of T type values
in	<i>title</i>	optional title to print before the vector

Definition at line 352 of file Insilico.h.

7.113.4.11 ProtectedLog()

```
double ProtectedLog (
    double x )
```

protected log function returns 0 for 0

Definition at line 358 of file Insilico.cpp.

7.113.4.12 scoresSortAsc()

```
bool scoresSortAsc (
    const ScoreVarPair & p1,
    const ScoreVarPair & p2 )
```

Definition at line 26 of file Insilico.cpp.

7.113.4.13 scoresSortAscByName()

```
bool scoresSortAscByName (
    const ScoreVarPair & p1,
    const ScoreVarPair & p2 )
```

Definition at line 34 of file Insilico.cpp.

7.113.4.14 scoresSortDesc()

```
bool scoresSortDesc (
    const ScoreVarPair & p1,
    const ScoreVarPair & p2 )
```

Definition at line 30 of file Insilico.cpp.

7.113.4.15 Timestamp()

```
std::string Timestamp ( )
```

Return a timestamp string for logging purposes.

Returns

fixed-length, formatted timestamp as a string

Definition at line 38 of file Insilico.cpp.

7.113.5 Variable Documentation

7.113.5.1 COMMAND_LINE_ERROR

```
const int COMMAND_LINE_ERROR = EXIT_FAILURE [static]
```

CONSTANTS.

Error codes.

Definition at line 73 of file Insilico.h.

7.113.5.2 DATASET_LOAD_ERROR

```
const int DATASET_LOAD_ERROR = EXIT_FAILURE [static]
```

Definition at line 74 of file Insilico.h.

7.113.5.3 datasetTypeToExt

```
std::map<std::string, std::string> datasetTypeToExt [static]
```

Definition at line 261 of file Insilico.h.

7.113.5.4 DEFAULT_FDR

```
const double DEFAULT_FDR = 0.05 [static]
```

Definition at line 104 of file Insilico.h.

7.113.5.5 DEFAULT_PVALUE

```
const double DEFAULT_PVALUE = 1.0 [static]
```

default p-value threshold

Definition at line 102 of file Insilico.h.

7.113.5.6 DEFAULT_PVALUE_THRESHOLD

```
const double DEFAULT_PVALUE_THRESHOLD = 0.05 [static]
```

Definition at line 103 of file Insilico.h.

7.113.5.7 DEFAULT_ZVALUE

```
const double DEFAULT_ZVALUE = 0.0 [static]
```

Definition at line 105 of file Insilico.h.

7.113.5.8 INVALID_ATTRIBUTE_VALUE

```
const AttributeLevel INVALID_ATTRIBUTE_VALUE = INT_MIN [static]
```

invalid attribute value

Definition at line 84 of file Insilico.h.

7.113.5.9 INVALID_DISCRETE_CLASS_VALUE

```
const ClassLevel INVALID_DISCRETE_CLASS_VALUE = INT_MIN [static]
```

stored value for missing discrete class

Definition at line 88 of file Insilico.h.

7.113.5.10 INVALID_DISTANCE

```
const int INVALID_DISTANCE = INT_MAX [static]
```

return value for invalid distance

Definition at line 77 of file Insilico.h.

7.113.5.11 INVALID_INDEX

```
const int INVALID_INDEX = INT_MAX [static]
```

return value for invalid index into attributes

Definition at line 79 of file Insilico.h.

7.113.5.12 INVALID_INT_VALUE

```
const unsigned int INVALID_INT_VALUE = UINT_MAX [static]
```

return value for invalid index into attributes

Definition at line 81 of file Insilico.h.

7.113.5.13 INVALID_NUMERIC_CLASS_VALUE

```
const NumericLevel INVALID_NUMERIC_CLASS_VALUE = INT_MIN [static]
```

stored value for missing numeric class

Definition at line 90 of file Insilico.h.

7.113.5.14 INVALID_NUMERIC_VALUE

```
const NumericLevel INVALID_NUMERIC_VALUE = INT_MIN [static]
```

invalid attribute value

Definition at line 86 of file Insilico.h.

7.113.5.15 MIN_NUM_GENES

```
const uint MIN_NUM_GENES = 4 [static]
```

Definition at line 106 of file Insilico.h.

7.113.5.16 MIN_NUM_SUBJ_PER_GROUP

```
const uint MIN_NUM_SUBJ_PER_GROUP = 4 [static]
```

Definition at line 107 of file Insilico.h.

7.113.5.17 MISSING_ATTRIBUTE_VALUE

```
const AttributeLevel MISSING_ATTRIBUTE_VALUE = -9 [static]
```

stored value for missing discrete attribute

Definition at line 93 of file Insilico.h.

7.113.5.18 MISSING_DISCRETE_CLASS_VALUE

```
const ClassLevel MISSING_DISCRETE_CLASS_VALUE = -9 [static]
```

stored value for missing discrete class

Definition at line 97 of file Insilico.h.

7.113.5.19 MISSING_NUMERIC_CLASS_VALUE

```
const NumericLevel MISSING_NUMERIC_CLASS_VALUE = -9 [static]
```

stored value for missing numeric class

Definition at line 99 of file Insilico.h.

7.113.5.20 MISSING_NUMERIC_VALUE

```
const NumericLevel MISSING_NUMERIC_VALUE = -9 [static]
```

stored value for missing numeric attribute

Definition at line 95 of file Insilico.h.

7.113.5.21 PP

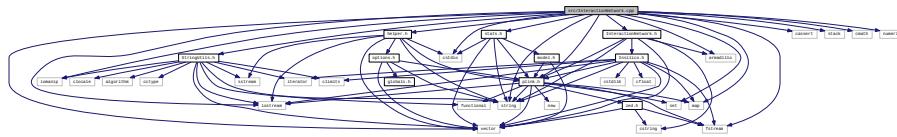
```
PLink* PP
```

Definition at line 85 of file inbix.cpp.

7.114 src/InteractionNetwork.cpp File Reference

```
#include <cstdio>
#include <cassert>
#include <iostream>
#include <fstream>
#include <iomanip>
#include <string>
#include <vector>
#include <stack>
#include <set>
#include <map>
#include <cmath>
#include <numeric>
#include <armadillo>
#include "plink.h"
#include "helper.h"
#include "stats.h"
#include "StringUtils.h"
#include "InteractionNetwork.h"

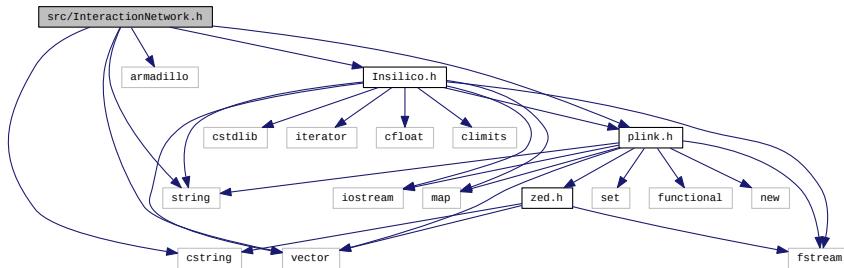
Include dependency graph for InteractionNetwork.cpp:
```



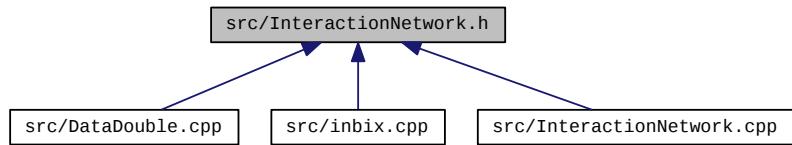
7.115 src/InteractionNetwork.h File Reference

```
#include <string>
#include <vector>
#include <cstring>
#include <armadillo>
#include "plink.h"
#include "Insilico.h"

Include dependency graph for InteractionNetwork.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- struct `_RipmResult`
- class `InteractionNetwork`

Typedefs

- typedef unsigned int `Indices`
- typedef std::vector<`Indices`> `ModuleIndices`
- typedef std::vector<`ModuleIndices`> `ModuleList`
- typedef std::pair<double, `ModuleList`> `ModularityResult`
- typedef std::pair<double, std::vector<double>> `HomophilyResult`
- typedef struct `_RipmResult` `RipmResult`

Enumerations

- enum `MatrixFileType` {
 `INVALID_FILE`, `REGAIN_FILE`, `CORR_1D_FILE`, `CSV_FILE`,
 `SIF_FILE` }
- enum `NetworkMatrixType` { `NET_MATRIX_INVALID`, `NET_MATRIX_ADJ`, `NET_MATRIX_CON`, `NET_MATRIX_BOTH` }

Variables

- const double `DEFAULT_CONNECTIVITY_THRESHOLD` = 0
- const double `MODULARITY_THRESHOLD` = 0

7.115.1 Typedef Documentation

7.115.1.1 HomophilyResult

```
typedef std::pair<double, std::vector<double> > HomophilyResult
```

Definition at line 26 of file InteractionNetwork.h.

7.115.1.2 Indices

```
typedef unsigned int Indices
```

Definition at line 22 of file InteractionNetwork.h.

7.115.1.3 ModularityResult

```
typedef std::pair<double, ModuleList> ModularityResult
```

Definition at line 25 of file InteractionNetwork.h.

7.115.1.4 ModuleIndices

```
typedef std::vector<Indices> ModuleIndices
```

Definition at line 23 of file InteractionNetwork.h.

7.115.1.5 ModuleList

```
typedef std::vector<ModuleIndices> ModuleList
```

Definition at line 24 of file InteractionNetwork.h.

7.115.1.6 RipmResult

```
typedef struct _RipmResult RipmResult
```

7.115.2 Enumeration Type Documentation

7.115.2.1 MatrixFileType

```
enum MatrixFileType
```

Enumerator

INVALID_FILE	
REGAIN_FILE	
CORR_1D_FILE	
CSV_FILE	
SIF_FILE	

Definition at line 36 of file InteractionNetwork.h.

7.115.2.2 NetworkMatrixType

```
enum NetworkMatrixType
```

Enumerator

NET_MATRIX_INVALID	
NET_MATRIX_ADJ	
NET_MATRIX_CON	
NET_MATRIX_BOTH	

Definition at line 44 of file InteractionNetwork.h.

7.115.3 Variable Documentation

7.115.3.1 DEFAULT_CONNECTIVITY_THRESHOLD

```
const double DEFAULT_CONNECTIVITY_THRESHOLD = 0
```

Definition at line 19 of file InteractionNetwork.h.

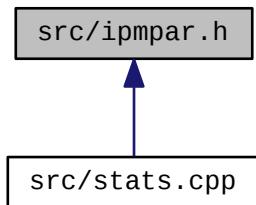
7.115.3.2 MODULARITY_THRESHOLD

```
const double MODULARITY_THRESHOLD = 0
```

Definition at line 20 of file InteractionNetwork.h.

7.116 src/ipmpar.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- int [ipmpar](#) (int *i)

7.116.1 Function Documentation

7.116.1.1 ipmpar()

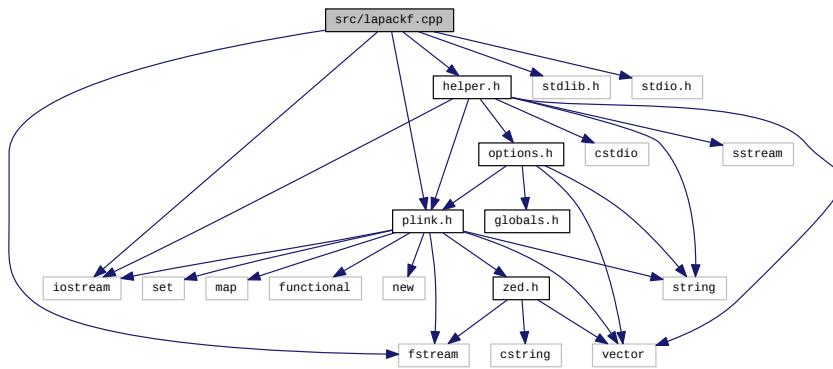
```
int ipmpar (
    int * i )
```

Definition at line 71 of file ipmpar.h.

7.117 src/lapackf.cpp File Reference

```
#include <iostream>
#include <fstream>
#include <stdlib.h>
#include <stdio.h>
#include "plink.h"
```

```
#include "helper.h"
Include dependency graph for lapackf.cpp:
```



Functions

- bool `svd_lapack` (int n, `vector_t` &A, `vector_t` &S, `matrix_t` &V)
- bool `eigen_lapack` (int n, `vector_t` &A, `vector_t` &S, `matrix_t` &V)

7.117.1 Function Documentation

7.117.1.1 eigen_lapack()

```
bool eigen_lapack (
    int n,
    vector_t & A,
    vector_t & S,
    matrix_t & V )
```

Definition at line 86 of file `lapackf.cpp`.

7.117.1.2 svd_lapack()

```
bool svd_lapack (
    int n,
    vector_t & A,
    vector_t & S,
    matrix_t & V )
```

Definition at line 40 of file `lapackf.cpp`.

7.118 src/lapackf.h File Reference

Functions

- bool `svd_lapack` (int, `vector_t` &A, `vector_t` &S, `matrix_t` &V)
- bool `eigen_lapack` (int, `vector_t` &A, `vector_t` &S, `matrix_t` &V)

7.118.1 Function Documentation

7.118.1.1 `eigen_lapack()`

```
bool eigen_lapack (
    int ,
    vector_t & A,
    vector_t & S,
    matrix_t & V )
```

Definition at line 86 of file `lapackf.cpp`.

7.118.1.2 `svd_lapack()`

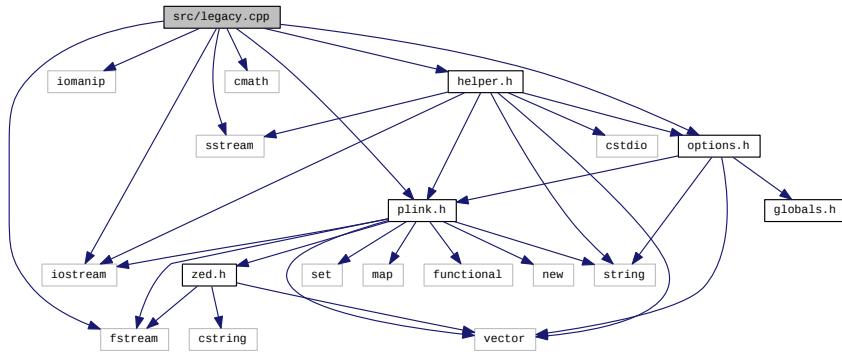
```
bool svd_lapack (
    int ,
    vector_t & A,
    vector_t & S,
    matrix_t & V )
```

Definition at line 40 of file `lapackf.cpp`.

7.119 src/legacy.cpp File Reference

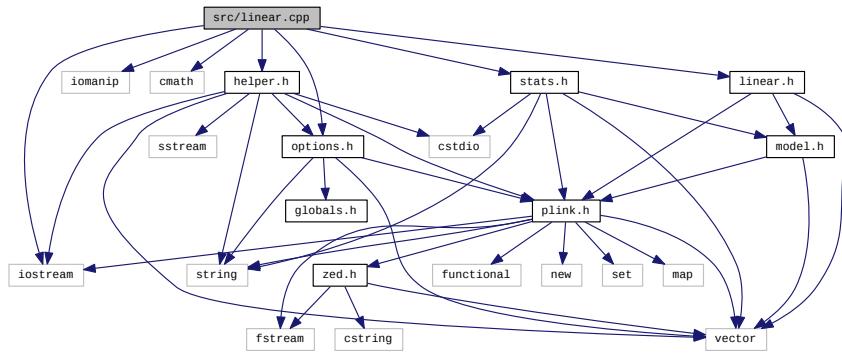
```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <cmath>
#include "plink.h"
#include "helper.h"
```

```
#include "options.h"
Include dependency graph for legacy.cpp:
```



7.120 src/linear.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <cmath>
#include "linear.h"
#include "helper.h"
#include "options.h"
#include "stats.h"
Include dependency graph for linear.cpp:
```



Functions

- void `covsrt (matrix_t &covar, vector< bool > &ia, const int mfit)`
- void `gaussj (matrix_t &a, matrix_t &b)`
- void `lfit (vector_t &x, vector_t &y, vector_t &sig, vector_t &a, vector< bool > &ia, matrix_t &covar, double &chisq, matrix_t &X)`

7.120.1 Function Documentation

7.120.1.1 covsrt()

```
void covsrt (
    matrix_t & covar,
    vector< bool > & ia,
    const int mfit )
```

Definition at line 50 of file linear.cpp.

7.120.1.2 gaussj()

```
void gaussj (
    matrix_t & a,
    matrix_t & b )
```

Definition at line 66 of file linear.cpp.

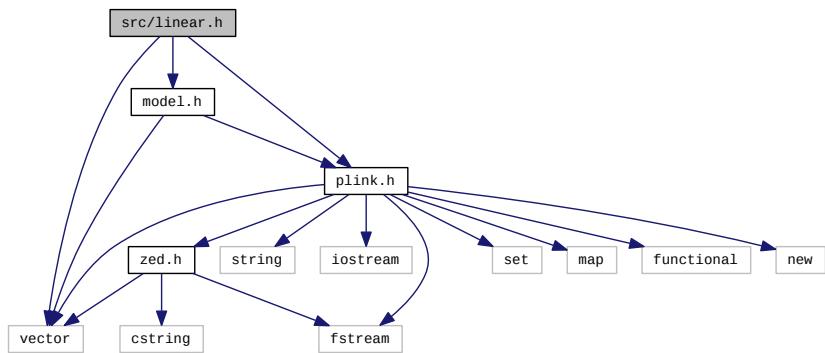
7.120.1.3 lfit()

```
void lfit (
    vector_t & x,
    vector_t & y,
    vector_t & sig,
    vector_t & a,
    vector< bool > & ia,
    matrix_t & covar,
    double & chisq,
    matrix_t & X )
```

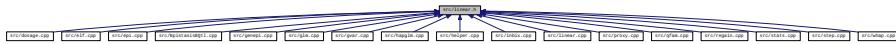
Definition at line 114 of file linear.cpp.

7.121 src/linear.h File Reference

```
#include <vector>
#include "plink.h"
#include "model.h"
Include dependency graph for linear.h:
```



This graph shows which files directly or indirectly include this file:



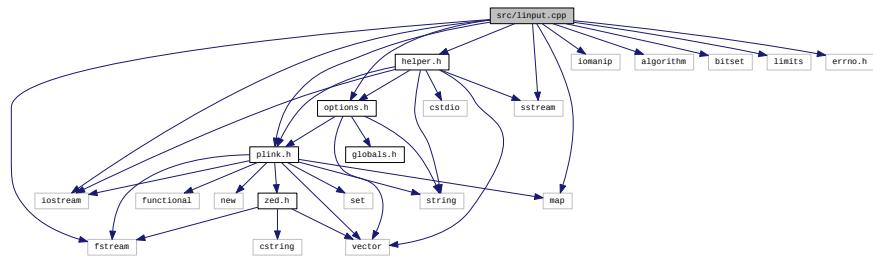
Classes

- class [LinearModel](#)

7.122 src/linput.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <map>
#include <algorithm>
#include <set>
#include <limits>
#include <errno.h>
#include "plink.h"
#include "options.h"
```

```
#include "helper.h"
Include dependency graph for linput.cpp:
```



Variables

- `ofstream LOG`

7.122.1 Variable Documentation

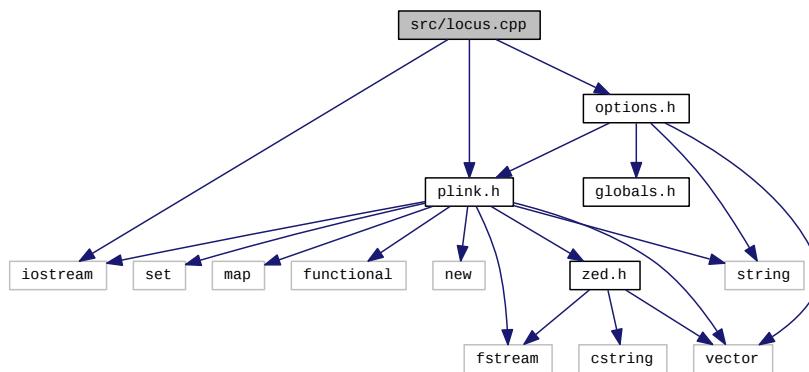
7.122.1.1 LOG

`ofstream LOG`

Definition at line 81 of file `inbix.cpp`.

7.123 src/locus.cpp File Reference

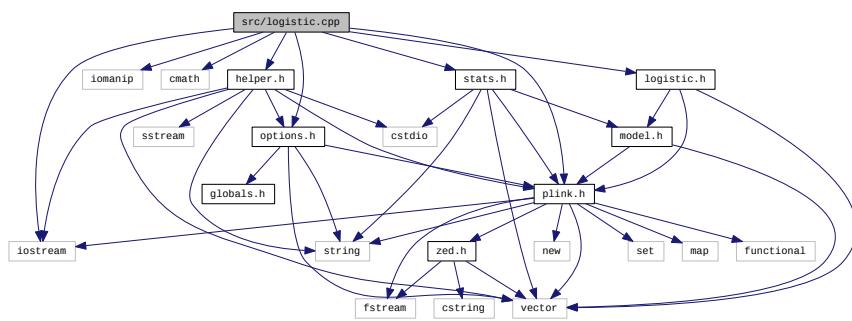
```
#include <iostream>
#include "plink.h"
#include "options.h"
Include dependency graph for locus.cpp:
```



7.124 src/logistic.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <cmath>
#include "logistic.h"
#include "plink.h"
#include "helper.h"
#include "options.h"
#include "stats.h"
#include "stats.h"

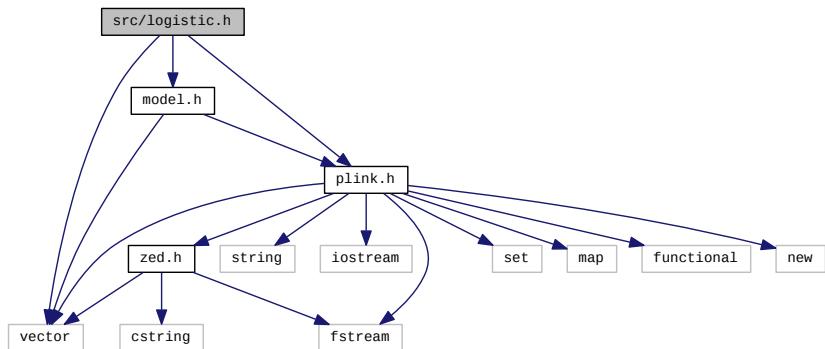
Include dependency graph for logistic.cpp:
```



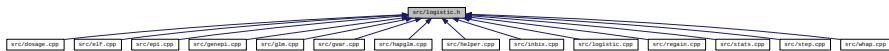
7.125 src/logistic.h File Reference

```
#include <vector>
#include "plink.h"
#include "model.h"

Include dependency graph for logistic.h:
```



This graph shows which files directly or indirectly include this file:

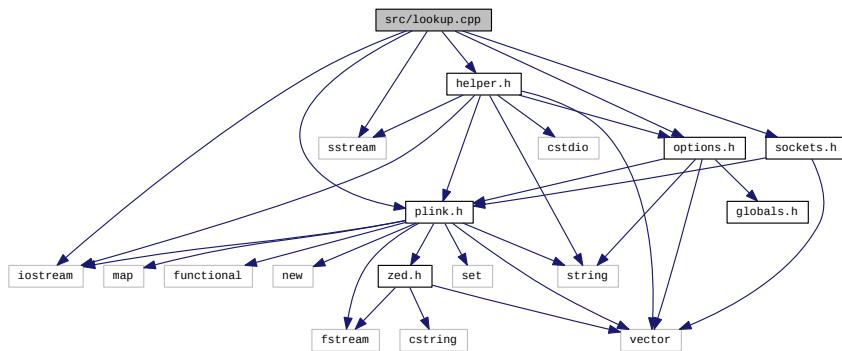


Classes

- class [LogisticModel](#)

7.126 src/lookup.cpp File Reference

```
#include <iostream>
#include <sstream>
#include "plink.h"
#include "helper.h"
#include "options.h"
#include "sockets.h"
Include dependency graph for lookup.cpp:
```



Macros

- `#define PORT_NUM 80`
- `#define IP_ADDR "132.183.161.22"`

7.126.1 Macro Definition Documentation

7.126.1.1 IP_ADDR

```
#define IP_ADDR "132.183.161.22"
```

Definition at line 25 of file lookup.cpp.

7.126.1.2 PORT_NUM

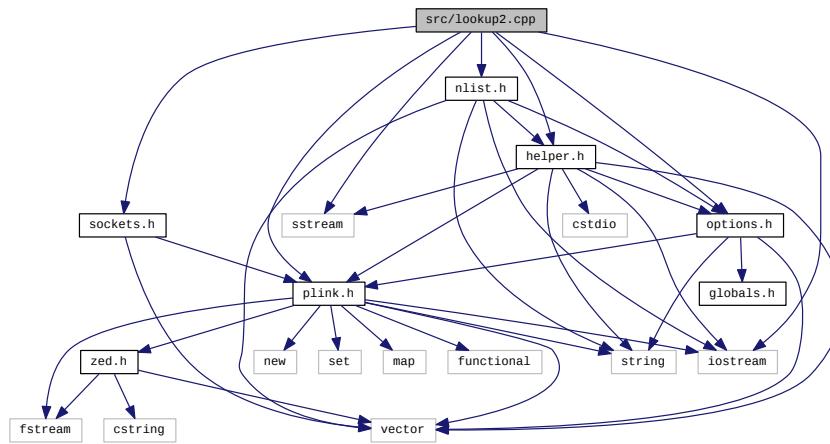
```
#define PORT_NUM 80
```

Definition at line 24 of file lookup.cpp.

7.127 src/lookup2.cpp File Reference

```
#include <iostream>
#include <sstream>
#include "plink.h"
#include "helper.h"
#include "options.h"
#include "sockets.h"
#include "nlist.h"
```

Include dependency graph for lookup2.cpp:



Macros

- #define PORT_NUM 80
- #define IP_ADDR "152.19.78.148"

Functions

- void [convertPosition](#) (string pquery, int &chr, int &bp1, int &bp2, bool useKb, bool useMb)

7.127.1 Macro Definition Documentation

7.127.1.1 IP_ADDR

```
#define IP_ADDR "152.19.78.148"
```

Definition at line 26 of file lookup2.cpp.

7.127.1.2 PORT_NUM

```
#define PORT_NUM 80
```

Definition at line 25 of file lookup2.cpp.

7.127.2 Function Documentation

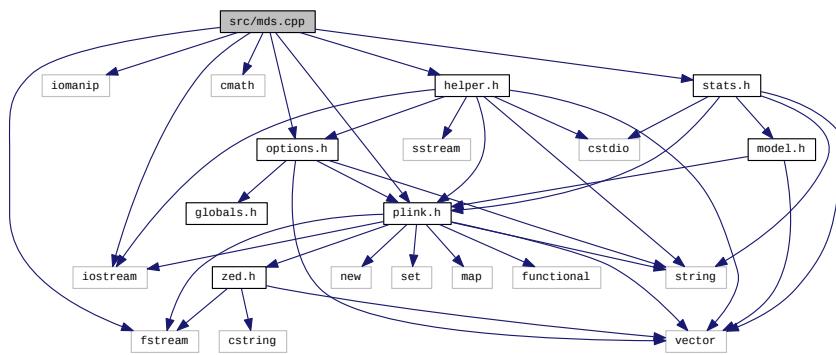
7.127.2.1 convertPosition()

```
void convertPosition (
    string pquery,
    int & chr,
    int & bp1,
    int & bp2,
    bool useKb,
    bool useMb )
```

Definition at line 30 of file lookup2.cpp.

7.128 src/mds.cpp File Reference

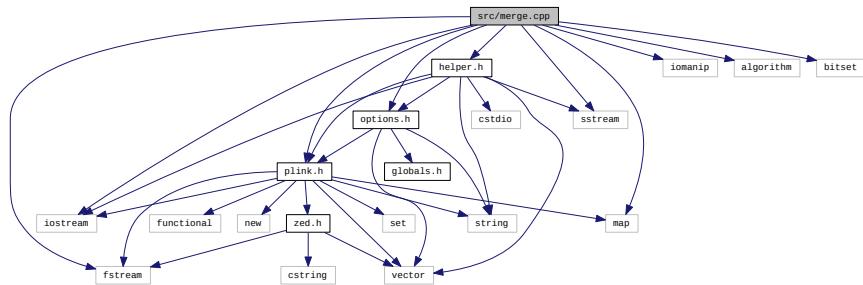
```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <cmath>
#include "plink.h"
#include "options.h"
#include "helper.h"
#include "stats.h"
Include dependency graph for mds.cpp:
```



7.129 src/merge.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <map>
#include <algorithm>
#include <set>
#include "plink.h"
#include "options.h"
#include "helper.h"
```

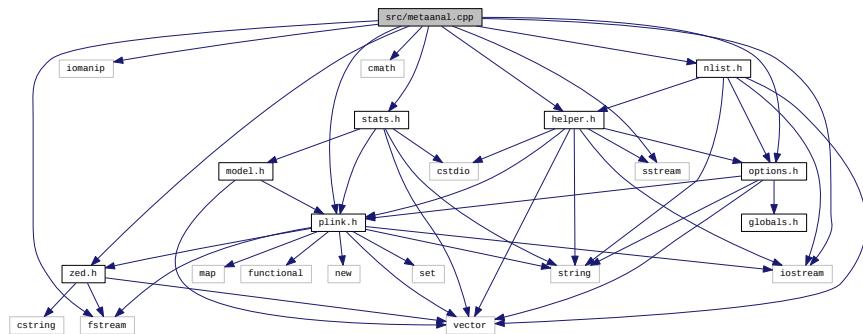
Include dependency graph for merge.cpp:



7.130 src/metaanal.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <cmath>
#include "stats.h"
#include "options.h"
#include "plink.h"
#include "helper.h"
#include "zed.h"
#include "nlist.h"
```

Include dependency graph for metaanal.cpp:



Classes

- class [SInfo](#)
- class [Alleles](#)

Typedefs

- `typedef map< Alleles, map< int, SInfo > > mymap_t`

Variables

- `Plink * PP`

7.130.1 Typedef Documentation

7.130.1.1 mymap_t

```
typedef map<Alleles, map<int, SInfo>> mymap_t
```

Definition at line 91 of file metaanal.cpp.

7.130.2 Variable Documentation

7.130.2.1 PP

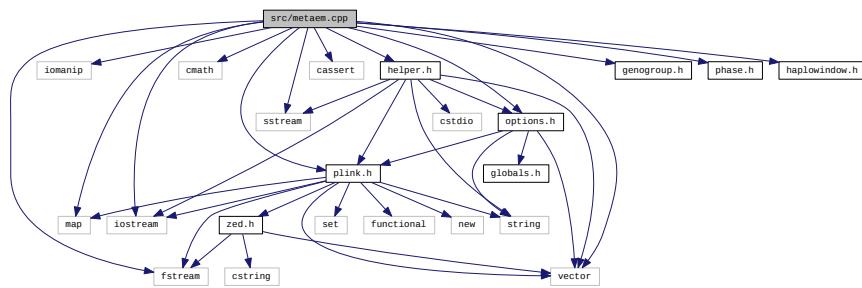
```
Plink* PP
```

Definition at line 85 of file inbix.cpp.

7.131 src/metaem.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <cmath>
#include <vector>
#include <map>
#include <cassert>
#include "plink.h"
#include "options.h"
#include "helper.h"
#include "genogroup.h"
#include "phase.h"
```

```
#include "haplowindow.h"
Include dependency graph for metaem.cpp:
```



Variables

- `ofstream LOG`

7.131.1 Variable Documentation

7.131.1.1 LOG

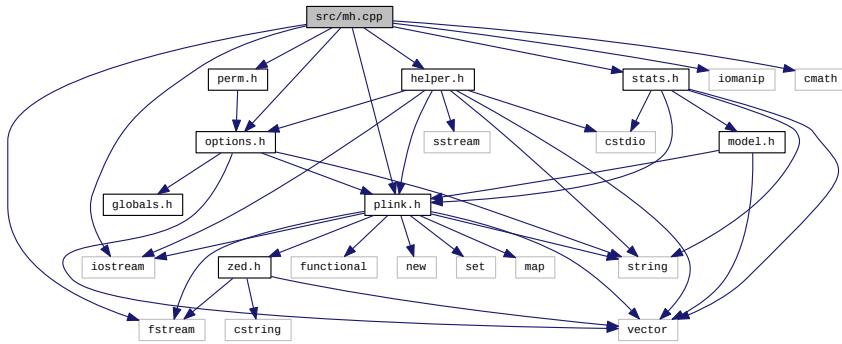
`ofstream LOG`

Definition at line 81 of file `inbix.cpp`.

7.132 src/mh.cpp File Reference

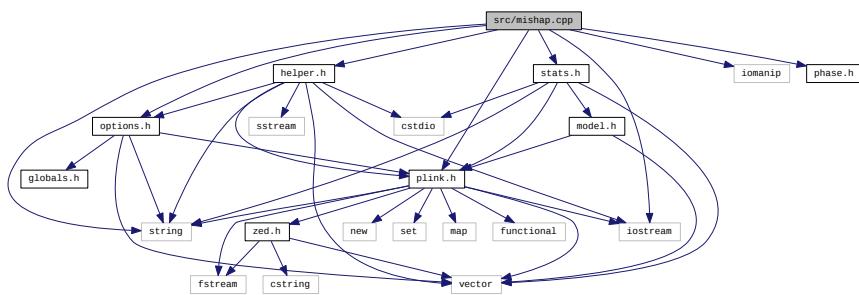
```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <cmath>
#include "plink.h"
#include "perm.h"
#include "options.h"
#include "helper.h"
```

```
#include "stats.h"
Include dependency graph for mh.cpp:
```



7.133 src/mishap.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <string>
#include "options.h"
#include "helper.h"
#include "plink.h"
#include "phase.h"
#include "stats.h"
Include dependency graph for mishap.cpp:
```



Variables

- [Plink * PP](#)

7.133.1 Variable Documentation

7.133.1.1 PP

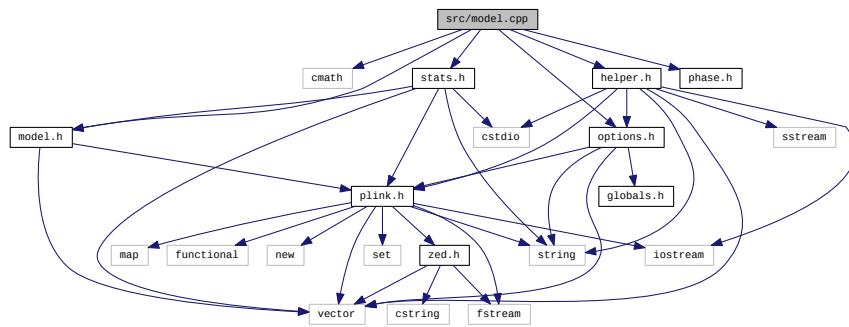
[Plink*](#) PP

Definition at line 85 of file inbix.cpp.

7.134 src/model.cpp File Reference

```
#include <cmath>
#include "model.h"
#include "options.h"
#include "helper.h"
#include "phase.h"
#include "stats.h"
```

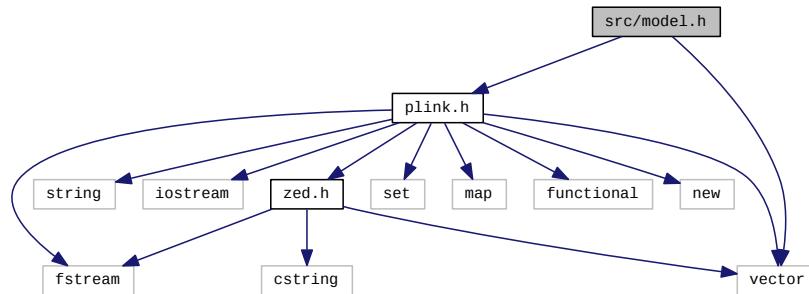
Include dependency graph for model.cpp:



7.135 src/model.h File Reference

```
#include <vector>
#include "plink.h"
```

Include dependency graph for model.h:



This graph shows which files directly or indirectly include this file:



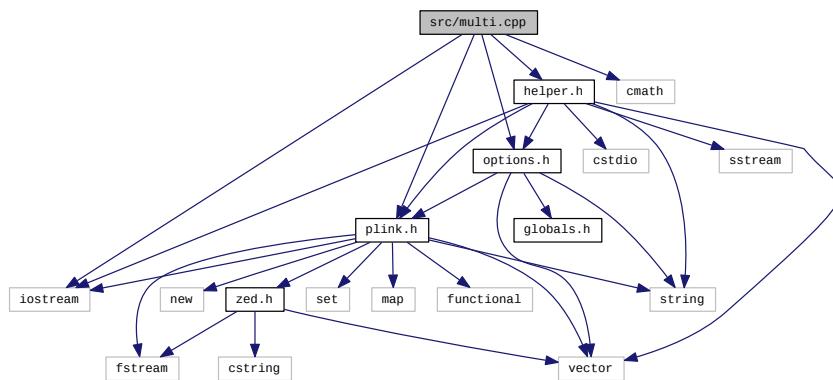
Classes

- class [Model](#)

7.136 src/multi.cpp File Reference

```
#include <iostream>
#include <cmath>
#include "options.h"
#include "plink.h"
#include "helper.h"
```

Include dependency graph for multi.cpp:



Macros

- #define [EPS](#) 0.00001

7.136.1 Macro Definition Documentation

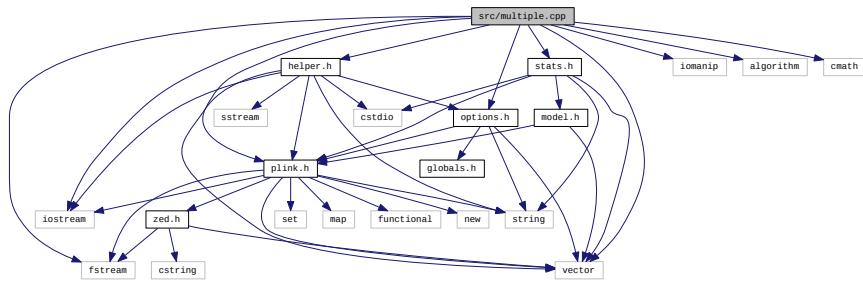
7.136.1.1 EPS

```
#define EPS 0.00001
```

Definition at line 20 of file multi.cpp.

7.137 src/multiple.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <vector>
#include <algorithm>
#include <cmath>
#include "plink.h"
#include "helper.h"
#include "stats.h"
#include "options.h"
Include dependency graph for multiple.cpp:
```



Classes

- class [Pair](#)

Functions

- void [pprint](#) (ofstream &MT, double p)

7.137.1 Function Documentation

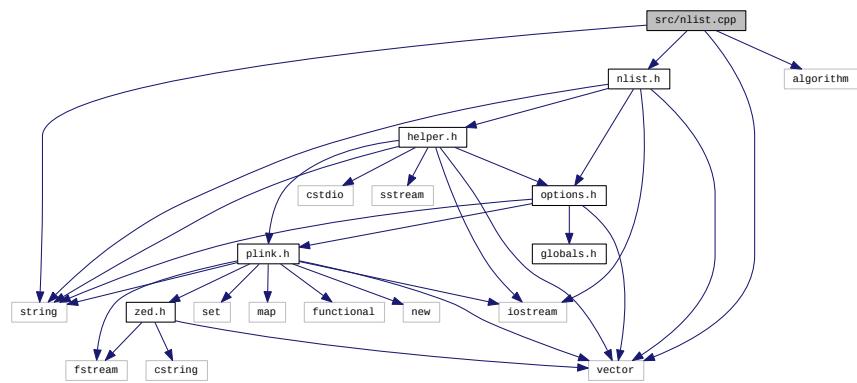
7.137.1.1 [pprint\(\)](#)

```
void pprint (
    ofstream & MT,
    double p )
```

Definition at line 51 of file multiple.cpp.

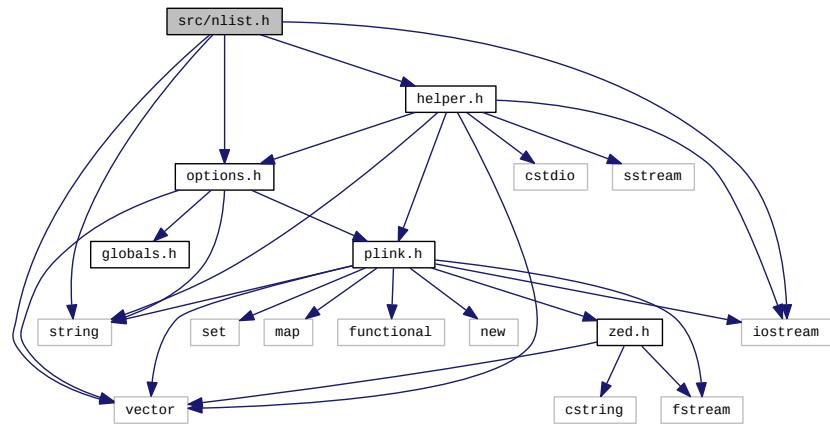
7.138 src/nlist.cpp File Reference

```
#include <string>
#include <vector>
#include <algorithm>
#include "nlist.h"
Include dependency graph for nlist.cpp:
```

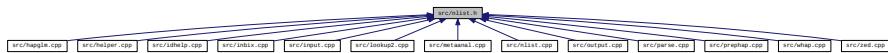


7.139 src/nlist.h File Reference

```
#include <string>
#include <vector>
#include <iostream>
#include "helper.h"
#include "options.h"
Include dependency graph for nlist.h:
```



This graph shows which files directly or indirectly include this file:



Classes

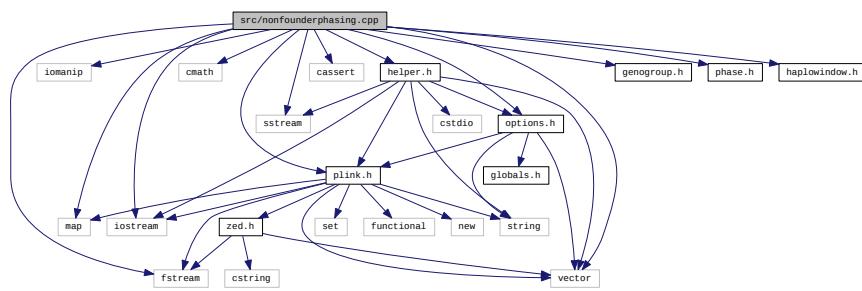
- class [NList](#)

7.140 src/nonfounderphasing.cpp File Reference

```

#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <cmath>
#include <vector>
#include <map>
#include <cassert>
#include "plink.h"
#include "options.h"
#include "helper.h"
#include "genogroup.h"
#include "phase.h"
#include "haplowindow.h"
  
```

Include dependency graph for nonfounderphasing.cpp:



Functions

- void [displayFamTran](#) (map< FamilyTransmissions, double > &pmap, int fi, [HaploPhase](#) *HP)

Variables

- [Plink](#) * PP

7.140.1 Function Documentation

7.140.1.1 displayFamTran()

```
void displayFamTran (
    map< FamilyTransmissions, double > & pmap,
    int fi,
    HaploPhase * HP )
```

Definition at line 33 of file nonfounderphasing.cpp.

7.140.2 Variable Documentation

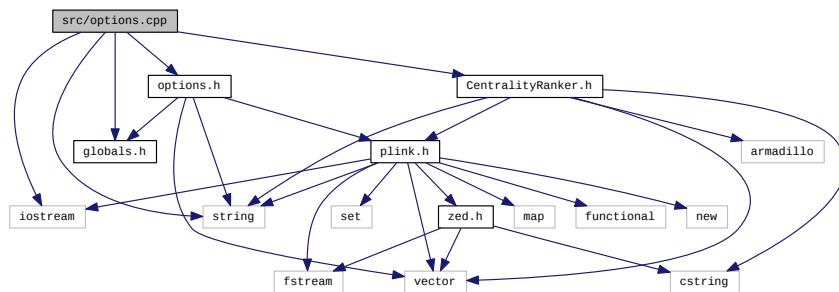
7.140.2.1 PP

```
Plink* PP
```

Definition at line 85 of file inbix.cpp.

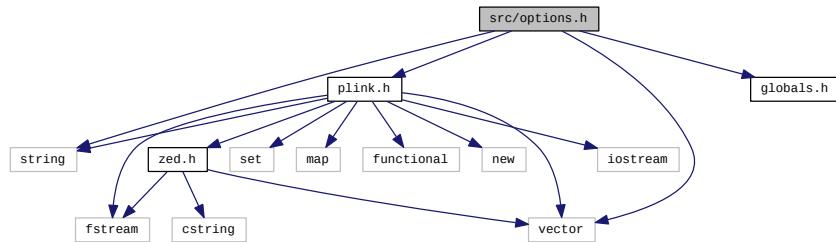
7.141 src/options.cpp File Reference

```
#include <iostream>
#include <string>
#include "options.h"
#include "CentralityRanker.h"
#include "globals.h"
Include dependency graph for options.cpp:
```



7.142 src/options.h File Reference

```
#include <string>
#include <vector>
#include "plink.h"
#include "globals.h"
Include dependency graph for options.h:
```



Classes

- class [OptionSet](#)
- class [Options](#)
- class [par](#)

Functions

- void [setOptions \(CArgs &\)](#)
- void [getOutputFilename \(CArgs &\)](#)

7.142.1 Function Documentation

7.142.1.1 [getOutputFilename\(\)](#)

```
void getOutputFilename (
    CArgs & )
```

Definition at line 21 of file `parse.cpp`.

7.142.1.2 setOptions()

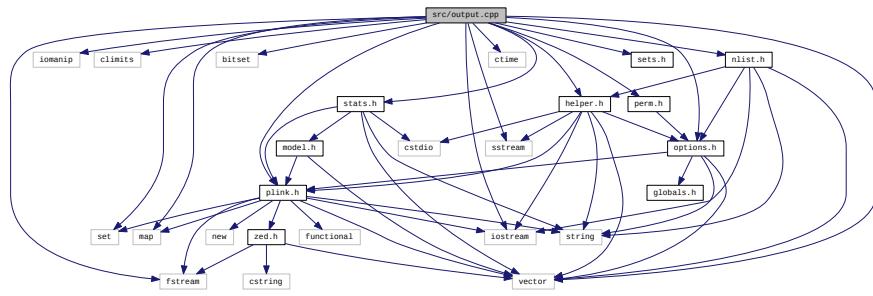
```
void setOptions (
    CArgs & )
```

Definition at line 62 of file parse.cpp.

7.143 src/output.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <climits>
#include <fstream>
#include <sstream>
#include <set>
#include <map>
#include <vector>
#include <ctime>
#include "plink.h"
#include "options.h"
#include "perm.h"
#include "sets.h"
#include "helper.h"
#include "nlist.h"
#include "stats.h"
```

Include dependency graph for output.cpp:



Functions

- void [make2LTable \(ofstream &TWOL, Plink &P, int m1, int m2, vector_t count, bool percent\)](#)

Variables

- ofstream [LOG](#)

7.143.1 Function Documentation

7.143.1.1 make2LTable()

```
void make2LTable (
    ofstream & TWOL,
    Plink & P,
    int m1,
    int m2,
    vector_t count,
    bool percent )
```

Definition at line 271 of file output.cpp.

7.143.2 Variable Documentation

7.143.2.1 LOG

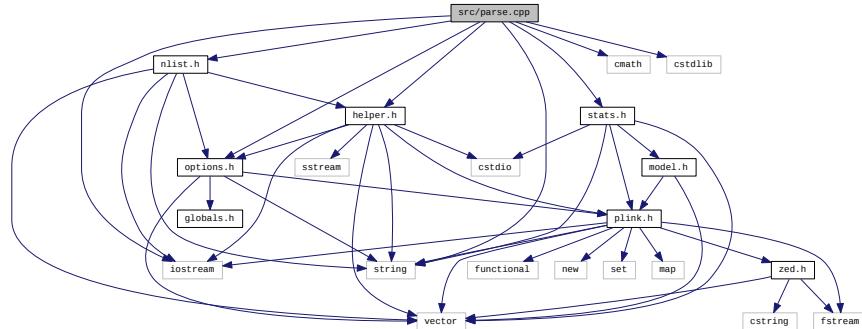
```
ofstream LOG
```

Definition at line 81 of file inbix.cpp.

7.144 src/parse.cpp File Reference

```
#include <iostream>
#include <string>
#include <cmath>
#include <cstdlib>
#include "options.h"
#include "helper.h"
#include "stats.h"
#include "nlist.h"
```

Include dependency graph for parse.cpp:



Functions

- void [getOutputFilename \(CArgs &a\)](#)
- void [setOptions \(CArgs &a\)](#)

7.144.1 Function Documentation

7.144.1.1 getOutputFilename()

```
void getOutputFilename (
    CArgs & a )
```

Definition at line 21 of file parse.cpp.

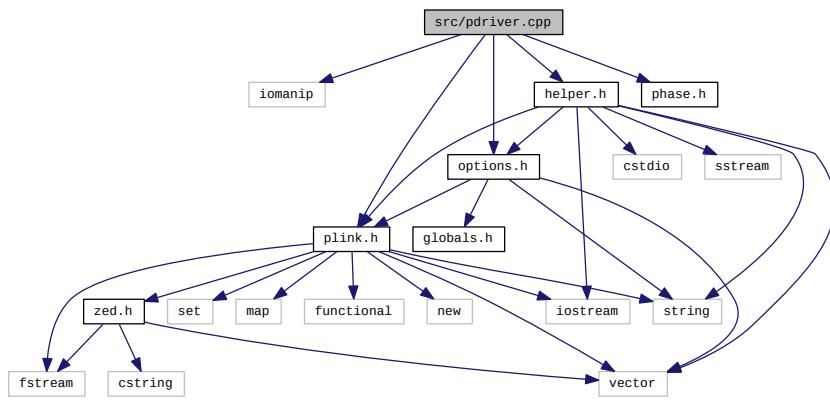
7.144.1.2 setOptions()

```
void setOptions (
    CArgs & a )
```

Definition at line 62 of file parse.cpp.

7.145 src/pdriver.cpp File Reference

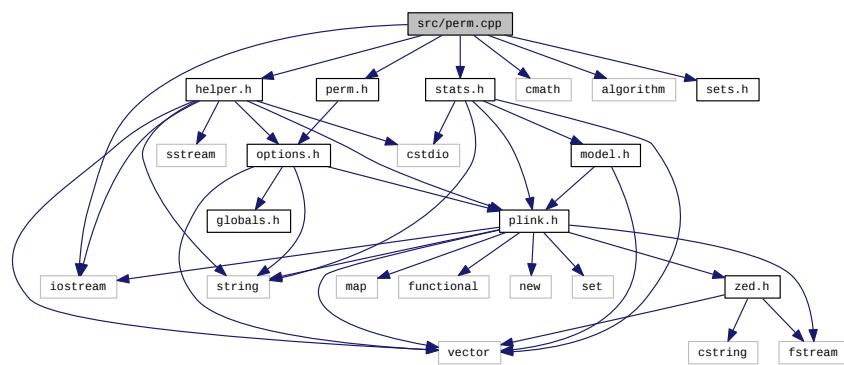
```
#include <iomanip>
#include "plink.h"
#include "options.h"
#include "helper.h"
#include "phase.h"
Include dependency graph for pdriver.cpp:
```



7.146 src/perm.cpp File Reference

```
#include <iostream>
#include <cmath>
#include <algorithm>
#include "perm.h"
#include "helper.h"
#include "stats.h"
#include "sets.h"

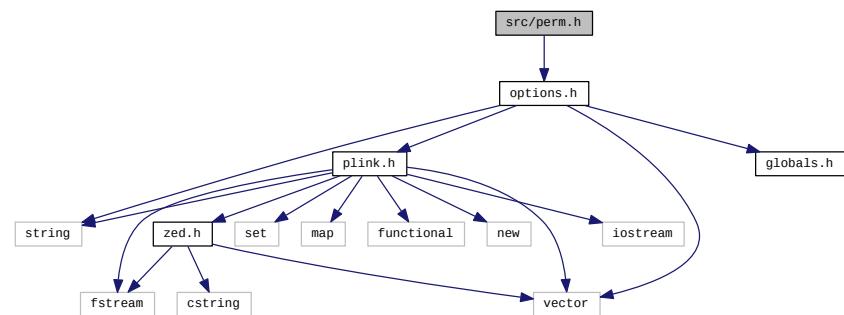
Include dependency graph for perm.cpp:
```



7.147 src/perm.h File Reference

```
#include "options.h"

Include dependency graph for perm.h:
```



This graph shows which files directly or indirectly include this file:



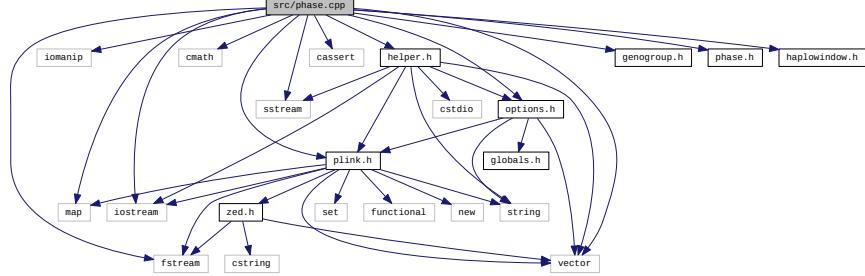
Classes

- class [Perm](#)

7.148 src/phase.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <cmath>
#include <vector>
#include <map>
#include <cassert>
#include "plink.h"
#include "options.h"
#include "helper.h"
#include "genogroup.h"
#include "phase.h"
#include "haplindow.h"
```

Include dependency graph for phase.cpp:



Variables

- ofstream [LOG](#)

7.148.1 Variable Documentation

7.148.1.1 LOG

ofstream LOG

Definition at line 81 of file inbix.cpp.

7.149 src/phase.h File Reference

This graph shows which files directly or indirectly include this file:

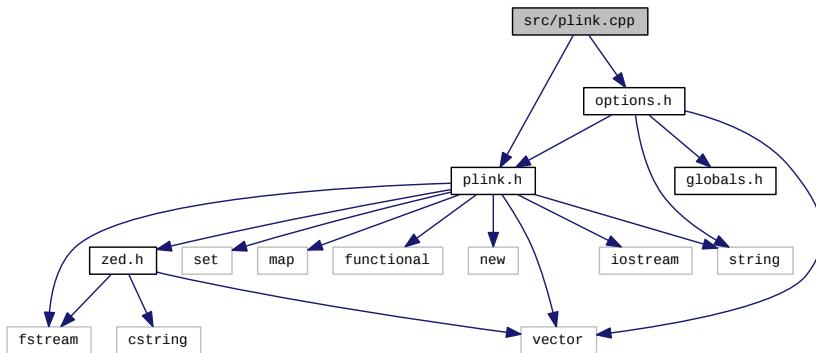


Classes

- class [FamilyTransmissions](#)
- class [HaploPhase](#)

7.150 src/plink.cpp File Reference

```
#include "plink.h"
#include "options.h"
Include dependency graph for plink.cpp:
```

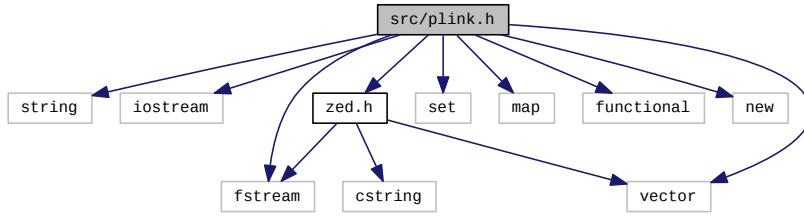


7.151 src/plink.h File Reference

```
#include <string>
#include <iostream>
#include <fstream>
#include <vector>
#include <set>
#include <map>
#include <functional>
#include <new>
```

```
#include "zed.h"
```

Include dependency graph for plink.h:



Classes

- class [RegressionRankResults](#)
- class [int2](#)
- class [double2](#)
- class [Pair2](#)
- class [indivPair](#)
- class [Range](#)
- class [WMLocus](#)
- class [Individual](#)
- class [CSNP](#)
- class [Cluster](#)
- class [Family](#)
- class [Locus](#)
- class [Transcript](#)
- class [std::less< Locus * >](#)
- class [MainExitException](#)
- class [Z](#)
- class [CInfo](#)
- class [Segment](#)
- class [ZZ](#)
- class [Plink](#)

Typedefs

- typedef `vector< vector< int > > table_t`
- typedef `vector< vector< double > > matrix_t`
- typedef `vector< double > vector_t`
- typedef `vector< bool > boolevec_t`
- typedef `vector< vector< bool > > boolmatrix_t`
- typedef `vector< int > intvec_t`
- typedef `vector< vector< double > > fmatrix_t`
- typedef `vector< float > floatvec_t`
- typedef `vector< Individual * >::iterator iIndividual`

- `typedef vector< Locus * >::iterator iLocus`
- `typedef vector< CSNP * >::iterator iSNP`
- `typedef vector< bool >::iterator iAllele`
- `typedef vector< pair< double, string > > rankedlist_t`
- `typedef pair< int, ModelTermType > interactionVar_t`
- `typedef pair< interactionVar_t, interactionVar_t > interaction_t`

Enumerations

- `enum ModelTermType {
INTERCEPT, ADDITIVE, DOMDEV, HAPLOTYPE,
SEX, COVARIATE, INTERACTION, TYPED_INTERACTION,
QFAM, NUMERIC }`
- `enum RegressionRankType { REGRESSION_RANK_STAT, REGRESSION_RANK_BETA, REGRESSION_RANK_PVAL }`

7.151.1 Typedef Documentation

7.151.1.1 boolmatrix_t

```
typedef vector<vector<bool> > boolmatrix_t
```

Definition at line 46 of file plink.h.

7.151.1.2 boolvec_t

```
typedef vector<bool> boolvec_t
```

Definition at line 45 of file plink.h.

7.151.1.3 floatvec_t

```
typedef vector<float> floatvec_t
```

Definition at line 49 of file plink.h.

7.151.1.4 fmatrix_t

```
typedef vector<vector<double>> fmatrix_t
```

Definition at line 48 of file plink.h.

7.151.1.5 iAllele

```
typedef vector<bool>::iterator iAllele
```

Definition at line 54 of file plink.h.

7.151.1.6 iIndividual

```
typedef vector<Individual*>::iterator iIndividual
```

Definition at line 51 of file plink.h.

7.151.1.7 iLocus

```
typedef vector<Locus*>::iterator iLocus
```

Definition at line 52 of file plink.h.

7.151.1.8 interaction_t

```
typedef pair<interactionVar_t, interactionVar_t> interaction_t
```

Definition at line 90 of file plink.h.

7.151.1.9 interactionVar_t

```
typedef pair<int, ModelTermType> interactionVar_t
```

Definition at line 89 of file plink.h.

7.151.1.10 intvec_t

```
typedef vector<int> intvec_t
```

Definition at line 47 of file plink.h.

7.151.1.11 iSNP

```
typedef vector<CSNP*>::iterator iSNP
```

Definition at line 53 of file plink.h.

7.151.1.12 matrix_t

```
typedef vector<vector<double>> matrix_t
```

Definition at line 43 of file plink.h.

7.151.1.13 rankedlist_t

```
typedef vector<pair<double, string>> rankedlist_t
```

Definition at line 56 of file plink.h.

7.151.1.14 table_t

```
typedef vector<vector<int>> table_t
```

Definition at line 42 of file plink.h.

7.151.1.15 vector_t

```
typedef vector<double> vector_t
```

Definition at line 44 of file plink.h.

7.151.2 Enumeration Type Documentation**7.151.2.1 ModelTermType**

```
enum ModelTermType
```

Enumerator

INTERCEPT
ADITIVE
DOMDEV
HAPLOTYPE
SEX
COVARIATE
INTERACTION
TYPED_INTERACTION
QFAM
NUMERIC

Definition at line 59 of file plink.h.

7.151.2.2 RegressionRankType

```
enum RegressionRankType
```

Enumerator

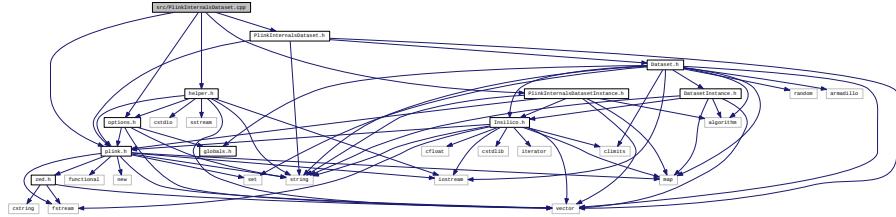
REGRESSION_RANK_STAT
REGRESSION_RANK_BETA
REGRESSION_RANK_PVAL

Definition at line 73 of file plink.h.

7.152 src/PlinkInternalsDataset.cpp File Reference

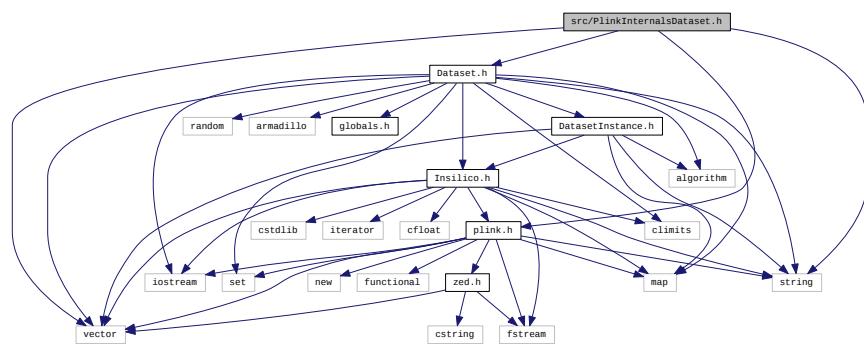
```
#include "plink.h"
#include "options.h"
#include "PlinkInternalsDataset.h"
#include "PlinkInternalsDatasetInstance.h"
#include "helper.h"
Include dependency graph for PlinkInternalsDataset.cpp:
```

Include dependency graph for PlinkInternalsDataset.cpp:

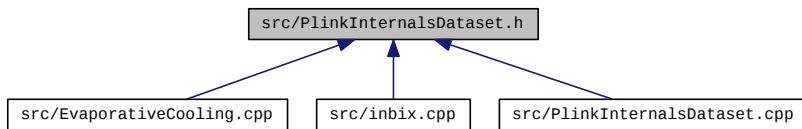


7.153 src/PlinkInternalsDataset.h File Reference

```
#include <string>
#include <vector>
#include "plink.h"
#include "Dataset.h"
Include dependency graph for PlinkInternalsDataset.h:
```



This graph shows which files directly or indirectly include this file:



Classes

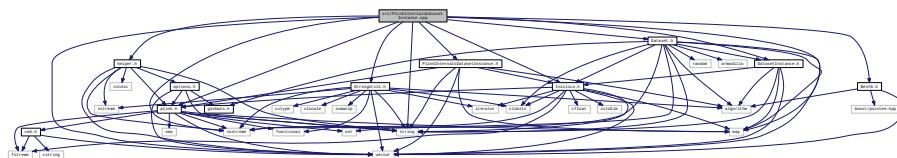
- class [PlinkInternalsDataset](#)
Plink internals adapter class.

7.154 src/PlinkInternalsDatasetInstance.cpp File Reference

```
#include <iostream>
#include <string>
#include <vector>
#include <map>
#include "plink.h"
#include "helper.h"
```

```
#include "Dataset.h"
#include "DatasetInstance.h"
#include "PlinkInternalsDatasetInstance.h"
#include "StringUtils.h"
#include "BestN.h"
#include "Insilico.h"

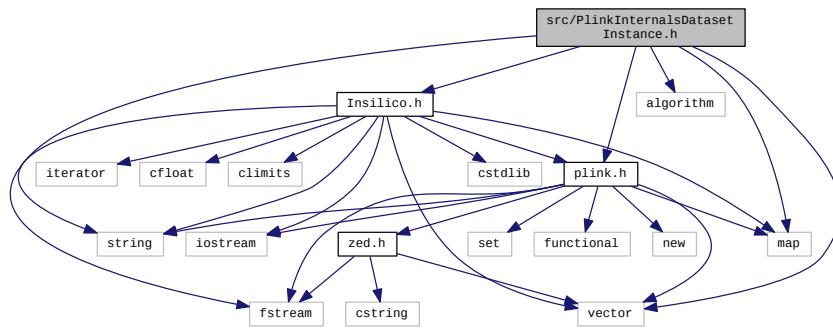
Include dependency graph for PlinkInternalsDatasetInstance.cpp:
```



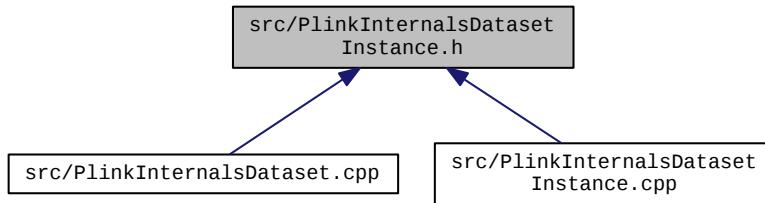
7.155 src/PlinkInternalsDatasetInstance.h File Reference

```
#include <string>
#include <vector>
#include <map>
#include <algorithm>
#include "plink.h"
#include "Insilico.h"

Include dependency graph for PlinkInternalsDatasetInstance.h:
```



This graph shows which files directly or indirectly include this file:



Classes

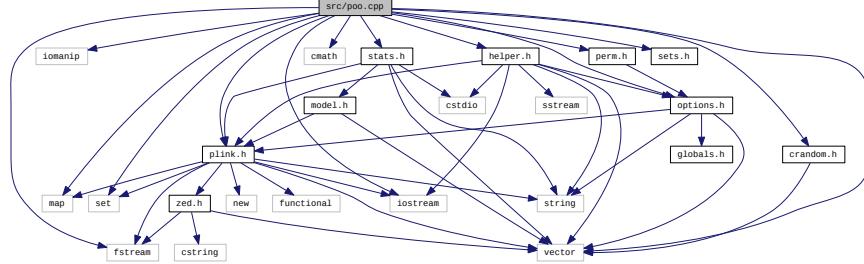
- class [PlinkInternalsDatasetInstance](#)

Class to hold dataset instances (rows of attributes).

7.156 src/poo.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <map>
#include <vector>
#include <set>
#include <cmath>
#include "plink.h"
#include "options.h"
#include "helper.h"
#include "crandom.h"
#include "sets.h"
#include "perm.h"
#include "stats.h"
```

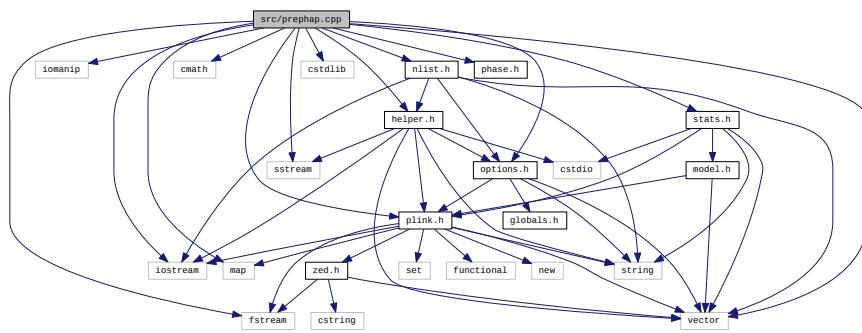
Include dependency graph for `poo.cpp`:



7.157 src/prephap.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <cmath>
#include <vector>
#include <map>
#include <cstdlib>
#include "plink.h"
#include "options.h"
#include "phase.h"
#include "helper.h"
#include "nlist.h"
#include "stats.h"
```

Include dependency graph for prephap.cpp:



Variables

- ofstream LOG

7.157.1 Variable Documentation

7.157.1.1 LOG

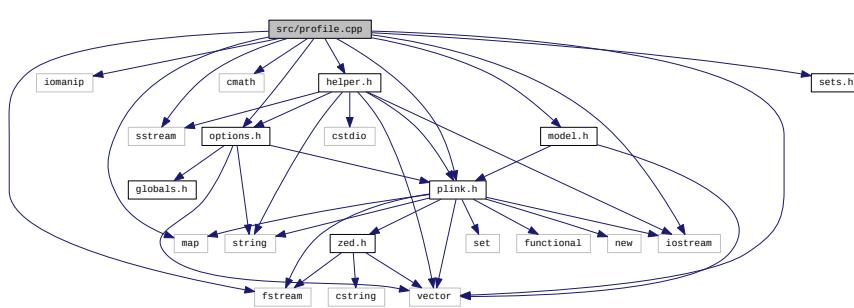
```
ofstream LOG
```

Definition at line 81 of file inbix.cpp.

7.158 src/profile.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <cmath>
#include <vector>
#include <map>
#include "plink.h"
#include "options.h"
#include "helper.h"
#include "model.h"
#include "sets.h"
```

Include dependency graph for profile.cpp:



Functions

- void `scoreRanges` (int, vector< int > &, map< int, set< Range *> > &, map< Range *, int > &, ofstream &)

Variables

- ofstream `LOG`
- Plink * `PP`

7.158.1 Function Documentation

7.158.1.1 scoreRanges()

```
void scoreRanges (
    int i,
    vector< int > & f,
    map< int, set< Range *> > & snp2range,
    map< Range *, int > & rangeCount,
    ofstream & ROUT )
```

Definition at line 651 of file profile.cpp.

7.158.2 Variable Documentation

7.158.2.1 LOG

ofstream LOG

Definition at line 81 of file inbix.cpp.

7.158.2.2 PP

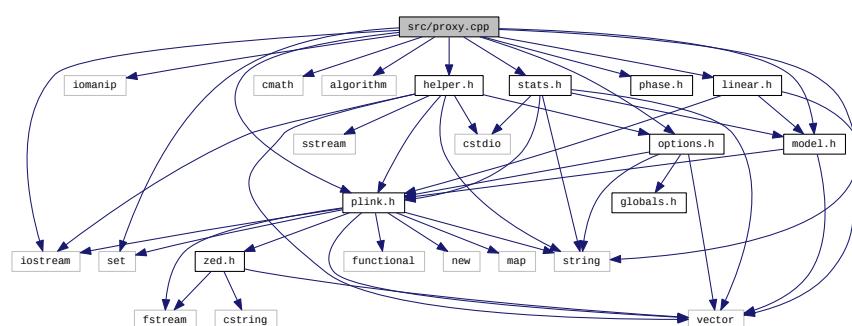
Plink* PP

Definition at line 85 of file inbix.cpp.

7.159 src/proxy.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <string>
#include <set>
#include <cmath>
#include <algorithm>
#include "options.h"
#include "helper.h"
#include "plink.h"
#include "phase.h"
#include "model.h"
#include "linear.h"
#include "stats.h"
```

Include dependency graph for proxy.cpp:



Classes

- class [ProxyResult](#)
- class [LDPair](#)

Functions

- void [combinations_recursive](#) (const vector< int > &elems, unsigned long req_len, vector< unsigned long > &pos, unsigned long depth, unsigned long margin, vector< vector< int > > &collection)

7.159.1 Function Documentation

7.159.1.1 combinations_recursive()

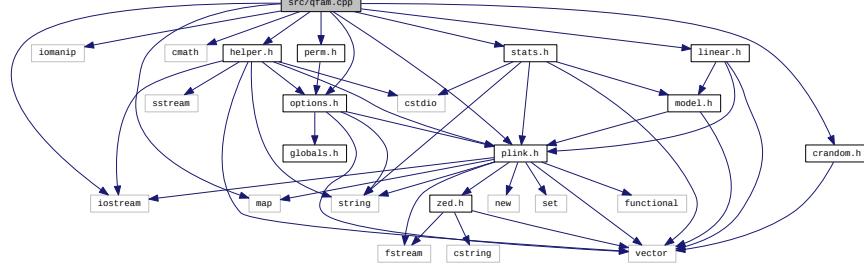
```
void combinations_recursive (
    const vector< int > & elems,
    unsigned long req_len,
    vector< unsigned long > & pos,
    unsigned long depth,
    unsigned long margin,
    vector< vector< int > > & collection )
```

Definition at line 71 of file proxy.cpp.

7.160 src/qfam.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <map>
#include <cmath>
#include "plink.h"
#include "helper.h"
#include "options.h"
#include "linear.h"
#include "perm.h"
#include "crandom.h"
#include "stats.h"
```

Include dependency graph for qfam.cpp:



Functions

- void `setCovariatesForSNP` (`Plink &P, int l`)
- void `scoreBetween` (`Plink &P, int l`)
- void `scoreBandW` (`Plink &P, int l, vector< bool > &include`)

7.160.1 Function Documentation

7.160.1.1 `scoreBandW()`

```
void scoreBandW (
    Plink & P,
    int l,
    vector< bool > & include )
```

Definition at line 175 of file qfam.cpp.

7.160.1.2 `scoreBetween()`

```
void scoreBetween (
    Plink & P,
    int l )
```

Definition at line 64 of file qfam.cpp.

7.160.1.3 `setCovariatesForSNP()`

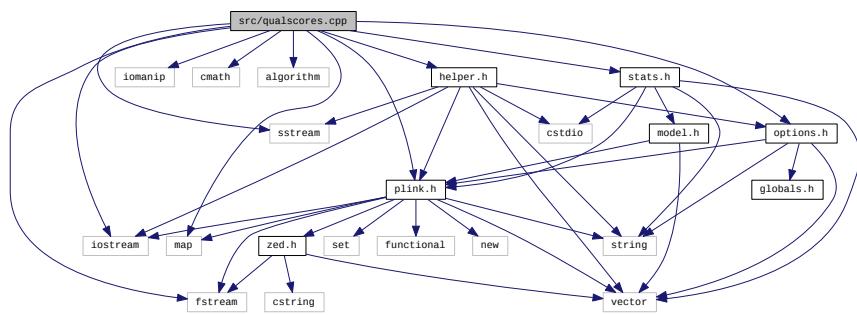
```
void setCovariatesForSNP (
    Plink & P,
    int l )
```

Definition at line 29 of file qfam.cpp.

7.161 src/qualscores.cpp File Reference

```
#include <iostream>
#include <fstream>
#include <sstream>
#include <iomanip>
#include <cmath>
#include <algorithm>
#include <map>
#include "plink.h"
#include "stats.h"
#include "helper.h"
#include "options.h"
```

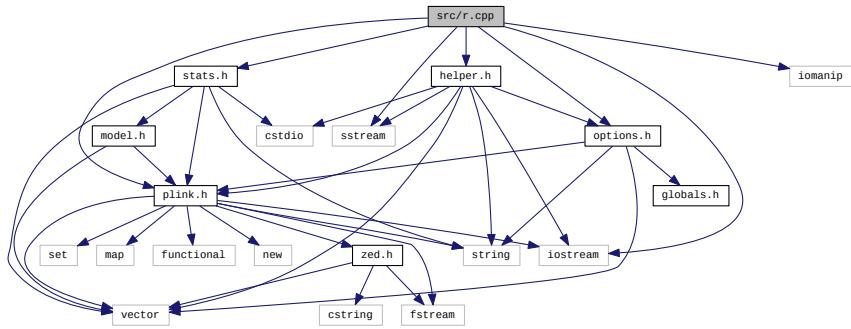
Include dependency graph for qualscores.cpp:



7.162 src/r.cpp File Reference

```
#include <iostream>
#include <sstream>
#include <iomanip>
#include "plink.h"
#include "helper.h"
#include "options.h"
#include "stats.h"
```

Include dependency graph for r.cpp:



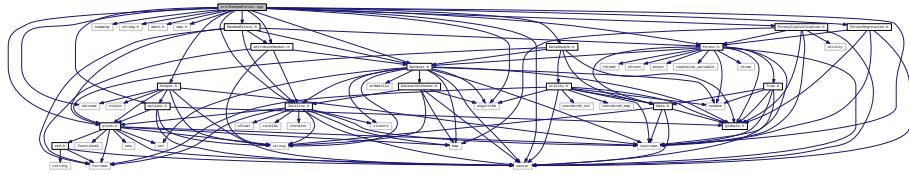
7.163 src/RandomForest.cpp File Reference

```

#include <iostream>
#include <algorithm>
#include <vector>
#include <iomanip>
#include <string.h>
#include <math.h>
#include <omp.h>
#include <sstream>
#include "Insilico.h"
#include "plink.h"
#include "options.h"
#include "helper.h"
#include "RandomForest.h"
#include "AttributeRanker.h"
#include "Dataset.h"
#include "DataDouble.h"
#include "Forest.h"
#include "ForestClassification.h"
#include "ForestRegression.h"

```

Include dependency graph for RandomForest.cpp:



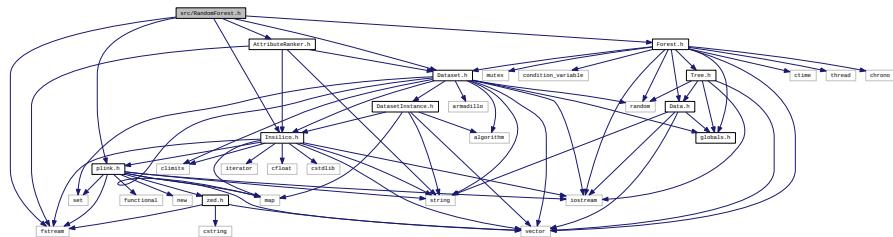
7.164 src/RandomForest.h File Reference

```

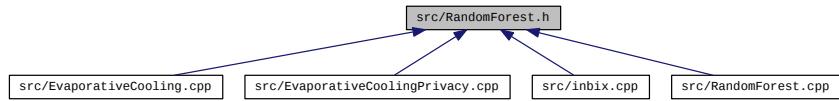
#include <fstream>
#include "Insilico.h"

```

```
#include "Dataset.h"
#include "AttributeRanker.h"
#include "plink.h"
#include "Forest.h"
Include dependency graph for RandomForest.h:
```



This graph shows which files directly or indirectly include this file:



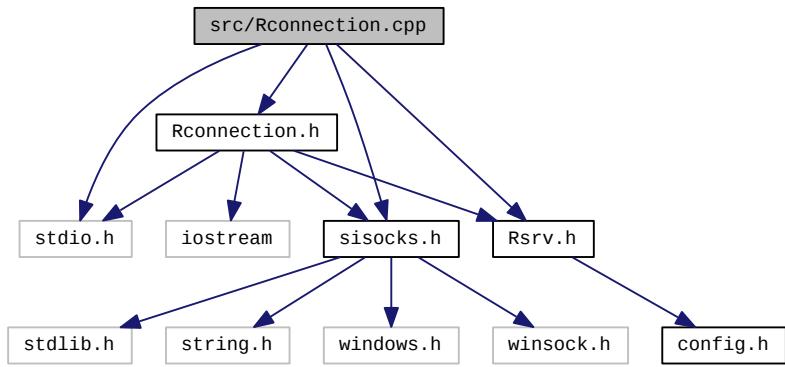
Classes

- class **RandomForest**
RandomForest attribute ranking algorithm.

7.165 src/Rconnection.cpp File Reference

```
#include "Rconnection.h"
#include <stdio.h>
#include "sisocks.h"
#include "Rsry.h"
```

Include dependency graph for Rconnection.cpp:



Macros

- `#define AF_LOCAL -1`

Functions

- static `Rexp * new_parsed_Rexp (unsigned int *d, Rmessage *msg)`
- static `Rexp * new_parsed_Rexp_from_Msg (Rmessage *msg)`

Variables

- static char `myID [16] = "Rsrv0103QAP1"`

7.165.1 Macro Definition Documentation

7.165.1.1 AF_LOCAL

```
#define AF_LOCAL -1
```

Definition at line 38 of file Rconnection.cpp.

7.165.2 Function Documentation

7.165.2.1 new_parsed_Rexp()

```
static Rexp* new_parsed_Rexp (
    unsigned int * d,
    Rmessage * msg ) [static]
```

Definition at line 58 of file Rconnection.cpp.

7.165.2.2 new_parsed_Rexp_from_Msg()

```
static Rexp* new_parsed_Rexp_from_Msg (
    Rmessage * msg ) [static]
```

Definition at line 80 of file Rconnection.cpp.

7.165.3 Variable Documentation

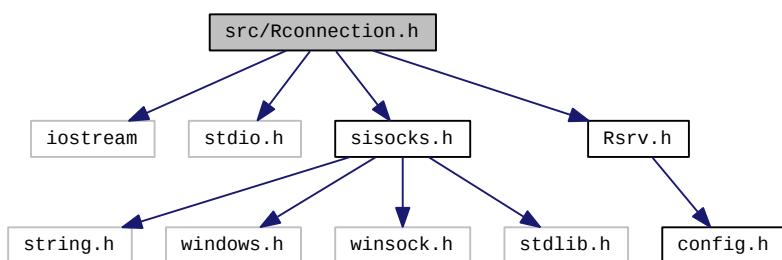
7.165.3.1 myID

```
char myID[16] = "Rsrv0103QAP1" [static]
```

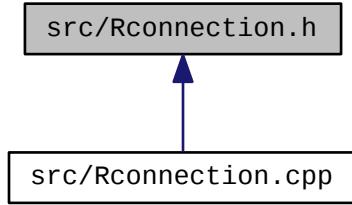
Definition at line 56 of file Rconnection.cpp.

7.166 src/Rconnection.h File Reference

```
#include <iostream>
#include <stdio.h>
#include "sisocks.h"
#include "Rsrv.h"
Include dependency graph for Rconnection.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [Rmessage](#)
- class [Rexp](#)
- class [Rinteger](#)
- class [Rdouble](#)
- class [Rsymbol](#)
- class [Rstrings](#)
- class [Rstring](#)
- class [Rlist](#)
- class [Rvector](#)
- class [Rconnection](#)

Macros

- #define [CERR_connect_failed](#) -1
- #define [CERR_handshake_failed](#) -2
- #define [CERR_invalid_id](#) -3
- #define [CERR_protocol_not_supp](#) -4
- #define [CERR_not_connected](#) -5
- #define [CERR_peer_closed](#) -7
- #define [CERR_malformed_packet](#) -8
- #define [CERR_send_error](#) -9
- #define [CERR_out_of_mem](#) -10
- #define [CERR_not_supported](#) -11
- #define [CERR_io_error](#) -12
- #define [CERR_auth_unsupported](#) -20
- #define [A_required](#) 0x001
- #define [A_crypt](#) 0x002
- #define [A_plain](#) 0x004

Typedefs

- `typedef unsigned int Rsize_t`

7.166.1 Macro Definition Documentation

7.166.1.1 A_crypt

```
#define A_crypt 0x002
```

Definition at line 66 of file Rconnection.h.

7.166.1.2 A_plain

```
#define A_plain 0x004
```

Definition at line 67 of file Rconnection.h.

7.166.1.3 A_required

```
#define A_required 0x001
```

Definition at line 65 of file Rconnection.h.

7.166.1.4 CERR_auth_unsupported

```
#define CERR_auth_unsupported -20
```

Definition at line 62 of file Rconnection.h.

7.166.1.5 CERR_connect_failed

```
#define CERR_connect_failed -1
```

Definition at line 48 of file Rconnection.h.

7.166.1.6 CERR_handshake_failed

```
#define CERR_handshake_failed -2
```

Definition at line 49 of file Rconnection.h.

7.166.1.7 CERR_invalid_id

```
#define CERR_invalid_id -3
```

Definition at line 50 of file Rconnection.h.

7.166.1.8 CERR_io_error

```
#define CERR_io_error -12
```

Definition at line 58 of file Rconnection.h.

7.166.1.9 CERR_malformed_packet

```
#define CERR_malformed_packet -8
```

Definition at line 54 of file Rconnection.h.

7.166.1.10 CERR_not_connected

```
#define CERR_not_connected -5
```

Definition at line 52 of file Rconnection.h.

7.166.1.11 CERR_not_supported

```
#define CERR_not_supported -11
```

Definition at line 57 of file Rconnection.h.

7.166.1.12 CERR_out_of_mem

```
#define CERR_out_of_mem -10
```

Definition at line 56 of file Rconnection.h.

7.166.1.13 CERR_peer_closed

```
#define CERR_peer_closed -7
```

Definition at line 53 of file Rconnection.h.

7.166.1.14 CERR_protocol_not_supp

```
#define CERR_protocol_not_supp -4
```

Definition at line 51 of file Rconnection.h.

7.166.1.15 CERR_send_error

```
#define CERR_send_error -9
```

Definition at line 55 of file Rconnection.h.

7.166.2 Typedef Documentation

7.166.2.1 Rsize_t

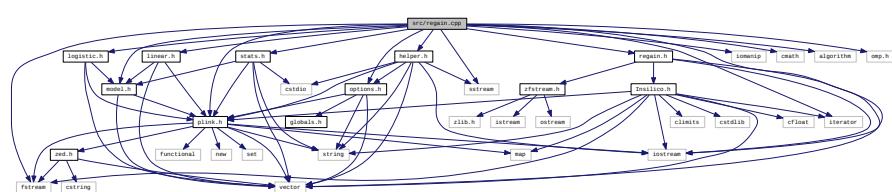
```
typedef unsigned int Rsize_t
```

Definition at line 44 of file Rconnection.h.

7.167 src/regain.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <cmath>
#include <algorithm>
#include <iterator>
#include <sstream>
#include <omp.h>
#include "plink.h"
#include "options.h"
#include "model.h"
#include "logistic.h"
#include "linear.h"
#include "stats.h"
#include "helper.h"
#include "regain.h"
```

Include dependency graph for regain.cpp:



Functions

- bool `pvalComparator` (const `matrixElement` &l, const `matrixElement` &r)

Variables

- `Plink * PP`

7.167.1 Function Documentation

7.167.1.1 pvalComparator()

```
bool pvalComparator (
    const matrixElement & l,
    const matrixElement & r )
```

Definition at line 35 of file regain.cpp.

7.167.2 Variable Documentation

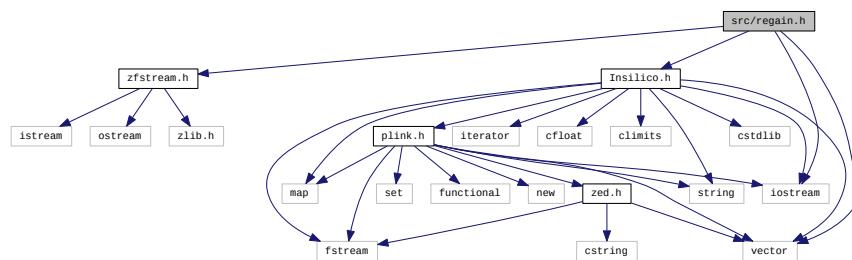
7.167.2.1 PP

```
Plink* PP
```

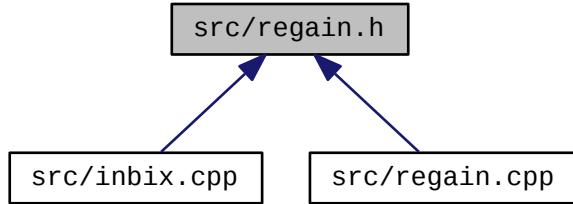
Definition at line 85 of file inbix.cpp.

7.168 src/regain.h File Reference

```
#include <iostream>
#include <vector>
#include "zfstream.h"
#include "Insilico.h"
Include dependency graph for regain.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [Regain](#)

Enumerations

- enum [RegainOutputFormat](#) { REGAIN_OUTPUT_FORMAT_UPPER, REGAIN_OUTPUT_FORMAT_FULL }
- enum [RegainOutputTransform](#) { REGAIN_OUTPUT_TRANSFORM_NONE, REGAIN_OUTPUT_TRANSFORM_M_ABS, REGAIN_OUTPUT_TRANSFORM_THRESH }

7.168.1 Enumeration Type Documentation

7.168.1.1 RegainOutputFormat

```
enum RegainOutputFormat
```

Enumerator

REGAIN_OUTPUT_FORMAT_UPPER	
REGAIN_OUTPUT_FORMAT_FULL	

Definition at line 27 of file regain.h.

7.168.1.2 RegainOutputTransform

enum RegainOutputTransform

Enumerator

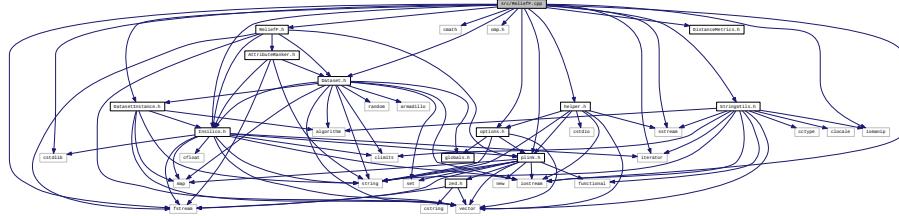
REGAIN_OUTPUT_TRANSFORM_NONE
REGAIN_OUTPUT_TRANSFORM_ABS
REGAIN_OUTPUT_TRANSFORM_THRESH

Definition at line 32 of file regain.h.

7.169 src/ReliefF.cpp File Reference

```
#include <cstdlib>
#include <iostream>
#include <fstream>
#include <iomanip>
#include <iterator>
#include <cmath>
#include <sstream>
#include <omp.h>
#include "ReliefF.h"
#include "Dataset.h"
#include "DatasetInstance.h"
#include "StringUtils.h"
#include "DistanceMetrics.h"
#include "Insilico.h"
#include "plink.h"
#include "options.h"
#include "helper.h"
```

Include dependency graph for ReliefF.cpp:



Classes

- class deref less

Typedefs

- `typedef vector< pair< double, unsigned int > > ScoresMap`
scores map: score->attribute index
- `typedef vector< pair< double, unsigned int > >::iterator ScoresMapIt`
scores map iterator
- `typedef vector< pair< unsigned int, double > > AttributeIndex`
attribute index map: attribute index->score
- `typedef vector< pair< unsigned int, double > >::const_iterator AttributeIndexIt`
attribute index map iterator
- `typedef pair< unsigned int, DatasetInstance * > T`
functor for T comparison - instance-to-instance distance

Functions

- `bool scoreSort (const ScoreVarPair &p1, const ScoreVarPair &p2)`
attribute score sorting functor
- `bool attributeSort (const pair< unsigned int, double > &p1, const pair< unsigned int, double > &p2)`
attribute index sorting functor

7.169.1 Typedef Documentation

7.169.1.1 AttributeIndex

```
typedef vector<pair<unsigned int, double> > AttributeIndex
```

attribute index map: attribute index->score

Definition at line 40 of file ReliefF.cpp.

7.169.1.2 AttributeIndexIt

```
typedef vector<pair<unsigned int, double> >::const_iterator AttributeIndexIt
```

attribute index map iterator

Definition at line 42 of file ReliefF.cpp.

7.169.1.3 ScoresMap

```
typedef vector<pair<double, unsigned int> > ScoresMap
```

scores map: score->attribute index

Definition at line 36 of file ReliefF.cpp.

7.169.1.4 ScoresMapIt

```
typedef vector<pair<double, unsigned int> >::iterator ScoresMapIt
```

scores map iterator

Definition at line 38 of file ReliefF.cpp.

7.169.1.5 T

```
typedef pair<unsigned int, DatasetInstance*> T
```

functor for T comparison - instance-to-instance distance

Definition at line 56 of file ReliefF.cpp.

7.169.2 Function Documentation

7.169.2.1 attributeSort()

```
bool attributeSort (
    const pair< unsigned int, double > & p1,
    const pair< unsigned int, double > & p2 )
```

attribute index sorting functor

Definition at line 50 of file ReliefF.cpp.

7.169.2.2 scoreSort()

```
bool scoreSort (
    const ScoreVarPair & p1,
    const ScoreVarPair & p2 )
```

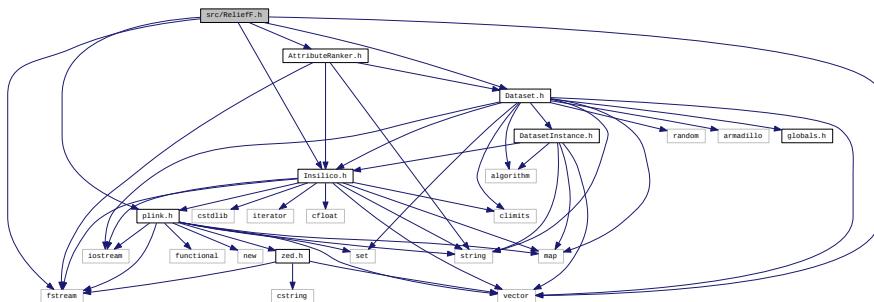
attribute score sorting functor

Definition at line 45 of file ReliefF.cpp.

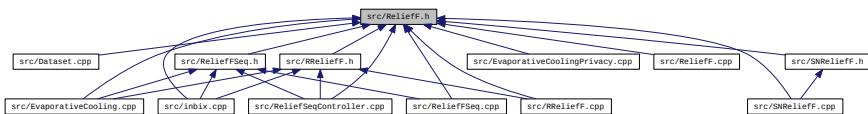
7.170 src/ReliefF.h File Reference

```
#include <vector>
#include <fstream>
#include "plink.h"
#include "AttributeRanker.h"
#include "Dataset.h"
#include "Insilico.h"
```

Include dependency graph for ReliefF.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [ReliefF](#)

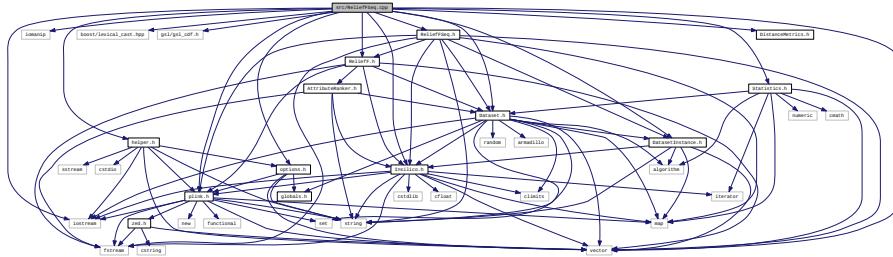
ReliefF attribute ranking algorithm.

7.171 src/ReliefFSeq.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <vector>
#include <boost/lexical_cast.hpp>
#include <gsl/gsl_cdf.h>
#include "plink.h"
#include "helper.h"
#include "ReliefF.h"
#include "ReliefFSeq.h"
#include "Dataset.h"
#include "DatasetInstance.h"
#include "DistanceMetrics.h"
#include "Insilico.h"
#include "Statistics.h"
#include "options.h"
```

Include dependency graph for ReliefFSeq.cpp:

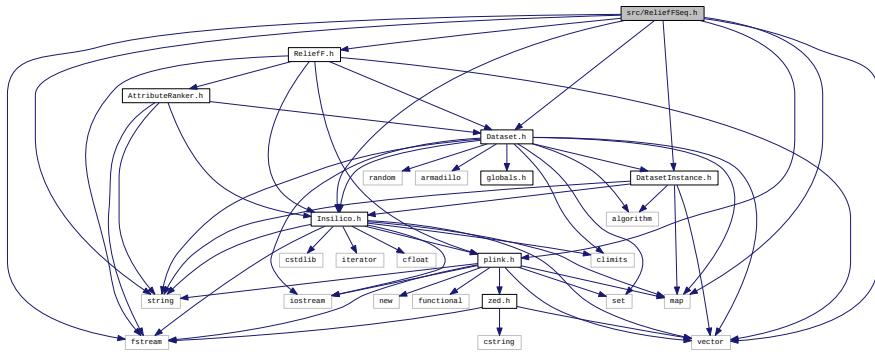
Include dependency graph for ReliefFSeq.cpp:



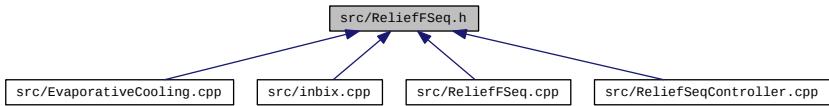
7.172 src/ReliefFSeq.h File Reference

```
#include <vector>
#include <map>
#include <string>
#include <fstream>
#include "plink.h"
#include "ReliefF.h"
#include "Dataset.h"
#include "DatasetInstance.h"
#include "Insilico.h"
```

Include dependency graph for ReliefFSeq.h:



This graph shows which files directly or indirectly include this file:



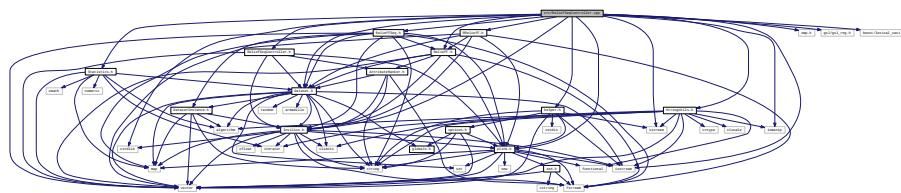
Classes

- class [ReliefFSeq](#)
ReliefFSeq attribute ranking algorithm.

7.173 src/ReliefSeqController.cpp File Reference

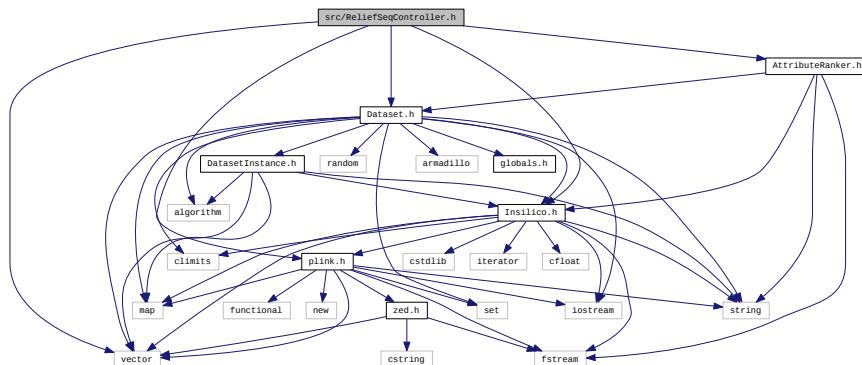
```
#include <cstdlib>
#include <iostream>
#include <iomanip>
#include <sstream>
#include <omp.h>
#include <gsl/gsl_rng.h>
#include <boost/lexical_cast.hpp>
#include "plink.h"
#include "helper.h"
#include "ReliefSeqController.h"
#include "Insilico.h"
#include "Dataset.h"
#include "Statistics.h"
#include "StringUtils.h"
#include "ReliefF.h"
```

```
#include "RReliefF.h"
#include "ReliefFSeq.h"
Include dependency graph for ReliefSeqController.cpp:
```

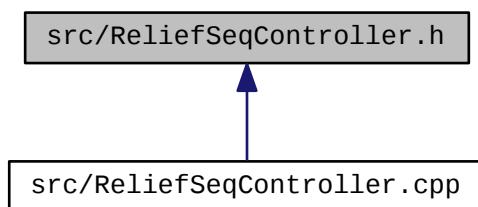


7.174 src/ReliefSeqController.h File Reference

```
#include <vector>
#include "plink.h"
#include "AttributeRanker.h"
#include "Dataset.h"
#include "Insilico.h"
Include dependency graph for ReliefSeqController.h:
```



This graph shows which files directly or indirectly include this file:



Classes

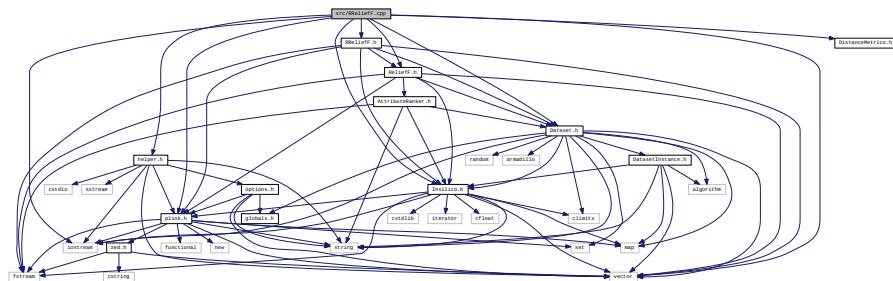
- class [ReliefSeqController](#)

Interface for running ReliefSeq algorithms.

7.175 src/RReliefF.cpp File Reference

```
#include <iostream>
#include <vector>
#include "plink.h"
#include "helper.h"
#include "ReliefF.h"
#include "RReliefF.h"
#include "Dataset.h"
#include "DistanceMetrics.h"
#include "Insilico.h"
```

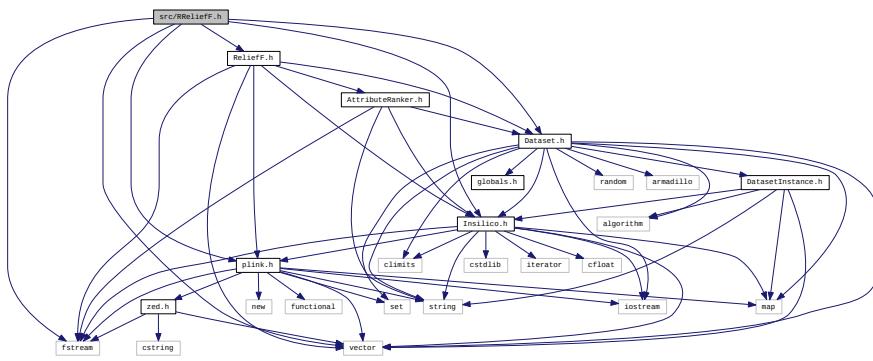
Include dependency graph for RReliefF.cpp:



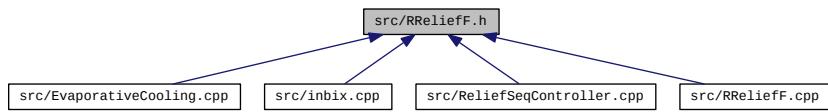
7.176 src/RReliefF.h File Reference

```
#include <vector>
#include <fstream>
#include "plink.h"
#include "ReliefF.h"
#include "Dataset.h"
#include "Insilico.h"
```

Include dependency graph for RReliefF.h:



This graph shows which files directly or indirectly include this file:

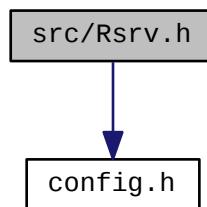


Classes

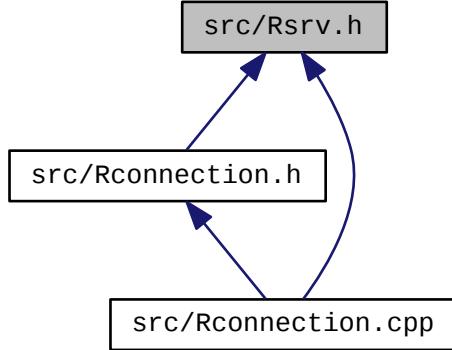
- class [RReliefF](#)
Regression ReliefF attribute ranking algorithm.

7.177 src/Rsrv.h File Reference

```
#include "config.h"
Include dependency graph for Rsrv.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- struct [phdr](#)

Macros

- `#define RSRV_VER 0x000503 /* Rserve v0.5-3 */`
- `#define default_Rsrv_port 6311`
- `#define PAR_TYPE(X) ((X)&255)`
- `#define PAR_LEN(X) ((X)>>8)`
- `#define PAR_LENGTH PAR_LEN`
- `#define SET_PAR(TY, LEN) (((LEN)&0xfffff)<<8)|((TY)&255))`
- `#define CMD_STAT(X) (((X)>>24)&127) /* returns the stat code of the response */`
- `#define SET_STAT(X, s) ((X)|((s)&127)<<24)) /* sets the stat code */`
- `#define CMD_RESP 0x10000 /* all responses have this flag set */`
- `#define RESP_OK`
- `#define RESP_ERR`
- `#define ERR_auth_failed`
- `#define ERR_conn_broken 0x42 /* connection closed or broken packet killed it */`
- `#define ERR_inv_cmd 0x43 /* unsupported/invalid command */`
- `#define ERR_inv_par 0x44 /* some parameters are invalid */`
- `#define ERR_Rerror`
- `#define ERR_IError 0x46 /* I/O error */`
- `#define ERR_notOpen`
- `#define ERR_accessDenied`
- `#define ERR_unsupportedCmd 0x49 /* unsupported command */`
- `#define ERR_unknownCmd`
- `#define ERR_data_overflow`

- #define **ERR_object_too_big**
- #define **ERR_out_of_mem**
- #define **ERR_session_busy** 0x50 /* session is still busy */
- #define **ERR_detach_failed**
- #define **CMD_login** 0x001 /* "name\npwd" : - */
- #define **CMD_voidEval** 0x002 /* string : - */
- #define **CMD_eval** 0x003 /* string : encoded SEXP */
- #define **CMD_shutdown** 0x004 /* [admin-pwd] : - */
- #define **CMD_openFile** 0x010 /* fn : - */
- #define **CMD_createFile** 0x011 /* fn : - */
- #define **CMD_closeFile** 0x012 /* - : - */
- #define **CMD_readFile**
- #define **CMD_writeFile** 0x014 /* data : - */
- #define **CMD_removeFile** 0x015 /* fn : - */
- #define **CMD_setSEXP** 0x020 /* string(name), REXP : - */
- #define **CMD_assignSEXP**
- #define **CMD_detachSession** 0x030 /* : session key */
- #define **CMD_detachedVoidEval** 0x031 /* string : session key; doesn't */
- #define **CMD_attachSession** 0x032 /* session key : - */
- #define **CMD_setBufferSize**
- #define **CMD_setEncoding** 0x082 /* string (one of "native","latin1","utf8") : -; since 0.5-3 */
- #define **CMD_SPECIAL_MASK** 0xf0
- #define **CMD_serEval** 0xf5 /* serialized eval - the packets are raw serialized data without data header */
- #define **CMD_serAssign** 0xf6 /* serialized assign - serialized list with [[1]]=name, [[2]]=value */
- #define **CMD_serEEval** 0xf7 /* serialized expression eval - like serEval with one additional evaluation round */
- #define **DT_INT** 1 /* int */
- #define **DT_CHAR** 2 /* char */
- #define **DT_DOUBLE** 3 /* double */
- #define **DT_STRING** 4 /* 0 terminted string */
- #define **DT_BYTESTREAM** 5 /* stream of bytes (unlike **DT_STRING** may contain 0) */
- #define **DT_SEXP** 10 /* encoded SEXP */
- #define **DT_ARRAY**
- #define **DT_LARGE**
- #define **XT_NULL** 0 /* P data: [0] */
- #define **XT_INT** 1 /* - data: [4]int */
- #define **XT_DOUBLE** 2 /* - data: [8]double */
- #define **XT_STR** 3 /* P data: [n]char null-term. strg. */
- #define **XT_LANG** 4 /* - data: same as **XT_LIST** */
- #define **XT_SYM** 5 /* - data: [n]char symbol name */
- #define **XT_BOOL**
- #define **XT_S4** 7 /* P data: [0] */
- #define **XT_VECTOR** 16 /* P data: [?]REXP,REXP,... */
- #define **XT_LIST** 17 /* - X head, X vals, X tag (since 0.1-5) */
- #define **XT_CLOS** 18 /* P X formals, X body (closure; since 0.1-5) */
- #define **XT_SYMNAME** 19 /* s same as **XT_STR** (since 0.5) */
- #define **XT_LIST_NOTAG** 20 /* s same as **XT_VECTOR** (since 0.5) */
- #define **XT_LIST_TAG** 21 /* P X tag, X val, Y tag, Y val, ... (since 0.5) */
- #define **XT_LANG_NOTAG** 22 /* s same as **XT_LIST_NOTAG** (since 0.5) */
- #define **XT_LANG_TAG** 23 /* s same as **XT_LIST_TAG** (since 0.5) */
- #define **XT_VECTOR_EXP** 26 /* s same as **XT_VECTOR** (since 0.5) */

- #define XT_VECTOR_STR 27 /* - same as XT_VECTOR (since 0.5 but unused, use XT_ARRAY_STR instead) */
- #define XT_ARRAY_INT 32 /* P data: [n*4]int,int,... */
- #define XT_ARRAY_DOUBLE 33 /* P data: [n*8]double,double,... */
- #define XT_ARRAY_STR 34 /* P data: string,string,... (string=byte,byte,...,0) padded with '01' */
- #define XT_ARRAY_BOOL_UA 35 /* - data: [n]byte,byte,... (unaligned! NOT supported anymore) */
- #define XT_ARRAY_BOOL 36 /* P data: int(n),byte,byte,... */
- #define XT_RAW 37 /* P data: int(n).byte,byte,... */
- #define XT_ARRAY_CPLX 38 /* P data: [n*16]double,double,... (Re,Im,Re,Im,...) */
- #define XT_UNKNOWN 48 /* P data: [4]int - SEXP type (as from TYPEOF(x)) */
- #define XT_LARGE
- #define XT_HAS_ATTR
- #define BOOL_TRUE 1
- #define BOOL_FALSE 0
- #define BOOL_NA 2
- #define GET_XT(X) ((X)&63)
- #define GET_DT(X) ((X)&63)
- #define HAS_ATTR(X) (((X)&XT_HAS_ATTR)>0)
- #define IS_LARGE(X) (((X)&XT_LARGE)>0)
- #define itop(X) (X)
- #define ptoi(X) (X)
- #define dtop(X) (X)
- #define ptod(X) (X)
- #define fixdcpy(T, S) ((double*)(T))[0]=((double*)(S))[0];

Functions

- int isByteSexOk ()

7.177.1 Macro Definition Documentation

7.177.1.1 BOOL_FALSE

```
#define BOOL_FALSE 0
```

Definition at line 286 of file Rsrv.h.

7.177.1.2 BOOL_NA

```
#define BOOL_NA 2
```

Definition at line 287 of file Rsrv.h.

7.177.1.3 BOOL_TRUE

```
#define BOOL_TRUE 1
```

Definition at line 285 of file Rsrv.h.

7.177.1.4 CMD_assign SEXP

```
#define CMD_assign SEXP
```

Value:

```
0x021 /* string(name), REXP : - ; same as set SEXP  
except that the name is parsed */
```

Definition at line 191 of file Rsrv.h.

7.177.1.5 CMD_attachSession

```
#define CMD_attachSession 0x032 /* session key : - - */
```

Definition at line 197 of file Rsrv.h.

7.177.1.6 CMD_closeFile

```
#define CMD_closeFile 0x012 /* - - : - - */
```

Definition at line 182 of file Rsrv.h.

7.177.1.7 CMD_createFile

```
#define CMD_createFile 0x011 /* fn : - - */
```

Definition at line 181 of file Rsrv.h.

7.177.1.8 CMD_detachVoidEval

```
#define CMD_detachVoidEval 0x031 /* string : session key; doesn't */
```

Definition at line 196 of file Rsrv.h.

7.177.1.9 CMD_detachSession

```
#define CMD_detachSession 0x030 /* : session key */
```

Definition at line 195 of file Rsrv.h.

7.177.1.10 CMD_eval

```
#define CMD_eval 0x003 /* string : encoded SEXP */
```

Definition at line 176 of file Rsrv.h.

7.177.1.11 CMD_login

```
#define CMD_login 0x001 /* "name\pwd" : - */
```

Definition at line 174 of file Rsrv.h.

7.177.1.12 CMD_openFile

```
#define CMD_openFile 0x010 /* fn : - */
```

Definition at line 180 of file Rsrv.h.

7.177.1.13 CMD_readFile

```
#define CMD_readFile
```

Value:

```
0x013 /* [int size] : data... ; if size not present,  
server is free to choose any value - usually  
it uses the size of its static buffer */
```

Definition at line 183 of file Rsrv.h.

7.177.1.14 CMD_removeFile

```
#define CMD_removeFile 0x015 /* fn : - */
```

Definition at line 187 of file Rsrv.h.

7.177.1.15 CMD_RESP

```
#define CMD_RESP 0x10000 /* all responses have this flag set */
```

Definition at line 112 of file Rsrv.h.

7.177.1.16 CMD_serAssign

```
#define CMD_serAssign 0xf6 /* serialized assign - serialized list with [[1]]=name, [[2]]=value */
```

Definition at line 213 of file Rsrv.h.

7.177.1.17 CMD_serEEval

```
#define CMD_serEEval 0xf7 /* serialized expression eval - like serEval with one additional evaluation  
round */
```

Definition at line 214 of file Rsrv.h.

7.177.1.18 CMD_serEval

```
#define CMD_serEval 0xf5 /* serialized eval - the packets are raw serialized data without data header */
```

Definition at line 212 of file Rsrv.h.

7.177.1.19 CMD_setBufferSize

```
#define CMD_setBufferSize
```

Value:

```
0x081 /* [int sendBufSize]
          this command allow clients to request
          bigger buffer sizes if large data is to be
          transported from Rserve to the client.
          (incoming buffer is resized automatically)
        */
```

Definition at line 200 of file Rsrv.h.

7.177.1.20 CMD_setEncoding

```
#define CMD_setEncoding 0x082 /* string (one of "native","latin1","utf8") : -; since 0.5-3 */
```

Definition at line 206 of file Rsrv.h.

7.177.1.21 CMD_setSEXP

```
#define CMD_setSEXP 0x020 /* string(name), REXP : - */
```

Definition at line 190 of file Rsrv.h.

7.177.1.22 CMD_shutdown

```
#define CMD_shutdown 0x004 /* [admin-pwd] : - */
```

Definition at line 177 of file Rsrv.h.

7.177.1.23 CMD_SPECIAL_MASK

```
#define CMD_SPECIAL_MASK 0xf0
```

Definition at line 210 of file Rsrv.h.

7.177.1.24 CMD_STAT

```
#define CMD_STAT( X ) (((X)>>24)&127) /* returns the stat code of the response */
```

Definition at line 109 of file Rsrv.h.

7.177.1.25 CMD_voidEval

```
#define CMD_voidEval 0x002 /* string : - */
```

Definition at line 175 of file Rsrv.h.

7.177.1.26 CMD_writeFile

```
#define CMD_writeFile 0x014 /* data : - */
```

Definition at line 186 of file Rsrv.h.

7.177.1.27 default_Rsrv_port

```
#define default_Rsrv_port 6311
```

Definition at line 37 of file Rsrv.h.

7.177.1.28 DT_ARRAY

```
#define DT_ARRAY
```

Value:

```
11 /* array of objects (i.e. first 4 bytes specify how many  
   subsequent objects are part of the array; 0 is legitimate) */
```

Definition at line 225 of file Rsrv.h.

7.177.1.29 DT_BYTESTREAM

```
#define DT_BYTESTREAM 5 /* stream of bytes (unlike DT_STRING may contain 0) */
```

Definition at line 223 of file Rsrv.h.

7.177.1.30 DT_CHAR

```
#define DT_CHAR 2 /* char */
```

Definition at line 220 of file Rsrv.h.

7.177.1.31 DT_DOUBLE

```
#define DT_DOUBLE 3 /* double */
```

Definition at line 221 of file Rsrv.h.

7.177.1.32 DT_INT

```
#define DT_INT 1 /* int */
```

Definition at line 219 of file Rsrv.h.

7.177.1.33 DT_LARGE

```
#define DT_LARGE
```

Value:

```
64 /* new in 0102: if this flag is set then the length of the object  
is coded as 56-bit integer enlarging the header by 4 bytes */
```

Definition at line 227 of file Rsrv.h.

7.177.1.34 DT_SEXP

```
#define DT_SEXP 10 /* encoded SEXP */
```

Definition at line 224 of file Rsrv.h.

7.177.1.35 DT_STRING

```
#define DT_STRING 4 /* 0 terminted string */
```

Definition at line 222 of file Rsrv.h.

7.177.1.36 dtop

```
#define dtop(  
    X ) (X)
```

Definition at line 344 of file Rsrv.h.

7.177.1.37 ERR_accessDenied

```
#define ERR_accessDenied
```

Value:

```
0x48 /* this answer is also valid on  
CMD_login; otherwise it's sent  
if the server deosn;t allow the user  
to issue the specified command.  
(e.g. some server admins may block  
file I/O operations for some users) */
```

Definition at line 138 of file Rsrv.h.

7.177.1.38 ERR_auth_failed

```
#define ERR_auth_failed
```

Value:

```
0x41 /* auth.failed or auth.requested but no
       login came. in case of authentication
       failure due to name/pwd mismatch,
       server may send CMD_accessDenied instead
     */
```

Definition at line 125 of file Rsrv.h.

7.177.1.39 ERR_conn_broken

```
#define ERR_conn_broken 0x42 /* connection closed or broken packet killed it */
```

Definition at line 130 of file Rsrv.h.

7.177.1.40 ERR_data_overflow

```
#define ERR_data_overflow
```

Value:

```
0x4b /* incoming packet is too big.
       currently there is a limit as of the
       size of an incoming packet. */
```

Definition at line 153 of file Rsrv.h.

7.177.1.41 ERR_detach_failed

```
#define ERR_detach_failed
```

Value:

```
0x51 /* unable to detach session (cannot determine
       peer IP or problems creating a listening
       socket for resume) */
```

Definition at line 168 of file Rsrv.h.

7.177.1.42 ERR_inv_cmd

```
#define ERR_inv_cmd 0x43 /* unsupported/invalid command */
```

Definition at line 131 of file Rsrv.h.

7.177.1.43 ERR_inv_par

```
#define ERR_inv_par 0x44 /* some parameters are invalid */
```

Definition at line 132 of file Rsrv.h.

7.177.1.44 ERR_IOerror

```
#define ERR_IOerror 0x46 /* I/O error */
```

Definition at line 135 of file Rsrv.h.

7.177.1.45 ERR_notOpen

```
#define ERR_notOpen
```

Value:

```
0x47 /* attempt to perform fileRead/Write  
on closed file */
```

Definition at line 136 of file Rsrv.h.

7.177.1.46 ERR_object_too_big

```
#define ERR_object_too_big
```

Value:

```
0x4c /* the requested object is too big  
to be transported in that way.  
If received after CMD_eval then  
the evaluation itself was successful.  
optional parameter is the size of the object  
*/
```

Definition at line 156 of file Rsrv.h.

7.177.1.47 ERR_out_of_mem

```
#define ERR_out_of_mem
```

Value:

```
0x4d /* out of memory. the connection is usually  
closed after this error was sent */
```

Definition at line 163 of file Rsrv.h.

7.177.1.48 ERR_Rerror

```
#define ERR_Rerror
```

Value:

```
0x45 /* R-error occured, usually followed by  
connection shutdown */
```

Definition at line 133 of file Rsrv.h.

7.177.1.49 ERR_session_busy

```
#define ERR_session_busy 0x50 /* session is still busy */
```

Definition at line 167 of file Rsrv.h.

7.177.1.50 ERR_unknownCmd

```
#define ERR_unknownCmd
```

Value:

```
0x4a /* unknown command - the difference  
between unsupported and unknown is that  
unsupported commands are known to the  
server but for some reasons (e.g.  
platform dependent) it's not supported.  
unknown commands are simply not recognized  
by the server at all. */
```

Definition at line 145 of file Rsrv.h.

7.177.1.51 ERR_unsupportedCmd

```
#define ERR_unsupportedCmd 0x49 /* unsupported command */
```

Definition at line 144 of file Rsrv.h.

7.177.1.52 fixdcpy

```
#define fixdcpy(  
    T,  
    S) ((double*) (T)) [0]=((double*) (S)) [0];
```

Definition at line 346 of file Rsrv.h.

7.177.1.53 GET_DT

```
#define GET_DT(  
    X) ((X) &63)
```

Definition at line 290 of file Rsrv.h.

7.177.1.54 GET_XT

```
#define GET_XT(  
    X) ((X) &63)
```

Definition at line 289 of file Rsrv.h.

7.177.1.55 HAS_ATTR

```
#define HAS_ATTR(  
    X) (((X) &XT\_HAS\_ATTR)>0)
```

Definition at line 291 of file Rsrv.h.

7.177.1.56 IS_LARGE

```
#define IS_LARGE(  
    X ) (( (X) &XT_LARGE) >0)
```

Definition at line 292 of file Rsrv.h.

7.177.1.57 itop

```
#define itop(  
    X ) (X)
```

Definition at line 342 of file Rsrv.h.

7.177.1.58 PAR_LEN

```
#define PAR_LEN(  
    X ) ((X) >>8)
```

Definition at line 105 of file Rsrv.h.

7.177.1.59 PAR_LENGTH

```
#define PAR_LENGTH PAR_LEN
```

Definition at line 106 of file Rsrv.h.

7.177.1.60 PAR_TYPE

```
#define PAR_TYPE(  
    X ) ((X) &255)
```

Definition at line 104 of file Rsrv.h.

7.177.1.61 ptod

```
#define ptod(  
    X ) (X)
```

Definition at line 345 of file Rsrv.h.

7.177.1.62 ptoi

```
#define ptoi(  
    X ) (X)
```

Definition at line 343 of file Rsrv.h.

7.177.1.63 RESP_ERR

```
#define RESP_ERR
```

Value:

```
(CMD_RESP|0x0002) /* command failed, check stats code  
attached string may describe the error */
```

Definition at line 116 of file Rsrv.h.

7.177.1.64 RESP_OK

```
#define RESP_OK
```

Value:

```
(CMD_RESP|0x0001) /* command succeeded; returned parameters depend  
on the command issued */
```

Definition at line 114 of file Rsrv.h.

7.177.1.65 RSRV_VER

```
#define RSRV_VER 0x000503 /* Rserve v0.5-3 */
```

Definition at line 35 of file Rsrv.h.

7.177.1.66 SET_PAR

```
#define SET_PAR(  
    TY,  
    LEN ) (((LEN)&0xffffffff)<<8)|((TY)&255))
```

Definition at line 107 of file Rsrv.h.

7.177.1.67 SET_STAT

```
#define SET_STAT(  
    X,  
    s ) ((X)|(((s)&127)<<24)) /* sets the stat code */
```

Definition at line 110 of file Rsrv.h.

7.177.1.68 XT_ARRAY_BOOL

```
#define XT_ARRAY_BOOL 36 /* P data: int(n),byte,byte,... */
```

Definition at line 262 of file Rsrv.h.

7.177.1.69 XT_ARRAY_BOOL_UA

```
#define XT_ARRAY_BOOL_UA 35 /* - data: [n]byte,byte,.. (unaligned! NOT supported anymore) */
```

Definition at line 261 of file Rsrv.h.

7.177.1.70 XT_ARRAY_CPLX

```
#define XT_ARRAY_CPLX 38 /* P data: [n*16]double,double,... (Re,Im,Re,Im,...) */
```

Definition at line 264 of file Rsrv.h.

7.177.1.71 XT_ARRAY_DOUBLE

```
#define XT_ARRAY_DOUBLE 33 /* P data: [n*8]double,double,... */
```

Definition at line 259 of file Rsrv.h.

7.177.1.72 XT_ARRAY_INT

```
#define XT_ARRAY_INT 32 /* P data: [n*4]int,int,... */
```

Definition at line 258 of file Rsrv.h.

7.177.1.73 XT_ARRAY_STR

```
#define XT_ARRAY_STR 34 /* P data: string,string,... (string=byte,byte,...,0) padded with '\01' */
```

Definition at line 260 of file Rsrv.h.

7.177.1.74 XT_BOOL

```
#define XT_BOOL
```

Value:

```
6 /* - data: [1]byte boolean  
   (1=TRUE, 0=FALSE, 2=NA) */
```

Definition at line 243 of file Rsrv.h.

7.177.1.75 XT_CLOS

```
#define XT_CLOS 18 /* P X formals, X body (closure; since 0.1-5) */
```

Definition at line 249 of file Rsrv.h.

7.177.1.76 XT_DOUBLE

```
#define XT_DOUBLE 2 /* - data: [8]double */
```

Definition at line 239 of file Rsrv.h.

7.177.1.77 XT_HAS_ATTR

```
#define XT_HAS_ATTR
```

Value:

```
128 /* flag; if set, the following REXP is the
       attribute */
```

Definition at line 281 of file Rsrv.h.

7.177.1.78 XT_INT

```
#define XT_INT 1 /* - data: [4]int */
```

Definition at line 238 of file Rsrv.h.

7.177.1.79 XT_LANG

```
#define XT_LANG 4 /* - data: same as XT_LIST */
```

Definition at line 241 of file Rsrv.h.

7.177.1.80 XT_LANG_NOTAG

```
#define XT_LANG_NOTAG 22 /* s same as XT_LIST_NOTAG (since 0.5) */
```

Definition at line 253 of file Rsrv.h.

7.177.1.81 XT_LANG_TAG

```
#define XT_LANG_TAG 23 /* s same as XT_LIST_TAG (since 0.5) */
```

Definition at line 254 of file Rsrv.h.

7.177.1.82 XT_LARGE

```
#define XT_LARGE
```

Value:

```
64 /* new in 0102: if this flag is set then the length of the object  
is coded as 56-bit integer enlarging the header by 4 bytes */
```

Definition at line 279 of file Rsrv.h.

7.177.1.83 XT_LIST

```
#define XT_LIST 17 /* - X head, X vals, X tag (since 0.1-5) */
```

Definition at line 248 of file Rsrv.h.

7.177.1.84 XT_LIST_NOTAG

```
#define XT_LIST_NOTAG 20 /* s same as XT_VECTOR (since 0.5) */
```

Definition at line 251 of file Rsrv.h.

7.177.1.85 XT_LIST_TAG

```
#define XT_LIST_TAG 21 /* P X tag, X val, Y tag, Y val, ... (since 0.5) */
```

Definition at line 252 of file Rsrv.h.

7.177.1.86 XT_NULL

```
#define XT_NULL 0 /* P data: [0] */
```

Definition at line 237 of file Rsrv.h.

7.177.1.87 XT_RAW

```
#define XT_RAW 37 /* P data: int(n),byte,byte,... */
```

Definition at line 263 of file Rsrv.h.

7.177.1.88 XT_S4

```
#define XT_S4 7 /* P data: [0] */
```

Definition at line 245 of file Rsrv.h.

7.177.1.89 XT_STR

```
#define XT_STR 3 /* P data: [n]char null-term. strg. */
```

Definition at line 240 of file Rsrv.h.

7.177.1.90 XT_SYM

```
#define XT_SYM 5 /* - data: [n]char symbol name */
```

Definition at line 242 of file Rsrv.h.

7.177.1.91 XT_SYMNAME

```
#define XT_SYMNAME 19 /* s same as XT_STR (since 0.5) */
```

Definition at line 250 of file Rsrv.h.

7.177.1.92 XT_UNKNOWN

```
#define XT_UNKNOWN 48 /* P data: [4]int - SEXP type (as from TYPEOF(x)) */
```

Definition at line 266 of file Rsrv.h.

7.177.1.93 XT_VECTOR

```
#define XT_VECTOR 16 /* P data: [?]REXP,REXP,... */
```

Definition at line 247 of file Rsrv.h.

7.177.1.94 XT_VECTOR_EXP

```
#define XT_VECTOR_EXP 26 /* s same as XT_VECTOR (since 0.5) */
```

Definition at line 255 of file Rsrv.h.

7.177.1.95 XT_VECTOR_STR

```
#define XT_VECTOR_STR 27 /* - same as XT_VECTOR (since 0.5 but unused, use XT_ARRAY_STR instead) */
```

Definition at line 256 of file Rsrv.h.

7.177.2 Function Documentation

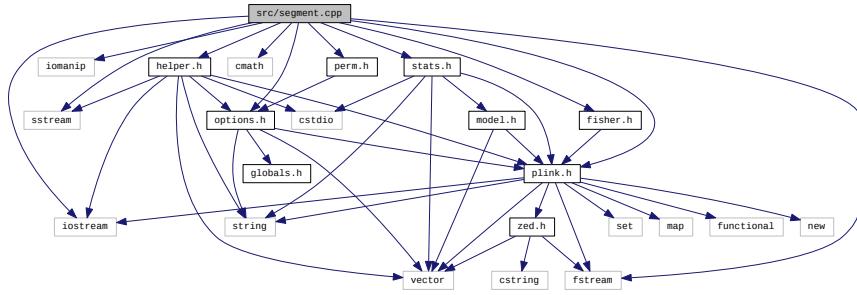
7.177.2.1 isByteSexOk()

```
int isByteSexOk ( )
```

7.178 src/segment.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <cmath>
#include "plink.h"
#include "helper.h"
#include "options.h"
#include "perm.h"
#include "fisher.h"
#include "stats.h"
```

Include dependency graph for segment.cpp:



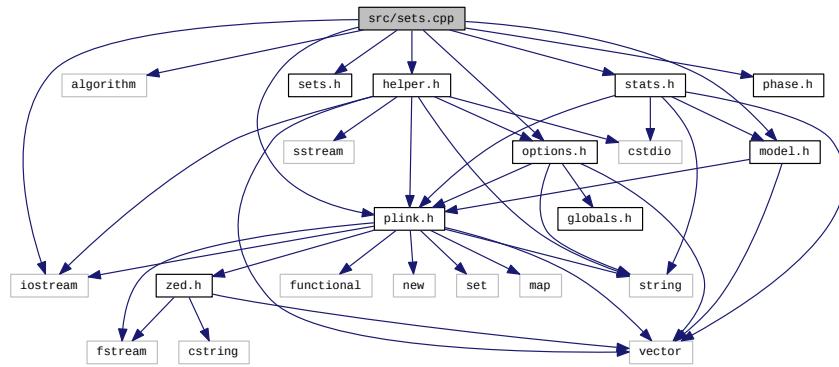
Classes

- class [SegmentSizeCmp](#)

7.179 src/sets.cpp File Reference

```
#include <iostream>
#include <algorithm>
#include "plink.h"
#include "sets.h"
#include "options.h"
#include "helper.h"
#include "model.h"
#include "stats.h"
```

```
#include "phase.h"
Include dependency graph for sets.cpp:
```



Variables

- `Plink * PP`

7.179.1 Variable Documentation

7.179.1.1 PP

`Plink* PP`

Definition at line 85 of file `inbix.cpp`.

7.180 src/sets.h File Reference

This graph shows which files directly or indirectly include this file:



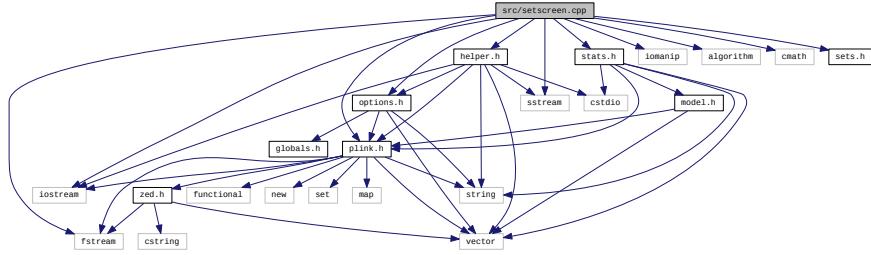
Classes

- class `SetSortedSNP`
- class `Set`

7.181 src/setscreen.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <algorithm>
#include <cmath>
#include "options.h"
#include "plink.h"
#include "helper.h"
#include "sets.h"
#include "stats.h"
```

Include dependency graph for setscreen.cpp:



Classes

- class [SortedResult](#)

Variables

- [Plink * PP](#)

7.181.1 Variable Documentation

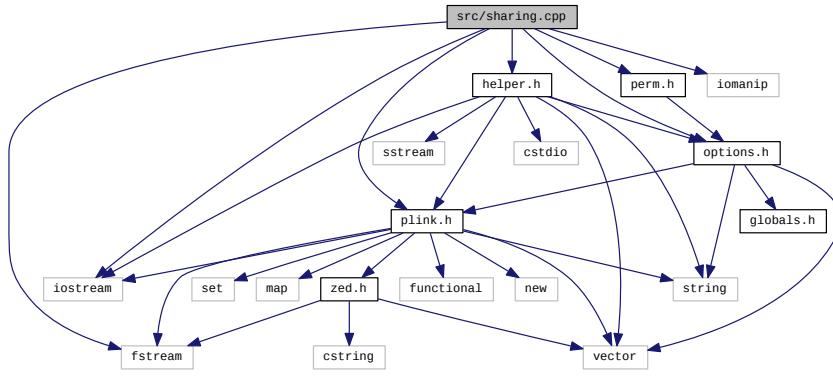
7.181.1.1 PP

[Plink* PP](#)

Definition at line 85 of file inbix.cpp.

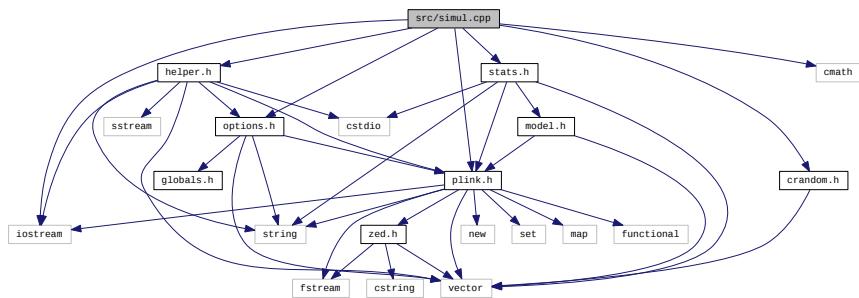
7.182 src/sharing.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include "plink.h"
#include "perm.h"
#include "options.h"
#include "helper.h"
Include dependency graph for sharing.cpp:
```



7.183 src/simul.cpp File Reference

```
#include <iostream>
#include "plink.h"
#include "helper.h"
#include "options.h"
#include "crandom.h"
#include "stats.h"
#include <cmath>
Include dependency graph for simul.cpp:
```



Classes

- class [SimParameters](#)
- class [SimParametersQT](#)

Functions

- `vector_t instanceSNP (SimParameters &s)`
- `vector_t instanceSNP_QT (SimParametersQT &s)`

7.183.1 Function Documentation

7.183.1.1 instanceSNP()

```
vector_t instanceSNP (
    SimParameters & s )
```

Definition at line 90 of file simul.cpp.

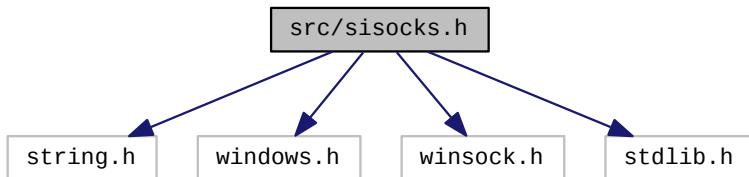
7.183.1.2 instanceSNP_QT()

```
vector_t instanceSNP_QT (
    SimParametersQT & s )
```

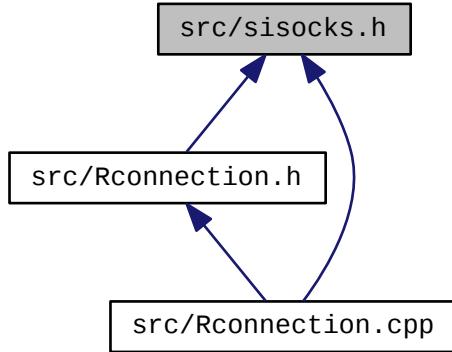
Definition at line 403 of file simul.cpp.

7.184 src/sisocks.h File Reference

```
#include <string.h>
#include <windows.h>
#include <winsock.h>
#include <stdlib.h>
Include dependency graph for sisocks.h:
```



This graph shows which files directly or indirectly include this file:



Macros

- `#define windows`
- `#define inet_aton(A, B) (0, B.s_addr=inet_addr(A))`
- `#define sockerrno WSAGetLastError()`
- `#define ECONNREFUSED WSAECONNREFUSED`
- `#define EADDRINUSE WSAEADDRINUSE`
- `#define ENOTSOCK WSAENOTSOCK`
- `#define EISCONN WSAEISCONN`
- `#define ETIMEDOUT WSAETIMEDOUT`
- `#define ENETUNREACH WSAENETUNREACH`
- `#define EINPROGRESS WSAEINPROGRESS`
- `#define EALREADY WSAEALREADY`
- `#define EAFNOSUPPORT WSAEAFNOSUPPORT`
- `#define EBADF WSAEBADF`
- `#define EINVAL WSAEINVAL`
- `#define EOPNOTSUPP WSAEOPNOTSUPP`
- `#define EFAULT WSAEFAULT`
- `#define EWOULDBLOCK WSAEWOULDBLOCK`
- `#define EACCES WSAEACCES`
- `#define SA struct sockaddr`
- `#define SAIN struct sockaddr_in`
- `#define donesocks() WSACleanup()`

Functions

- `int initsocks (void)`
- `struct sockaddr * build_sin (struct sockaddr_in *sa, char *ip, int port)`

7.184.1 Macro Definition Documentation

7.184.1.1 **donesocks**

```
#define donesocks( ) WSACleanup()
```

Definition at line 109 of file sisocks.h.

7.184.1.2 **EACCES**

```
#define EACCES WSAEACCES
```

Definition at line 71 of file sisocks.h.

7.184.1.3 **EADDRINUSE**

```
#define EADDRINUSE WSAEADDRINUSE
```

Definition at line 58 of file sisocks.h.

7.184.1.4 **EAFNOSUPPORT**

```
#define EAFNOSUPPORT WSAEAFNOSUPPORT
```

Definition at line 65 of file sisocks.h.

7.184.1.5 **EALREADY**

```
#define EALREADY WSAEALREADY
```

Definition at line 64 of file sisocks.h.

7.184.1.6 EBADF

```
#define EBADF WSAEBADF
```

Definition at line 66 of file sisocks.h.

7.184.1.7 ECONNREFUSED

```
#define ECONNREFUSED WSAECONNREFUSED
```

Definition at line 57 of file sisocks.h.

7.184.1.8 EFAULT

```
#define EFAULT WSAEFAULT
```

Definition at line 69 of file sisocks.h.

7.184.1.9 EINPROGRESS

```
#define EINPROGRESS WSAEINPROGRESS
```

Definition at line 63 of file sisocks.h.

7.184.1.10 EINVAL

```
#define EINVAL WSAEINVAL
```

Definition at line 67 of file sisocks.h.

7.184.1.11 EISCONN

```
#define EISCONN WSAEISCONN
```

Definition at line 60 of file sisocks.h.

7.184.1.12 ENETUNREACH

```
#define ENETUNREACH WSAENETUNREACH
```

Definition at line 62 of file sisocks.h.

7.184.1.13 ENOTSOCK

```
#define ENOTSOCK WSAENOTSOCK
```

Definition at line 59 of file sisocks.h.

7.184.1.14 EOPNOTSUPP

```
#define EOPNOTSUPP WSAEOPNOTSUPP
```

Definition at line 68 of file sisocks.h.

7.184.1.15 ETIMEDOUT

```
#define ETIMEDOUT WSAETIMEDOUT
```

Definition at line 61 of file sisocks.h.

7.184.1.16 EWOULDBLOCK

```
#define EWOULDBLOCK WSAEWOULDBLOCK
```

Definition at line 70 of file sisocks.h.

7.184.1.17 inet_aton

```
#define inet_aton( A,  
                 B ) ( 0, B.s_addr=inet_addr(A) )
```

Definition at line 53 of file sisocks.h.

7.184.1.18 SA

```
#define SA struct sockaddr
```

Definition at line 93 of file sisocks.h.

7.184.1.19 SAIN

```
#define SAIN struct sockaddr_in
```

Definition at line 94 of file sisocks.h.

7.184.1.20 sockerrno

```
#define sockerrno WSAGetLastError()
```

Definition at line 55 of file sisocks.h.

7.184.1.21 windows

```
#define windows
```

Definition at line 48 of file sisocks.h.

7.184.2 Function Documentation

7.184.2.1 build_sin()

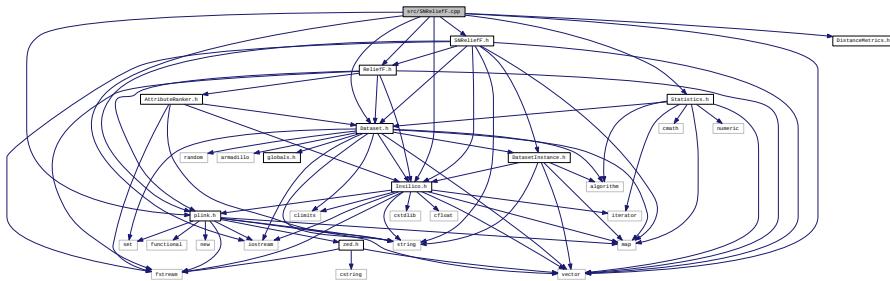
```
struct sockaddr* build_sin (
    struct sockaddr_in * sa,
    char * ip,
    int port )
```

7.184.2.2 initsocks()

```
int initsocks (
    void )
```

7.185 src/SNReliefF.cpp File Reference

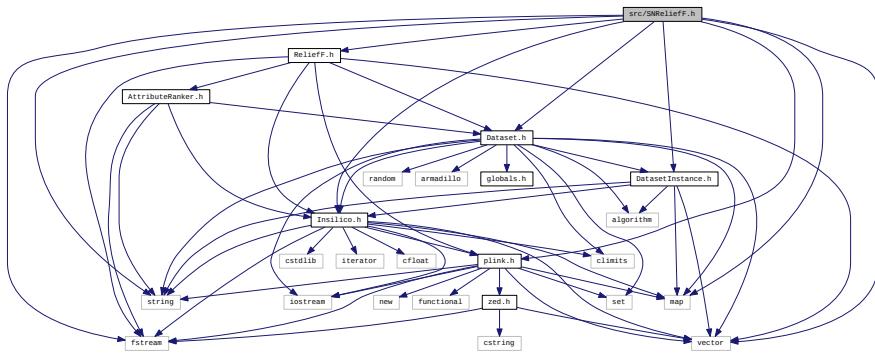
```
#include <iostream>
#include <vector>
#include "ReliefF.h"
#include "SNReliefF.h"
#include "Dataset.h"
#include "DistanceMetrics.h"
#include "Insilico.h"
#include "Statistics.h"
#include "plink.h"
Include dependency graph for SNReliefF.cpp:
```



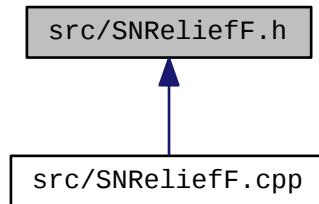
7.186 src/SNReliefF.h File Reference

```
#include <vector>
#include <map>
#include <string>
#include <fstream>
#include "plink.h"
#include "ReliefF.h"
#include "Dataset.h"
#include "DatasetInstance.h"
#include "Insilico.h"
```

Include dependency graph for SNReliefF.h:



This graph shows which files directly or indirectly include this file:



Classes

- class **SNReliefF**
*Signal-to-Noise **ReliefF** attribute ranking algorithm.*

TypeDefs

- `typedef std::vector< std::pair< double, double > > InstanceAttributeStats`
a pair of vectors for hit and miss statistics for each instance
 - `typedef std::vector< std::pair< double, double > >::const_iterator InstanceAttributeStatsIt`
 - `typedef std::pair< InstanceAttributeStats, InstanceAttributeStats > InstanceHitMissStats`
 - `typedef std::vector< InstanceHitMissStats > NeighborStats`
a map from instance ID to neighbor statistics
 - `typedef std::vector< InstanceHitMissStats >::iterator NeighborStatsIt`
 - `typedef std::vector< InstanceHitMissStats >::const_iterator NeighborStatsCIt`

7.186.1 Typedef Documentation

7.186.1.1 InstanceAttributeStats

```
typedef std::vector<std::pair<double, double> > InstanceAttributeStats
```

a pair of vectors for hit and miss statistics for each instance

Definition at line 39 of file SNReliefF.h.

7.186.1.2 InstanceAttributeStatsIt

```
typedef std::vector<std::pair<double, double> >::const_iterator InstanceAttributeStatsIt
```

Definition at line 41 of file SNReliefF.h.

7.186.1.3 InstanceHitMissStats

```
typedef std::pair<InstanceAttributeStats, InstanceAttributeStats> InstanceHitMissStats
```

Definition at line 44 of file SNReliefF.h.

7.186.1.4 NeighborStats

```
typedef std::vector<InstanceHitMissStats> NeighborStats
```

a map from instance ID to neighbor statistics

Definition at line 47 of file SNReliefF.h.

7.186.1.5 NeighborStatsCIt

```
typedef std::vector<InstanceHitMissStats>::const_iterator NeighborStatsCIt
```

Definition at line 49 of file SNReliefF.h.

7.186.1.6 NeighborStatsIt

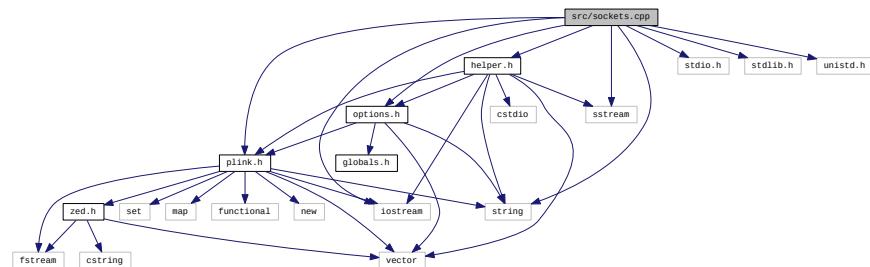
```
typedef std::vector<InstanceHitMissStats>::iterator NeighborStatsIt
```

Definition at line 48 of file SNReliefF.h.

7.187 src/sockets.cpp File Reference

```
#include <iostream>
#include <sstream>
#include "plink.h"
#include "helper.h"
#include "options.h"
#include <stdio.h>
#include <stdlib.h>
#include <string>
#include <unistd.h>
```

Include dependency graph for sockets.cpp:



Functions

- `vector< string > socketConnection (Plink *P, string ip_addr, int port, string message)`

Variables

- `string PVERSION`

7.187.1 Function Documentation

7.187.1.1 socketConnection()

```
vector<string> socketConnection (
    Plink * P,
    string ip_addr,
    int port,
    string message )
```

Definition at line 56 of file sockets.cpp.

7.187.2 Variable Documentation

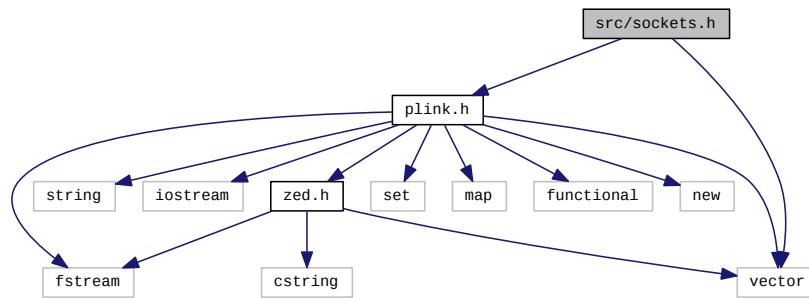
7.187.2.1 PVERSION

```
string PVERSION
```

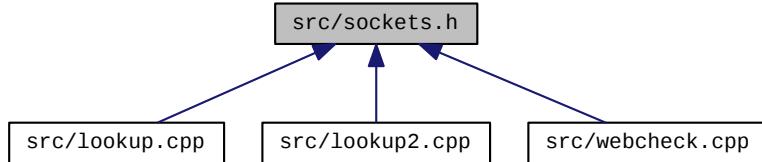
Definition at line 82 of file inbix.cpp.

7.188 src/sockets.h File Reference

```
#include <vector>
#include "plink.h"
Include dependency graph for sockets.h:
```



This graph shows which files directly or indirectly include this file:



Functions

- `vector< string > socketConnection (Plink *P, string ip_addr, int port, string message)`

7.188.1 Function Documentation

7.188.1.1 socketConnection()

```
vector<string> socketConnection (
```

	<code>Plink * P,</code>
	<code>string ip_addr,</code>
	<code>int port,</code>
	<code>string message)</code>

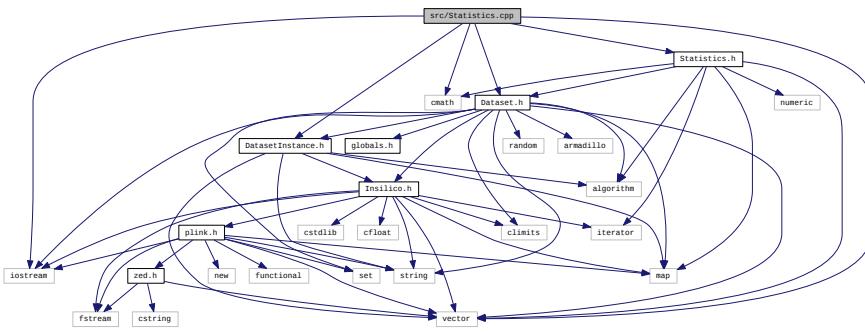
Definition at line 56 of file sockets.cpp.

7.189 src/Statistics.cpp File Reference

```
#include <iostream>
#include <vector>
#include <cmath>
#include "Dataset.h"
#include "DatasetInstance.h"
```

```
#include "Statistics.h"
```

Include dependency graph for Statistics.cpp:



Macros

- #define DEBUG_Z 0
- #define DEBUG_E 1

Functions

- void PrintHistogram (Histogram histogram)
Print a Histogram to cout.
- bool ZTransform (const VectorDouble &inputValues, VectorDouble &outputValues)
ZTransform input values.
- double SelfEntropy (const vector< AttributeLevel > &a, const vector< AttributeLevel > &c)
Calculates the entropy of a sequence with itself and the class.
- double Entropy (const vector< AttributeLevel > &sequenceValues)
Calculates the entropy of a sequence of unsigned integers.
- double condentropy (const vector< AttributeLevel > &X, const vector< AttributeLevel > &Y)
- double ConditionalEntropy (const vector< AttributeLevel > &sequenceValues, const vector< AttributeLevel > &givenValues)
Calculates the conditional entropy of a sequence of unsigned integers based (conditioned) on another sequence of unsigned integers (the givens).
- bool ConstructAttributeCart (const vector< AttributeLevel > &a, const vector< AttributeLevel > &b, vector< AttributeLevel > &ab)
Create a new attribute that is the Cartesian product of a and b.
- double KendallTau (vector< string > X, vector< string > Y)
Compute KendallTau for two ranked vectors of strings.
- double KendallTau (vector< double > X, vector< double > Y)
Compute KendallTau for two ranked vectors of doubles.
- double KendallTau (vector< int > X, vector< int > Y)
Compute KendallTau for two ranked vectors of integers.
- double hypot (double x, double y)
Compute the hypotenuse.

7.189.1 Macro Definition Documentation

7.189.1.1 DEBUG_E

```
#define DEBUG_E 1
```

Definition at line 18 of file Statistics.cpp.

7.189.1.2 DEBUG_Z

```
#define DEBUG_Z 0
```

Definition at line 17 of file Statistics.cpp.

7.189.2 Function Documentation

7.189.2.1 condentropy()

```
double condentropy (
    const vector< AttributeLevel > & X,
    const vector< AttributeLevel > & Y )
```

Definition at line 117 of file Statistics.cpp.

7.189.2.2 ConditionalEntropy()

```
double ConditionalEntropy (
    const std::vector< AttributeLevel > & attributeValues,
    const std::vector< AttributeLevel > & givenValues )
```

Calculates the conditional entropy of a sequence of unsigned integers based (conditioned) on another sequence of unsigned integers (the givens).

$P(\text{sequenceValues} \mid \text{givenValues})$

Parameters

in	<i>attributeValues</i>	vector of values
in	<i>givenValues</i>	vector of givens

Returns

conditional entropy as a double-precision float

convert from base e to base 2

Definition at line 126 of file Statistics.cpp.

7.189.2.3 ConstructAttributeCart()

```
bool ConstructAttributeCart (
    const std::vector< AttributeLevel > & a,
    const std::vector< AttributeLevel > & b,
    std::vector< AttributeLevel > & ab )
```

Create a new attribute that is the Cartesian product of a and b.

NOTE: works for genotypes; need to verify for missing data levels, etc.

Parameters

in	<i>a</i>	attributes vector a
in	<i>b</i>	attributes vector b
out	<i>vector</i>	ab, the Cartesian product of a and b

Returns

success

Get the number of levels in a for a multiplier

Definition at line 207 of file Statistics.cpp.

7.189.2.4 Entropy()

```
double Entropy (
    const std::vector< AttributeLevel > & attributeValues )
```

Calculates the entropy of a sequence of unsigned integers.

Parameters

in	<i>attributeValues</i>	vector of sequence values - unsigned ints - positive categorical
----	------------------------	--

Returns

entropy as a double-precision float

Definition at line 92 of file Statistics.cpp.

7.189.2.5 hypot()

```
double hypot (
    double x,
    double y )
```

Compute the hypotenuse.

Definition at line 351 of file Statistics.cpp.

7.189.2.6 KendallTau() [1/3]

```
double KendallTau (
    std::vector< std::string > X,
    std::vector< std::string > Y )
```

Compute KendallTau for two ranked vectors of strings.

Why Kendall Tau - G. E. NOETHER <http://www.rsscse-edu.org.uk/tsj/bts/noether/text.html>

Parameters

in	<i>X</i>	ranked attribute vector X
in	<i>Y</i>	ranked attribute vector Y

Returns

Kendall Tau value (-1, 1)

Definition at line 247 of file Statistics.cpp.

7.189.2.7 KendallTau() [2/3]

```
double KendallTau (
    std::vector< double > X,
    std::vector< double > Y )
```

Compute KendallTau for two ranked vectors of doubles.

Why Kendall Tau - G. E. NOETHER <http://www.rsscse-edu.org.uk/tsj/bts/noether/text.html>

Parameters

in	X	ranked attribute vector X
in	Y	ranked attribute vector Y

Returns

Kendall Tau value (-1, 1)

Definition at line 283 of file Statistics.cpp.

7.189.2.8 KendallTau() [3/3]

```
double KendallTau (
    std::vector< int > X,
    std::vector< int > Y )
```

Compute KendallTau for two ranked vectors of integers.

Why Kendall Tau - G. E. NOETHER <http://www.rsscse-edu.org.uk/tsj/bts/noether/text.html>

Parameters

in	X	ranked attribute vector X
in	Y	ranked attribute vector Y

Returns

Kendall Tau value (-1, 1)

Definition at line 317 of file Statistics.cpp.

7.189.2.9 PrintHistogram()

```
void PrintHistogram (
    Histogram histogram )
```

Print a Histogram to cout.

Parameters

in	<i>histogram</i>	Histogram to print
----	------------------	--------------------

Definition at line 20 of file Statistics.cpp.

7.189.2.10 SelfEntropy()

```
double SelfEntropy (
    const std::vector< AttributeLevel > & a,
    const std::vector< AttributeLevel > & c )
```

Calculates the entropy of a sequence with itself and the class.

Parameters

in	<i>a</i>	vector of values
in	<i>c</i>	vector of class levels

Returns

entropy as a double-precision float

Definition at line 86 of file Statistics.cpp.

7.189.2.11 ZTransform()

```
bool ZTransform (
    const VectorDouble & inputValues,
    VectorDouble & outputValues )
```

ZTransform input values.

Parameters

in	<i>inputValues</i>	const vector of double input values
out	<i>outputValues</i>	transformed input values to z-scores with mean=0, stddev=1

Returns

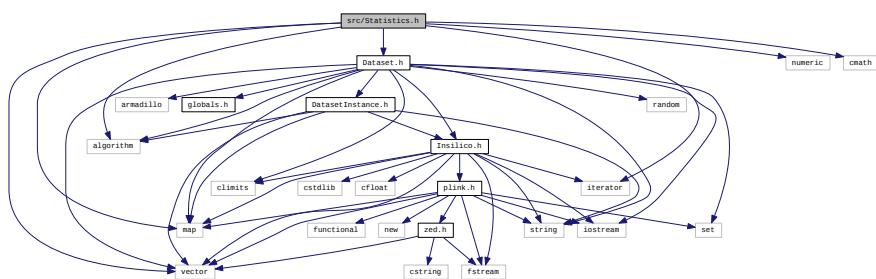
success

Definition at line 27 of file Statistics.cpp.

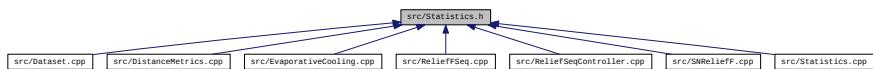
7.190 src/Statistics.h File Reference

Statistical utilities.

```
#include <vector>
#include <map>
#include <numeric>
#include <iterator>
#include <cmath>
#include <algorithm>
#include "Dataset.h"
Include dependency graph for Statistics.h:
```



This graph shows which files directly or indirectly include this file:



Typedefs

- `typedef std::vector< double > VectorDouble`
vector of doubles type
- `typedef std::vector< double >::const_iterator VectorDoubleIt`
vector of doubles iterator
- `typedef std::map< AttributeLevel, unsigned int > Histogram`
histogram type is a map: value->count
- `typedef std::map< AttributeLevel, unsigned int >::const_iterator HistogramIt`
histogram iterator

Functions

- `void PrintHistogram (Histogram histogram)`
Print a Histogram to cout.
- `bool ZTransform (const VectorDouble &inputValues, VectorDouble &outputValues)`
ZTransform input values.
- `double SelfEntropy (const std::vector< AttributeLevel > &a, const std::vector< AttributeLevel > &c)`
Calculates the entropy of a sequence with itself and the class.
- `double Entropy (const std::vector< AttributeLevel > &attributeValues)`
Calculates the entropy of a sequence of unsigned integers.
- `double ConditionalEntropy (const std::vector< AttributeLevel > &attributeValues, const std::vector< AttributeLevel > &givenValues)`
Calculates the conditional entropy of a sequence of unsigned integers based (conditioned) on another sequence of unsigned integers (the givens).
- `double condentropy (const std::vector< AttributeLevel > &X, const std::vector< AttributeLevel > &Y)`
- `bool ConstructAttributeCart (const std::vector< AttributeLevel > &a, const std::vector< AttributeLevel > &b, std::vector< AttributeLevel > &ab)`
Create a new attribute that is the Cartesian product of a and b.
- `double KendallTau (std::vector< std::string > X, std::vector< std::string > Y)`
Compute KendallTau for two ranked vectors of strings.
- `double KendallTau (std::vector< double > X, std::vector< double > Y)`
Compute KendallTau for two ranked vectors of doubles.
- `double KendallTau (std::vector< int > X, std::vector< int > Y)`
Compute KendallTau for two ranked vectors of integers.
- `double hypot (double x, double y)`
Compute the hypotenuse.
- `template<class T > std::pair< double, double > VarStd (std::vector< T > &values, double &average)`
Calculate variance and standard deviation of a vector of values.

7.190.1 Detailed Description

Statistical utilities.

Author

Bill White

Version

1.0

Contact: bill.c.white@gmail.com Created on: 2/13/06

7.190.2 Typedef Documentation

7.190.2.1 Histogram

```
typedef std::map<AttributeLevel, unsigned int> Histogram
```

histogram type is a map: value->count

Definition at line 30 of file Statistics.h.

7.190.2.2 HistogramIt

```
typedef std::map<AttributeLevel, unsigned int>::const_iterator HistogramIt
```

histogram iterator

Definition at line 32 of file Statistics.h.

7.190.2.3 VectorDouble

```
typedef std::vector<double> VectorDouble
```

vector of doubles type

Definition at line 26 of file Statistics.h.

7.190.2.4 VectorDoubleIt

```
typedef std::vector<double>::const_iterator VectorDoubleIt
```

vector of doubles iterator

Definition at line 28 of file Statistics.h.

7.190.3 Function Documentation

7.190.3.1 condentropy()

```
double condentropy (
    const std::vector< AttributeLevel > & X,
    const std::vector< AttributeLevel > & Y )
```

Definition at line 117 of file Statistics.cpp.

7.190.3.2 ConditionalEntropy()

```
double ConditionalEntropy (
    const std::vector< AttributeLevel > & attributeValues,
    const std::vector< AttributeLevel > & givenValues )
```

Calculates the conditional entropy of a sequence of unsigned integers based (conditioned) on another sequence of unsigned integers (the givens).

$P(\text{sequenceValues} \mid \text{givenValues})$

Parameters

in	<i>attributeValues</i>	vector of values
in	<i>givenValues</i>	vector of givens

Returns

conditional entropy as a double-precision float

convert from base e to base 2

Definition at line 126 of file Statistics.cpp.

7.190.3.3 ConstructAttributeCart()

```
bool ConstructAttributeCart (
    const std::vector< AttributeLevel > & a,
    const std::vector< AttributeLevel > & b,
    std::vector< AttributeLevel > & ab )
```

Create a new attribute that is the Cartesian product of a and b.

NOTE: works for genotypes; need to verify for missing data levels, etc.

Parameters

in	<i>a</i>	attributes vector a
in	<i>b</i>	attributes vector b
out	<i>vector</i>	ab, the Cartesian product of a and b

Returns

success

Get the number of levels in a for a multiplier

Definition at line 207 of file Statistics.cpp.

7.190.3.4 Entropy()

```
double Entropy (
    const std::vector< AttributeLevel > & attributeValues )
```

Calculates the entropy of a sequence of unsigned integers.

Parameters

in	<i>attributeValues</i>	vector of sequence values - unsigned ints - positive categorical
----	------------------------	--

Returns

entropy as a double-precision float

Definition at line 92 of file Statistics.cpp.

7.190.3.5 hypot()

```
double hypot (
    double x,
    double y )
```

Compute the hypotenuse.

Definition at line 351 of file Statistics.cpp.

7.190.3.6 KendallTau() [1/3]

```
double KendallTau (
    std::vector< std::string > X,
    std::vector< std::string > Y )
```

Compute KendallTau for two ranked vectors of strings.

Why Kendall Tau - G. E. NOETHER <http://www.rsscse-edu.org.uk/tsj/bts/noether/text.html>

Parameters

in	X	ranked attribute vector X
in	Y	ranked attribute vector Y

Returns

Kendall Tau value (-1, 1)

Definition at line 247 of file Statistics.cpp.

7.190.3.7 KendallTau() [2/3]

```
double KendallTau (
    std::vector< double > X,
    std::vector< double > Y )
```

Compute KendallTau for two ranked vectors of doubles.

Why Kendall Tau - G. E. NOETHER <http://www.rsscse-edu.org.uk/tsj/bts/noether/text.html>

Parameters

in	X	ranked attribute vector X
in	Y	ranked attribute vector Y

Returns

Kendall Tau value (-1, 1)

Definition at line 283 of file Statistics.cpp.

7.190.3.8 KendallTau() [3/3]

```
double KendallTau (
    std::vector< int > X,
    std::vector< int > Y )
```

Compute KendallTau for two ranked vectors of integers.

Why Kendall Tau - G. E. NOETHER <http://www.rsscse-edu.org.uk/tsj/bts/noether/text.html>

Parameters

in	X	ranked attribute vector X
in	Y	ranked attribute vector Y

Returns

Kendall Tau value (-1, 1)

Definition at line 317 of file Statistics.cpp.

7.190.3.9 PrintHistogram()

```
void PrintHistogram (
    Histogram histogram )
```

Print a Histogram to cout.

Parameters

in	histogram	Histogram to print
----	-----------	--------------------

Definition at line 20 of file Statistics.cpp.

7.190.3.10 SelfEntropy()

```
double SelfEntropy (
    const std::vector< AttributeLevel > & a,
    const std::vector< AttributeLevel > & c )
```

Calculates the entropy of a sequence with itself and the class.

Parameters

in	<i>a</i>	vector of values
in	<i>c</i>	vector of class levels

Returns

entropy as a double-precision float

Definition at line 86 of file Statistics.cpp.

7.190.3.11 VarStd()

```
template<class T >
std::pair<double, double> VarStd (
    std::vector< T > & values,
    double & average )
```

Calculate variance and standard deviation of a vector of values.

Parameters

in	<i>ranked</i>	attribute lists X and Y
----	---------------	-------------------------

Returns

pair: variance and standard deviation

Definition at line 118 of file Statistics.h.

7.190.3.12 ZTransform()

```
bool ZTransform (
    const VectorDouble & inputValues,
    VectorDouble & outputValues )
```

ZTransform input values.

Parameters

in	<i>inputValues</i>	const vector of double input values
out	<i>outputValues</i>	transformed input values to z-scores with mean=0, stddev=1

Returns

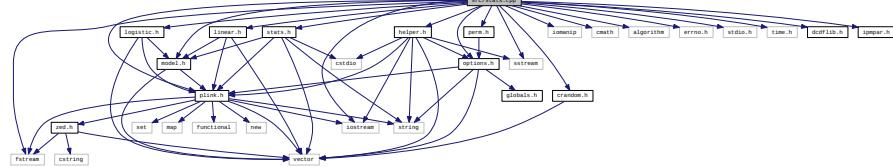
success

Definition at line 27 of file Statistics.cpp.

7.191 src/stats.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <cmath>
#include <algorithm>
#include <errno.h>
#include <stdio.h>
#include <time.h>
#include "stats.h"
#include "helper.h"
#include "crandom.h"
#include "options.h"
#include "plink.h"
#include "perm.h"
#include "dcdflib.h"
#include "ipmpar.h"
#include "model.h"
#include "linear.h"
#include "logistic.h"
```

Include dependency graph for stats.cpp:



Macros

- #define FPMIN 1.0e-30
 - #define LOW 0.02425
 - #define HIGH 0.97575

Functions

- bool `realnum` (double `d`)
- long double `factorial` (int `x`)
- double `normdist` (double `z`)
- double `chiprobP` (double `x`, double `df`)
- double `inverse_chiprob` (double `q`, double `df`)
- double `gammln` (double `xx`)
- double `ltqnorm` (double `p`)
- double `pT` (double `T`, double `df`)
- double `pF` (const double `F`, const int `df1`, const int `df2`)
- double `betai` (const double `a`, const double `b`, const double `x`)
- double `betacf` (const double `a`, const double `b`, const double `x`)
- vector<vector<double>> `inverse` (vector<vector<double>> &`m`)
- vector<double> `eigenvalues` (vector<vector<double>> &`a`)
- void `tred2` (vector<vector<double>> &`a`, vector<double> &`d`, vector<double> &`e`)
- void `tqli` (vector<double> &`d`, vector<double> &`e`, vector<vector<double>> &`z`)
- Eigen `eigenvectors` (vector<vector<double>> &`a`)
- void `EV_tred2` (vector<vector<double>> &`a`, vector<double> &`d`, vector<double> &`e`)
- void `EV_tqli` (vector<double> &`d`, vector<double> &`e`, vector<vector<double>> &`z`)
- double `qromb` (double func(const double), double `a`, double `b`)
- void `polint` (vector_t &`xa`, vector_t &`ya`, const double `x`, double &`y`, double &`dy`)
- double `trapzd` (double func(const double), const double `a`, const double `b`, const int `n`)
- void `svdvar` (vector<vector<double>> &`v`, vector<double> &`w`, vector<vector<double>> &`cvm`)
- void `svbksb` (vector<vector<double>> &`u`, vector<double> &`w`, vector<vector<double>> &`v`, vector<double> &`b`, vector<double> &`x`)
- bool `svd` (matrix_t &`u`, vector_t &`w`, matrix_t &`v`)
- vector<vector<double>> `svd_inverse` (vector<vector<double>> &`u`, bool &`flag`)
- vector<vector<double>> `msqrt` (vector<vector<double>> &`u`)
- void `ludcmp` (vector<vector<double>> &`a`, vector<int> &`indx`, double &`d`)
- void `lubksb` (vector<vector<double>> &`a`, vector<int> &`indx`, vector<double> &`b`)
- bool `svdcmp` (vector<vector<double>> &`a`, vector<double> &`w`, vector<vector<double>> &`v`)
- double `pythag` (const double `a`, const double `b`)
- double `SQR` (double `a`)
- void `multMatrix` (vector<vector<double>> &`a`, vector<vector<double>> &`b`, vector<vector<double>> &`c`)
- double `symTable` (table_t `t`)
- double `chiTable` (table_t `t`)
- double `chi2x2` (table_t `t`)
- double `chi2x2` (matrix_t `t`)
- double `chi2x2` (double `a`, double `b`, double `c`, double `d`)
- int `pca` (matrix_t &`x`, boolmatrix_t &`mask`, vector_t &`p`, matrix_t &`s`, matrix_t &`v`, bool `mean_centre=true`)
- matrix_t `vec2diag` (vector_t &`v`)
- double `rnorm` ()
- bool `matrixComputeCovariance` (matrix_t `X`, matrix_t &`covMatrix`, matrix_t &`corMatrix`)
- bool `matrixFill` (matrix_t &`m`, double `f`)
- bool `matrixMultiplyScalar` (matrix_t &`m`, double `s`)
- bool `matrixDivideScalar` (matrix_t &`m`, double `s`)
- bool `matrixAdd` (matrix_t `m1`, matrix_t `m2`, matrix_t &`result`)
- bool `matrixSubtract` (matrix_t `m1`, matrix_t `m2`, matrix_t &`result`)
- bool `matrixTranspose` (matrix_t `in`, matrix_t &`out`)

- bool `matrixGetDiag (matrix_t m, vector_t &d)`
- bool `matrixSetDiag (matrix_t &m, vector_t d)`
- bool `matrixRead (string mFilename, matrix_t &m, vector< string > &variableNames)`
- bool `matrixWrite (matrix_t m, string mFilename, vector< string > variableNames)`
- bool `matrixSums (matrix_t m, vector_t &sums, int dim)`
- bool `matrixComputeNodeDegrees (matrix_t a, vector_t &ad)`
- bool `matrixExtractRowColIdx (matrix_t m, intvec_t rowIdx, intvec_t colIdx, matrix_t &nm)`
- bool `matrixGetTrace (matrix_t m, double &t)`
- bool `matrixElementWiseMultiply (matrix_t m, matrix_t n, matrix_t &m_out)`
- bool `matrixConnectivityThreshold (matrix_t &m, double t, bool binary)`
- bool `reportNumericSummaryStats ()`
- bool `vectorSummary (vector_t values, vector_t &summary)`
- bool `rankByRegression (RegressionRankType rankType, rankedlist_t &ranks, RegressionRankResults &results)`
- pair< double, double > `fitModel (Model *mainEffectModel)`
- bool `numericLowValueFilter (double percentile, boolevec_t &varFlags)`
- bool `numericVarianceFilter (double percentile, boolevec_t &varFlags)`
- bool `quantile (vector_t values, double percentile, double &percentileValue)`
- bool `numericMeanCenter ()`
- bool `numericStandardize ()`
- bool `getNumericCaseControl (int varIndex, vector_t &cases, vector_t &controls)`
- bool `tTest (int varIndex, double &t)`
- bool `zTest (int varIndex, double &z)`

Variables

- ofstream `LOG`
- `Plink * PP`
- static const double `a []`
- static const double `b []`
- static const double `c []`
- static const double `d []`

7.191.1 Macro Definition Documentation

7.191.1.1 FPMIN

```
#define FPMIN 1.0e-30
```

Definition at line 39 of file stats.cpp.

7.191.1.2 HIGH

```
#define HIGH 0.97575
```

Definition at line 187 of file stats.cpp.

7.191.1.3 LOW

```
#define LOW 0.02425
```

Definition at line 186 of file stats.cpp.

7.191.2 Function Documentation

7.191.2.1 betacf()

```
double betacf (
    const double a,
    const double b,
    const double x )
```

Definition at line 258 of file stats.cpp.

7.191.2.2 betai()

```
double betai (
    const double a,
    const double b,
    const double x )
```

Definition at line 245 of file stats.cpp.

7.191.2.3 chi2x2() [1/3]

```
double chi2x2 (
    table\_t t )
```

Definition at line 1224 of file stats.cpp.

7.191.2.4 chi2x2() [2/3]

```
double chi2x2 (
    matrix_t t )
```

Definition at line 1228 of file stats.cpp.

7.191.2.5 chi2x2() [3/3]

```
double chi2x2 (
    double a,
    double b,
    double c,
    double d )
```

Definition at line 1232 of file stats.cpp.

7.191.2.6 chiprobP()

```
double chiprobP (
    double x,
    double df )
```

Definition at line 74 of file stats.cpp.

7.191.2.7 chiTable()

```
double chiTable (
    table_t t )
```

Definition at line 1192 of file stats.cpp.

7.191.2.8 eigenvalues()

```
vector<double> eigenvalues (
    vector< vector< double > > & a )
```

Definition at line 325 of file stats.cpp.

7.191.2.9 eigenvectors()

```
Eigen eigenvectors (
    vector< vector< double > > & a )
```

Definition at line 469 of file stats.cpp.

7.191.2.10 EV_tqli()

```
void EV_tqli (
    vector< double > & d,
    vector< double > & e,
    vector< vector< double > > & z )
```

Definition at line 556 of file stats.cpp.

7.191.2.11 EV_tred2()

```
void EV_tred2 (
    vector< vector< double > > & a,
    vector< double > & d,
    vector< double > & e )
```

Definition at line 486 of file stats.cpp.

7.191.2.12 factorial()

```
long double factorial (
    int x )
```

Definition at line 52 of file stats.cpp.

7.191.2.13 fitModel()

```
pair<double, double> fitModel (
    Model * mainEffectModel )
```

Definition at line 2021 of file stats.cpp.

7.191.2.14 gammln()

```
double gammln (
    double xx )
```

Definition at line 114 of file stats.cpp.

7.191.2.15 getNumericCaseControl()

```
bool getNumericCaseControl (
    int varIndex,
    vector_t & cases,
    vector_t & controls )
```

Definition at line 2187 of file stats.cpp.

7.191.2.16 inverse()

```
vector< vector<double> > inverse (
    vector< vector< double > > & m )
```

Definition at line 296 of file stats.cpp.

7.191.2.17 inverse_chiprob()

```
double inverse_chiprob (
    double q,
    double df )
```

Definition at line 93 of file stats.cpp.

7.191.2.18 ltqnorm()

```
double ltqnorm (
    double p )
```

Definition at line 189 of file stats.cpp.

7.191.2.19 lubksb()

```
void lubksb (
    vector< vector< double > > & a,
    vector< int > & indx,
    vector< double > & b )
```

Definition at line 920 of file stats.cpp.

7.191.2.20 ludcmp()

```
void ludcmp (
    vector< vector< double > > & a,
    vector< int > & indx,
    double & d )
```

Definition at line 869 of file stats.cpp.

7.191.2.21 matrixAdd()

```
bool matrixAdd (
    matrix_t m1,
    matrix_t m2,
    matrix_t & result )
```

Definition at line 1478 of file stats.cpp.

7.191.2.22 matrixComputeCovariance()

```
bool matrixComputeCovariance (
    matrix_t X,
    matrix_t & covMatrix,
    matrix_t & corMatrix )
```

Definition at line 1407 of file stats.cpp.

7.191.2.23 matrixComputeNodeDegrees()

```
bool matrixComputeNodeDegrees (
    matrix_t a,
    vector_t & ad )
```

Definition at line 1687 of file stats.cpp.

7.191.2.24 matrixConnectivityThreshold()

```
bool matrixConnectivityThreshold (
    matrix_t & m,
    double t,
    bool binary )
```

Definition at line 1748 of file stats.cpp.

7.191.2.25 matrixDivideScalar()

```
bool matrixDivideScalar (
    matrix_t & m,
    double s )
```

Definition at line 1468 of file stats.cpp.

7.191.2.26 matrixElementWiseMultiply()

```
bool matrixElementWiseMultiply (
    matrix_t m,
    matrix_t n,
    matrix_t & m_out )
```

Definition at line 1727 of file stats.cpp.

7.191.2.27 matrixExtractRowColIdx()

```
bool matrixExtractRowColIdx (
    matrix_t m,
    intvec_t rowIdx,
    intvec_t colIdx,
    matrix_t & nm )
```

Definition at line 1703 of file stats.cpp.

7.191.2.28 matrixFill()

```
bool matrixFill (
    matrix_t & m,
    double f )
```

Definition at line 1448 of file stats.cpp.

7.191.2.29 matrixGetDiag()

```
bool matrixGetDiag (
    matrix_t m,
    vector_t & d )
```

Definition at line 1538 of file stats.cpp.

7.191.2.30 matrixGetTrace()

```
bool matrixGetTrace (
    matrix_t m,
    double & t )
```

Definition at line 1719 of file stats.cpp.

7.191.2.31 matrixMultiplyScalar()

```
bool matrixMultiplyScalar (
    matrix_t & m,
    double s )
```

Definition at line 1458 of file stats.cpp.

7.191.2.32 matrixRead()

```
bool matrixRead (
    string mFilename,
    matrix_t & m,
    vector< string > & variableNames )
```

Definition at line 1553 of file stats.cpp.

7.191.2.33 matrixSetDiag()

```
bool matrixSetDiag (
    matrix_t & m,
    vector_t d )
```

Definition at line 1546 of file stats.cpp.

7.191.2.34 matrixSubtract()

```
bool matrixSubtract (
    matrix_t m1,
    matrix_t m2,
    matrix_t & result )
```

Definition at line 1501 of file stats.cpp.

7.191.2.35 matrixSums()

```
bool matrixSums (
    matrix_t m,
    vector_t & sums,
    int dim )
```

Definition at line 1667 of file stats.cpp.

7.191.2.36 matrixTranspose()

```
bool matrixTranspose (
    matrix_t in,
    matrix_t & out )
```

Definition at line 1524 of file stats.cpp.

7.191.2.37 matrixWrite()

```
bool matrixWrite (
    matrix_t m,
    string mFilename,
    vector< string > variableNames )
```

Definition at line 1626 of file stats.cpp.

7.191.2.38 msqrt()

```
vector<vector<double> > msqrt (
    vector< vector< double > > & u )
```

Definition at line 827 of file stats.cpp.

7.191.2.39 multMatrix()

```
void multMatrix (
    vector< vector< double > > & a,
    vector< vector< double > > & b,
    vector< vector< double > > & c )
```

Definition at line 1145 of file stats.cpp.

7.191.2.40 normdist()

```
double normdist (
    double z )
```

Definition at line 60 of file stats.cpp.

7.191.2.41 numericLowValueFilter()

```
bool numericLowValueFilter (
    double percentile,
    boolevec_t & varFlags )
```

Definition at line 2050 of file stats.cpp.

7.191.2.42 numericMeanCenter()

```
bool numericMeanCenter ( )
```

Definition at line 2151 of file stats.cpp.

7.191.2.43 numericStandardize()

```
bool numericStandardize ( )
```

Definition at line 2165 of file stats.cpp.

7.191.2.44 numericVarianceFilter()

```
bool numericVarianceFilter (
    double percentile,
    boolevec_t & varFlags )
```

Definition at line 2093 of file stats.cpp.

7.191.2.45 pca()

```
int pca (
    matrix_t & x,
    boolmatrix_t & mask,
    vector_t & p,
    matrix_t & s,
    matrix_t & v,
    bool mean_centre = true )
```

Definition at line 1254 of file stats.cpp.

7.191.2.46 pF()

```
double pF (
    const double F,
    const int df1,
    const int df2 )
```

Definition at line 241 of file stats.cpp.

7.191.2.47 polint()

```
void polint (
    vector_t & xa,
    vector_t & ya,
    const double x,
    double & y,
    double & dy )
```

Definition at line 635 of file stats.cpp.

7.191.2.48 pT()

```
double pT (
    double T,
    double df )
```

Definition at line 219 of file stats.cpp.

7.191.2.49 pythag()

```
double pythag (
    const double a,
    const double b )
```

Definition at line 1132 of file stats.cpp.

7.191.2.50 qromb()

```
double qromb (
    double funcconst double,
    double a,
    double b )
```

Definition at line 611 of file stats.cpp.

7.191.2.51 quantile()

```
bool quantile (
    vector_t values,
    double percentile,
    double & percentileValue )
```

Definition at line 2136 of file stats.cpp.

7.191.2.52 rankByRegression()

```
bool rankByRegression (
    RegressionRankType rankType,
    rankedlist_t & ranks,
    RegressionRankResults & results )
```

Definition at line 1847 of file stats.cpp.

7.191.2.53 realnum()

```
bool realnum (
    double d )
```

Definition at line 44 of file stats.cpp.

7.191.2.54 reportNumericSummaryStats()

```
bool reportNumericSummaryStats ( )
```

Definition at line 1766 of file stats.cpp.

7.191.2.55 rnorm()

```
double rnorm ( )
```

Definition at line 1395 of file stats.cpp.

7.191.2.56 SQR()

```
double SQR (
    double a )
```

Definition at line 1141 of file stats.cpp.

7.191.2.57 svbksb()

```
void svbksb (
    vector< vector< double > > & u,
    vector< double > & w,
    vector< vector< double > > & v,
    vector< double > & b,
    vector< double > & x )
```

Definition at line 707 of file stats.cpp.

7.191.2.58 svd()

```
bool svd (
    matrix_t & u,
    vector_t & w,
    matrix_t & v )
```

Definition at line 743 of file stats.cpp.

7.191.2.59 svd_inverse()

```
vector< vector< double > > svd_inverse (
    vector< vector< double > > & u,
    bool & flag )
```

Definition at line 778 of file stats.cpp.

7.191.2.60 svdcmp()

```
bool svdcmp (
    vector< vector< double > > & a,
    vector< double > & w,
    vector< vector< double > > & v )
```

Definition at line 943 of file stats.cpp.

7.191.2.61 svdvar()

```
void svdvar (
    vector< vector< double > > & v,
    vector< double > & w,
    vector< vector< double > > & cvm )
```

Definition at line 685 of file stats.cpp.

7.191.2.62 symTable()

```
double symTable (
    table_t t )
```

Definition at line 1171 of file stats.cpp.

7.191.2.63 tqli()

```
void tqli (
    vector< double > & d,
    vector< double > & e,
    vector< vector< double > > & z )
```

Definition at line 413 of file stats.cpp.

7.191.2.64 trapzd()

```
double trapzd (
    double funcconst double,
    const double a,
    const double b,
    const int n )
```

Definition at line 666 of file stats.cpp.

7.191.2.65 tred2()

```
void tred2 (
    vector< vector< double > > & a,
    vector< double > & d,
    vector< double > & e )
```

Definition at line 341 of file stats.cpp.

7.191.2.66 tTest()

```
bool tTest (
    int varIndex,
    double & t )
```

Definition at line 2235 of file stats.cpp.

7.191.2.67 vec2diag()

```
matrix_t vec2diag (
    vector_t & v )
```

Definition at line 1387 of file stats.cpp.

7.191.2.68 vectorSummary()

```
bool vectorSummary (
    vector_t values,
    vector_t & summary )
```

Definition at line 1823 of file stats.cpp.

7.191.2.69 zTest()

```
bool zTest (
    int varIndex,
    double & z )
```

Definition at line 2268 of file stats.cpp.

7.191.3 Variable Documentation

7.191.3.1 a

```
const double a[] [static]
```

Initial value:

```
= {
    -3.969683028665376e+01,
    2.209460984245205e+02,
    -2.759285104469687e+02,
    1.383577518672690e+02,
    -3.066479806614716e+01,
    2.506628277459239e+00
}
```

Definition at line 153 of file stats.cpp.

7.191.3.2 b

```
const double b[] [static]
```

Initial value:

```
= {
    -5.447609879822406e+01,
    1.615858368580409e+02,
    -1.556989798598866e+02,
    6.680131188771972e+01,
    -1.328068155288572e+01
}
```

Definition at line 162 of file stats.cpp.

7.191.3.3 c

```
const double c[ ] [static]
```

Initial value:

```
= {  
-7.784894002430293e-03,  
-3.223964580411365e-01,  
-2.400758277161838e+00,  
-2.549732539343734e+00,  
4.374664141464968e+00,  
2.938163982698783e+00  
}
```

Definition at line 170 of file stats.cpp.

7.191.3.4 d

```
const double d[ ] [static]
```

Initial value:

```
= {  
7.784695709041462e-03,  
3.224671290700398e-01,  
2.445134137142996e+00,  
3.754408661907416e+00  
}
```

Definition at line 179 of file stats.cpp.

7.191.3.5 LOG

```
ofstream LOG
```

Definition at line 81 of file inbix.cpp.

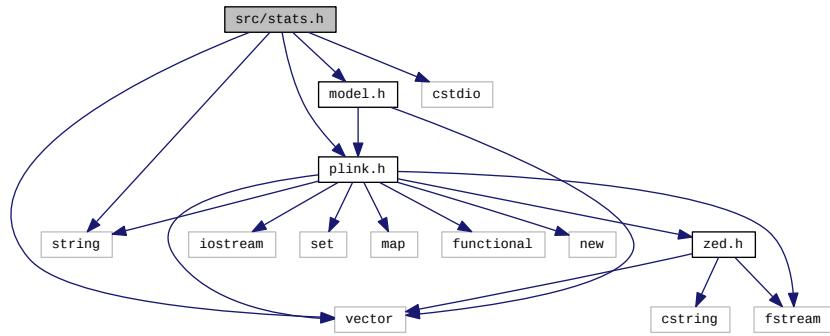
7.191.3.6 PP

```
Plink* PP
```

Definition at line 85 of file inbix.cpp.

7.192 src/stats.h File Reference

```
#include <string>
#include <vector>
#include <cstdio>
#include "plink.h"
#include "model.h"
Include dependency graph for stats.h:
```



Classes

- class [Eigen](#)

Functions

- void [sizeMatrix \(matrix_t &, int, int\)](#)
- void [sizeTable \(table_t &, int, int\)](#)
- void [multMatrix \(matrix_t &a, matrix_t &b, matrix_t &c\)](#)
- [matrix_t vec2diag \(vector_t &\)](#)
- bool [matrixDivideScalar \(matrix_t &m, double s\)](#)
- bool [matrixFill \(matrix_t &m, double f\)](#)
- bool [matrixSubtract \(matrix_t m1, matrix_t m2, matrix_t &result\)](#)
- bool [matrixTranspose \(matrix_t in, matrix_t &out\)](#)
- bool [matrixGetDiag \(matrix_t m, vector_t &d\)](#)
- bool [matrixSetDiag \(matrix_t &m, vector_t d\)](#)
- bool [matrixComputeCovariance \(matrix_t X, matrix_t &covMatrix, matrix_t &corMatrix\)](#)
- bool [matrixRead \(string mFilename, matrix_t &m, vector< string > &variableNames\)](#)
- bool [matrixWrite \(matrix_t m, string mFilename, vector< string > variableNames\)](#)
- bool [matrixSums \(matrix_t m, vector_t &sums, int dim\)](#)
- bool [matrixExtractRowColIdx \(matrix_t m, intvec_t rowIdx, intvec_t colIdx, matrix_t &nM\)](#)
- bool [matrixGetTrace \(matrix_t m, double &t\)](#)
- bool [matrixElementWiseMultiply \(matrix_t m, matrix_t n, matrix_t &out\)](#)
- bool [matrixMultiplyScalar \(matrix_t &m, double s\)](#)
- bool [matrixAdd \(matrix_t m1, matrix_t m2, matrix_t &result\)](#)

- bool `matrixComputeNodeDegrees (matrix_t a, vector_t &ad)`
- bool `matrixConnectivityThreshold (matrix_t &m, double t, bool binary)`
- bool `reportNumericSummaryStats ()`
- bool `vectorSummary (vector_t values, vector_t &summary)`
- bool `rankByRegression (RegressionRankType rankType, rankedlist_t &ranks, RegressionRankResults &results)`
- pair< double, double > `fitModel (Model *m)`
- bool `numericLowValueFilter (double percentile, boolevec_t &varFlags)`
- bool `numericVarianceFilter (double percentile, boolevec_t &varFlags)`
- bool `quantile (vector_t values, double percentile, double &percentileValue)`
- bool `numericMeanCenter ()`
- bool `numericStandardize ()`
- bool `getNumericCaseControl (int varIndex, vector_t &cases, vector_t &controls)`
- bool `tTest (int varIndex, double &t)`
- bool `zTest (int varIndex, double &z)`
- bool `realnum (double)`
- long double `factorial (int)`
- double `normdist (double)`
- double `ltqnorm (double)`
- double `chi2x2 (double, double, double, double)`
- double `chi2x2 (table_t)`
- double `chi2x2 (matrix_t)`
- double `chiTable (table_t)`
- double `chiprobP (double, double)`
- double `symTable (table_t)`
- double `inverse_chiprob (double, double)`
- double `gammp (double a, double x)`
- void `gser (double *gamser, double a, double x, double *gln)`
- void `gcf (double *gammcf, double a, double x, double *gln)`
- double `gammln (double xx)`
- double `rnorm ()`
- void `lubksb (vector< vector< double > > &a, vector< int > &indx, vector< double > &b)`
- void `ludcmp (vector< vector< double > > &a, vector< int > &indx, double &d)`
- vector< vector< double > > `inverse (vector< vector< double > > &m)`
- vector< double > `eigenvalues (vector< vector< double > > &a)`
- void `tred2 (vector< vector< double > > &, vector< double > &, vector< double > &)`
- void `tqli (vector< double > &d, vector< double > &e, vector< vector< double > > &z)`
- `Eigen eigenvectors (vector< vector< double > > &a)`
- void `EV_tred2 (vector< vector< double > > &, vector< double > &, vector< double > &)`
- void `EV_tqli (vector< double > &d, vector< double > &e, vector< vector< double > > &z)`
- vector< vector< double > > `svd_inverse (vector< vector< double > > &, bool &)`
- bool `svd (matrix_t &, vector_t &, matrix_t &)`
- bool `svdcmp (vector< vector< double > > &, vector< double > &, vector< vector< double > > &)`
- void `svbksb (vector< vector< double > > &u, vector< double > &w, vector< vector< double > > &v, vector< double > &b, vector< double > &x)`
- vector< vector< double > > `msqrt (vector< vector< double > > &u)`
- double `qromb (double func(const double), double a, double b)`
- void `polint (vector_t &xa, vector_t &ya, const double x, double &y, double &dy)`
- double `trapzd (double func(const double), const double a, const double b, const int n)`
- void `svdvar (vector< vector< double > > &v, vector< double > &w, vector< vector< double > > &cvm)`
- int `pca (matrix_t &x, boolmatrix_t &mask, vector_t &p, matrix_t &s, matrix_t &v, bool)`
- double `pythag (const double a, const double b)`

- double **betacf** (const double **a**, const double **b**, const double **x**)
- double **betai** (const double **a**, const double **b**, const double **x**)
- double **pF** (const double **F**, const int **df1**, const int **df2**)
- double **pT** (const double **T**, const double **df**)

7.192.1 Function Documentation

7.192.1.1 betacf()

```
double betacf (
    const double a,
    const double b,
    const double x )
```

Definition at line 258 of file stats.cpp.

7.192.1.2 betai()

```
double betai (
    const double a,
    const double b,
    const double x )
```

Definition at line 245 of file stats.cpp.

7.192.1.3 chi2x2() [1/3]

```
double chi2x2 (
    double ,
    double ,
    double ,
    double )
```

Definition at line 1232 of file stats.cpp.

7.192.1.4 chi2x2() [2/3]

```
double chi2x2 (
    table_t   )
```

Definition at line 1224 of file stats.cpp.

7.192.1.5 chi2x2() [3/3]

```
double chi2x2 (
    matrix_t  )
```

Definition at line 1228 of file stats.cpp.

7.192.1.6 chiprobP()

```
double chiprobP (
    double ,
    double  )
```

Definition at line 74 of file stats.cpp.

7.192.1.7 chiTable()

```
double chiTable (
    table_t   )
```

Definition at line 1192 of file stats.cpp.

7.192.1.8 eigenvalues()

```
vector<double> eigenvalues (
    vector< vector< double > > & a )
```

Definition at line 325 of file stats.cpp.

7.192.1.9 eigenvectors()

```
Eigen eigenvectors (
    vector< vector< double > > & a )
```

Definition at line 469 of file stats.cpp.

7.192.1.10 EV_tqli()

```
void EV_tqli (
    vector< double > & d,
    vector< double > & e,
    vector< vector< double > > & z )
```

Definition at line 556 of file stats.cpp.

7.192.1.11 EV_tred2()

```
void EV_tred2 (
    vector< vector< double > > & ,
    vector< double > & ,
    vector< double > & )
```

Definition at line 486 of file stats.cpp.

7.192.1.12 factorial()

```
long double factorial (
    int )
```

Definition at line 52 of file stats.cpp.

7.192.1.13 fitModel()

```
pair<double, double> fitModel (
    Model * m )
```

Definition at line 2021 of file stats.cpp.

7.192.1.14 gammln()

```
double gammln (
    double xx )
```

Definition at line 114 of file stats.cpp.

7.192.1.15 gammp()

```
double gammp (
    double a,
    double x )
```

7.192.1.16 gcf()

```
void gcf (
    double * gammcf,
    double a,
    double x,
    double * gln )
```

7.192.1.17 getNumericCaseControl()

```
bool getNumericCaseControl (
    int varIndex,
    vector_t & cases,
    vector_t & controls )
```

Definition at line 2187 of file stats.cpp.

7.192.1.18 gser()

```
void gser (
    double * gamser,
    double a,
    double x,
    double * gln )
```

7.192.1.19 inverse()

```
vector< vector<double> > inverse (
    vector< vector< double > > & m )
```

Definition at line 296 of file stats.cpp.

7.192.1.20 inverse_chiprob()

```
double inverse_chiprob (
    double ,
    double )
```

Definition at line 93 of file stats.cpp.

7.192.1.21 ltqnorm()

```
double ltqnorm (
    double )
```

Definition at line 189 of file stats.cpp.

7.192.1.22 lubksb()

```
void lubksb (
    vector< vector< double > > & a,
    vector< int > & indx,
    vector< double > & b )
```

Definition at line 920 of file stats.cpp.

7.192.1.23 ludcmp()

```
void ludcmp (
    vector< vector< double > > & a,
    vector< int > & indx,
    double & d )
```

Definition at line 869 of file stats.cpp.

7.192.1.24 matrixAdd()

```
bool matrixAdd (
    matrix_t m1,
    matrix_t m2,
    matrix_t & result )
```

Definition at line 1478 of file stats.cpp.

7.192.1.25 matrixComputeCovariance()

```
bool matrixComputeCovariance (
    matrix_t X,
    matrix_t & covMatrix,
    matrix_t & corMatrix )
```

Definition at line 1407 of file stats.cpp.

7.192.1.26 matrixComputeNodeDegrees()

```
bool matrixComputeNodeDegrees (
    matrix_t a,
    vector_t & ad )
```

Definition at line 1687 of file stats.cpp.

7.192.1.27 matrixConnectivityThreshold()

```
bool matrixConnectivityThreshold (
    matrix_t & m,
    double t,
    bool binary )
```

Definition at line 1748 of file stats.cpp.

7.192.1.28 matrixDivideScalar()

```
bool matrixDivideScalar (
    matrix_t & m,
    double s )
```

Definition at line 1468 of file stats.cpp.

7.192.1.29 matrixElementWiseMultiply()

```
bool matrixElementWiseMultiply (
    matrix_t m,
    matrix_t n,
    matrix_t & out )
```

Definition at line 1727 of file stats.cpp.

7.192.1.30 matrixExtractRowColIdx()

```
bool matrixExtractRowColIdx (
    matrix_t m,
    intvec_t rowIdx,
    intvec_t colIdx,
    matrix_t & nm )
```

Definition at line 1703 of file stats.cpp.

7.192.1.31 matrixFill()

```
bool matrixFill (
    matrix_t & m,
    double f )
```

Definition at line 1448 of file stats.cpp.

7.192.1.32 matrixGetDiag()

```
bool matrixGetDiag (
    matrix_t m,
    vector_t & d )
```

Definition at line 1538 of file stats.cpp.

7.192.1.33 matrixGetTrace()

```
bool matrixGetTrace (
    matrix_t m,
    double & t )
```

Definition at line 1719 of file stats.cpp.

7.192.1.34 matrixMultiplyScalar()

```
bool matrixMultiplyScalar (
    matrix_t & m,
    double s )
```

Definition at line 1458 of file stats.cpp.

7.192.1.35 matrixRead()

```
bool matrixRead (
    string mFilename,
    matrix_t & m,
    vector< string > & variableNames )
```

Definition at line 1553 of file stats.cpp.

7.192.1.36 matrixSetDiag()

```
bool matrixSetDiag (
    matrix_t & m,
    vector_t d )
```

Definition at line 1546 of file stats.cpp.

7.192.1.37 matrixSubtract()

```
bool matrixSubtract (
    matrix_t m1,
    matrix_t m2,
    matrix_t & result )
```

Definition at line 1501 of file stats.cpp.

7.192.1.38 matrixSums()

```
bool matrixSums (
    matrix_t m,
    vector_t & sums,
    int dim )
```

Definition at line 1667 of file stats.cpp.

7.192.1.39 matrixTranspose()

```
bool matrixTranspose (
    matrix_t in,
    matrix_t & out )
```

Definition at line 1524 of file stats.cpp.

7.192.1.40 matrixWrite()

```
bool matrixWrite (
    matrix_t m,
    string mFilename,
    vector< string > variableNames )
```

Definition at line 1626 of file stats.cpp.

7.192.1.41 msqrt()

```
vector<vector<double> > msqrt (
    vector< vector< double > > & u )
```

Definition at line 827 of file stats.cpp.

7.192.1.42 multMatrix()

```
void multMatrix (
    matrix_t & a,
    matrix_t & b,
    matrix_t & c )
```

7.192.1.43 normdist()

```
double normdist (
    double )
```

Definition at line 60 of file stats.cpp.

7.192.1.44 numericLowValueFilter()

```
bool numericLowValueFilter (
    double percentile,
    boolevec_t & varFlags )
```

Definition at line 2050 of file stats.cpp.

7.192.1.45 numericMeanCenter()

```
bool numericMeanCenter ( )
```

Definition at line 2151 of file stats.cpp.

7.192.1.46 numericStandardize()

```
bool numericStandardize ( )
```

Definition at line 2165 of file stats.cpp.

7.192.1.47 numericVarianceFilter()

```
bool numericVarianceFilter (
    double percentile,
    boolevec_t & varFlags )
```

Definition at line 2093 of file stats.cpp.

7.192.1.48 pca()

```
int pca (
    matrix_t & x,
    boolmatrix_t & mask,
    vector_t & p,
    matrix_t & s,
    matrix_t & v,
    bool )
```

Definition at line 1254 of file stats.cpp.

7.192.1.49 pF()

```
double pF (
    const double F,
    const int df1,
    const int df2 )
```

Definition at line 241 of file stats.cpp.

7.192.1.50 polint()

```
void polint (
    vector_t & xa,
    vector_t & ya,
    const double x,
    double & y,
    double & dy )
```

Definition at line 635 of file stats.cpp.

7.192.1.51 pT()

```
double pT (
    const double T,
    const double df )
```

Definition at line 219 of file stats.cpp.

7.192.1.52 pythag()

```
double pythag (
    const double a,
    const double b )
```

Definition at line 1132 of file stats.cpp.

7.192.1.53 qromb()

```
double qromb (
    double funcconst double,
    double a,
    double b )
```

Definition at line 611 of file stats.cpp.

7.192.1.54 quantile()

```
bool quantile (
    vector_t values,
    double percentile,
    double & percentileValue )
```

Definition at line 2136 of file stats.cpp.

7.192.1.55 rankByRegression()

```
bool rankByRegression (
    RegressionRankType rankType,
    rankedlist_t & ranks,
    RegressionRankResults & results )
```

Definition at line 1847 of file stats.cpp.

7.192.1.56 realnum()

```
bool realnum (
    double )
```

Definition at line 44 of file stats.cpp.

7.192.1.57 reportNumericSummaryStats()

```
bool reportNumericSummaryStats ( )
```

Definition at line 1766 of file stats.cpp.

7.192.1.58 rnorm()

```
double rnorm ( )
```

Definition at line 1395 of file stats.cpp.

7.192.1.59 sizeMatrix()

```
void sizeMatrix (
    matrix_t & ,
    int ,
    int )
```

Definition at line 1598 of file helper.cpp.

7.192.1.60 sizeTable()

```
void sizeTable (
    table_t & ,
    int ,
    int )
```

Definition at line 1612 of file helper.cpp.

7.192.1.61 svbksb()

```
void svbksb (
    vector< vector< double > > & u,
    vector< double > & w,
    vector< vector< double > > & v,
    vector< double > & b,
    vector< double > & x )
```

Definition at line 707 of file stats.cpp.

7.192.1.62 svd()

```
bool svd (
    matrix_t & ,
    vector_t & ,
    matrix_t & )
```

Definition at line 743 of file stats.cpp.

7.192.1.63 svd_inverse()

```
vector< vector<double> > svd_inverse (
    vector< vector< double > > & ,
    bool & )
```

Definition at line 778 of file stats.cpp.

7.192.1.64 svdcmp()

```
bool svdcmp (
    vector< vector< double > > & ,
    vector< double > & ,
    vector< vector< double > > & )
```

Definition at line 943 of file stats.cpp.

7.192.1.65 svdvar()

```
void svdvar (
    vector< vector< double > > & v,
    vector< double > & w,
    vector< vector< double > > & cvm )
```

Definition at line 685 of file stats.cpp.

7.192.1.66 symTable()

```
double symTable (
    table_t    )
```

Definition at line 1171 of file stats.cpp.

7.192.1.67 tqli()

```
void tqli (
    vector< double > & d,
    vector< double > & e,
    vector< vector< double > > & z )
```

Definition at line 413 of file stats.cpp.

7.192.1.68 trapzd()

```
double trapzd (
    double   funcconst double,
    const double a,
    const double b,
    const int n )
```

Definition at line 666 of file stats.cpp.

7.192.1.69 tred2()

```
void tred2 (
    vector< vector< double > > & ,
    vector< double > & ,
    vector< double > & )
```

Definition at line 341 of file stats.cpp.

7.192.1.70 tTest()

```
bool tTest (
    int varIndex,
    double & t )
```

Definition at line 2235 of file stats.cpp.

7.192.1.71 vec2diag()

```
matrix_t vec2diag (
    vector_t & )
```

Definition at line 1387 of file stats.cpp.

7.192.1.72 vectorSummary()

```
bool vectorSummary (
    vector_t values,
    vector_t & summary )
```

Definition at line 1823 of file stats.cpp.

7.192.1.73 zTest()

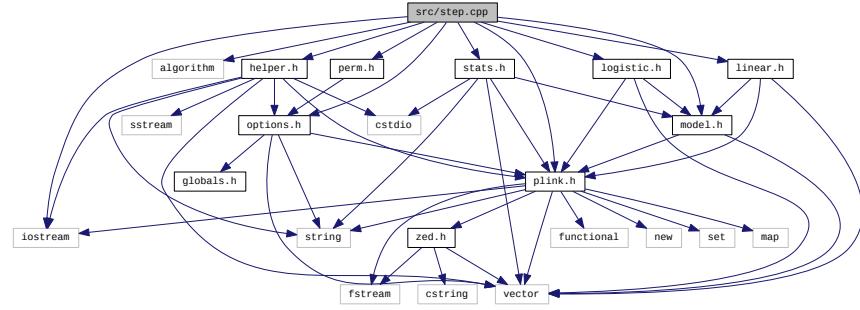
```
bool zTest (
    int varIndex,
    double & z )
```

Definition at line 2268 of file stats.cpp.

7.193 src/step.cpp File Reference

```
#include <iostream>
#include <algorithm>
#include "plink.h"
#include "helper.h"
#include "options.h"
#include "perm.h"
#include "model.h"
#include "linear.h"
#include "logistic.h"
#include "stats.h"
```

Include dependency graph for step.cpp:

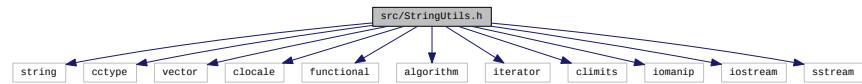


7.194 src/StringUtils.h File Reference

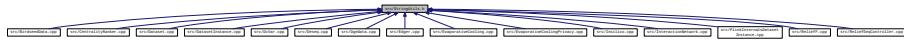
Various string-related utilities.

```
#include <string>
#include <cctype>
#include <vector>
#include <clocale>
#include <functional>
#include <algorithm>
#include <iterator>
#include <climits>
#include <iomanip>
#include <iostream>
#include <sstream>
```

Include dependency graph for StringUtils.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `insilico::is_classified< Type, charT >`
- class `insilico::do_to_upper< charT >`
- class `insilico::do_to_lower< charT >`

Namespaces

- `insilico`

Functions

- template<typename stringT >
stringT `insilico::trim_left` (const stringT &s, const std::locale &loc=std::locale())
- template<typename stringT >
stringT `insilico::trim_right` (const stringT &s, const std::locale &loc=std::locale())
- template<typename stringT >
stringT `insilico::trim` (const stringT &s, const std::locale &loc=std::locale())
- template<typename Container , typename stringT >
void `insilico::split` (Container &cont, const stringT &s, const std::locale &loc=std::locale())
- template<typename Container , typename stringT >
void `insilico::split` (Container &cont, const stringT &s, const stringT &delim)
- template<typename Container , typename stringT , typename Pred >
void `insilico::split_if` (Container &cont, const stringT &s, const Pred &pred)
- template<typename It , typename stringT >
stringT `insilico::join` (const It &begin, const It &end, const stringT &delim)
- template<typename stringT >
stringT `insilico::to_upper` (const stringT &str, const std::locale &loc=std::locale())
- template<typename stringT >
stringT `insilico::to_lower` (const stringT &str, const std::locale &loc=std::locale())
- std::string `insilico::trim_left` (const char *s, const std::locale &loc=std::locale())
- std::wstring `insilico::trim_left` (const wchar_t *s, const std::locale &loc=std::locale())
- std::string `insilico::trim_right` (const char *s, const std::locale &loc=std::locale())
- std::wstring `insilico::trim_right` (const wchar_t *s, const std::locale &loc=std::locale())
- std::string `insilico::trim` (const char *s, const std::locale &loc=std::locale())
- std::wstring `insilico::trim` (const wchar_t *s, const std::locale &loc=std::locale())
- template<typename Container >
void `insilico::split` (Container &cont, const char *s, const std::locale &loc=std::locale())
- template<typename Container >
void `insilico::split` (Container &cont, const wchar_t *s, const std::locale &loc=std::locale())
- template<typename Container >
void `insilico::split` (Container &cont, const std::string &s, const char *delim)

- template<typename Container >
void **insilico::split** (Container &cont, const char *s, const std::string &delim)
- template<typename Container >
void **insilico::split** (Container &cont, const char *s, const char *delim)
- template<typename Container >
void **insilico::split** (Container &cont, const std::wstring &s, const wchar_t *delim)
- template<typename Container >
void **insilico::split** (Container &cont, const wchar_t *s, const std::wstring &delim)
- template<typename Container >
void **insilico::split** (Container &cont, const wchar_t *s, const wchar_t *delim)
- template<typename Container , typename Pred >
void **insilico::split_if** (Container &cont, const char *s, const Pred &pred)
- template<typename Container , typename Pred >
void **insilico::split_if** (Container &cont, const wchar_t *s, const Pred &pred)
- template<typename It >
std::string **insilico::join** (const It &begin, const It &end, const char *delim)
- template<typename It >
std::wstring **insilico::join** (const It &begin, const It &end, const wchar_t *delim)
- std::string **insilico::to_upper** (const char *s, const std::locale &loc=std::locale())
- std::wstring **insilico::to_upper** (const wchar_t *s, const std::locale &loc=std::locale())
- std::string **insilico::to_lower** (const char *s, const std::locale &loc=std::locale())
- std::wstring **insilico::to_lower** (const wchar_t *s, const std::locale &loc=std::locale())
- template<typename T >
std::string **insilico::get_bits** (T value)
- template<typename T >
std::string **insilico::zeroPadNumber** (T num, int padSize)

7.194.1 Detailed Description

Various string-related utilities.

This is originally from Nate Barney circa Moore Lab days 2003-2007. His function naming follows lowercase with underscores style, while my additions are camelCase.

Author

Bill White, Nate Barney

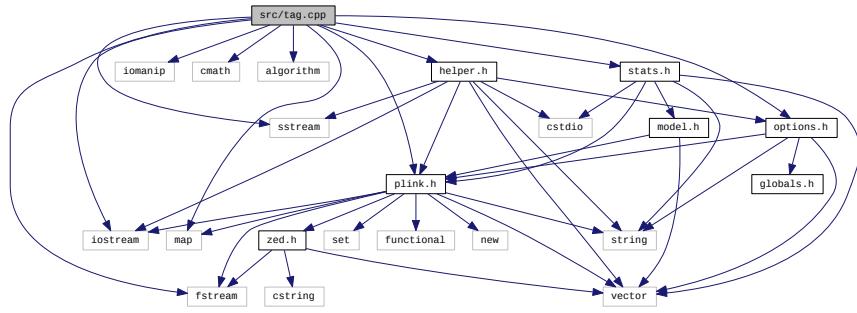
Version

1.0

Contact: bill.c.white@gmail.com Created on: 10/7/04

7.195 src/tag.cpp File Reference

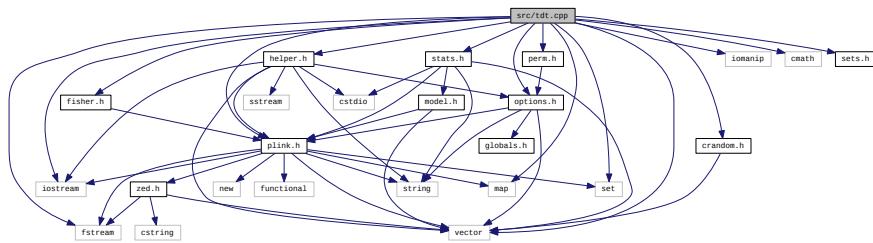
```
#include <iostream>
#include <fstream>
#include <sstream>
#include <iomanip>
#include <cmath>
#include <algorithm>
#include <map>
#include "plink.h"
#include "stats.h"
#include "helper.h"
#include "options.h"
Include dependency graph for tag.cpp:
```



7.196 src/tdt.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <map>
#include <vector>
#include <set>
#include <cmath>
#include "plink.h"
#include "options.h"
#include "helper.h"
#include "crandom.h"
#include "sets.h"
#include "perm.h"
#include "fisher.h"
#include "stats.h"
```

Include dependency graph for tdt.cpp:

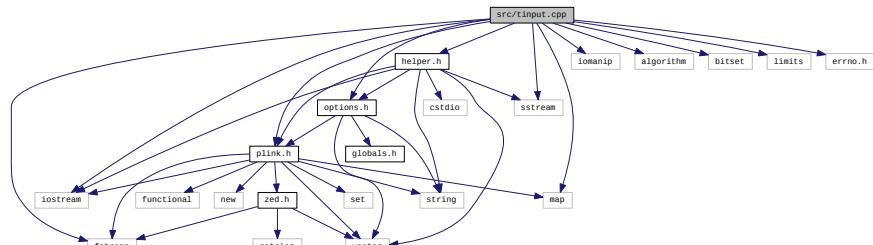


7.197 src/tinput.cpp File Reference

```

#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <map>
#include <algorithm>
#include <bitset>
#include <limits>
#include <errno.h>
#include "plink.h"
#include "options.h"
#include "helper.h"
  
```

Include dependency graph for tinput.cpp:



Variables

- ofstream [LOG](#)

7.197.1 Variable Documentation

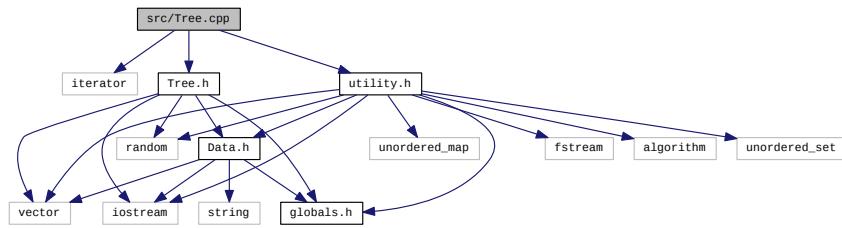
7.197.1.1 LOG

`ofstream LOG`

Definition at line 81 of file `inbbox.cpp`.

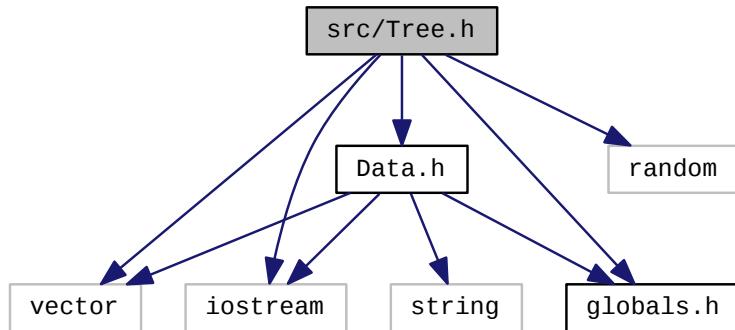
7.198 src/Tree.cpp File Reference

```
#include <iterator>
#include "Tree.h"
#include "utility.h"
Include dependency graph for Tree.cpp:
```

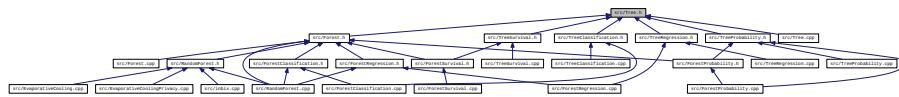


7.199 src/Tree.h File Reference

```
#include <vector>
#include <random>
#include <iostream>
#include "globals.h"
#include "Data.h"
Include dependency graph for Tree.h:
```



This graph shows which files directly or indirectly include this file:



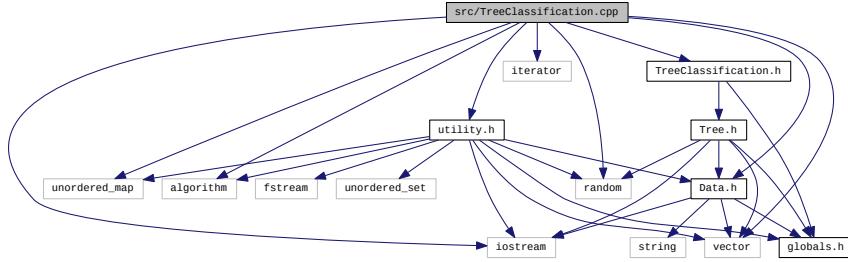
Classes

- class Tree

7.200 src/TreeClassification.cpp File Reference

```
#include <unordered_map>
#include <random>
#include <algorithm>
#include <iostream>
#include <iterator>
#include <vector>
#include "TreeClassification.h"
#include "utility.h"
#include "Data.h"
```

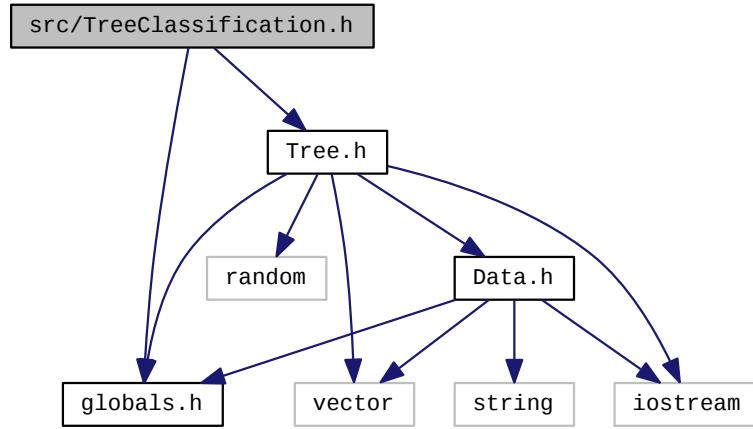
Include dependency graph for TreeClassification.cpp:



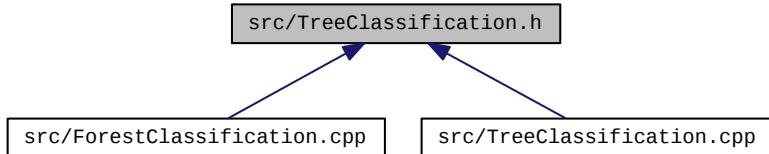
7.201 src/TreeClassification.h File Reference

```
#include "globals.h"  
#include "Tree.h"
```

Include dependency graph for TreeClassification.h:



This graph shows which files directly or indirectly include this file:



Classes

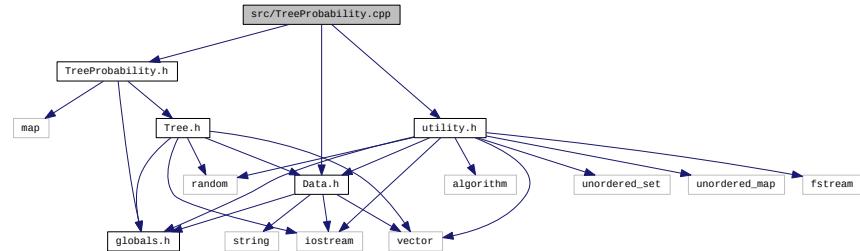
- class [TreeClassification](#)

7.202 src/TreeProbability.cpp File Reference

```
#include "TreeProbability.h"
#include "utility.h"
```

```
#include "Data.h"
```

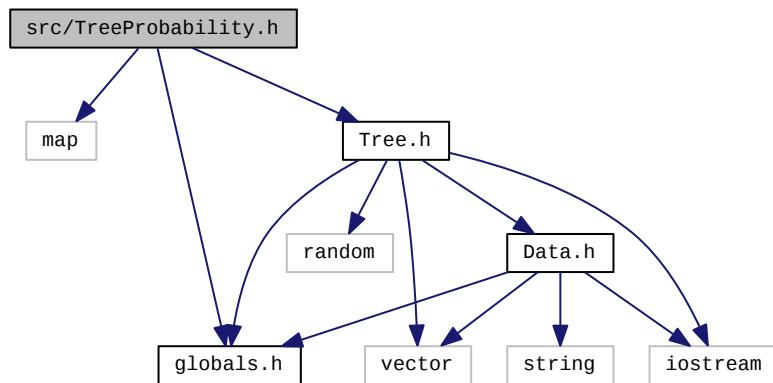
Include dependency graph for TreeProbability.cpp:



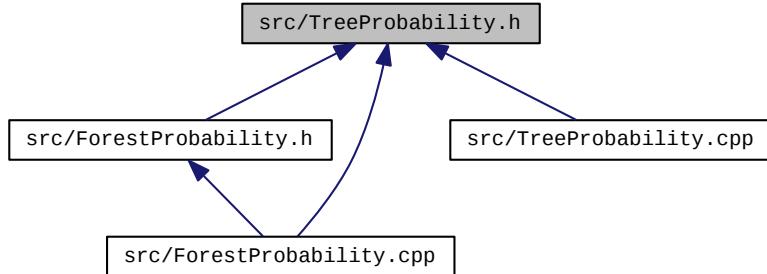
7.203 src/TreeProbability.h File Reference

```
#include <map>
#include "globals.h"
#include "Tree.h"
```

Include dependency graph for TreeProbability.h:



This graph shows which files directly or indirectly include this file:



Classes

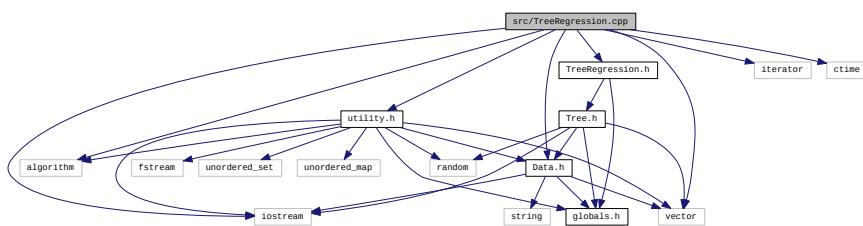
- class [TreeProbability](#)

7.204 src/TreeRegression.cpp File Reference

```

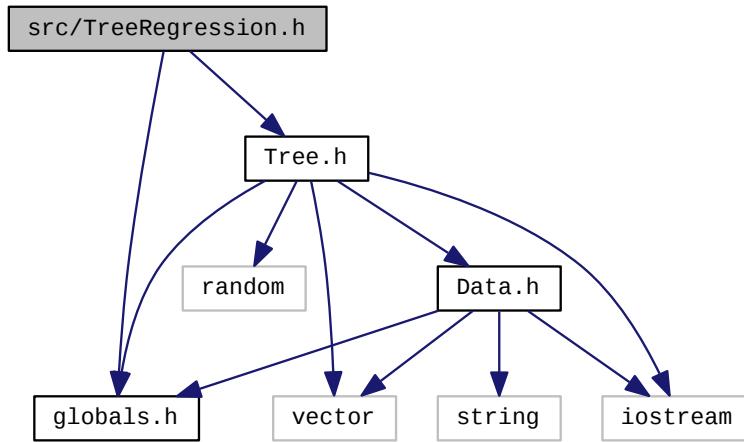
#include <algorithm>
#include <iostream>
#include <iterator>
#include <vector>
#include <ctime>
#include "utility.h"
#include "TreeRegression.h"
#include "Data.h"
  
```

Include dependency graph for `TreeRegression.cpp`:

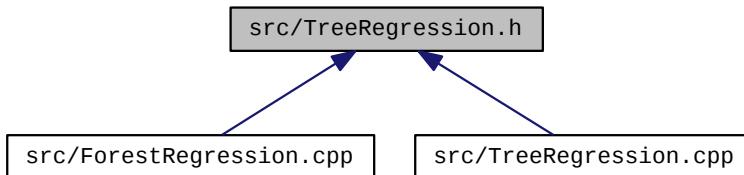


7.205 src/TreeRegression.h File Reference

```
#include "globals.h"
#include "Tree.h"
Include dependency graph for TreeRegression.h:
```



This graph shows which files directly or indirectly include this file:

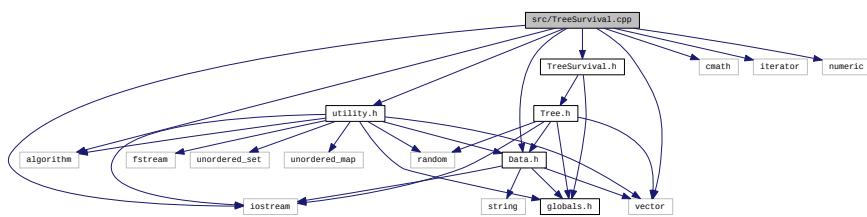


Classes

- class [TreeRegression](#)

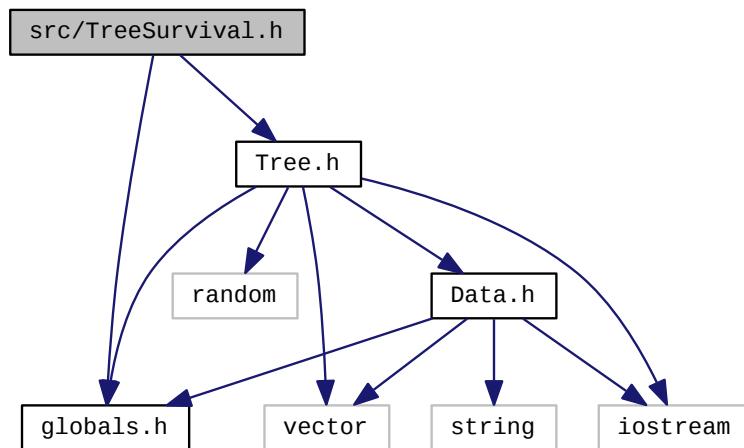
7.206 src/TreeSurvival.cpp File Reference

```
#include <algorithm>
#include <cmath>
#include <iostream>
#include <iterator>
#include <numeric>
#include <vector>
#include "utility.h"
#include "TreeSurvival.h"
#include "Data.h"
Include dependency graph for TreeSurvival.cpp:
```

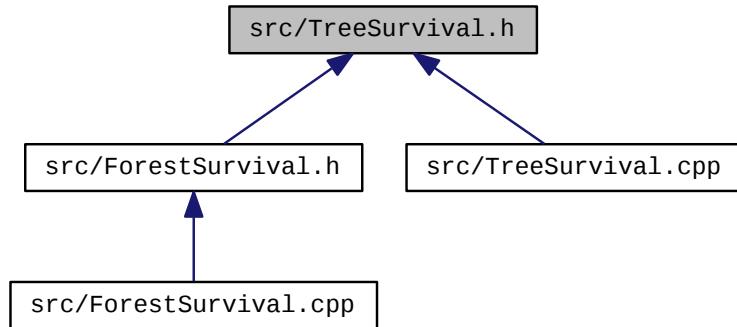


7.207 src/TreeSurvival.h File Reference

```
#include "globals.h"
#include "Tree.h"
Include dependency graph for TreeSurvival.h:
```



This graph shows which files directly or indirectly include this file:



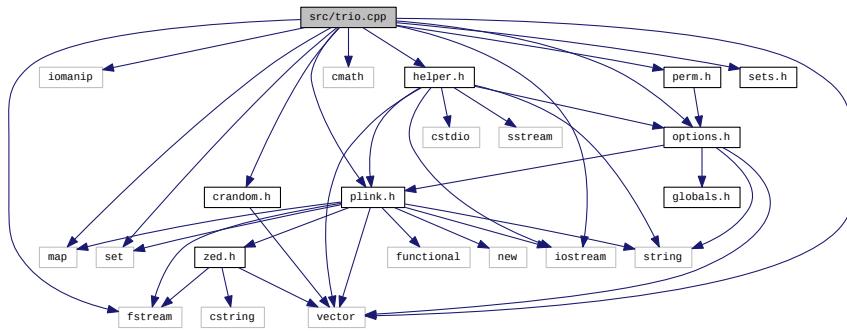
Classes

- class [TreeSurvival](#)

7.208 src/trio.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <map>
#include <vector>
#include <set>
#include <cmath>
#include "plink.h"
#include "options.h"
#include "helper.h"
#include "crandom.h"
#include "sets.h"
#include "perm.h"
```

Include dependency graph for trio.cpp:



Functions

- `string gprint (int l, bool s1, bool s2)`
- `void addParent (Family *f, Individual *person)`
- `void addPerson (Family *f, Individual *person)`

Variables

- `Plink * PP`

7.208.1 Function Documentation

7.208.1.1 addParent()

```
void addParent (
    Family * f,
    Individual * person )
```

Definition at line 36 of file `trio.cpp`.

7.208.1.2 addPerson()

```
void addPerson (
    Family * f,
    Individual * person )
```

Definition at line 86 of file `trio.cpp`.

7.208.1.3 gprint()

```
string gprint (
    int l,
    bool s1,
    bool s2 )
```

Definition at line 1156 of file trio.cpp.

7.208.2 Variable Documentation

7.208.2.1 PP

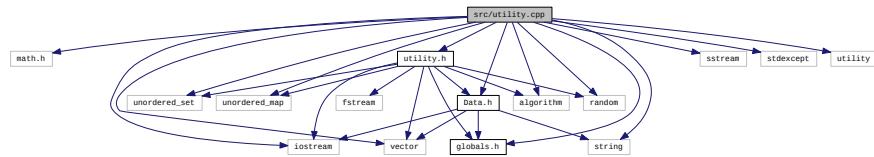
[Plink*](#) PP

Definition at line 85 of file inbix.cpp.

7.209 src/utility.cpp File Reference

```
#include <math.h>
#include <iostream>
#include <sstream>
#include <unordered_set>
#include <unordered_map>
#include <algorithm>
#include <random>
#include <stdexcept>
#include <string>
#include <utility>
#include <vector>
#include "utility.h"
#include "globals.h"
#include "Data.h"
```

Include dependency graph for utility.cpp:



Functions

- void `equalSplit` (`std::vector< uint > &result, uint start, uint end, uint num_parts)`

Split sequence start..end in num_parts parts with sizes as equal as possible.
- void `loadDoubleVectorFromFile` (`std::vector< double > &result, std::string filename)`

Read a double vector from text file.
- void `drawWithoutReplacementFisherYates` (`std::vector< size_t > &result, std::mt19937_64 &random_number_generator, size_t max, std::vector< size_t > &skip, size_t num_samples)`
- void `drawWithoutReplacementSkip` (`std::vector< size_t > &result, std::mt19937_64 &random_number_generator, size_t max, std::vector< size_t > &skip, size_t num_samples)`

Draw random numbers in a range without replacement and skip values.
- void `drawWithoutReplacementSimple` (`std::vector< size_t > &result, std::mt19937_64 &random_number_generator, size_t max, std::vector< size_t > &skip, size_t num_samples)`

Simple algorithm for sampling without replacement, faster for smaller num_samples.
- void `drawWithoutReplacementKnuth` (`std::vector< size_t > &result, std::mt19937_64 &random_number_generator, size_t max, std::vector< size_t > &skip, size_t num_samples)`

Knuth's algorithm for sampling without replacement, faster for larger num_samples Idea from Knuth 1985, The Art of Computer Programming, Vol.
- void `drawWithoutReplacementWeighted` (`std::vector< size_t > &result, std::mt19937_64 &random_number_generator, std::vector< size_t > &indices, size_t num_samples, std::vector< double > &weights)`

Draw random numbers without replacement and with weighted probabilities from vector of indices.
- void `drawWithoutReplacementWeighted` (`std::vector< size_t > &result, std::mt19937_64 &random_number_generator, size_t max_index, size_t num_samples, std::vector< double > &weights)`

Draw random numbers without replacement and with weighted probabilities from 0..n-1.
- double `mostFrequentValue` (`std::unordered_map< double, size_t > &class_count, std::mt19937_64 random_number_generator)`

Returns the most frequent value of a map with counts for the values.
- double `computeConcordanceIndex` (`Data *data, std::vector< double > &sum_chf, size_t dependent_varID, size_t status_varID, std::vector< size_t > &sample_IDs)`

Compute concordance index for given data and summed cumulative hazard function/estimate.
- `std::string uintToString` (`uint number)`

Convert a unsigned integer to string.
- `std::string beautifyTime` (`uint seconds)`

Beautify output of time.
- `size_t roundToNextMultiple` (`size_t value, uint multiple)`

Round up to next multiple of a number.
- void `splitString` (`std::vector< std::string > &result, std::string input, char split_char)`

Split string in parts separated by character.
- void `shuffleAndSplit` (`std::vector< size_t > &first_part, std::vector< size_t > &second_part, size_t n_all, size_t n_first, std::mt19937_64 random_number_generator)`

Create numbers from 0 to n_all-1, shuffle and split in two parts.
- `std::string checkUnorderedVariables` (`Data *data, std::vector< std::string > unordered_variable_names)`

Check if not too many factor levels and all values in unordered categorical variables are positive integers.
- `bool checkPositiveIntegers` (`std::vector< double > &all_values)`

Check if all values in double vector are positive integers.
- double `maxstatPValueLau92` (`double b, double minprop, double maxprop)`

Compute p-value for maximally selected rank statistics using Lau92 approximation See Lausen, B.
- double `maxstatPValueLau94` (`double b, double minprop, double maxprop, size_t N, std::vector< size_t > &m)`

Compute p-value for maximally selected rank statistics using Lau94 approximation See Lausen, B., Sauerbrei, W.

- double **maxstatPValueUnadjusted** (double **b**)
Compute unadjusted p-value for rank statistics.
- double **dstdnorm** (double **x**)
Standard normal density.
- double **pstdnorm** (double **x**)
Standard normal distribution.
- std::vector< double > **adjustPvalues** (std::vector< double > &unadjusted_pvalues)
Adjust p-values with Benjamini/Hochberg.
- std::vector< size_t > **orderInData** (**Data** *data, std::vector< size_t > &sampleIDs, size_t varID, bool decreasing)
*Get indices of sorted values, compute in-place of **Data**.*
- std::vector< double > **logrankScores** (std::vector< double > &time, std::vector< double > &status)
Compute Logrank scores for survival times.
- std::vector< double > **logrankScoresData** (**Data** *data, size_t time_varID, size_t status_varID, std::vector< size_t > &sampleIDs)
*Compute Logrank scores for survival times directly from **Data** object.*
- void **maxstat** (std::vector< double > &scores, std::vector< double > &x, std::vector< size_t > &indices, double &best_maxstat, double &best_split_value, double minprop, double maxprop)
Compute maximally selected rank statistics.
- void **maxstatInData** (std::vector< double > &scores, **Data** *data, std::vector< size_t > &sampleIDs, size_t varID, std::vector< size_t > &indices, double &best_maxstat, double &best_split_value, double minprop, double maxprop)
*Compute maximally selected rank statistics in **Data** object.*
- std::vector< size_t > **numSamplesLeftOfCutpoint** (std::vector< double > &x, std::vector< size_t > &indices)
Compute number of samples smaller or equal than each unique value in x.
- std::vector< size_t > **numSamplesLeftOfCutpointInData** (**Data** *data, std::vector< size_t > &sampleIDs, size_t varID, std::vector< size_t > &indices)
Compute number of samples smaller or equal than each unique value in data.

7.209.1 Function Documentation

7.209.1.1 **adjustPvalues()**

```
std::vector<double> adjustPvalues (
    std::vector< double > & unadjusted_pvalues )
```

Adjust p-values with Benjamini/Hochberg.

Parameters

<i>unadjusted_pvalues</i>	Unadjusted p-values (input)
<i>adjusted_pvalues</i>	Adjusted p-values (result)

Definition at line 465 of file utility.cpp.

7.209.1.2 `beautifyTime()`

```
std::string beautifyTime (
    uint seconds )
```

Beautify output of time.

Parameters

<code>seconds</code>	Time in seconds
----------------------	-----------------

Returns

Time in days, hours, minutes and seconds as string

Definition at line 310 of file utility.cpp.

7.209.1.3 `checkPositiveIntegers()`

```
bool checkPositiveIntegers (
    std::vector< double > & all_values )
```

Check if all values in double vector are positive integers.

Parameters

<code>all_values</code>	Double vector to check
-------------------------	------------------------

Returns

True if all values are positive integers

Definition at line 411 of file utility.cpp.

7.209.1.4 `checkUnorderedVariables()`

```
std::string checkUnorderedVariables (
    Data * data,
    std::vector< std::string > unordered_variable_names )
```

Check if not too many factor levels and all values in unordered categorical variables are positive integers.

Parameters

<i>data</i>	Pointer to data object
<i>unordered_variable_names</i>	Names of unordered variables

Returns

Error message, empty if no problem occurred

Definition at line 385 of file utility.cpp.

7.209.1.5 computeConcordanceIndex()

```
double computeConcordanceIndex (
    Data * data,
    std::vector< double > & sum_chf,
    size_t dependent_varID,
    size_t status_varID,
    std::vector< size_t > & sample_IDs )
```

Compute concordance index for given data and summed cumulative hazard function/estimate.

Parameters

<i>data</i>	Pointer to Data object
<i>sum_chf</i>	Summed chf over timepoints for each sample
<i>dependent_varID</i>	ID of dependent variable
<i>status_varID</i>	ID of status variable
<i>sample_IDs</i>	IDs of samples, for example OOB samples

Returns

concordance index

Definition at line 251 of file utility.cpp.

7.209.1.6 drawWithoutReplacementFisherYates()

```
void drawWithoutReplacementFisherYates (
    std::vector< size_t > & result,
    std::mt19937_64 & random_number_generator,
```

```
size_t max,
std::vector< size_t > & skip,
size_t num_samples )
```

Definition at line 99 of file utility.cpp.

7.209.1.7 drawWithoutReplacementKnuth()

```
void drawWithoutReplacementKnuth (
    std::vector< size_t > & result,
    std::mt19937_64 & random_number_generator,
    size_t max,
    std::vector< size_t > & skip,
    size_t num_samples )
```

Knuth's algorithm for sampling without replacement, faster for larger num_samples Idea from Knuth 1985, The Art of Computer Programming, Vol.

2, Sec. 3.4.2 Algorithm S

Parameters

<i>result</i>	Vector to add results to. Will not be cleaned before filling.
<i>random_number_generator</i>	Random number generator
<i>range_length</i>	Length of range. Interval to draw from: 0..max-1
<i>skip</i>	Values to skip
<i>num_samples</i>	Number of samples to draw

Definition at line 156 of file utility.cpp.

7.209.1.8 drawWithoutReplacementSimple()

```
void drawWithoutReplacementSimple (
    std::vector< size_t > & result,
    std::mt19937_64 & random_number_generator,
    size_t max,
    std::vector< size_t > & skip,
    size_t num_samples )
```

Simple algorithm for sampling without replacement, faster for smaller num_samples.

Parameters

<i>result</i>	Vector to add results to. Will not be cleaned before filling.
---------------	---

Parameters

<i>random_number_generator</i>	Random number generator
<i>range_length</i>	Length of range. Interval to draw from: 0..max-1
<i>skip</i>	Values to skip
<i>num_samples</i>	Number of samples to draw

Definition at line 131 of file utility.cpp.

7.209.1.9 drawWithoutReplacementSkip()

```
void drawWithoutReplacementSkip (
    std::vector< size_t > & result,
    std::mt19937_64 & random_number_generator,
    size_t range_length,
    std::vector< size_t > & skip,
    size_t num_samples )
```

Draw random numbers in a range without replacement and skip values.

Parameters

<i>result</i>	Vector to add results to. Will not be cleaned before filling.
<i>random_number_generator</i>	Random number generator
<i>range_length</i>	Length of range. Interval to draw from: 0..max-1
<i>skip</i>	Values to skip
<i>num_samples</i>	Number of samples to draw

Definition at line 121 of file utility.cpp.

7.209.1.10 drawWithoutReplacementWeighted() [1/2]

```
void drawWithoutReplacementWeighted (
    std::vector< size_t > & result,
    std::mt19937_64 & random_number_generator,
    std::vector< size_t > & indices,
    size_t num_samples,
    std::vector< double > & weights )
```

Draw random numbers without replacement and with weighted probabilities from vector of indices.

Parameters

<i>result</i>	Vector to add results to. Will not be cleaned before filling.
<i>random_number_generator</i>	Random number generator
<i>indices</i>	Vector with numbers to draw
<i>num_samples</i>	Number of samples to draw
<i>weights</i>	A weight for each element of indices

Definition at line 187 of file utility.cpp.

7.209.1.11 drawWithoutReplacementWeighted() [2/2]

```
void drawWithoutReplacementWeighted (
    std::vector< size_t > & result,
    std::mt19937_64 & random_number_generator,
    size_t max_index,
    size_t num_samples,
    std::vector< double > & weights )
```

Draw random numbers without replacement and with weighted probabilities from 0..n-1.

Parameters

<i>result</i>	Vector to add results to. Will not be cleaned before filling.
<i>random_number_generator</i>	Random number generator
<i>max_index</i>	Maximum index to draw
<i>num_samples</i>	Number of samples to draw
<i>weights</i>	A weight for each element of indices

Definition at line 207 of file utility.cpp.

7.209.1.12 dstdnorm()

```
double dstdnorm (
    double x )
```

Standard normal density.

Parameters

x	Quantile
---	----------

Returns

Standard normal density at quantile x

Definition at line 457 of file utility.cpp.

7.209.1.13 equalSplit()

```
void equalSplit (
    std::vector< uint > & result,
    uint start,
    uint end,
    uint num_parts )
```

Split sequence start..end in num_parts parts with sizes as equal as possible.

Parameters

<i>result</i>	Result vector of size num_parts+1. Ranges for the parts are then result[0]..result[1]-1, result[1]..result[2]-1, ..
<i>start</i>	minimum value
<i>end</i>	maximum value
<i>num_parts</i>	number of parts

Definition at line 45 of file utility.cpp.

7.209.1.14 loadDoubleVectorFromFile()

```
void loadDoubleVectorFromFile (
    std::vector< double > & result,
    std::string filename )
```

Read a double vector from text file.

Reads only the first line.

Parameters

<i>result</i>	Result vector of doubles with contents
<i>filename</i>	filename of input file

Definition at line 80 of file utility.cpp.

7.209.1.15 logrankScores()

```
std::vector<double> logrankScores (
    std::vector< double > & time,
    std::vector< double > & status )
```

Compute Logrank scores for survival times.

Parameters

<i>time</i>	Survival time
<i>status</i>	Censoring indicator

Returns

Logrank scores

Definition at line 500 of file utility.cpp.

7.209.1.16 logrankScoresData()

```
std::vector<double> logrankScoresData (
    Data * data,
    size_t time_varID,
    size_t status_varID,
    std::vector< size_t > sampleIDs )
```

Compute Logrank scores for survival times directly from [Data](#) object.

Parameters

<i>data</i>	Pointer to Data object
<i>time_varID</i>	variable ID for time column
<i>status_varID</i>	Variable ID for status column
<i>sampleIDs</i>	IDs of samples to include

Returns

Logrank scores

Definition at line 532 of file utility.cpp.

7.209.1.17 maxstat()

```
void maxstat (
    std::vector< double > & scores,
    std::vector< double > & x,
    std::vector< size_t > & indices,
    double & best_maxstat,
    double & best_split_value,
    double minprop,
    double maxprop )
```

Compute maximally selected rank statistics.

Parameters

<i>scores</i>	Scores for dependent variable (y)
<i>x</i>	Independent variable
<i>indices</i>	Ordering of x values
<i>best_maxstat</i>	Maximally selected statistic (output)
<i>best_split_value</i>	Split value for maximally selected statistic (output)
<i>minprop</i>	Minimal proportion of observations left of cutpoint
<i>maxprop</i>	Maximal proportion of observations left of cutpoint

Definition at line 569 of file utility.cpp.

7.209.1.18 maxstatInData()

```
void maxstatInData (
    std::vector< double > & scores,
    Data * data,
    std::vector< size_t > & sampleIDs,
    size_t varID,
    std::vector< size_t > & indices,
    double & best_maxstat,
    double & best_split_value,
    double minprop,
    double maxprop )
```

Compute maximally selected rank statistics in [Data](#) object.

Parameters

<i>scores</i>	Scores for dependent variable (y)
<i>data</i>	Data object
<i>sampleIDs</i>	IDs of samples to consider
<i>varID</i>	ID of variable to consider
<i>indices</i>	Ordering of x values

Parameters

<i>best_maxstat</i>	Maximally selected statistic (output)
<i>best_split_value</i>	Split value for maximally selected statistic (output)
<i>minprop</i>	Minimal proportion of observations left of cutpoint
<i>maxprop</i>	Maximal proportion of observations left of cutpoint

Definition at line 629 of file utility.cpp.

7.209.1.19 maxstatPValueLau92()

```
double maxstatPValueLau92 (
    double b,
    double minprop,
    double maxprop )
```

Compute p-value for maximally selected rank statistics using Lau92 approximation See Lausen, B.

& Schumacher, M. (1992). Biometrics 48, 73-85.

Parameters

<i>b</i>	Quantile
<i>minprop</i>	Minimal proportion of observations left of cutpoint
<i>maxprop</i>	Maximal proportion of observations left of cutpoint

Returns

p-value for quantile b

Definition at line 420 of file utility.cpp.

7.209.1.20 maxstatPValueLau94()

```
double maxstatPValueLau94 (
    double b,
    double minprop,
    double maxprop,
    size_t N,
    std::vector< size_t > & m )
```

Compute p-value for maximally selected rank statistics using Lau94 approximation See Lausen, B., Sauerbrei, W.

& Schumacher, M. (1994). Computational Statistics. 483-496.

Parameters

<i>b</i>	Quantile
<i>minprop</i>	Minimal proportion of observations left of cutpoint
<i>maxprop</i>	Maximal proportion of observations left of cutpoint
<i>N</i>	Number of observations
<i>m</i>	Vector with number of observations smaller or equal than cutpoint, sorted, only for unique cutpoints

Returns

p-value for quantile b

Definition at line 439 of file utility.cpp.

7.209.1.21 maxstatPValueUnadjusted()

```
double maxstatPValueUnadjusted (
    double b )
```

Compute unadjusted p-value for rank statistics.

Parameters

<i>b</i>	Quantile
----------	----------

Returns

p-value for quantile b

Definition at line 453 of file utility.cpp.

7.209.1.22 mostFrequentValue()

```
double mostFrequentValue (
    std::unordered_map< double, size_t > & class_count,
    std::mt19937_64 random_number_generator )
```

Returns the most frequent value of a map with counts for the values.

Returns a random class if counts are equal.

Parameters

<i>class_count</i>	Map with classes and counts
<i>random_number_generator</i>	Random number generator

Returns

Most frequent value

Definition at line 227 of file utility.cpp.

7.209.1.23 numSamplesLeftOfCutpoint()

```
std::vector<size_t> numSamplesLeftOfCutpoint (
    std::vector< double > & x,
    std::vector< size_t > & indices )
```

Compute number of samples smaller or equal than each unique value in x.

Parameters

<i>x</i>	Value vector
<i>indices</i>	Ordering of x

Returns

Vector of number of samples smaller or equal than each unique value in x

Definition at line 690 of file utility.cpp.

7.209.1.24 numSamplesLeftOfCutpointInData()

```
std::vector<size_t> numSamplesLeftOfCutpointInData (
    Data * data,
    std::vector< size_t > & sampleIDs,
    size_t varID,
    std::vector< size_t > & indices )
```

Compute number of samples smaller or equal than each unique value in data.

Parameters

<i>data</i>	Data object
<i>sampleIDs</i>	IDs of samples to consider
<i>varID</i>	ID of variable to consider
<i>indices</i>	Ordering of x

Returns

Vector of number of samples smaller or equal than each unique value in x

Definition at line 707 of file utility.cpp.

7.209.1.25 orderInData()

```
std::vector<size_t> orderInData (
    Data * data,
    std::vector< size_t > & sampleIDs,
    size_t varID,
    bool decreasing )
```

Get indices of sorted values, compute in-place of [Data](#).

Parameters

<i>data</i>	Data object
<i>sampleIDs</i>	IDs of samples to sort
<i>varID</i>	ID of variable to sort by
<i>decreasing</i>	Order decreasing

Returns

Indices of sorted values

Definition at line 484 of file utility.cpp.

7.209.1.26 pstdnorm()

```
double pstdnorm (
    double x )
```

Standard normal distribution.

Parameters

x	Quantile
---	----------

Returns

Standard normal distribution at quantile x

Definition at line 461 of file utility.cpp.

7.209.1.27 roundToNextMultiple()

```
size_t roundToNextMultiple (
    size_t value,
    uint multiple )
```

Round up to next multiple of a number.

Parameters

<i>value</i>	Value to be rounded up.
<i>multiple</i>	Number to multiply.

Returns

Rounded number

Definition at line 343 of file utility.cpp.

7.209.1.28 shuffleAndSplit()

```
void shuffleAndSplit (
    std::vector< size_t > & first_part,
    std::vector< size_t > & second_part,
    size_t n_all,
    size_t n_first,
    std::mt19937_64 random_number_generator )
```

Create numbers from 0 to n_all-1, shuffle and split in two parts.

Parameters

<i>first_part</i>	First part
<i>second_part</i>	Second part
<i>n_all</i>	Number elements
<i>n_first</i>	Number of elements of first part
<i>random_number_generator</i>	Random number generator

Definition at line 367 of file utility.cpp.

7.209.1.29 splitString()

```
void splitString (
    std::vector< std::string > & result,
    std::string input,
    char split_char )
```

Split string in parts separated by character.

Parameters

<i>result</i>	Splitted string
<i>input</i>	String to be splitted
<i>split_char</i>	Char to separate parts

Definition at line 357 of file utility.cpp.

7.209.1.30 uintToString()

```
std::string uintToString (
    uint number )
```

Convert a unsigned integer to string.

Parameters

<i>number</i>	Number to convert
---------------	-------------------

Returns

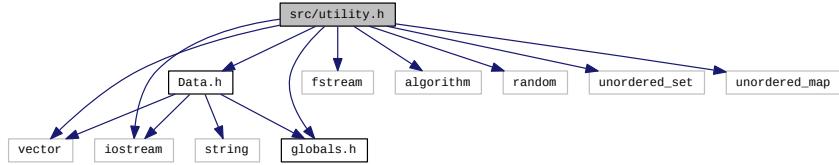
Converted number as string

Definition at line 300 of file utility.cpp.

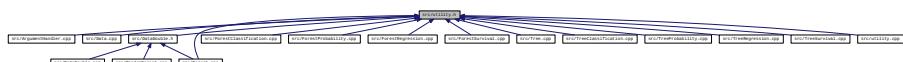
7.210 src/utility.h File Reference

```
#include <vector>
#include <iostream>
#include <fstream>
#include <algorithm>
#include <random>
#include <unordered_set>
#include <unordered_map>
#include "globals.h"
#include "Data.h"
Include dependency graph for utility.h:
```

Include dependency graph for utility.h:



This graph shows which files directly or indirectly include this file:



Functions

- void **equalSplit** (std::vector< uint > &result, uint start, uint end, uint num_parts)
Split sequence start..end in num_parts parts with sizes as equal as possible.
 - template<typename T>
void **saveVector1D** (std::vector< T > &vector, std::ofstream &file)
Write a 1d vector to filestream.
 - template<>
void **saveVector1D** (std::vector< bool > &vector, std::ofstream &file)
 - template<typename T>
void **readVector1D** (std::vector< T > &result, std::ifstream &file)
Read a 1d vector written by [saveVector1D\(\)](#) from filestream.
 - template<>
void **readVector1D** (std::vector< bool > &result, std::ifstream &file)
 - template<typename T>
void **saveVector2D** (std::vector< std::vector< T > > &vector, std::ofstream &file)

Write a 2d vector to filestream.

- template<typename T >
void **readVector2D** (std::vector< std::vector< T > > &result, std::ifstream &file)

Read a 2d vector written by [saveVector2D\(\)](#) from filestream.

- void **loadDoubleVectorFromFile** (std::vector< double > &result, std::string filename)

Read a double vector from text file.

- void **drawWithoutReplacementSkip** (std::vector< size_t > &result, std::mt19937_64 &random_number_generator, size_t range_length, std::vector< size_t > &skip, size_t num_samples)

Draw random numbers in a range without replacement and skip values.

- void **drawWithoutReplacementSimple** (std::vector< size_t > &result, std::mt19937_64 &random_number_generator, size_t max, std::vector< size_t > &skip, size_t num_samples)

Simple algorithm for sampling without replacement, faster for smaller num_samples.

- void **drawWithoutReplacementKnuth** (std::vector< size_t > &result, std::mt19937_64 &random_number_generator, size_t max, std::vector< size_t > &skip, size_t num_samples)

Knuth's algorithm for sampling without replacement, faster for larger num_samples Idea from Knuth 1985, The Art of Computer Programming, Vol.

- void **drawWithoutReplacementWeighted** (std::vector< size_t > &result, std::mt19937_64 &random_number_generator, std::vector< size_t > &indices, size_t num_samples, std::vector< double > &weights)

Draw random numbers without replacement and with weighted probabilities from vector of indices.

- void **drawWithoutReplacementWeighted** (std::vector< size_t > &result, std::mt19937_64 &random_number_generator, size_t max_index, size_t num_samples, std::vector< double > &weights)

Draw random numbers without replacement and with weighted probabilities from 0..n-1.

- template<typename T >
size_t **mostFrequentClass** (std::vector< T > &class_count, std::mt19937_64 random_number_generator)

Returns the most frequent class index of a vector with counts for the classes.

- double **mostFrequentValue** (std::unordered_map< double, size_t > &class_count, std::mt19937_64 random_number_generator)

Returns the most frequent value of a map with counts for the values.

- double **computeConcordanceIndex** (Data *data, std::vector< double > &sum_chf, size_t dependent_varID, size_t status_varID, std::vector< size_t > &sample_IDs)

Compute concordance index for given data and summed cumulative hazard function/estimate.

- std::string **uintToString** (uint number)

Convert a unsigned integer to string.

- std::string **beautifyTime** (uint seconds)

Beautify output of time.

- size_t **roundToNextMultiple** (size_t value, uint multiple)

Round up to next multiple of a number.

- void **splitString** (std::vector< std::string > &result, std::string input, char split_char)

Split string in parts separated by character.

- void **shuffleAndSplit** (std::vector< size_t > &first_part, std::vector< size_t > &second_part, size_t n_all, size_t n_first, std::mt19937_64 random_number_generator)

Create numbers from 0 to n_all-1, shuffle and split in two parts.

- std::string **checkUnorderedVariables** (Data *data, std::vector< std::string > unordered_variable_names)

Check if not too many factor levels and all values in unordered categorical variables are positive integers.

- bool **checkPositiveIntegers** (std::vector< double > &all_values)

Check if all values in double vector are positive integers.

- double **maxstatPValueLau92** (double b, double minprop, double maxprop)

Compute p-value for maximally selected rank statistics using Lau92 approximation See Lausen, B.

- double **maxstatPValueLau94** (double b, double minprop, double maxprop, size_t N, std::vector< size_t > &m)

- Compute p-value for maximally selected rank statistics using Lau94 approximation See Lausen, B., Sauerbrei, W.*
- double **maxstatPValueUnadjusted** (double b)

Compute unadjusted p-value for rank statistics.
 - double **dstdnorm** (double x)

Standard normal density.
 - double **pstdnorm** (double x)

Standard normal distribution.
 - std::vector< double > **adjustPvalues** (std::vector< double > &unadjusted_pvalues)

Adjust p-values with Benjamini/Hochberg.
 - template<typename T>
 std::vector< size_t > **order** (std::vector< T > &values, bool decreasing)

Get indices of sorted values.
 - std::vector< size_t > **orderInData** (Data *data, std::vector< size_t > &sampleIDs, size_t varID, bool decreasing)

Get indices of sorted values, compute in-place of Data.
 - template<typename T>
 std::vector< double > **rank** (std::vector< T > &values)

Sample ranks starting from 1.
 - std::vector< double > **logrankScores** (std::vector< double > &time, std::vector< double > &status)

Compute Logrank scores for survival times.
 - std::vector< double > **logrankScoresData** (Data *data, size_t time_varID, size_t status_varID, std::vector< size_t > &sampleIDs)

Compute Logrank scores for survival times directly from Data object.
 - void **maxstat** (std::vector< double > &scores, std::vector< double > &x, std::vector< size_t > &indices, double &best_maxstat, double &best_split_value, double minprop, double maxprop)

Compute maximally selected rank statistics.
 - void **maxstatInData** (std::vector< double > &scores, Data *data, std::vector< size_t > &sampleIDs, size_t varID, std::vector< size_t > &indices, double &best_maxstat, double &best_split_value, double minprop, double maxprop)

Compute maximally selected rank statistics in Data object.
 - std::vector< size_t > **numSamplesLeftOfCutpoint** (std::vector< double > &x, std::vector< size_t > &indices)

Compute number of samples smaller or equal than each unique value in x.
 - std::vector< size_t > **numSamplesLeftOfCutpointInData** (Data *data, std::vector< size_t > &sampleIDs, size_t varID, std::vector< size_t > &indices)

Compute number of samples smaller or equal than each unique value in data.

7.210.1 Function Documentation

7.210.1.1 **adjustPvalues()**

```
std::vector<double> adjustPvalues (
    std::vector< double > & unadjusted_pvalues )
```

Adjust p-values with Benjamini/Hochberg.

Parameters

<i>unadjusted_pvalues</i>	Unadjusted p-values (input)
<i>adjusted_pvalues</i>	Adjusted p-values (result)

Definition at line 465 of file utility.cpp.

7.210.1.2 beautifyTime()

```
std::string beautifyTime (
    uint seconds )
```

Beautify output of time.

Parameters

<i>seconds</i>	Time in seconds
----------------	-----------------

Returns

Time in days, hours, minutes and seconds as string

Definition at line 310 of file utility.cpp.

7.210.1.3 checkPositiveIntegers()

```
bool checkPositiveIntegers (
    std::vector< double > & all_values )
```

Check if all values in double vector are positive integers.

Parameters

<i>all_values</i>	Double vector to check
-------------------	------------------------

Returns

True if all values are positive integers

Definition at line 411 of file utility.cpp.

7.210.1.4 checkUnorderedVariables()

```
std::string checkUnorderedVariables (
    Data * data,
    std::vector< std::string > unordered_variable_names )
```

Check if not too many factor levels and all values in unordered categorical variables are positive integers.

Parameters

<i>data</i>	Pointer to data object
<i>unordered_variable_names</i>	Names of unordered variables

Returns

Error message, empty if no problem occurred

Definition at line 385 of file utility.cpp.

7.210.1.5 computeConcordanceIndex()

```
double computeConcordanceIndex (
    Data * data,
    std::vector< double > & sum_chf,
    size_t dependent_varID,
    size_t status_varID,
    std::vector< size_t > & sample_IDs )
```

Compute concordance index for given data and summed cumulative hazard function/estimate.

Parameters

<i>data</i>	Pointer to Data object
<i>sum_chf</i>	Summed chf over timepoints for each sample
<i>dependent_varID</i>	ID of dependent variable
<i>status_varID</i>	ID of status variable
<i>sample_IDs</i>	IDs of samples, for example OOB samples

Returns

concordance index

Definition at line 251 of file utility.cpp.

7.210.1.6 drawWithoutReplacementKnuth()

```
void drawWithoutReplacementKnuth (
    std::vector< size_t > & result,
    std::mt19937_64 & random_number_generator,
    size_t max,
    std::vector< size_t > & skip,
    size_t num_samples )
```

Knuth's algorithm for sampling without replacement, faster for larger num_samples Idea from Knuth 1985, The Art of Computer Programming, Vol.

2, Sec. 3.4.2 Algorithm S

Parameters

<i>result</i>	Vector to add results to. Will not be cleaned before filling.
<i>random_number_generator</i>	Random number generator
<i>range_length</i>	Length of range. Interval to draw from: 0..max-1
<i>skip</i>	Values to skip
<i>num_samples</i>	Number of samples to draw

Definition at line 156 of file utility.cpp.

7.210.1.7 drawWithoutReplacementSimple()

```
void drawWithoutReplacementSimple (
    std::vector< size_t > & result,
    std::mt19937_64 & random_number_generator,
    size_t max,
    std::vector< size_t > & skip,
    size_t num_samples )
```

Simple algorithm for sampling without replacement, faster for smaller num_samples.

Parameters

<i>result</i>	Vector to add results to. Will not be cleaned before filling.
<i>random_number_generator</i>	Random number generator
<i>range_length</i>	Length of range. Interval to draw from: 0..max-1
<i>skip</i>	Values to skip
<i>num_samples</i>	Number of samples to draw

Definition at line 131 of file utility.cpp.

7.210.1.8 drawWithoutReplacementSkip()

```
void drawWithoutReplacementSkip (
    std::vector< size_t > & result,
    std::mt19937_64 & random_number_generator,
    size_t range_length,
    std::vector< size_t > & skip,
    size_t num_samples )
```

Draw random numbers in a range without replacement and skip values.

Parameters

<i>result</i>	Vector to add results to. Will not be cleaned before filling.
<i>random_number_generator</i>	Random number generator
<i>range_length</i>	Length of range. Interval to draw from: 0..max-1
<i>skip</i>	Values to skip
<i>num_samples</i>	Number of samples to draw

Definition at line 121 of file utility.cpp.

7.210.1.9 drawWithoutReplacementWeighted() [1/2]

```
void drawWithoutReplacementWeighted (
    std::vector< size_t > & result,
    std::mt19937_64 & random_number_generator,
    std::vector< size_t > & indices,
    size_t num_samples,
    std::vector< double > & weights )
```

Draw random numbers without replacement and with weighted probabilities from vector of indices.

Parameters

<i>result</i>	Vector to add results to. Will not be cleaned before filling.
<i>random_number_generator</i>	Random number generator
<i>indices</i>	Vector with numbers to draw
<i>num_samples</i>	Number of samples to draw
<i>weights</i>	A weight for each element of indices

Definition at line 187 of file utility.cpp.

7.210.1.10 drawWithoutReplacementWeighted() [2/2]

```
void drawWithoutReplacementWeighted (
    std::vector< size_t > & result,
    std::mt19937_64 & random_number_generator,
    size_t max_index,
    size_t num_samples,
    std::vector< double > & weights )
```

Draw random numbers without replacement and with weighted probabilities from 0..n-1.

Parameters

<i>result</i>	Vector to add results to. Will not be cleaned before filling.
<i>random_number_generator</i>	Random number generator
<i>max_index</i>	Maximum index to draw
<i>num_samples</i>	Number of samples to draw
<i>weights</i>	A weight for each element of indices

Definition at line 207 of file utility.cpp.

7.210.1.11 dstdnorm()

```
double dstdnorm (
    double x )
```

Standard normal density.

Parameters

x	Quantile
---	----------

Returns

Standard normal density at quantile x

Definition at line 457 of file utility.cpp.

7.210.1.12 equalSplit()

```
void equalSplit (
    std::vector< uint > & result,
```

```
    uint start,
    uint end,
    uint num_parts )
```

Split sequence `start..end` in `num_parts` parts with sizes as equal as possible.

Parameters

<code>result</code>	Result vector of size <code>num_parts+1</code> . Ranges for the parts are then <code>result[0]..result[1]-1</code> , <code>result[1]..result[2]-1</code> , ..
<code>start</code>	minimum value
<code>end</code>	maximum value
<code>num_parts</code>	number of parts

Definition at line 45 of file utility.cpp.

7.210.1.13 loadDoubleVectorFromFile()

```
void loadDoubleVectorFromFile (
    std::vector< double > & result,
    std::string filename )
```

Read a double vector from text file.

Reads only the first line.

Parameters

<code>result</code>	Result vector of doubles with contents
<code>filename</code>	filename of input file

Definition at line 80 of file utility.cpp.

7.210.1.14 logrankScores()

```
std::vector<double> logrankScores (
    std::vector< double > & time,
    std::vector< double > & status )
```

Compute Logrank scores for survival times.

Parameters

<i>time</i>	Survival time
<i>status</i>	Censoring indicator

Returns

Logrank scores

Definition at line 500 of file utility.cpp.

7.210.1.15 logrankScoresData()

```
std::vector<double> logrankScoresData (
    Data * data,
    size_t time_varID,
    size_t status_varID,
    std::vector<size_t> sampleIDs )
```

Compute Logrank scores for survival times directly from [Data](#) object.**Parameters**

<i>data</i>	Pointer to Data object
<i>time_varID</i>	variable ID for time column
<i>status_varID</i>	Variable ID for status column
<i>sampleIDs</i>	IDs of samples to include

Returns

Logrank scores

Definition at line 532 of file utility.cpp.

7.210.1.16 maxstat()

```
void maxstat (
    std::vector<double> & scores,
    std::vector<double> & x,
    std::vector<size_t> & indices,
    double & best_maxstat,
```

```
    double & best_split_value,
    double minprop,
    double maxprop )
```

Compute maximally selected rank statistics.

Parameters

<i>scores</i>	Scores for dependent variable (y)
<i>x</i>	Independent variable
<i>indices</i>	Ordering of x values
<i>best_maxstat</i>	Maximally selected statistic (output)
<i>best_split_value</i>	Split value for maximally selected statistic (output)
<i>minprop</i>	Minimal proportion of observations left of cutpoint
<i>maxprop</i>	Maximal proportion of observations left of cutpoint

Definition at line 569 of file utility.cpp.

7.210.1.17 maxstatInData()

```
void maxstatInData (
    std::vector< double > & scores,
    Data * data,
    std::vector< size_t > & sampleIDs,
    size_t varID,
    std::vector< size_t > & indices,
    double & best_maxstat,
    double & best_split_value,
    double minprop,
    double maxprop )
```

Compute maximally selected rank statistics in [Data](#) object.

Parameters

<i>scores</i>	Scores for dependent variable (y)
<i>data</i>	Data object
<i>sampleIDs</i>	IDs of samples to consider
<i>varID</i>	ID of variable to consider
<i>indices</i>	Ordering of x values
<i>best_maxstat</i>	Maximally selected statistic (output)
<i>best_split_value</i>	Split value for maximally selected statistic (output)
<i>minprop</i>	Minimal proportion of observations left of cutpoint
<i>maxprop</i>	Maximal proportion of observations left of cutpoint

Definition at line 629 of file utility.cpp.

7.210.1.18 maxstatPValueLau92()

```
double maxstatPValueLau92 (
    double b,
    double minprop,
    double maxprop )
```

Compute p-value for maximally selected rank statistics using Lau92 approximation See Lausen, B.

& Schumacher, M. (1992). Biometrics 48, 73-85.

Parameters

<i>b</i>	Quantile
<i>minprop</i>	Minimal proportion of observations left of cutpoint
<i>maxprop</i>	Maximal proportion of observations left of cutpoint

Returns

p-value for quantile b

Definition at line 420 of file utility.cpp.

7.210.1.19 maxstatPValueLau94()

```
double maxstatPValueLau94 (
    double b,
    double minprop,
    double maxprop,
    size_t N,
    std::vector< size_t > & m )
```

Compute p-value for maximally selected rank statistics using Lau94 approximation See Lausen, B., Sauerbrei, W.

& Schumacher, M. (1994). Computational Statistics. 483-496.

Parameters

<i>b</i>	Quantile
<i>minprop</i>	Minimal proportion of observations left of cutpoint
<i>maxprop</i>	Maximal proportion of observations left of cutpoint
<i>N</i>	Number of observations
<i>m</i>	Vector with number of observations smaller or equal than cutpoint, sorted, only for unique cutpoints

Returns

p-value for quantile b

Definition at line 439 of file utility.cpp.

7.210.1.20 maxstatPValueUnadjusted()

```
double maxstatPValueUnadjusted (
    double b )
```

Compute unadjusted p-value for rank statistics.

Parameters

<i>b</i>	Quantile
----------	----------

Returns

p-value for quantile b

Definition at line 453 of file utility.cpp.

7.210.1.21 mostFrequentClass()

```
template<typename T >
size_t mostFrequentClass (
    std::vector< T > & class_count,
    std::mt19937_64 random_number_generator )
```

Returns the most frequent class index of a vector with counts for the classes.

Returns a random class if counts are equal.

Parameters

<i>class_count</i>	Vector with class counts
<i>random_number_generator</i>	Random number generator

Returns

Most frequent class index. Out of range index if all 0.

Definition at line 223 of file utility.h.

7.210.1.22 mostFrequentValue()

```
double mostFrequentValue (
    std::unordered_map< double, size_t > & class_count,
    std::mt19937_64 random_number_generator )
```

Returns the most frequent value of a map with counts for the values.

Returns a random class if counts are equal.

Parameters

<i>class_count</i>	Map with classes and counts
<i>random_number_generator</i>	Random number generator

Returns

Most frequent value

Definition at line 227 of file utility.cpp.

7.210.1.23 numSamplesLeftOfCutpoint()

```
std::vector<size_t> numSamplesLeftOfCutpoint (
    std::vector< double > & x,
    std::vector< size_t > & indices )
```

Compute number of samples smaller or equal than each unique value in x.

Parameters

<i>x</i>	Value vector
<i>indices</i>	Ordering of x

Returns

Vector of number of samples smaller or equal than each unique value in x

Definition at line 690 of file utility.cpp.

7.210.1.24 numSamplesLeftOfCutpointInData()

```
std::vector<size_t> numSamplesLeftOfCutpointInData (
    Data * data,
    std::vector< size_t > & sampleIDs,
    size_t varID,
    std::vector< size_t > & indices )
```

Compute number of samples smaller or equal than each unique value in data.

Parameters

<i>data</i>	Data object
<i>sampleIDs</i>	IDs of samples to consider
<i>varID</i>	ID of variable to consider
<i>indices</i>	Ordering of x

Returns

Vector of number of samples smaller or equal than each unique value in x

Definition at line 707 of file utility.cpp.

7.210.1.25 order()

```
template<typename T >
std::vector<size_t> order (
    std::vector< T > & values,
    bool decreasing )
```

Get indices of sorted values.

Parameters

<i>values</i>	Values to sort
<i>decreasing</i>	Order decreasing

Returns

Indices of sorted values

Definition at line 386 of file utility.h.

7.210.1.26 orderInData()

```
std::vector<size_t> orderInData (
    Data * data,
    std::vector< size_t > & sampleIDs,
    size_t varID,
    bool decreasing )
```

Get indices of sorted values, compute in-place of [Data](#).

Parameters

<i>data</i>	Data object
<i>sampleIDs</i>	IDs of samples to sort
<i>varID</i>	ID of variable to sort by
<i>decreasing</i>	Order decreasing

Returns

Indices of sorted values

Definition at line 484 of file [utility.cpp](#).

7.210.1.27 pstdnorm()

```
double pstdnorm (
    double x )
```

Standard normal distribution.

Parameters

<i>x</i>	Quantile
----------	----------

Returns

Standard normal distribution at quantile x

Definition at line 461 of file [utility.cpp](#).

7.210.1.28 rank()

```
template<typename T >
std::vector<double> rank (
    std::vector< T > & values )
```

Sample ranks starting from 1.

Ties are given the average rank.

Parameters

<code>values</code>	Values to rank
---------------------	----------------

Returns

Ranks of input values

Definition at line 416 of file utility.h.

7.210.1.29 readVector1D() [1/2]

```
template<typename T >
void readVector1D (
    std::vector< T > & result,
    std::ifstream & file ) [inline]
```

Read a 1d vector written by [saveVector1D\(\)](#) from ifstream.

Parameters

<code>result</code>	Result vector with elements of type T.
<code>file</code>	ifstream object to read from.

Definition at line 94 of file utility.h.

7.210.1.30 readVector1D() [2/2]

```
template<>
void readVector1D (
    std::vector< bool > & result,
    std::ifstream & file ) [inline]
```

Definition at line 103 of file utility.h.

7.210.1.31 `readVector2D()`

```
template<typename T >
void readVector2D (
    std::vector< std::vector< T >> & result,
    std::ifstream & file ) [inline]
```

Read a 2d vector written by [saveVector2D\(\)](#) from ifstream.

Parameters

<i>result</i>	Result vector of vectors with elements of type T.
<i>file</i>	ifstream object to read from.

Definition at line 140 of file utility.h.

7.210.1.32 `roundToNextMultiple()`

```
size_t roundToNextMultiple (
    size_t value,
    uint multiple )
```

Round up to next multiple of a number.

Parameters

<i>value</i>	Value to be rounded up.
<i>multiple</i>	Number to multiply.

Returns

Rounded number

Definition at line 343 of file utility.cpp.

7.210.1.33 `saveVector1D()` [1/2]

```
template<typename T >
void saveVector1D (
    std::vector< T > & vector,
    std::ofstream & file ) [inline]
```

Write a 1d vector to ifstream.

First the size is written as size_t, then all vector elements.

Parameters

<i>vector</i>	Vector with elements of type T to write to file.
<i>file</i>	ofstream object to write to. Write a 1d vector to filestream. First the size is written, then all vector elements.
<i>vector</i>	Vector of type T to save
<i>file</i>	ofstream object to write to.

Definition at line 68 of file utility.h.

7.210.1.34 saveVector1D() [2/2]

```
template<>
void saveVector1D (
    std::vector< bool > & vector,
    std::ofstream & file ) [inline]
```

Definition at line 76 of file utility.h.

7.210.1.35 saveVector2D()

```
template<typename T >
void saveVector2D (
    std::vector< std::vector< T >> & vector,
    std::ofstream & file ) [inline]
```

Write a 2d vector to filestream.

First the size of the first dim is written as size_t, then for all inner vectors the size and elements.

Parameters

<i>vector</i>	Vector of vectors of type T to write to file.
<i>file</i>	ofstream object to write to.

Definition at line 122 of file utility.h.

7.210.1.36 shuffleAndSplit()

```
void shuffleAndSplit (
    std::vector< size_t > & first_part,
```

```
std::vector< size_t > & second_part,
size_t n_all,
size_t n_first,
std::mt19937_64 random_number_generator )
```

Create numbers from 0 to n_all-1, shuffle and split in two parts.

Parameters

<i>first_part</i>	First part
<i>second_part</i>	Second part
<i>n_all</i>	Number elements
<i>n_first</i>	Number of elements of first part
<i>random_number_generator</i>	Random number generator

Definition at line 367 of file utility.cpp.

7.210.1.37 splitString()

```
void splitString (
    std::vector< std::string > & result,
    std::string input,
    char split_char )
```

Split string in parts separated by character.

Parameters

<i>result</i>	Splitted string
<i>input</i>	String to be splitted
<i>split_char</i>	Char to separate parts

Definition at line 357 of file utility.cpp.

7.210.1.38 uintToString()

```
std::string uintToString (
    uint number )
```

Convert a unsigned integer to string.

Parameters

<code>number</code>	Number to convert
---------------------	-------------------

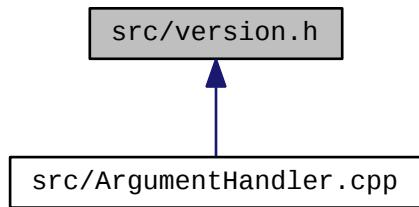
Returns

Converted number as string

Definition at line 300 of file utility.cpp.

7.211 src/version.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- `#define RANGER_VERSION "0.6.3"`

7.211.1 Macro Definition Documentation

7.211.1.1 RANGER_VERSION

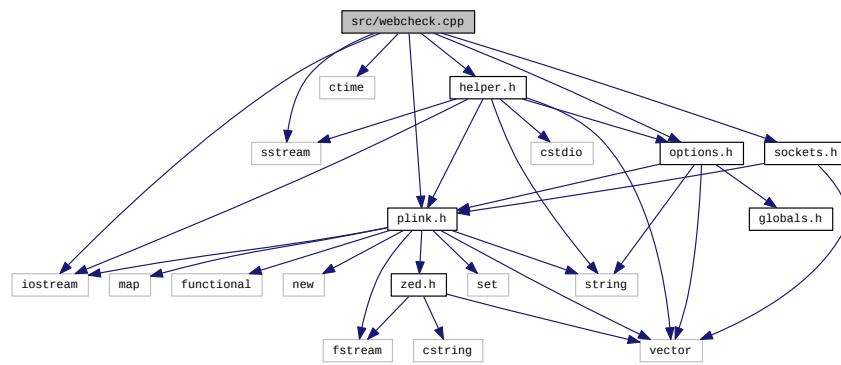
```
#define RANGER_VERSION "0.6.3"
```

Definition at line 2 of file version.h.

7.212 src/webcheck.cpp File Reference

```
#include <iostream>
#include <sstream>
#include <ctime>
#include "plink.h"
#include "helper.h"
#include "options.h"
#include "sockets.h"

Include dependency graph for webcheck.cpp:
```



Macros

- #define PORT_NUM 80
- #define IP_ADDR "155.52.206.11"
- #define GET_STRING "GET /~purcell/plink/version2.txt HTTP/1.1\nHost: pngu.mgh.harvard.edu\nConnection: close\n\n"

Variables

- string PVERSION
- string PREL

7.212.1 Macro Definition Documentation

7.212.1.1 GET_STRING

```
#define GET_STRING "GET /~purcell/plink/version2.txt HTTP/1.1\nHost: pngu.mgh.harvard.edu\n←
Connection: close\n\n"
```

Definition at line 30 of file webcheck.cpp.

7.212.1.2 IP_ADDR

```
#define IP_ADDR "155.52.206.11"
```

Definition at line 29 of file webcheck.cpp.

7.212.1.3 PORT_NUM

```
#define PORT_NUM 80
```

Definition at line 28 of file webcheck.cpp.

7.212.2 Variable Documentation

7.212.2.1 PREL

```
string PREL
```

Definition at line 84 of file inbix.cpp.

7.212.2.2 PVERSION

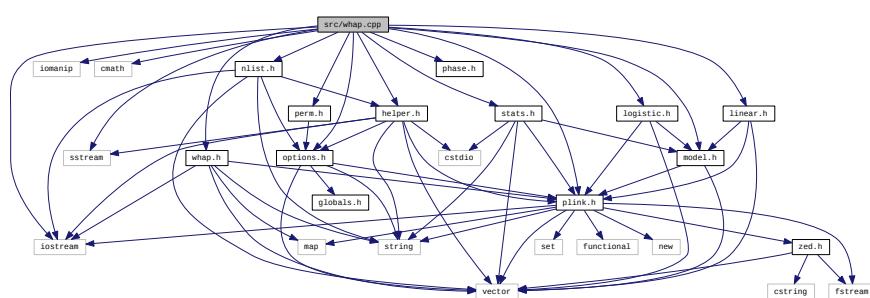
```
string PVERSION
```

Definition at line 82 of file inbix.cpp.

7.213 src/whap.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <cmath>
#include <sstream>
#include "whap.h"
#include "helper.h"
#include "plink.h"
#include "options.h"
#include "perm.h"
#include "nlist.h"
#include "phase.h"
#include "model.h"
#include "linear.h"
#include "logistic.h"
#include "stats.h"
```

Include dependency graph for whap.cpp:



Functions

- void [displayHaploGroups](#) (ofstream &, ChapModel &, HaploPhase *)
- string [ci](#) (double coef, double se)

7.213.1 Function Documentation

7.213.1.1 ci()

```
string ci (
    double coef,
    double se )
```

Definition at line 58 of file whap.cpp.

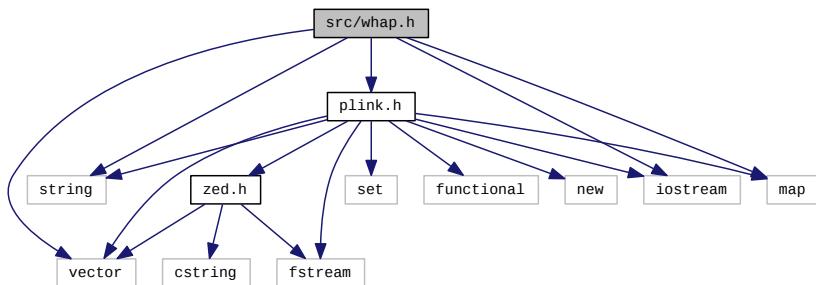
7.213.1.2 displayHaploGroups()

```
void displayHaploGroups (
    ofstream & CH,
    ChapModel & m,
    HaploPhase * haplo )
```

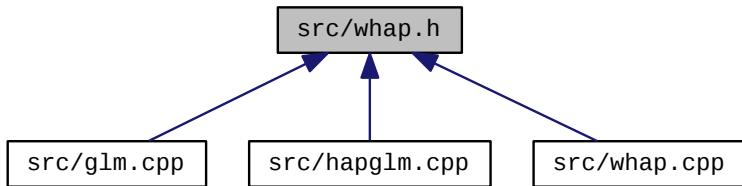
Definition at line 1420 of file whap.cpp.

7.214 src/whap.h File Reference

```
#include <string>
#include <vector>
#include <iostream>
#include <map>
#include "plink.h"
Include dependency graph for whap.h:
```



This graph shows which files directly or indirectly include this file:

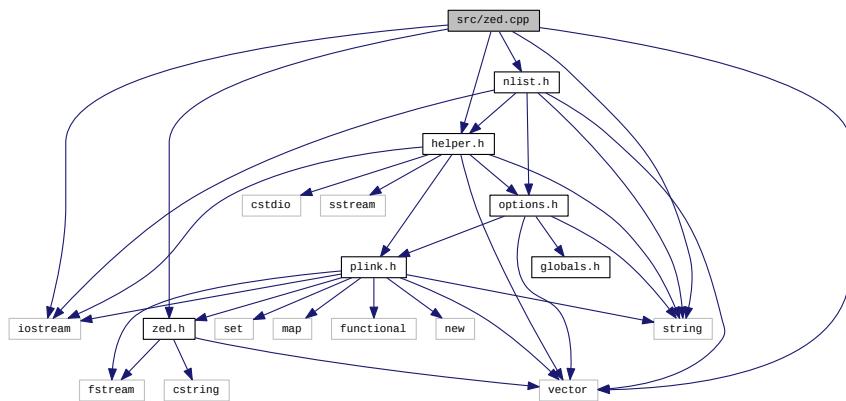


Classes

- class [ChapModel](#)
- class [Chap](#)

7.215 src/zed.cpp File Reference

```
#include <iostream>
#include <string>
#include <vector>
#include "zed.h"
#include "helper.h"
#include "nlist.h"
Include dependency graph for zed.cpp:
```



Functions

- void [fileCompress \(\)](#)
- void [fileUncompress \(\)](#)

Variables

- [Plink * PP](#)

7.215.1 Function Documentation

7.215.1.1 fileCompress()

```
void fileCompress ( )
```

Definition at line 201 of file zed.cpp.

7.215.1.2 fileUncompress()

```
void fileUncompress ( )
```

Definition at line 224 of file zed.cpp.

7.215.2 Variable Documentation

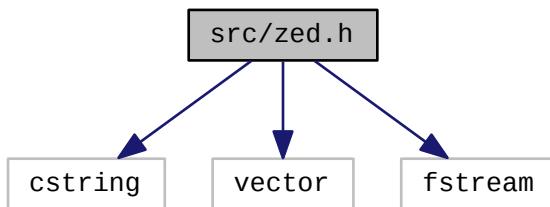
7.215.2.1 PP

```
PLink* PP
```

Definition at line 85 of file inbix.cpp.

7.216 src/zed.h File Reference

```
#include <cstring>
#include <vector>
#include <fstream>
Include dependency graph for zed.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [ZInput](#)
- class [ZOutput](#)

Functions

- void [fileCompress \(\)](#)
- void [fileUncompress \(\)](#)

Variables

- const int [MAX_LINE_LENGTH](#) = 1000000

7.216.1 Function Documentation

7.216.1.1 [fileCompress\(\)](#)

```
void fileCompress ( )
```

Definition at line 201 of file zed.cpp.

7.216.1.2 [fileUncompress\(\)](#)

```
void fileUncompress ( )
```

Definition at line 224 of file zed.cpp.

7.216.2 Variable Documentation

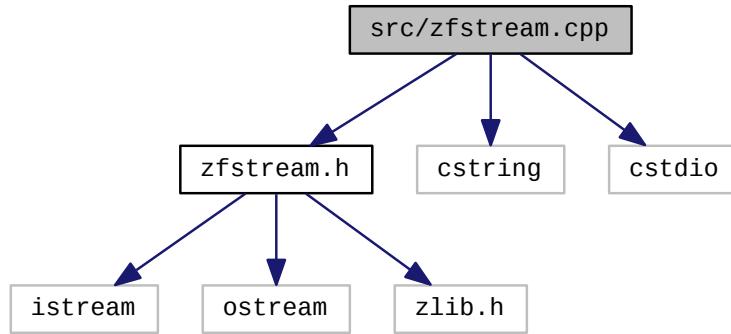
7.216.2.1 [MAX_LINE_LENGTH](#)

```
const int MAX_LINE_LENGTH = 1000000
```

Definition at line 17 of file zed.h.

7.217 src/zfstream.cpp File Reference

```
#include "zfstream.h"
#include <cstring>
#include <cstdio>
Include dependency graph for zfstream.cpp:
```



Macros

- #define BIGBUFSIZE BUFSIZ
- #define SMALLBUFSIZE 1

7.217.1 Macro Definition Documentation

7.217.1.1 BIGBUFSIZE

```
#define BIGBUFSIZE BUFSIZ
```

Definition at line 15 of file zfstream.cpp.

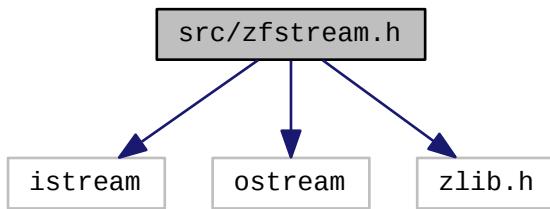
7.217.1.2 SMALLBUFSIZE

```
#define SMALLBUFSIZE 1
```

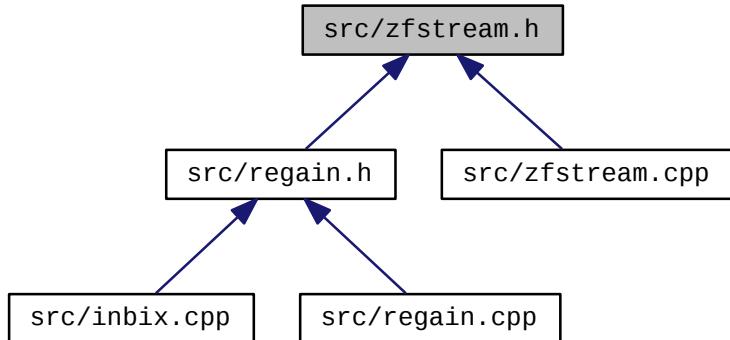
Definition at line 16 of file zfstream.cpp.

7.218 src/zfstream.h File Reference

```
#include <iostream>
#include <ostream>
#include "zlib.h"
Include dependency graph for zfstream.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [gzfilebuf](#)
Gzipped file stream buffer class.
- class [gzifstream](#)
Gzipped file input stream class.
- class [gzofstream](#)
Gzipped file output stream class.
- class [gzomanip2< T1, T2 >](#)
Gzipped file output stream manipulator class.

Functions

- `gzofstream & setcompression (gzofstream &gzs, int l, int s=Z_DEFAULT_STRATEGY)`
- `template<typename T1 , typename T2 >`
`gzofstream & operator<< (gzofstream &s, const gzomanip2< T1, T2 > &m)`
- `gzomanip2< int, int > setcompression (int l, int s=Z_DEFAULT_STRATEGY)`

7.218.1 Function Documentation

7.218.1.1 `operator<<()`

```
template<typename T1 , typename T2 >
gzofstream& operator<< (
    gzofstream & s,
    const gzomanip2< T1, T2 > & m ) [inline]
```

Definition at line 458 of file zfstream.h.

7.218.1.2 `setcompression()` [1/2]

```
gzofstream& setcompression (
    gzofstream & gzs,
    int l,
    int s = Z_DEFAULT_STRATEGY ) [inline]
```

Definition at line 440 of file zfstream.h.

7.218.1.3 `setcompression()` [2/2]

```
gzomanip2<int,int> setcompression (
    int l,
    int s = Z_DEFAULT_STRATEGY ) [inline]
```

Definition at line 463 of file zfstream.h.

Index

- ~RipmResult, 27
 - degrees, 28
 - hubDegrees, 28
 - hubs, 28
 - modules, 28
 - sizes, 28
- ~ArgumentHandler
 - ArgumentHandler, 35
- ~AttributeRanker
 - AttributeRanker, 44
- ~BirdseedData
 - BirdseedData, 51
- ~CentralityRanker
 - CentralityRanker, 69
- ~ChiSquared
 - ChiSquared, 86
- ~Data
 - Data, 114
- ~DataChar
 - DataChar, 123
- ~DataDouble
 - DataDouble, 128
- ~DataFloat
 - DataFloat, 133
- ~Dataset
 - Dataset, 144
- ~DatasetInstance
 - DatasetInstance, 203
- ~DcVar
 - DcVar, 217
- ~Dseq
 - Dseq, 235
- ~DgeData
 - DgeData, 240
- ~Edger
 - Edger, 258
- ~EpistasisEQtl
 - EpistasisEQtl, 265
- ~EvaporativeCooling
 - EvaporativeCooling, 276
- ~EvaporativeCoolingPrivacy
 - EvaporativeCoolingPrivacy, 289
- ~ExpressionDataSimulator
 - ExpressionDataSimulator, 302
- ~Forest
 - Forest, 316
 - ~ForestClassification
 - ForestClassification, 335
 - ~ForestProbability
 - ForestProbability, 341
 - ~ForestRegression
 - ForestRegression, 348
 - ~ForestSurvival
 - ForestSurvival, 354
 - ~GSLRandomBase
 - GSLRandomBase, 363
 - ~GSLRandomFlat
 - GSLRandomFlat, 366
 - ~HaploWindow
 - HaploWindow, 427
 - ~InteractionNetwork
 - InteractionNetwork, 473
 - ~LinearModel
 - LinearModel, 499
 - ~LogisticModel
 - LogisticModel, 513
 - ~Model
 - Model, 521
 - ~Options
 - Options, 548
 - ~PlinkInternalsDataset
 - PlinkInternalsDataset, 844
 - ~PlinkInternalsDatasetInstance
 - PlinkInternalsDatasetInstance, 847
 - ~RReliefF
 - RReliefF, 964
 - ~RandomForest
 - RandomForest, 863
 - ~Rconnection
 - Rconnection, 872
 - ~Regain
 - Regain, 891
 - ~ReliefF
 - ReliefF, 908
 - ~ReliefFSeq
 - ReliefFSeq, 922
 - ~ReliefSeqController
 - ReliefSeqController, 928
 - ~Rexp
 - Rexp, 940

~Rlist
 Rlist, 954
 ~Rmessage
 Rmessage, 958
 ~Rvector
 Rvector, 982
 ~SNReliefF
 SNReliefF, 1013
 ~Tree
 Tree, 1024
 ~TreeClassification
 TreeClassification, 1039
 ~TreeProbability
 TreeProbability, 1047
 ~TreeRegression
 TreeRegression, 1056
 ~TreeSurvival
 TreeSurvival, 1064
 ~gzfilebuf
 gzfilebuf, 373

a
 CArgs, 65
 PairwiseLinkage, 558
 stats.cpp, 1495

a1
 Alleles, 31
 fullGenotype, 360
 Var, 1072

a2
 Alleles, 32
 fullGenotype, 360

A_crypt
 Rconnection.h, 1410

A_plain
 Rconnection.h, 1410

A_required
 Rconnection.h, 1410

AF_LOCAL
 Rconnection.cpp, 1407

AA
 probabilisticGenotype, 855

AB
 probabilisticGenotype, 855

acnt
 RCount, 880

acnt2
 RCount, 880

acode
 Variant, 1075

actual_map
 HaploPhase, 412

actual_nw
 HaploPhase, 412

adaptive
 Perm, 767
 adaptive_alpha
 par, 582
 adaptive_ci
 par, 582
 adaptive_interval
 par, 583
 adaptive_interval2
 par, 583
 adaptive_max
 par, 583
 adaptive_min
 par, 583
 adaptive_perm
 par, 583
 addAdditiveSNP
 Model, 522
 addCovariate
 Model, 522
 addCovariates
 Regain, 892
 addDominanceSNP
 Model, 522
 addGiniImportance
 TreeClassification, 1039
 addHaplotypeDosage
 Model, 522
 addImpurityImportance
 TreeProbability, 1047
 TreeRegression, 1056
 AddInfluenceFactorD
 DatasetInstance, 203
 addInteraction
 Model, 522
 AddModule
 InteractionNetwork, 473
 AddNumeric
 DatasetInstance, 203
 addNumeric
 Model, 522
 addOption
 Options, 548
 addParent
 trio.cpp, 1528
 addPerson
 trio.cpp, 1528
 addSNP
 RCount, 878
 addSexEffect
 Model, 523
 addSparseData
 Data, 115
 addToTerminalNodes

TreeProbability, 1047
addTypedInteraction
 Model, 523
additive
 Model, 531
adjMatrix
 InteractionNetwork, 482
adjustPvalues
 utility.cpp, 1531
 utility.h, 1548
af_count
 par, 583
af_file
 par, 584
af_read
 par, 584
af_write
 par, 584
aff
 Individual, 459
affCoding
 helper.cpp, 1280
 helper.h, 1301
affpair
 par, 584
afni1dFile
 par, 584
afniNetwork
 par, 584
algediv
 cdflib.h, 1110
 dcdflib.cpp, 1171
algorithm_verbose
 par, 585
algorithmMode
 par, 585
 ReliefSeqController, 931
algorithmSeqMode
 par, 585
algorithmSeqS0
 par, 585
algorithmSnrMode
 par, 585
algorithmTstatMode
 par, 585
algorithmType
 EvaporativeCooling, 282
alias_delimit
 IDFile, 444
aliasList
 IDField, 441
all_pheno
 par, 586
all_valid

Model, 531
allSegmentsIntersecting
 cnv.cpp, 1142
 cnv.h, 1145
allele
 WMLocus, 1077
allele1
 GVariant, 368
 Locus, 509
allele2
 GVariant, 368
 Locus, 509
allelePairs
 clump_LD, 93
alleleRecoding
 Plink, 780
Alleles, 29
 a1, 31
 a2, 32
 Alleles, 30
 bp, 32
 chr, 32
 matches, 31
 operator<, 31
 operator==, 31
 snp, 32
 swap, 31
alleles
 Variant, 1075
allellicVariation
 Variant, 1075
alngam
 cdflib.h, 1110
 dcdflib.cpp, 1171
alnrel
 cdflib.h, 1111
 dcdflib.cpp, 1172
alogam
 fisher.cpp, 1233
alpha
 ArgumentHandler, 36
 Forest, 325
 par, 586
 Tree, 1030
alternate
 Chap, 75
alternatePhenotypesFilename
 Dataset, 187
alwayssplitvars
 ArgumentHandler, 36
 par, 586
AM
 CRandom, 108
ambig

HaplPhase, 413
 HaplWindow, 430
analyse_cluster
 par, 586
analyseModel
 gvar.cpp, 1270
AnalysisType
 Insilico.h, 1343
analysisType
 EvaporativeCooling, 283
 ReliefSeqController, 931
 ReliefF, 914
annot
 ClumpResults, 100
 ResultTrio, 936
annot.cpp
 filterRanges, 1094
 PP, 1094
annot_file
 par, 586
annot_filename
 par, 586
annotateFile
 Plink, 780
any
 CArgs, 62
appendToFile
 Tree, 1025
appendToFileInternal
 Tree, 1025
 TreeClassification, 1039
 TreeProbability, 1047
 TreeRegression, 1056
 TreeSurvival, 1064
ApplyFisherTransform
 InteractionNetwork, 473
ApplyPowerTransform
 InteractionNetwork, 474
apser
 cdflib.h, 1111
 dcdflib.cpp, 1172
argc
 ArgumentHandler, 36
ArgumentHandler, 33
 ~ArgumentHandler, 35
 alpha, 36
 alwaysplitvars, 36
 argc, 36
 ArgumentHandler, 35
 argv, 37
 caseweights, 37
 catvars, 37
 checkArguments, 35
 DISALLOW_COPY_AND_ASSIGN, 35
 depvarname, 37
 displayHelp, 35
 displayVersion, 36
 file, 37
 fraction, 37
 holdout, 38
 impmeasure, 38
 memmode, 38
 minprop, 38
 mtry, 38
 nthreads, 38
 ntree, 39
 outprefix, 39
 predall, 39
 predict, 39
 predictiontype, 39
 probability, 39
 processArguments, 36
 replace, 40
 savemem, 40
 seed, 40
 splitrule, 40
 splitweights, 40
 statusvarname, 40
 targetpartitionsize, 41
 treetype, 41
 verbose, 41
 write, 41
argv
 ArgumentHandler, 37
armaComputeCovariance
 ArmadilloFuncs.cpp, 1096
 ArmadilloFuncs.h, 1099
armaComputeSparseCovariance
 ArmadilloFuncs.cpp, 1096
 ArmadilloFuncs.h, 1099
armaDcgain
 ArmadilloFuncs.cpp, 1097
 ArmadilloFuncs.h, 1099
armaGetPlinkNumericToMatrixAll
 ArmadilloFuncs.cpp, 1097
 ArmadilloFuncs.h, 1099
armaGetPlinkNumericToMatrixCaseControl
 ArmadilloFuncs.cpp, 1097
 ArmadilloFuncs.h, 1100
armaReadMatrix
 ArmadilloFuncs.cpp, 1097
 ArmadilloFuncs.h, 1100
armaWriteMatrix
 ArmadilloFuncs.cpp, 1097
 ArmadilloFuncs.h, 1100
armaWriteSparseMatrix
 ArmadilloFuncs.cpp, 1098
 ArmadilloFuncs.h, 1100

ArmadilloFuncs.cpp
armaComputeCovariance, 1096
armaComputeSparseCovariance, 1096
armaDcgain, 1097
armaGetPlinkNumericToMatrixAll, 1097
armaGetPlinkNumericToMatrixCaseControl, 1097
armaReadMatrix, 1097
armaWriteMatrix, 1097
armaWriteSparseMatrix, 1098
ArmadilloFuncs.h
armaComputeCovariance, 1099
armaComputeSparseCovariance, 1099
armaDcgain, 1099
armaGetPlinkNumericToMatrixAll, 1099
armaGetPlinkNumericToMatrixCaseControl, 1100
armaReadMatrix, 1100
armaWriteMatrix, 1100
armaWriteSparseMatrix, 1100
assign
 Rconnection, 872
assoc.cpp
 LOG, 1101
assoc_counts
 par, 587
assoc_glm
 par, 587
assoc_glm_without_main_snp
 par, 587
assoc_gxe
 par, 587
assoc_results
 clump_LD, 95
assoc_test
 par, 587
assoc_test_alt_perm
 par, 587
atol
 dcdflib.cpp, 1162, 1163
attach
 gzfilebuf, 373
 gzifstream, 384
 gzofstream, 390
attr
 Rexp, 943
attribFields
 IDHelper, 451
attribFilterInd
 Plink, 780
attribFilterSNP
 Plink, 780
attribs
 Rexp, 943
attribute
 IDField, 441
 Rexp, 940
 attributeAlleleCounts
 Dataset, 187
 attributeAlleles
 Dataset, 188
 AttributeIndex
 ReliefF.cpp, 1417
 AttributeIndexIt
 ReliefF.cpp, 1417
 AttributeInteractionInformation
 Dataset, 145
 AttributeLevel
 Insilico.h, 1340
 attributeLevelsSeen
 Dataset, 188
 attributeMinorAllele
 Dataset, 188
 attributeMutationMap
 Dataset, 188
 AttributeMutationType
 Insilico.h, 1343
 attributeMutationTypes
 Dataset, 188
 attributeNames
 Dataset, 189
 Regain, 896
 Rexp, 941
 attributeProbability
 EvaporativeCoolingPrivacy, 293
AttributeRanker, 42
 ~AttributeRanker, 44
 AttributeRanker, 44
 classificationAccuracy, 47
 ComputeScores, 44
 dataset, 47
 GetClassificationError, 44
 GetNormalizeFlag, 44
 GetScores, 45
 InitializeData, 45
 k, 47
 NormalizeScores, 45
 normalizeScores, 47
 PrintScores, 45
 ResetForNextIteration, 46
 scoreNames, 48
 scores, 48
 SetNormalize, 46
 SetK, 46
 WriteScores, 46
 AttributeScores
 Insilico.h, 1340
 AttributeScoresClt
 Insilico.h, 1341
 AttributeScoresIt

Insilico.h, 1341
 attributeSort
 ReliefF.cpp, 1418
 AttributeType
 Insilico.h, 1344
 attributes
 DatasetInstance, 212
 attributesMask
 Dataset, 189
 attributesMaskPushed
 Dataset, 189
 attrnames
 Rexp, 943
 auth
 Rconnection, 876
 aval
 RCount, 880
 awin
 RCount, 880

B

Family, 308
 Individual, 459

b

PairwiseLinkage, 559
 stats.cpp, 1495

BATCH_SIZE

par, 588

BETAS

Regain, 896

BIGBUFSIZE

zfstream.cpp, 1575

BOOL_FALSE

Rsrv.h, 1427

BOOL_NA

Rsrv.h, 1427

BOOL_TRUE

Rsrv.h, 1427

BA

probabilisticGenotype, 855

badModels

EpistasisEQtl, 269

bartlett

genepi.cpp, 1254

baseline

Segment, 986

basym

cdflib.h, 1111
 dcdflib.cpp, 1172

BB

probabilisticGenotype, 855

bcorr

cdflib.h, 1111
 dcdflib.cpp, 1172

bcovar
 Individual, 459
 beautifyTime
 utility.cpp, 1532
 utility.h, 1549
 best_n
 insilico, 18
 bestClassificationError
 EvaporativeCooling, 283
 bestKs
 ReliefSeqController, 931
 ReliefF, 914
 bestNeighborhoods
 DatasetInstance, 212
 bestNeighborhoodsDiffClass
 DatasetInstance, 212
 bestNeighborhoodsSameClass
 DatasetInstance, 212
 betacf
 stats.cpp, 1480
 stats.h, 1499
 betai
 stats.cpp, 1480
 stats.h, 1499
 betaln
 cdflib.h, 1111
 dcdflib.cpp, 1172
 genepi.cpp, 1254
 bfrac
 cdflib.h, 1112
 dcdflib.cpp, 1173
 bgrat
 cdflib.h, 1112
 dcdflib.cpp, 1173
 bico
 genepi.cpp, 1254
 binput.cpp
 LOG, 1105
 BirdseedData, 48
 ~BirdseedData, 51
 BirdseedData, 51
 confidences, 56
 excludeSnps, 56
 excludeSnpsFilename, 56
 genotypeCounts, 56
 GetAlleleCounts, 51
 GetGenotypeCounts, 51
 GetMajorAlleleFrequency, 51
 GetMajorMinorAlleles, 52
 GetMissingValues, 52
 GetNumSNPs, 52
 GetNumSubjects, 52
 GetSNPNames, 53
 GetSamplePhenotype, 53

GetSubjectCallConfidences, 53
GetSubjectGenotypes, 53
GetSubjectLabels, 54
GetSubjectNames, 54
hasExcludedSnps, 57
hasIncludedSnps, 57
HasPhenotypes, 54
hasPhenotypes, 57
HasSubjectLabels, 54
hasSubjectLabels, 57
includeSnps, 57
includeSnpsFilename, 58
LoadData, 54
missingValues, 58
phenosFilename, 58
phenotypes, 58
PrintAlleleCounts, 55
PrintInfo, 56
snpAlleleCounts, 58
snpGenotypes, 59
snpMajorAlleleFreq, 59
snpMajorMinorAlleles, 59
snpNames, 59
snpsFilename, 59
subjectLabels, 60
subjectLabelsFilename, 60
subjectNames, 60
bitfilename
 par, 588
bitfilename_map
 par, 588
blox.cpp
 PP, 1107
bmatch
 par, 588
bmatch_direction_filename
 par, 588
bmatch_filename
 par, 588
bmatch_usertype
 par, 589
bnumeric
 Individual, 460
boolmatrix_t
 plink.h, 1392
boolvec_t
 plink.h, 1392
boot
 par, 589
bootstrap
 Tree, 1025
bootstrapWeighted
 Tree, 1025
bootstrapWithoutReplacement
 Tree, 1025
bootstrapWithoutReplacementWeighted
 Tree, 1026
bound
 par, 589
bp
 Alleles, 32
 Locus, 509
bpEnd
 Transcript, 1020
bpStart
 Transcript, 1020
bpser
 cdflib.h, 1112
 dcdflib.cpp, 1173
bpstart
 CInfo, 91
bpstop
 CInfo, 91
bratio
 cdflib.h, 1112
 dcdflib.cpp, 1173
brcmp1
 cdflib.h, 1113
 dcdflib.cpp, 1174
brcomp
 cdflib.h, 1113
 dcdflib.cpp, 1174
breslowday
 par, 589
bt
 par, 589
buf
 ZInput, 1084
 ZOutput, 1089
buffer
 gzfilebuf, 378
buffer_size
 gzfilebuf, 378
build
 Chap, 74
build_sin
 sisocks.h, 1456
buildAdditive
 Model, 523
buildCluster
 Plink, 781
buildCovariate
 Model, 523
buildDesignMatrix
 Model, 523
buildDominance
 Model, 524
buildFromGroups

ChapModel, 77
 buildFromSNPs
 ChapModel, 77
 buildHaplotype
 Model, 524
 buildInteraction
 Model, 524
 buildIntercept
 Model, 524
 buildNumeric
 Model, 524
 buildQFAM
 Model, 525
 buildSex
 Model, 525
 buildTypedInteraction
 Model, 525
 buildT
 Plink, 781
 built_families
 par, 589
 bup
 cdflib.h, 1113
 dcdflib.cpp, 1174
 byName
 Rvector, 982

C

genepi.cpp, 1254
 LinearModel, 503

c

stats.cpp, 1495

CArgs, 61

a, 65
 any, 62
 CArgs, 62
 check_unused_options, 63
 count, 63
 find, 63
 fromPriorLog, 63
 fromScript, 63
 n, 65
 option, 65
 optionLabel, 66
 original, 66
 parseOptions, 64
 parsed, 66
 root_command, 66
 value, 64
 value_double, 64
 value_int, 64
 value_lui, 65
 varValue, 65

CCA_caseonly

genepi.cpp, 1255
 CCA_logit
 genepi.cpp, 1255
 CERR_auth_unsupported
 Rconnection.h, 1410
 CERR_connect_failed
 Rconnection.h, 1410
 CERR_handshake_failed
 Rconnection.h, 1411
 CERR_invalid_id
 Rconnection.h, 1411
 CERR_io_error
 Rconnection.h, 1411
 CERR malformed_packet
 Rconnection.h, 1411
 CERR_not_connected
 Rconnection.h, 1411
 CERR_not_supported
 Rconnection.h, 1411
 CERR_out_of_mem
 Rconnection.h, 1412
 CERR_peer_closed
 Rconnection.h, 1412
 CERR_protocol_not_supp
 Rconnection.h, 1412
 CERR_send_error
 Rconnection.h, 1412
 CHECKPOINT_FILENAME
 DcVar.h, 1199
 CHIP_SEQ_EXTRACT_FIELD_IDX
 DcVar.h, 1198
 CHIP_SEQ_INFO_LIST
 DcVar.h, 1198
 CHIP_SEQ_INFO, 80
 chrom, 81
 position, 81
 rsnum, 81
 totalRegionReads, 81

CInfo, 90

bpstart, 91
 bpstop, 91
 lstart, 91
 lstop, 91

CMD_RESP

Rsrv.h, 1430

CMD_SPECIAL_MASK

Rsrv.h, 1431

CMD_STAT

Rsrv.h, 1432

CMD_assignSEXP

Rsrv.h, 1428

CMD_attachSession

Rsrv.h, 1428

CMD_closeFile

Rsrv.h, 1428
CMD_createFile
 Rsrv.h, 1428
CMD_detachSession
 Rsrv.h, 1429
CMD_detachedVoidEval
 Rsrv.h, 1428
CMD_eval
 Rsrv.h, 1429
CMD_login
 Rsrv.h, 1429
CMD_openFile
 Rsrv.h, 1429
CMD_readFile
 Rsrv.h, 1429
CMD_removeFile
 Rsrv.h, 1430
CMD_serAssign
 Rsrv.h, 1430
CMD_serEEval
 Rsrv.h, 1430
CMD_serEval
 Rsrv.h, 1430
CMD_setBufferSize
 Rsrv.h, 1431
CMD_setEncoding
 Rsrv.h, 1431
CMD_setSEXP
 Rsrv.h, 1431
CMD_shutdown
 Rsrv.h, 1431
CMD_voidEval
 Rsrv.h, 1432
CMD_writeFile
 Rsrv.h, 1432
CMH_test_1
 par, 600
CMH_test_2
 par, 601
CMH_test_ORD
 par, 601
CNVIndivReport, 102
 CNVIndivReport, 103
 calculateResults, 103
 count, 103
 count_baseline, 103
 n, 103
 segCount, 104
 t1, 104
 t10, 104
 t11, 104
 t12, 104
 t2, 104
 t3, 105
t4, 105
t5, 105
t6, 105
t7, 105
t8, 105
t9, 106
COMMAND_LINE_ERROR
 Insilico.h, 1353
COORD_FIELDS
 EpistasisEQtl.h, 1224
CRandom, 106
 AM, 108
 EPS, 109
 IA, 109
 idum, 109
 IM, 109
 IQ, 109
 IR, 109
 iv, 110
 iy, 110
 NDIV, 110
 NTAB, 110
 RNMX, 110
 rand, 108
 srand, 108
CSNP, 111
 one, 112
 two, 112
calc_SNPSNP_LD
 par, 590
calcAssociationWithBootstrap
 Plink, 781
calcAssociationWithPermutation
 Plink, 781
calcEpistasis
 Plink, 781
calcFlipScan
 Plink, 782
calcFst
 par, 590
 Plink, 782
calcGENEPIMeanVariance
 genepi.cpp, 1255
calcGXE
 Plink, 783
calcGenomeIBD
 Plink, 782
calcGenomeIBM
 Plink, 782
calcGenomeIBS
 Plink, 782
calcHomog
 Plink, 783
calcHotel

Plink, 783
 calcHotelSetMeanVariance
 hotel.cpp, 1323
 calcInbreeding
 Plink, 783
 calcInformative
 Plink, 783
 calcLDStatistics
 Plink, 784
 calcLocusIBD
 Plink, 784
 calcMantelHaenszel_2x2xK
 Plink, 784
 calcMantelHaenszel_IxJxK
 Plink, 784
 calcMantelHaenszel_ORD
 Plink, 784
 calcMH
 Plink, 785
 calcMultiPoint
 Plink, 785
 calcPairwiseLD
 Plink, 785
 calcPhenotypes
 Plink, 785
 calcQTDT
 Plink, 785
 calcRegression
 Plink, 786
 calcSetCovarianceMatrix
 helper.cpp, 1280
 helper.h, 1301
 calcSinglePoint
 Plink, 786
 calcStratifiedAlleleFreqs
 Plink, 786
 calculateAdjustedRSquared
 LinearModel, 499
 CalculateCentrality
 CentralityRanker, 69
 calculateCI
 PairwiseLinkage, 558
 CalculateDistanceMatrix
 Dataset, 145, 146
 calculateDp
 HaploPhase, 413
 calculateEmpiricalVariance
 HaploPhase, 399
 calculateFTest
 LinearModel, 500
 CalculateFdrBHThreshold
 DcVar, 217
 CalculateGainMatrix
 Dataset, 146
 calculateHaplotypeFrequencies
 HaploPhase, 400
 CalculateInteractionInformation
 Dataset, 147
 calculateLD
 PairwiseLinkage, 558
 calculateMallowC
 LinearModel, 500
 calculateProfile
 Plink, 786
 calculateRSquared
 LinearModel, 500
 calculateRSS
 LinearModel, 500
 CalculateRegainMatrix
 Dataset, 147
 calculateResults
 CNVIndivReport, 103
 calculated
 probabilisticGenotype, 856
 calculateHaplotypePhases
 HaploPhase, 400
 capacity
 Rvector, 983
 case_freq
 HaploPhase, 413
 case_weights
 Forest, 326
 Tree, 1030
 caseldxCol
 DcVar, 225
 caseweights
 ArgumentHandler, 37
 par, 590
 catvars
 ArgumentHandler, 37
 par, 590
 cdfbet
 cdflib.h, 1113
 dcdflib.cpp, 1174
 dcdflib.h, 1191
 cdfbin
 cdflib.h, 1114
 dcdflib.cpp, 1175
 dcdflib.h, 1191
 cdfchi
 cdflib.h, 1114
 dcdflib.cpp, 1175
 dcdflib.h, 1192
 cdfchn
 cdflib.h, 1114
 dcdflib.cpp, 1175
 dcdflib.h, 1192
 cdff

cdflib.h, 1115
dcdflib.cpp, 1176
dcdflib.h, 1192
cdffnc
 cdflib.h, 1115
 dcdflib.cpp, 1176
 dcdflib.h, 1193
cdfgam
 cdflib.h, 1115
 dcdflib.cpp, 1176
 dcdflib.h, 1193
cdflib.h
 algdiv, 1110
 alngam, 1110
 alnrel, 1111
 apser, 1111
 basym, 1111
 bcorr, 1111
 betaln, 1111
 bfrac, 1112
 bgrat, 1112
 bpser, 1112
 bratio, 1112
 brcmp1, 1113
 brcomp, 1113
 bup, 1113
 cdfbet, 1113
 cdfbin, 1114
 cdfchi, 1114
 cdfchn, 1114
 cdff, 1115
 cdfnnc, 1115
 cdfgam, 1115
 cdfnbn, 1116
 cdfnor, 1116
 cdfpoi, 1116
 cdft, 1117
 cumbet, 1117
 cumbin, 1117
 cumchi, 1118
 cumchn, 1118
 cumf, 1118
 cumfnc, 1118
 cumgam, 1119
 cumnbn, 1119
 cumnor, 1119
 cumpoi, 1119
 cumt, 1120
 dbetrm, 1120
 devlpl, 1120
 dexpm1, 1120
 dinvnr, 1121
 dinvr, 1121
 dlanor, 1121
 dln1mx, 1121
 dln1px, 1121
 dlnbet, 1122
 dlngam, 1122
 dstinv, 1122
 dstrem, 1122
 dstzr, 1122
 dt1, 1123
 dzror, 1123
 E0000, 1123
 E0001, 1123
 erf1, 1124
 erfc1, 1124
 esum, 1124
 exparg, 1124
 fidfint, 1125
 fidfmax1, 1125
 fidfmin1, 1125
 fidfsign, 1125
 fidfint, 1125
 fifmod, 1126
 fpser, 1126
 ftnstop, 1126
 gam1, 1126
 gaminv, 1126
 gamln, 1127
 gamln1, 1127
 grat1, 1127
 gratio, 1127
 gsumln, 1128
 ipmpar, 1128
 psi, 1128
 rcomp, 1128
 rexp, 1128
 rlog, 1129
 rlog1, 1129
 spmpar, 1129
 stvaln, 1129
 Xgamm, 1129
cdfnbn
 cdflib.h, 1116
 dcdflib.cpp, 1177
 dcdflib.h, 1193
cdfnor
 cdflib.h, 1116
 dcdflib.cpp, 1177
 dcdflib.h, 1194
cdfpoi
 cdflib.h, 1116
 dcdflib.cpp, 1177
 dcdflib.h, 1194
cdft
 cdflib.h, 1117
 dcdflib.cpp, 1178

dcdflib.h, 1194
 CentralityRanker, 67
 ~CentralityRanker, 69
 CalculateCentrality, 69
 CentralityRanker, 68
 colsum, 71
 D, 71
 G, 71
 gainFile, 71
 gamma, 72
 gammaVector, 72
 GaussEliminationSolver, 69
 Gdiag, 72
 GetResultsByVariable, 69
 Gtrace, 72
 n, 72
 Permute, 69
 PowerMethodSolver, 70
 r, 72
 ranks, 73
 ReadGainFile, 70
 SetGammaVector, 70
 SetGlobalGamma, 70
 topN, 73
 variableNames, 73
 WriteToConsole, 70
 WriteToFile, 71
 CentralityRanker.h
 SolverMethod, 1131
 cfamily.cpp
 genrel, 1132
 isAncestorOf, 1133
 listAllAncestors, 1133
 mCount, 1133
 cfamily.h
 genrel, 1134
 Chap, 73
 alternate, 75
 build, 74
 Chap, 74
 current, 75
 determineTestType, 74
 H, 76
 isNested, 75
 null, 76
 P, 76
 setModels, 75
 setSNPList, 75
 chap_add_grp_specifics
 par, 590
 chap_drop_snps
 par, 590
 chap_drop_snps_list
 par, 591
 chap_entity
 par, 591
 chap_haplotype_specific
 par, 591
 chap_independent_effect
 par, 591
 chap_model0
 par, 591
 chap_model1
 par, 591
 chap_sole_variant
 par, 592
 chap_sole_variant_specific_allele_list
 par, 592
 chap_sole_variant_specific_alleles
 par, 592
 chap_specified_groups
 par, 592
 chap_specified_snps
 par, 592
 chap_test
 par, 592
 ChapModel, 76
 buildFromGroups, 77
 buildFromSNPs, 77
 coef, 78
 df, 78
 group, 78
 haploGroup, 78
 haplotypesInSameGroup, 77
 label, 78
 lnLk, 79
 masked_conditioning_snps, 79
 model, 79
 p, 79
 rsq, 79
 se, 79
 check_sex
 par, 593
 check_unused_options
 CArgs, 63
 checkArguments
 ArgumentHandler, 35
 CheckDetectedAttributes
 EvaporativeCoolingPrivacy, 289
 checkDuplicates
 helper.cpp, 1280
 helper.h, 1301
 checkFileExists
 helper.cpp, 1280, 1281
 helper.h, 1301
 CheckHardyWeinbergEquilibrium
 Dataset, 148
 CheckIndices

InteractionNetwork, 474
CheckInputs
 DcVar, 217
 EpistasisEQtl, 265
checkMendel
 Plink, 787
CheckMergeResults
 InteractionNetwork, 474
CheckMissing
 DistanceMetrics.cpp, 1204
 DistanceMetrics.h, 1210
CheckMissingNumeric
 DistanceMetrics.cpp, 1204
 DistanceMetrics.h, 1211
checkPositiveIntegers
 utility.cpp, 1532
 utility.h, 1549
checkUnorderedVariables
 utility.cpp, 1532
 utility.h, 1549
checkVIF
 Model, 525
chf
 TreeSurvival, 1069
chi2x2
 stats.cpp, 1480, 1481
 stats.h, 1499, 1500
ChiSquared, 82
 ~ChiSquared, 86
 ChiSquared, 85
 chiSquaredValues, 89
 ClearTables, 86
 ComputeScore, 86
 ComputeScores, 87
 ComputeScoresWithPValues, 87
 expectedContingencyTable, 89
 GetFrequencyCounts, 87
 numClasses, 89
 numLevels, 89
 observedFreqTable, 89
 PrepareForAttribute, 87
 PrintScoresWithPValues, 88
 PrintTables, 88
 ResetForNextIteration, 88
 scoresPvalues, 90
 WriteScoresWithPValues, 88
chiSquaredValues
 ChiSquared, 89
chiTable
 stats.cpp, 1481
 stats.h, 1500
child_nodeIDs
 Tree, 1031
chipSeqExpression
 DcVar, 225
chipSeqMode
 DcVar, 225
chiprobP
 stats.cpp, 1481
 stats.h, 1500
chipseqComparatorAscending
 DcVar.cpp, 1196
chisq
 LinearModel, 504
 LogisticModel, 516
 ProxyResult, 858
 SetSortedSNP, 998
 SortedResult, 1016
ChooseSnpsDatasetByType
 Insilico.cpp, 1332
 Insilico.h, 1347
chr
 Alleles, 32
 Locus, 510
 Pool, 852
 Range, 869
 Transcript, 1020
 WMLocus, 1078
chr_code
 par, 593
chr_haploid
 par, 593
chr_map
 par, 593
chr_sex
 par, 593
chr_Y
 par, 593
chrom
 CHIP_SEQ_INFO, 81
 SNP_INFO, 1009
chromosomeName
 helper.cpp, 1281
 helper.h, 1302
ci
 whap.cpp, 1570
ci_level
 par, 594
ci_zt
 par, 594
class_values
 ForestClassification, 338
 ForestProbability, 344
 TreeClassification, 1043
 TreeProbability, 1051
classColumn
 Dataset, 189
classIndexes

Dataset, 189
 classLabel
 DatasetInstance, 213
 ClassLevel
 Insilico.h, 1341
 ClassType
 Insilico.h, 1344
 classification_table
 ForestClassification, 338
 ForestProbability, 344
 classificationAccuracy
 AttributeRanker, 47
 classificationErrors
 EvaporativeCooling, 283
 ClassifyAttributeSet
 EvaporativeCoolingPrivacy, 290
 cldist
 cluster.cpp, 1139
 cleanUp
 Plink, 787
 cleanUpInternal
 Tree, 1026
 TreeClassification, 1039
 TreeProbability, 1048
 TreeRegression, 1056
 TreeSurvival, 1064
 ClearInfluenceFactors
 DatasetInstance, 204
 ClearTables
 ChiSquared, 86
 cli
 par, 594
 clist
 Individual, 460
 par, 594
 clist_filename
 par, 594
 clist_number
 par, 594
 clist_selection
 par, 595
 clist_selection_name
 par, 595
 clist_selection_number
 par, 595
 clist_selection_string
 par, 595
 clistMissing
 Individual, 460
 clistname
 Plink, 824
 close
 gzfilebuf, 374
 gzfstream, 385
 gzofstream, 391
 ZInput, 1082
 ZOutput, 1087
 closeDUMP
 Perm, 763
 closeFile
 Rconnection, 873
 clst
 Model, 532
 clump
 clump_LD, 93
 clump_LD, 92
 allelePairs, 93
 assoc_results, 95
 clump, 93
 clump_LD, 93
 clumped, 95
 filename, 95
 hp, 95
 ld_distance, 95
 P, 96
 pval_cutoff, 96
 pvals, 96
 r2_cutoff, 96
 read_assoc_file, 94
 second_pval_cutoff, 96
 set_id, 94
 set_pval, 94
 set_r2, 94
 set_second_pval, 94
 snps, 96
 ClumpPair, 97
 f, 98
 operator<, 98
 snp, 98
 ClumpResults, 99
 annot, 100
 operator<, 100
 p, 100
 clumped
 clump_LD, 95
 clumpld
 par, 595
 clumpld.cpp
 returnFullRangeList, 1137
 clumpld_annot
 par, 595
 clumpld_annot_fields
 par, 596
 clumpld_best
 par, 596
 clumpld_column
 par, 596
 clumpld_indep

par, 596
clumpld_index1
 par, 596
clumpld_kb
 par, 596
clumpld_min
 par, 597
clumpld_only_show_replications
 par, 597
clumpld_only_show_replications_list
 par, 597
clumpld_p1
 par, 597
clumpld_p2
 par, 597
clumpld_r2
 par, 597
clumpld_range_annotate
 par, 598
clumpld_range_file
 par, 598
clumpld_results
 par, 598
clumpld_verbose
 par, 598
Cluster, 101
 person, 101
cluster
 Model, 532
 par, 598
cluster.cpp
 cldist, 1139
 groupAvgLink, 1140
 homogeneous_clusters, 1140
 LOG, 1141
 pairable_cluster, 1140
 selcon_inds, 1140
 spec_clusters, 1141
cluster_euclidean
 par, 598
cluster_group_avg
 par, 599
cluster_ibm_constraint
 par, 599
cluster_ibm_constraint_value
 par, 599
cluster_mds_dim
 par, 599
cluster_missing
 par, 599
cluster_on_mcc
 par, 599
cluster_on_phenotype
 par, 600
cluster_plot
 par, 600
cluster_selcon
 par, 600
cluster_selcon_file
 par, 600
cm_map
 par, 600
cmd
 phdr, 772
cnt_f
 HaploPhase, 413
 Plink, 824
cnv.cpp
 allSegmentsIntersecting, 1142
 count_intersects, 1142
 EPS_OVERLAP, 1144
 intersects, 1143
 PP, 1144
 probOverlap, 1143
 segmentCountCaseControls, 1143
 weighted_count_intersects, 1143
cnv.h
 allSegmentsIntersecting, 1145
 count_intersects, 1145
 intersects, 1145
 segmentCountCaseControls, 1145
 weighted_count_intersects, 1146
cnv_check_overlap
 par, 601
cnv_col
 par, 601
cnv_count
 par, 601
cnv_count_baseline
 par, 601
cnv_count_baseline_file
 par, 602
cnv_defined_overlap
 par, 602
cnv_del_only
 par, 602
cnv_disrupt
 par, 602
cnv_drop_no_segment
 par, 602
cnv_dup_only
 par, 602
cnv_en_model
 par, 603
cnv_enrichment_test
 par, 603
cnv_exclude
 par, 603

cnv_freq_include
 par, 603
 cnv_freq_include_below
 par, 603
 cnv_freq_include_cnt
 par, 603
 cnv_freq_include_exact
 par, 604
 cnv_freq_include_exact_exclude
 par, 604
 cnv_freq_method2
 par, 604
 cnv_freq_method2_threshold
 par, 604
 cnv_glm
 par, 604
 cnv_indiv_perm
 par, 604
 cnv_intersect
 par, 605
 cnv_intersect_file
 par, 605
 cnv_intersect_subset
 par, 605
 cnv_intersect_subset_file
 par, 605
 cnv_intersect_writeback
 par, 605
 cnv_intersect_writeback_verbose
 par, 605
 cnv_list
 par, 606
 cnv_listname
 par, 606
 cnv_makemap
 par, 606
 cnv_max_kb
 par, 606
 cnv_max_score
 par, 606
 cnv_max_sites
 par, 606
 cnv_min_kb
 par, 607
 cnv_min_score
 par, 607
 cnv_min_sites
 par, 607
 cnv_overlap
 par, 607
 cnv_pos_perm
 par, 607
 cnv_region_border
 par, 607

cnv_region_overlap
 par, 608
 cnv_type
 par, 608
 cnv_union_overlap
 par, 608
 cnv_unique
 par, 608
 cnv_weighted_gene_test
 par, 608
 cnv_write_freq
 par, 608
 cnv_writelist
 par, 609
 cnvqt.cpp
 PP, 1147
 coding01
 par, 609
 coef
 ChapModel, 78
 Model, 532
 coefs
 RegressionRankResults, 903
 colNames
 ExpressionDataSimulator, 304
 colsum
 CentralityRanker, 71
 combinations_recursive
 proxy.cpp, 1402
 commaParse
 helper.cpp, 1281
 helper.h, 1302
 command
 Rmessage, 958
 compareCNVs
 homozyg.cpp, 1320
 compareModels
 gvar.cpp, 1270
 complete
 Rmessage, 960
 compound_genotype_code
 par, 609
 compress_file
 par, 609
 compress_filename
 par, 609
 compress_genome
 par, 609
 compressed
 helper.cpp, 1281
 helper.h, 1302
 ZInput, 1084
 ZOutput, 1089
 ComputeAttributeProbabilities

EvaporativeCoolingPrivacy, 290
ComputeAttributeScores
 RReliefF, 964
 ReliefFSeq, 923
 ReliefF, 909
 SNReliefF, 1013
ComputeAttributeScoresIteratively
 ReliefF, 909
ComputeAttributeScoresKopt
 ReliefF, 910
computeAucSplit
 TreeSurvival, 1065
ComputeBestAttributesErrors
 EvaporativeCoolingPrivacy, 290
computeChildDeathCounts
 TreeSurvival, 1065
ComputeClassificationErrorRandomForest
 EvaporativeCooling, 276
computeConcordanceIndex
 utility.cpp, 1533
 utility.h, 1550
computeDeathCounts
 TreeSurvival, 1065
ComputeDifferentialCorrelationZsparse
 DcVar, 218
ComputeDifferentialCorrelationZvals
 DcVar, 218
ComputeDifferentialCorrelationZ
 DcVar, 218
ComputeECScores
 EvaporativeCooling, 276
ComputeFreeEnergy
 EvaporativeCooling, 276
ComputeGRM
 ReliefF, 910
ComputeImportance
 EvaporativeCoolingPrivacy, 290
ComputeInstanceStats
 SNReliefF, 1014
ComputeInstanceToInstanceDistance
 Dataset, 149
ComputeInverseImportance
 EvaporativeCoolingPrivacy, 290
computePermutationImportance
 Forest, 316
 Tree, 1026
computePermutationImportanceInternal
 TreeClassification, 1040
 TreeProbability, 1048
 TreeRegression, 1057
 TreeSurvival, 1065
computePredictionAccuracyInternal
 Tree, 1026
 TreeClassification, 1040
 TreeProbability, 1048
 TreeRegression, 1057
 TreeSurvival, 1065
computePredictionError
 Forest, 316
computePredictionErrorInternal
 Forest, 316
 ForestClassification, 335
 ForestProbability, 341
 ForestRegression, 348
 ForestSurvival, 354
computePredictionMSE
 TreeRegression, 1057
ComputeScore
 ChiSquared, 86
ComputeScores
 AttributeRanker, 44
 ChiSquared, 87
 Deseq, 236
 Edger, 259
 EvaporativeCoolingPrivacy, 291
 RandomForest, 864
 ReliefSeqController, 928
 ReliefF, 910
ComputeScoresKopt
 ReliefSeqController, 928
ComputeScoresWithPValues
 ChiSquared, 87
computeSurvival
 TreeSurvival, 1066
computeTreePermutationImportanceInThread
 Forest, 316
ComputeWeightByDistanceFactors
 ReliefF, 910
ComputeQ
 InteractionNetwork, 474
condentropy
 Statistics.cpp, 1464
 Statistics.h, 1471
condition_variable
 Forest, 326
ConditionalEntropy
 Statistics.cpp, 1464
 Statistics.h, 1472
conditionalHaplotypeTest
 Plink, 787
conditioner
 Plink, 824
conditioner_mask
 Plink, 824
conditioning_snp_name
 par, 610
conditioning_snp_single
 par, 610

conditioning_snps
 par, 610
 conditioning_snps_file
 par, 610
 confidences
 BirdseedData, 56
 ConfigMap
 Insilico.h, 1341
 connMatrix
 InteractionNetwork, 483
 connect
 Rconnection, 873
 connectivityThreshold
 InteractionNetwork, 483
 connectivityThresholdAbs
 InteractionNetwork, 483
 consistentNonfounderMalePhaseGivenXGenotypes
 HaploPhase, 400
 consistentNonfounderPhaseGivenGenotypes
 HaploPhase, 400
 consistentNonfounderPhaseGivenParents
 HaploPhase, 400
 ConstructAttributeCart
 Statistics.cpp, 1465
 Statistics.h, 1472
 cont
 Rstrings, 974
 Rvector, 983
 continuousPhenotypeMinMax
 Dataset, 190
 control_freq
 HaploPhase, 413
 converged
 HaploWindow, 430
 Model, 532
 convertGenericVariantData
 Plink, 787
 convertPosition
 lookup2.cpp, 1372
 CoordinateTable
 EpistasisEQtl.h, 1224
 CoordinateTableClt
 EpistasisEQtl.h, 1224
 coordinates
 DcVar, 225
 EpistasisEQtl, 270
 copy
 Family, 307
 Locus, 508
 copyNumberVariation
 Variant, 1075
 correlation2SNP
 Plink, 787
 count
 CArgs, 63
 CNVIndivReport, 103
 MultiLocusGenotype, 539
 Perm, 767
 Rstrings, 973
 Rvector, 984
 Segment, 987
 count_baseline
 CNVIndivReport, 103
 count_intersects
 cnv.cpp, 1142
 cnv.h, 1145
 countCNVPerRegion
 Plink, 788
 countMeioses
 Individual, 459
 counter
 TreeClassification, 1043
 TreeProbability, 1051
 TreeRegression, 1060
 counter_per_class
 TreeClassification, 1043
 TreeProbability, 1051
 counts
 DgeData, 243
 countsFilename
 DgeData, 243
 covar
 Individual, 460
 covar_file
 par, 610
 covar_filename
 par, 610
 covariate
 Model, 532
 covsrt
 linear.cpp, 1366
 CP
 par, 611
 CreateDefaultForestForPheno
 RandomForest, 864
 CreateDiffCoexpMatrixNoME
 ExpressionDataSimulator, 302
 CreateDummyAlleles
 Dataset, 149
 createEmptyNode
 Tree, 1026
 createEmptyNodeInternal
 Tree, 1027
 TreeClassification, 1040
 TreeProbability, 1048
 TreeRegression, 1057
 TreeSurvival, 1066
 createFile

Rconnection, 873
createPossibleSplitVarSubset
Tree, 1027
CreateSimulation
ExpressionDataSimulator, 303
ctrlIdxCol
DcVar, 225
cumbet
cdflib.h, 1117
dcdflib.cpp, 1178
cumbin
cdflib.h, 1117
dcdflib.cpp, 1178
cumchi
cdflib.h, 1118
dcdflib.cpp, 1179
cumchn
cdflib.h, 1118
dcdflib.cpp, 1179
cumf
cdflib.h, 1118
dcdflib.cpp, 1179
cumfnc
cdflib.h, 1118
dcdflib.cpp, 1179
cumgam
cdflib.h, 1119
dcdflib.cpp, 1180
cummulativeProbabilities
EvaporativeCoolingPrivacy, 293
cumnbn
cdflib.h, 1119
dcdflib.cpp, 1180
cumnor
cdflib.h, 1119
dcdflib.cpp, 1180
cumpoi
cdflib.h, 1119
dcdflib.cpp, 1180
cumt
cdflib.h, 1120
dcdflib.cpp, 1181
cumulativeSetSum_WITHLABELS
Set, 991
cumulativeSetSum_WITHOUTLABELS
Set, 991
cur
Set, 994
curVarMap
EvaporativeCoolingPrivacy, 293
curVarNames
EvaporativeCoolingPrivacy, 293
current
Chap, 75
HaploPhase, 413
current_reps
Perm, 763
CurrentNumberCorrect
EvaporativeCoolingPrivacy, 291
currentTemp
EvaporativeCoolingPrivacy, 293
CvSetType
Insilico.h, 1344
cvSetType
DatasetInstance, 213

D
CentralityRanker, 71
d
Eigen, 262
SIInfo, 1008
stats.cpp, 1496
DATASET_LOAD_ERROR
Insilico.h, 1353
DATASET_TYPE
EvaporativeCoolingPrivacy.h, 1228
DEBUG_E
Statistics.cpp, 1464
DEBUG_Z
Statistics.cpp, 1464
DEFAULT_ALPHA
globals.h, 1263
DEFAULT_CONNECTIVITY_THRESHOLD
InteractionNetwork.h, 1361
DEFAULT_FDR
Insilico.h, 1354
DEFAULT_IMPORTANCE_MODE
globals.h, 1264
DEFAULT_MIN_NODE_SIZE_CLASSIFICATION
globals.h, 1264
DEFAULT_MIN_NODE_SIZE_PROBABILITY
globals.h, 1264
DEFAULT_MIN_NODE_SIZE_REGRESSION
globals.h, 1264
DEFAULT_MIN_NODE_SIZE_SURVIVAL
globals.h, 1264
DEFAULT_MINPROP
globals.h, 1264
DEFAULT_NUM_THREADS
globals.h, 1265
DEFAULT_NUM_TREE
globals.h, 1265
DEFAULT_PREDICTIONTYPE
globals.h, 1265
DEFAULT_PVALUE_THRESHOLD
Insilico.h, 1354
DEFAULT_PVALUE
Insilico.h, 1354

DEFAULT_SPLITRULE
 globals.h, 1265

DEFAULT_ZVALUE
 Insilico.h, 1354

DISALLOW_COPY_AND_ASSIGN
 ArgumentHandler, 35
 Data, 115
 DataChar, 124
 DataDouble, 129
 DataFloat, 134
 Forest, 317
 ForestClassification, 335
 ForestProbability, 341
 ForestRegression, 348
 ForestSurvival, 354
 globals.h, 1261
 Tree, 1027
 TreeClassification, 1040
 TreeProbability, 1048
 TreeRegression, 1057
 TreeSurvival, 1066

DOUBLE_DIGITS
 fisher.h, 1240

DOUBLE_EPS
 fisher.h, 1240

DOUBLE_XMAX
 fisher.h, 1241

DOUBLE_XMIN
 fisher.h, 1241

DPrime, 254
 dp, 255
 dpl, 255
 dpu, 255
 lod, 255

DT_ARRAY
 Rsrv.h, 1432

DT_BYTESTREAM
 Rsrv.h, 1433

DT_CHAR
 Rsrv.h, 1433

DT_DOUBLE
 Rsrv.h, 1433

DT_INT
 Rsrv.h, 1433

DT_LARGE
 Rsrv.h, 1433

DT_SEXP
 Rsrv.h, 1434

DT_STRING
 Rsrv.h, 1434

Data, 112
 ~Data, 114
 addSparseData, 115
 DISALLOW_COPY_AND_ASSIGN, 115

 Data, 114
 externalData, 118
 get, 115
 getAllValues, 115
 getIndex, 115
 getMaxNumUniqueValues, 116
 getNumCols, 116
 getNumRows, 116
 getNumUniqueDataValues, 116
 getUniqueDataValue, 116
 getVariableID, 117
 getVariableNames, 117
 index_data, 118
 loadFromFile, 117
 loadFromFileOther, 117
 loadFromFileWhitespace, 117
 max_num_unique_values, 119
 num_cols, 119
 num_cols_no_sparse, 119
 num_rows, 119
 num_rows_rounded, 119
 reserveMemory, 118
 set, 118
 sort, 118
 sparse_data, 119
 unique_data_values, 120
 variable_names, 120

 data
 DataChar, 125
 DataDouble, 130
 DataFloat, 135
 Forest, 326
 Rexp, 943
 Rmessage, 960
 Tree, 1031

 DataChar, 120
 ~DataChar, 123
 DISALLOW_COPY_AND_ASSIGN, 124
 data, 125
 DataChar, 123
 get, 124
 reserveMemory, 124
 set, 124

 DataDouble, 125
 ~DataDouble, 128
 DISALLOW_COPY_AND_ASSIGN, 129
 data, 130
 DataDouble, 128
 get, 129
 loadFromDatasetMask, 129
 loadFromPlink, 129
 reserveMemory, 129
 set, 130

 DataFloat, 131

~DataFloat, 133
DISALLOW_COPY_AND_ASSIGN, 134
data, 135
DataFloat, 133
get, 134
reserveMemory, 134
set, 134
dataIsSimulated
EvaporativeCoolingPrivacy, 293
Dataset, 135
 ~Dataset, 144
 alternatePhenotypesFilename, 187
 attributeAlleleCounts, 187
 attributeAlleles, 188
 AttributeInteractionInformation, 145
 attributeLevelsSeen, 188
 attributeMinorAllele, 188
 attributeMutationMap, 188
 attributeMutationTypes, 188
 attributeNames, 189
 attributesMask, 189
 attributesMaskPushed, 189
 CalculateDistanceMatrix, 145, 146
 CalculateGainMatrix, 146
 CalculateInteractionInformation, 147
 CalculateRegainMatrix, 147
 CheckHardyWeinbergEquilibrium, 148
 classColumn, 189
 classIndexes, 189
 ComputeInstanceToInstanceDistance, 149
 continuousPhenotypeMinMax, 190
 CreateDummyAlleles, 149
 Dataset, 144
 engine, 190
 ExcludeMonomorphs, 149
 ExtractAttributes, 149
 genotypeCounts, 190
 GetAlternatePhenotypesFilename, 150
 GetAttribute, 150
 GetAttributeAlleles, 151
 GetAttributeIndexFromName, 151
 GetAttributeMAF, 151
 GetAttributeMutationType, 152
 GetAttributeNames, 152
 GetAttributeRowCol, 153
 GetAttributeTiTvCounts, 153
 GetAttributeValue, 153, 154
 GetClassColumn, 154
 GetClassIndexes, 155
 GetClassProbability, 155
 GetClassValues, 155
 GetDistanceMetrics, 157
 GetFileAttributeName, 157
 GetFileNumericsName, 157
 GetInstance, 157
 GetInstanceIds, 158
 GetInstanceIndexForID, 158
 GetIntForGenotype, 158
 GetJukesCantorDistance, 159
 GetKimuraDistance, 159
 GetMAFs, 159
 GetMeanForNumeric, 159
 GetMinMaxForContinuousPhenotype, 161
 GetMinMaxForNumeric, 161
 GetNumeric, 161
 GetNumericIndexFromName, 162
 GetNumericMatrix, 162
 GetNumericRowCol, 162
 GetNumericValues, 163, 164
 GetNumericsFilename, 163
 GetNumericsNames, 163
 GetProbabilityValueGivenClass, 164
 GetRandomInstance, 165
 GetSnpFilename, 165
 GetVariableNames, 165
 HasAllelicInfo, 165
 hasAllelicInfo, 190
 HasAlternatePhenotypes, 166
 hasAlternatePhenotypes, 190
 HasContinuousPhenotypes, 166
 hasContinuousPhenotypes, 191
 HasGenotypes, 166
 hasGenotypes, 191
 HasNumerics, 166, 167
 hasNumerics, 191
 HasPhenotypes, 167
 hasPhenotypes, 191
 instanceIds, 191
 instanceIdsToLoad, 192
 instances, 192
 instancesMask, 192
 instancesMaskPushed, 192
 IsLoadableInstanceId, 167
 levelCounts, 192
 levelCountsByClass, 193
 LoadAlternatePhenotypes, 167
 LoadDataset, 168, 169
 LoadNumerics, 170
 LoadOtherDatasetInstances, 170
 LoadPrivacySim, 171
 LoadSnp, 171
 MaskGetAllVariableNames, 172
 MaskGetAttributeIndices, 172
 MaskGetAttributeMask, 172
 MaskGetInstanceIds, 173
 MaskGetInstanceIndices, 173
 MaskGetInstanceMask, 173
 MaskIncludeAllAttributes, 174

MaskIncludeAllInstances, 174
 maskIsPushed, 193
 MaskPopAll, 174
 MaskPushAll, 175
 MaskRemoveInstance, 175
 MaskRemoveVariable, 175
 MaskRemoveVariableType, 176
 MaskSearchInstance, 176
 MaskSearchVariableType, 177
 MaskWriteNewDataset, 177
 missingNumericValues, 193
 missingValues, 193
 NumAttributes, 178
 NumClasses, 178
 numDiffFuncPtr, 193
 numDiffMetricName, 194
 NumInstances, 178
 NumLevels, 178
 NumNumerics, 179
 NumVariables, 179
 numericsFilename, 194
 numericsIds, 194
 numericsMask, 194
 numericsMaskPushed, 195
 numericsMinMax, 195
 numericsNames, 195
 numericsSums, 195
 phenotypesIds, 195
 Print, 179
 PrintAttributeLevelsSeen, 179
 PrintClassIndexInfo, 180
 PrintLevelCounts, 180
 PrintMaskStats, 180
 PrintMissingValuesStats, 180
 PrintNumericsStats, 180
 PrintStats, 181
 PrintStatsSimple, 181
 ProcessExclusionFile, 181
 ResetNearestNeighbors, 182
 RunSnpDiagnosticTests, 182
 SNPHWE, 183
 SetDistanceMetrics, 182
 .snpDiffFuncPtr, 196
 .snpDiffMetricName, 197
 .snpNearestNeighborFuncPtr, 197
 .snpNearestNeighborMetricName, 197
 snpsFilename, 198
 SwapAttributes, 183
 UpdateAllLevelCounts, 184
 UpdateLevelCounts, 184
 WriteLevelCounts, 185
 WriteNewDataset, 185
 WriteNewPlinkCovarDataset, 186
 WriteNewPlinkPedDataset, 187
 WriteSnpTiTvInfo, 187
 dataset
 AttributeRanker, 47
 DatasetInstance, 213
 Deseq, 237
 Edger, 260
 EvaporativeCooling, 283
 ReliefSeqController, 932
 Dataset.cpp
 PP, 1155
 datasetFile
 RandomForest, 865
 DatasetInstance, 198
 ~DatasetInstance, 203
 AddInfluenceFactorD, 203
 AddNumeric, 203
 attributes, 212
 bestNeighborhoods, 212
 bestNeighborhoodsDiffClass, 212
 bestNeighborhoodsSameClass, 212
 classLabel, 213
 ClearInfluenceFactors, 204
 cvSetType, 213
 dataset, 213
 DatasetInstance, 202
 GetAttribute, 204
 GetAttributes, 204
 GetClass, 204
 GetCvSetType, 205
 GetDatasetPtr, 205
 GetId, 205
 GetInfluenceFactorD, 205
 GetNNearestInstances, 205, 206
 GetNumeric, 207
 GetNumerics, 207
 GetPredictedValueTau, 207
 id, 213
 LoadInstanceFromInstancePtr, 208
 LoadInstanceFromVector, 208
 neighborInfluenceFactorDs, 213
 NumAttributes, 208
 NumNumerics, 208
 numerics, 214
 predictedValueTau, 214
 Print, 209
 PrintDistancePairs, 209
 ResetNearestNeighbors, 209
 SetClass, 209
 SetCvSetType, 210
 SetDistanceSums, 210
 SetId, 211
 SetPredictedValueTau, 211
 SwapAttributes, 211
 DatasetInstance.cpp

T, [1157](#)
datasetTypeToExt
 Insilico.h, [1354](#)
dbetrm
 cdflib.h, [1120](#)
 dcdflib.cpp, [1181](#)
dbl2str
 helper.cpp, [1281](#)
 helper.h, [1302](#)
dbl2str_fixed
 helper.cpp, [1282](#)
 helper.h, [1302](#)
DcVar, [215](#)
 ~DcVar, [217](#)
 CalculateFdrBHThreshold, [217](#)
 casIdxCol, [225](#)
 CheckInputs, [217](#)
 chipSeqExpression, [225](#)
 chipSeqMode, [225](#)
 ComputeDifferentialCorrelationZsparse, [218](#)
 ComputeDifferentialCorrelationZvals, [218](#)
 ComputeDifferentialCorrelationZ, [218](#)
 coordinates, [225](#)
 ctrlIdxCol, [225](#)
 DcVar, [217](#)
 expressionMatrix, [226](#)
 FilterPvalues, [218](#)
 FindSnps, [218](#)
 FlattenPvals, [219](#)
 geneExprNames, [226](#)
 geneExprSubjects, [226](#)
 genotypeMatrix, [226](#)
 genotypeSubjects, [226](#)
 GetLocalCis, [219](#)
 GetRadius, [219](#)
 GetSnpsForTFs, [219](#)
 GetSnpsForTranscript, [219](#)
 GetTFIinfo, [220](#)
 GetTFRadius, [220](#)
 GetTF, [220](#)
 IsSnplnTFs, [220](#)
 LoadDefaultTranscriptionFactorLUT, [220](#)
 localCis, [226](#)
 MapPhenosToModel, [221](#)
 MapSnplIndexToPlinkPhenos, [221](#)
 nInnerLoop, [227](#)
 nOuterLoop, [227](#)
 numCombs, [227](#)
 pVals, [227](#)
 PrintState, [221](#)
 radius, [227](#)
 ReadCheckpoint, [221](#)
 ReadChipSeqFile, [221](#)
 ReadGeneExpressionFile, [222](#)
 ReadGenotypesFile, [222](#)
 ReadSnpLocationsFile, [222](#)
 ReadTranscriptCoordinates, [222](#)
 ReadTranscriptFactorCoordinates, [222](#)
 Run, [222](#)
 RunOMRFChipSeq, [223](#)
 RunOMRF, [223](#)
 RunPlink, [223](#)
 SetLocalCis, [223](#)
 SetRadius, [223](#)
 SetTFRadius, [224](#)
 SetTF, [224](#)
 snplInputType, [227](#)
 snpLocations, [228](#)
 snpNames, [228](#)
 SplitExpressionCaseControl, [224](#)
 tfMode, [228](#)
 tfRadius, [228](#)
 tfTableLoaded, [228](#)
 thisTFSnplIndices, [228](#)
 thisTranscriptSnplIndices, [229](#)
 totalTests, [229](#)
 transcriptFactorLUT, [229](#)
 WriteCheckpoint, [224](#)
 WriteResults, [224](#)
 zVals, [229](#)
DcVar.cpp
 chipseqComparatorAscending, [1196](#)
 pvalComparatorAscending, [1196](#)
 snpinfoComparatorAscending, [1196](#)
DcVar.h
 CHECKPOINT_FILENAME, [1199](#)
 CHIP_SEQ_EXTRACT_FIELD_IDX, [1198](#)
 CHIP_SEQ_INFO_LIST, [1198](#)
 SNP_EXTRACT_FIELD_IDX, [1199](#)
 SNP_INFO_LIST_IT, [1198](#)
 SNP_INFO_LIST, [1198](#)
 SNP_INPUT_TYPE, [1199](#)
dcdflib.cpp
 algdiv, [1171](#)
 alngam, [1171](#)
 alnrel, [1172](#)
 apser, [1172](#)
 atol, [1162, 1163](#)
 basym, [1172](#)
 bcorr, [1172](#)
 betaln, [1172](#)
 bfrac, [1173](#)
 bgrat, [1173](#)
 bpser, [1173](#)
 bratio, [1173](#)
 brcmp1, [1174](#)
 brcomp, [1174](#)
 bup, [1174](#)

cdfbet, 1174
cdfbin, 1175
cdfchi, 1175
cdfchn, 1175
cdff, 1176
cdffnc, 1176
cdfgam, 1176
cdfnbn, 1177
cdfnor, 1177
cdfpoi, 1177
cdft, 1178
cumbet, 1178
cumbin, 1178
cumchi, 1179
cumchn, 1179
cumf, 1179
cumfnc, 1179
cumgam, 1180
cumnbn, 1180
cumnor, 1180
cumpoi, 1180
cumt, 1181
dbetrm, 1181
dennor, 1163
devlpl, 1181
dexpm1, 1181
dg, 1163
dinvnr, 1182
dinvr, 1182
dlanor, 1182
dln1mx, 1182
dln1px, 1182
dlnbet, 1183
dlngam, 1183
dlsqpi, 1163
done, 1164
dstinv, 1183
dstrem, 1183
dstrzr, 1183
dt1, 1184
dzror, 1184
E0000, 1184
E0001, 1184
eps, 1164
erf1, 1185
erfc1, 1185
esum, 1185
exparg, 1185
fifdint, 1186
fifdmax1, 1186
fifdmin1, 1186
fifdsign, 1186
fifidint, 1186
fifmod, 1187
fpser, 1187
ftnstop, 1187
ftol, 1164
gam1, 1187
gaminv, 1187
gamln, 1188
gamln1, 1188
grat1, 1188
gratio, 1188
gsumln, 1189
half, 1164
hln2pi, 1164, 1165
inf, 1165, 1166
maxdf, 1166
maxit, 1166
ncoef, 1166
nhalf, 1167
one, 1167
psi, 1189
qsmall, 1167, 1168
qtired, 1168
qxmon, 1168
r2pi, 1168
rcomp, 1189
rexp, 1189
rlog, 1189
rlog1, 1190
spmpar, 1190
stvaln, 1190
tent4, 1168
tol, 1169, 1170
Xgamm, 1190
zero, 1170, 1171

dcdflib.h

- cdfbet, 1191
- cdfbin, 1191
- cdfchi, 1192
- cdfchn, 1192
- cdff, 1192
- cdffnc, 1193
- cdfgam, 1193
- cdfnbn, 1193
- cdfnor, 1194
- cdfpoi, 1194
- cdft, 1194

- dcvar_chip_seq_file
- par, 611
- dcvar_gene_expression_file
- par, 611
- dcvar_genotypes_file
- par, 611
- dcvar_pfilter_type
- par, 611
- dcvar_pfilter_value

par, 611
dcvar_radius
 par, 612
dcvar_resume.snp
 par, 612
dcvar.snp.locations_file
 par, 612
dcvar.var.model
 par, 612
debug
 par, 612
DebugMessage
 InteractionNetwork, 474
debugMode
 EpistasisEQtl, 270
 InteractionNetwork, 483
decode
 Rstrings, 973
deconvolutionAlpha
 par, 612
deconvolutionBeta
 par, 613
deconvolutionControl
 par, 613
Deconvolve
 InteractionNetwork, 475
default_Rsrv_port
 Rsrv.h, 1432
defineCowChromosomes
 helper.cpp, 1282
 helper.h, 1303
defineDogChromosomes
 helper.cpp, 1282
 helper.h, 1303
defineHorseChromosomes
 helper.cpp, 1282
 helper.h, 1303
defineHumanChromosomes
 helper.cpp, 1282
 helper.h, 1303
defineMouseChromosomes
 helper.cpp, 1283
 helper.h, 1303
defineRiceChromosomes
 helper.cpp, 1283
 helper.h, 1304
defineSheepChromosomes
 helper.cpp, 1283
 helper.h, 1304
degrees
 _RipmResult, 28
 InteractionNetwork, 483
deleteIndividuals
 Plink, 788
deleteIndividualsNumeric
 Plink, 788
deleteSNPs
 Plink, 788, 789
delimit
 IDFile, 444
delimit_char
 NList, 545
deltaQ
 EvaporativeCoolingPrivacy, 294
dennor
 dcdflib.cpp, 1163
deparseNumberList
 NList, 543
deparseStringList
 NList, 544
dependent_varID
 Forest, 326
 Tree, 1031
depvarname
 ArgumentHandler, 37
 par, 613
derefl_less, 230
 operator(), 231
derefl_less_bcw, 231
 operator(), 232
Deseq, 233
 ~Deseq, 235
 ComputeScores, 236
 dataset, 237
 Deseq, 235
 PrintScores, 236
 ReadDeseqScores, 236
 scores, 237
 WriteDatasetInMayoFormat, 237
 WriteScores, 237
DetectClassType
 Insilico.cpp, 1332
 Insilico.h, 1348
DetermineRandomJungleTreeType
 Insilico.h, 1348
determineTestType
 Chap, 74
deterministic_varIDs
 Forest, 326
 Tree, 1031
devlpl
 cdflib.h, 1120
 dcdflib.cpp, 1181
dexpm1
 cdflib.h, 1120
 dcdflib.cpp, 1181
df
 ChapModel, 78

dfam_sibs
 par, 613
 dfam_tdt
 par, 613
 dfam_unrelateds
 par, 613
 dg
 dcdflib.cpp, 1163
 DgeData, 238
 ~DgeData, 240
 counts, 243
 countsFilename, 243
 DgeData, 240
 geneNames, 243
 GetCountsFilename, 240
 GetGeneCountsSum, 240
 GetGeneMinMax, 241
 GetGeneNames, 241
 GetNormalizationFactors, 241
 GetNumGenes, 241
 GetNumSamples, 241
 GetSampleCounts, 242
 GetSampleNames, 242
 GetSamplePhenotype, 242
 GetSequencingDepthForSample, 242
 hasNormFactors, 244
 LoadData, 242
 minMaxGeneCounts, 244
 minMaxSampleCounts, 244
 normFactors, 244
 normsFilename, 244
 phenosFilename, 245
 phenotypes, 245
 PrintSampleStats, 243
 sampleNames, 245
 sampleZeroes, 245
 sumGeneCounts, 245
 dict
 IDHelper, 451
 diff
 EvaporativeCoolingPrivacy, 294
 diff_concordant_overlap
 Plink, 824
 diff_nonmissing_overlap
 Plink, 824
 diff_overlap
 Plink, 825
 diffAMM
 DistanceMetrics.cpp, 1205
 DistanceMetrics.h, 1211
 diffEuclidean
 DistanceMetrics.cpp, 1205
 DistanceMetrics.h, 1212
 diffGMM
 DistanceMetrics.cpp, 1206
 DistanceMetrics.h, 1212
 diffImportance
 EvaporativeCoolingPrivacy, 294
 diffKM
 DistanceMetrics.cpp, 1206
 DistanceMetrics.h, 1213
 diffManhattan
 DistanceMetrics.cpp, 1206
 DistanceMetrics.h, 1213
 diffNCA
 DistanceMetrics.cpp, 1207
 DistanceMetrics.h, 1214
 diffPredictedValueTau
 DistanceMetrics.cpp, 1207
 DistanceMetrics.h, 1214
 diffScores
 EvaporativeCoolingPrivacy, 294
 diffTITV
 DistanceMetrics.cpp, 1208
 DistanceMetrics.h, 1215
 dimM
 ExpressionDataSimulator, 304
 dimN
 ExpressionDataSimulator, 304
 dinvr
 cdflib.h, 1121
 dcdflib.cpp, 1182
 dinvr
 cdflib.h, 1121
 dcdflib.cpp, 1182
 disable_buffer
 gzfilebuf, 374
 disconnect
 Rconnection, 873
 discordant_parents
 Family, 308
 par, 614
 disp_r1
 par, 614
 disp_r2
 par, 614
 disp_r_window
 par, 614
 disp_r_window_kb
 par, 614
 disp_r_window_r2
 par, 614
 disp_r_window.snp
 par, 615
 display
 helper.cpp, 1283, 1284
 helper.h, 1304, 1305
 IDGroup, 447

OptionSet, 551
display_ci
 par, 615
display_cnv_track
 par, 615
display_hap_freqs
 par, 615
display_indivReport
 Plink, 789
display_listByAllele
 Plink, 789
display_pairList
 Plink, 789
display_phase_probs
 par, 615
display_phase_probs_wide
 par, 615
display_recoded_LONG
 Plink, 790
display_recoded_MUTLIST
 Plink, 790
display_recoded_PEDFILE_AD
 Plink, 790
display_recoded_PEDFILE_transpose
 Plink, 790
display_recoded_PEDFILE
 Plink, 790
display_segment_long
 par, 616
display_twolocus
 Plink, 790
displayFamTran
 nonfounderphasing.cpp, 1383
displayGMULTI
 Plink, 791
displayGeneReport
 Plink, 791
displayGenomePV
 Plink, 791
displayGenomeWideInfo
 Plink, 791
displayHaploGroups
 whap.cpp, 1570
displayHelp
 ArgumentHandler, 35
displayLine
 helper.cpp, 1284
 helper.h, 1305
displayPoolVerbose
 homozyg.cpp, 1321
displayQTMeans
 Plink, 791
displayRareRange
 Plink, 792
displayResults
 LinearModel, 500
 LogisticModel, 513
 Model, 525
displayScoresPerson
 elf.cpp, 1219
displayScoresRegion
 elf.cpp, 1219
displaySegmentsBED
 Plink, 792
displaySegmentsLong
 Plink, 792
displayVersion
 ArgumentHandler, 36
displayWindow
 RCount, 879
dist
 LDPair, 491
 Neighbour, 541
distance_matrix
 par, 616
distanceMatrix
 ReliefF, 914
distanceMatrixFilename
 par, 616
DistanceMetrics.cpp
 CheckMissing, 1204
 CheckMissingNumeric, 1204
 diffAMM, 1205
 diffEuclidean, 1205
 diffGMM, 1206
 diffKM, 1206
 diffManhattan, 1206
 diffNCA, 1207
 diffPredictedValueTau, 1207
 diffTITV, 1208
 norm, 1208
DistanceMetrics.h
 CheckMissing, 1210
 CheckMissingNumeric, 1211
 diffAMM, 1211
 diffEuclidean, 1212
 diffGMM, 1212
 diffKM, 1213
 diffManhattan, 1213
 diffNCA, 1214
 diffPredictedValueTau, 1214
 diffTITV, 1215
 norm, 1215
DistancePair
 Insilico.h, 1341
DistancePairs
 Insilico.h, 1342
DistancePairsIt

Insilico.h, 1342
 dlanor
 cdflib.h, 1121
 dcdflib.cpp, 1182
 dln1mx
 cdflib.h, 1121
 dcdflib.cpp, 1182
 dln1px
 cdflib.h, 1121
 dcdflib.cpp, 1182
 dlnbet
 cdflib.h, 1122
 dcdflib.cpp, 1183
 dlngam
 cdflib.h, 1122
 dcdflib.cpp, 1183
 dlsqpi
 dcdflib.cpp, 1163
 do_coexpression_all
 par, 616
 do_coexpression_casecontrol
 par, 616
 do_covariance_matrix
 par, 616
 do_dcgain_abs
 par, 617
 do_dcvar
 par, 617
 do_dcvar_chipseq
 par, 617
 do_dcvar_pfilter
 par, 617
 do_deconvolution
 par, 617
 do_differential_coexpression
 par, 617
 do_differential_modularity
 par, 618
 do_dmgain_abs
 par, 618
 do_ec
 par, 618
 do_ec_privacy
 par, 618
 do_iqtl
 par, 618
 do_iqtl_tf
 par, 618
 do_iterative_removal
 par, 619
 do_iterative_removal_ec
 par, 619
 do_iterative_write_scores
 par, 619
 do_kopt
 par, 619
 do_modularity
 par, 619
 do_normalize_scores
 par, 619
 do_not_load_snps
 par, 620
 do_numeric_extract
 par, 620
 do_numeric_lowval_filter
 par, 620
 do_numeric_lowvar_filter
 par, 620
 do_numeric_standardize
 par, 620
 do_numeric_summary
 par, 620
 do_randomforest
 par, 621
 do_ranker_permutation
 par, 621
 do_ranking
 par, 621
 do_regain
 par, 621
 do_regain_post
 par, 621
 do_regain_pvalue_threshold
 par, 621
 do_relieff
 par, 622
 do_rfprobability
 par, 622
 do_ripm
 par, 622
 do_to_lower
 insilico::do_to_lower, 247, 248
 do_to_upper
 insilico::do_to_upper, 250, 251
 do_write_best_k
 par, 622
 do_write_each_k_scores
 par, 622
 doFdrPrune
 Regain, 896
 doRegression
 Plink, 792
 doRemovePercent
 ReliefSeqController, 932
 ReliefF, 915
 doesFileExist
 helper.cpp, 1284
 helper.h, 1305

dof
phdr, 772

dom
SimParametersQT, 1004

domModel
RCount, 881

dominance
Model, 532

done
dcdflib.cpp, 1164

done_global_pihat
par, 622

donesocks
sisocks.h, 1453

dosage
HaploPhase, 401

dosage.cpp
PP, 1217
setUpQScoring, 1217

dosage1
GVariant, 368

dosage2
GVariant, 368

dosage_assoc
par, 623

dosage_file
par, 623

dosage_hard_call
par, 623

dosage_hard_call_thresh
par, 623

dosage_hard_call_thresh2
par, 623

dosage_hasMap
par, 623

double2, 252
double2, 252, 253
operator<, 253
operator==, 253
p1, 253
p2, 254

doubleArray
Rdouble, 887

doubleAt
Rdouble, 887

downcoding
HaploPhase, 414

dp
DPrime, 255
PairwiseLinkage, 559

dp_lower
PairwiseLinkage, 559

dp_upper
PairwiseLinkage, 559

dpl
DPrime, 255

dprime
HaploPhase, 401
SimParameters, 1000
SimParametersQT, 1004

dpu
DPrime, 255

drawWithoutReplacementFisherYates
utility.cpp, 1533

drawWithoutReplacementKnuth
utility.cpp, 1534
utility.h, 1550

drawWithoutReplacementSimple
utility.cpp, 1534
utility.h, 1551

drawWithoutReplacementSkip
utility.cpp, 1535
utility.h, 1551

drawWithoutReplacementWeighted
utility.cpp, 1535, 1536
utility.h, 1552

driverSCREEPI
Plink, 792

drop_sets
par, 624

dropAlleles
Perm, 763

dropDownSamplePermuted
Tree, 1027

dropNotSet
Set, 992

dst
Plink, 825

dstdnorm
utility.cpp, 1536
utility.h, 1553

dstinv
cdflib.h, 1122
dcdflib.cpp, 1183

dstrem
cdflib.h, 1122
dcdflib.cpp, 1183

dstrzr
cdflib.h, 1122
dcdflib.cpp, 1183

dt1
cdflib.h, 1123
dcdflib.cpp, 1184

dtop
Rsrv.h, 1434

dummy
par, 624

dummy_nind

par, 624
 dummy_nsnp
 par, 624
 dummyLoader
 Plink, 793
 dump_all
 Perm, 767
 dump_best
 Perm, 767
 dump_clst
 par, 624
 dump_covar
 par, 624
 dump_covar_dummy_coding
 par, 625
 dump_covar_with_phenotype
 par, 625
 dump_gene
 par, 625
 dump_genename
 par, 625
 dumpped
 par, 625
 dwrk
 fisher.cpp, 1232
 dzror
 cdflib.h, 1123
 dcdflib.cpp, 1184

E00
 Plink, 825

E0000
 cdflib.h, 1123
 dcdflib.cpp, 1184

E0001
 cdflib.h, 1123
 dcdflib.cpp, 1184

E01
 Plink, 825

E02
 Plink, 825

E10
 Plink, 825

E11
 Plink, 826

E12
 Plink, 826

E20
 Plink, 826

E21
 Plink, 826

E22
 Plink, 826

EACCES

sisocks.h, 1453
EADDRINUSE
 sisocks.h, 1453
EAFNOSUPPORT
 sisocks.h, 1453
EALREADY
 sisocks.h, 1453
EBADF
 sisocks.h, 1453
ECONNREFUSED
 sisocks.h, 1454
EFAULT
 sisocks.h, 1454
EINPROGRESS
 sisocks.h, 1454
EINVAL
 sisocks.h, 1454
EISCONN
 sisocks.h, 1454
ENETUNREACH
 sisocks.h, 1454
ENOTSOCK
 sisocks.h, 1455
EOPNOTSUPP
 sisocks.h, 1455
EPS_OVERLAP
 cnv.cpp, 1144
EPS
 CRandom, 109
 multi.cpp, 1379
ERR_IOerror
 Rsrv.h, 1436
ERR_Rerror
 Rsrv.h, 1437
ERR_accessDenied
 Rsrv.h, 1434
ERR_auth_failed
 Rsrv.h, 1434
ERR_conn_broken
 Rsrv.h, 1435
ERR_data_overflow
 Rsrv.h, 1435
ERR_detach_failed
 Rsrv.h, 1435
ERR_inv_cmd
 Rsrv.h, 1435
ERR_inv_par
 Rsrv.h, 1436
ERR_notOpen
 Rsrv.h, 1436
ERR_object_too_big
 Rsrv.h, 1436
ERR_out_of_mem
 Rsrv.h, 1436

ERR_session_busy
 Rsrv.h, 1437

ERR_unknownCmd
 Rsrv.h, 1437

ERR_unsupportedCmd
 Rsrv.h, 1437

ETIMEDOUT
 sisocks.h, 1455

EV_tqli
 stats.cpp, 1482
 stats.h, 1501

EV_tred2
 stats.cpp, 1482
 stats.h, 1501

EWOULDBLOCK
 sisocks.h, 1455

ecAlgorithmSteps
 par, 625

EcAlgorithmType
 Insilico.h, 1345

ecltAlgorithm
 par, 626

EcltAlgorithmType
 Insilico.h, 1345

eclterNumToRemove
 par, 626

eclterPercentToRemove
 par, 626

ecMeAlgorithm
 par, 626

EcMeAlgorithmType
 Insilico.h, 1345

ecNumTarget
 par, 626

ecOptimizeTemp
 par, 626

ecPrivacyFinalTemp
 par, 627

ecPrivacyHoldoutFile
 par, 627

ecPrivacyPercentSignal
 par, 627

ecPrivacyStartTemp
 par, 627

ecPrivacyTau
 par, 627

ecPrivacyTestFile
 par, 627

ecPrivacyTrainFile
 par, 628

ecPrivacyUpdateFrequency
 par, 628

ecScores
 EvaporativeCooling, 283

Edger, 256
 ~Edger, 258

ComputeScores, 259

dataset, 260

Edger, 258

PrintScores, 259

ReadEdgerScores, 259

scores, 260

WriteDatasetInMayoFormat, 260

WriteScores, 260

Eigen, 261
 d, 262
 set, 262
 z, 262

eigen_lapack
 lapackf.cpp, 1363
 lapackf.h, 1364

eigenvalues
 stats.cpp, 1481
 stats.h, 1500

eigenvectors
 stats.cpp, 1481
 stats.h, 1500

elf.cpp
 displayScoresPerson, 1219
 displayScoresRegion, 1219
 PP, 1220

elf_baseline
 par, 628

elf_pcemode
 par, 628

elf_pcemode_2sided
 par, 628

elfBaseline
 Plink, 793

em.cpp
 LOG, 1221

empiricalSetPValues
 Set, 992

empiricalVariance
 HaploPhase, 414

enable_buffer
 gzfilebuf, 374

endOfFile
 ZInput, 1083

engine
 Dataset, 190
 EvaporativeCoolingPrivacy, 294

Entropy
 Statistics.cpp, 1465
 Statistics.h, 1473

entryByTagName
 Rlist, 954

enumerateAllPhases

HaploPhase, 401
 enumerateGenogroups
 HaploWindow, 427
 enumerateHaplotypes
 HaploPhase, 401
 HaploWindow, 427
 enumerateNonfounderPhase
 HaploPhase, 401
 enumeratePhase
 HaploWindow, 427
 enumeratePhasedWindows
 HaploPhase, 402
 epi.cpp
 LOG, 1222
 epi1
 Plink, 826
 epi2
 Plink, 827
 epi_alpha1
 par, 628
 epi_alpha2
 par, 629
 epi_caseonly
 par, 629
 epi_caseonly_kb_gap
 par, 629
 epi_filter
 par, 629
 epi_genebased
 par, 629
 epi_quickscan
 par, 629
 epistasis
 par, 630
 EpistasisEQtl, 263
 ~EpistasisEQtl, 265
 badModels, 269
 CheckInputs, 265
 coordinates, 270
 debugMode, 270
 EpistasisEQtl, 265
 GetLocalCis, 265
 GetRadius, 265
 GetSnpsForTFs, 266
 GetSnpsForTranscript, 266
 GetTFIInfo, 266
 GetTRadius, 266
 GetTF, 266
 goodModels, 270
 IsSnpInTFs, 267
 LoadDefaultTranscriptionFactorLUT, 267
 localCis, 270
 nInnerLoop, 270
 nOuterLoop, 270
 outerLoopSnps, 271
 PrintState, 267
 radius, 271
 ReadTranscriptCoordinates, 267
 ReadTranscriptFactorCoordinates, 267
 resultsMatrixBetas, 271
 resultsMatrixPvals, 271
 Run, 268
 RunEqtl, 268
 RunIqlCisTrans, 268
 RunIqlFull, 268
 SetDebugMode, 268
 SetLocalCis, 269
 SetRadius, 269
 SetTFRadius, 269
 SetTF, 269
 tfMode, 271
 tfRadius, 271
 tfTableLoaded, 272
 thisTFSnpIndices, 272
 thisTranscriptSnpIndices, 272
 transcriptFactorLUT, 272
 EpistasisEQtl.h
 COORD_FIELDS, 1224
 CoordinateTable, 1224
 CoordinateTableCl, 1224
 TransciptFactorTable, 1224
 TransciptFactorTableCl, 1224
 eps
 dcdflib.cpp, 1164
 epsilon
 par, 630
 eqid
 IDField, 441
 equalSplit
 utility.cpp, 1537
 utility.h, 1553
 equiv
 IDField, 441
 IDFile, 444
 erf1
 cdflib.h, 1124
 dcdflib.cpp, 1185
 erfc1
 cdflib.h, 1124
 dcdflib.cpp, 1185
 error
 helper.cpp, 1284
 helper.h, 1305
 estimate
 TreeClassification, 1040
 TreeRegression, 1058
 esum
 cdflib.h, 1124

dcdflib.cpp, 1185
eval
 Rconnection, 873
EvaporateWorstAttributes
 EvaporativeCoolingPrivacy, 291
evaporatedAttributes
 EvaporativeCooling, 284
EvaporativeCooling, 273
 ~EvaporativeCooling, 276
algorithmType, 282
analysisType, 283
bestClassificationError, 283
classificationErrors, 283
ComputeClassificationErrorRandomForest, 276
ComputeECScores, 276
ComputeFreeEnergy, 276
dataset, 283
ecScores, 283
evaporatedAttributes, 284
EvaporativeCooling, 275
freeEnergyScores, 284
GetAlgorithmType, 277
GetECScores, 277
GetInteractionScores, 277
GetMaineffectScores, 277
interactionAlgorithm, 284
interactionScores, 284
itAlgorithmType, 284
maineffectAlgorithm, 285
maineffectScores, 285
meAlgorithmType, 285
numRFThreads, 285
numTargetAttributes, 285
numToRemoveNextIteration, 286
numToRemovePerIteration, 286
optimalTemperature, 286
OptimizeTemperature, 278
optimizeTemperature, 286
PP, 286
PrintAllAttributeScores, 278
PrintAllScoresTabular, 278
PrintAttributeScores, 278
PrintInteractionAttributeScores, 279
PrintKendallTaus, 279
PrintMainEffectAttributeScores, 279
RemoveWorstAttributes, 280
RunRandomForest, 280
RunReliefF, 280
temperatures, 287
WriteAttributeScores, 280
WriteClassificationErrors, 282
WriteTemperatures, 282
EvaporativeCoolingPrivacy, 287
 ~EvaporativeCoolingPrivacy, 289
attributeProbability, 293
CheckDetectedAttributes, 289
ClassifyAttributeSet, 290
ComputeAttributeProbabilities, 290
ComputeBestAttributesErrors, 290
ComputeImportance, 290
ComputeInverseImportance, 290
ComputeScores, 291
cummulativeProbabilities, 293
curVarMap, 293
curVarNames, 293
CurrentNumberCorrect, 291
currentTemp, 293
dataIsSimulated, 293
deltaQ, 294
diff, 294
diffImportance, 294
diffScores, 294
engine, 294
EvaporateWorstAttributes, 291
EvaporativeCoolingPrivacy, 289
finalTemp, 294
GenerateRandomUniformProbabilities, 291
GetKeptRemoved, 291
holdError, 295
holdout, 295
holdoutErrors, 295
holdoutImportance, 295
holdoutInvImportance, 295
iteration, 295
keepAttrs, 296
MAX_ITERATIONS, 296
numInstances, 296
numSignalsInData, 296
numVariables, 296
possiblyRemove, 296
PP, 297
PrintState, 292
Q_EPS, 297
randUniformProbs, 297
randUniformValue, 297
randomForestPredictError, 297
removeAttrs, 297
RemoveImportanceScore, 292
scaledProbabilities, 298
signalNames, 298
startTemp, 298
summedProbabilities, 298
tau, 298
test, 298
testError, 299
testErrors, 299
threshold, 299
tolerance, 299

train, 299
 trainError, 299
 trainErrors, 300
 trainImportance, 300
 trainInvImportance, 300
 update, 300
 updateInterval, 300
 UpdateTemperature, 292
 UsingSimData, 292
 WriteBestAttributes, 292
EvaporativeCoolingPrivacy.h
 DATASET_TYPE, 1228
 ResultsLists, 1228
exclude_file
 par, 630
exclude_set
 par, 630
ExcludeMonomorphs
 Dataset, 149
excludeSnps
 BirdseedData, 56
excludeSnpsFilename
 BirdseedData, 56
exitMessage
 MainExitException, 517
expandGenogroups
 HaploWindow, 428
expandNumberList
 NList, 544
exparg
 cdflib.h, 1124
 dcdflib.cpp, 1185
expectedContingencyTable
 ChiSquared, 89
expectedOverlap
 Plink, 827
expectedOverlapBaseline
 Plink, 827
exportArff
 par, 630
exportDelimited
 par, 630
exportDelimiter
 par, 631
ExpressionDataSimulator, 301
 ~ExpressionDataSimulator, 302
 colNames, 304
 CreateDiffCoexpMatrixNoME, 302
 CreateSimulation, 303
 dimM, 304
 dimN, 304
 ExpressionDataSimulator, 302
 GetData, 303
 n1, 304
 n2, 304
 phenosBase, 304
 PP, 305
 SimulateData, 303
 simulatedDataBase, 305
 simulatedDataHoldout, 305
 simulatedDataTest, 305
 simulatedDataTrain, 305
 subIds, 305
ExpressionDataSimulator.h
 SplitType, 1229
expressionMatrix
 DcVar, 226
externalData
 Data, 118
extract_before_exclude
 par, 631
extract_file
 par, 631
extract_set
 par, 631
ExtractAttributes
 Dataset, 149
extractExcludeSet
 Plink, 793

f

ClumpPair, 98
 HaploPhase, 414
 HaploWindow, 430
 ProxyResult, 858
 ResultTrio, 936

f10act

fisher.cpp, 1234

f11act

fisher.cpp, 1234

f2xact

fisher.cpp, 1234

f3xact

fisher.cpp, 1235

f4xact

fisher.cpp, 1236

f5xact

fisher.cpp, 1236

f6xact

fisher.cpp, 1236

f7xact

fisher.cpp, 1237

f8xact

fisher.cpp, 1237

f9xact

fisher.cpp, 1237

FDR_BH

helper.h, 1305

FIX_IBD
 par, 634
FIX_p
 par, 634
FIXED_p
 par, 634
FIXED
 par, 634
FPMIN
 helper.cpp, 1279
 stats.cpp, 1479
factln
 genepi.cpp, 1255
factorial
 stats.cpp, 1482
 stats.h, 1501
failureValue
 Regain, 897
failures
 Regain, 896
famfile
 par, 631
Family, 306
 B, 308
 copy, 307
 discordant_parents, 308
 Family, 307
 include, 308
 kid, 308
 mat, 308
 operator=, 307
 parents, 309
 pat, 309
 sibship, 309
 singleton, 309
 TDT, 309
family
 Individual, 460
 Plink, 827
 Rconnection, 876
FamilyTransmissions, 310
 mt, 311
 mu, 311
 operator<, 310
 pt, 311
 pu, 311
fast_binary
 par, 631
fast_epistasis
 par, 632
fdrPrune
 Regain, 892
fexact
 fisher.cpp, 1237
 fisher.h, 1243
fid
 Individual, 460
field
 IDValue, 456
fieldMap
 IDHelper, 452
fields
 IDFile, 444
 IDHelper, 452
fifdint
 cdflib.h, 1125
 dcdflib.cpp, 1186
fifdmax1
 cdflib.h, 1125
 dcdflib.cpp, 1186
fifdmin1
 cdflib.h, 1125
 dcdflib.cpp, 1186
fifdsign
 cdflib.h, 1125
 dcdflib.cpp, 1186
fifidint
 cdflib.h, 1125
 dcdflib.cpp, 1186
fifmod
 cdflib.h, 1126
 dcdflib.cpp, 1187
file
 ArgumentHandler, 37
 gzfilebuf, 379
 IDGroup, 447
fileCompress
 zed.cpp, 1572
 zed.h, 1574
fileUncompress
 zed.cpp, 1573
 zed.h, 1574
filename
 clump_LD, 95
 IDFile, 444
 ZInput, 1084
 ZOutput, 1089
fileroot
 par, 632
files
 IDHelper, 452
filter_cases
 par, 632
filter_controls
 par, 632
filter_females
 par, 632
filter_filename

par, 632
 filter_founders
 par, 633
 filter_males
 par, 633
 filter_nonfounders
 par, 633
 filter_on_covar
 par, 633
 filter_value
 par, 633
 filterOnCase
 Plink, 793
 filterOnControl
 Plink, 793
 filterOnCovariate
 Plink, 793
 filterOnFemale
 Plink, 794
 filterOnFounder
 Plink, 794
 filterOnMale
 Plink, 794
 filterOnNonFounder
 Plink, 794
 FilterPvalues
 DcVar, 218
 filterQualGenotypes
 Plink, 794
 filterQualSNPs
 Plink, 794
 filterRanges
 annot.cpp, 1094
 helper.cpp, 1285
 helper.h, 1306
 filterSNPs
 Plink, 795
 filters.cpp
 MISSING, 1230
 finalTemp
 EvaporativeCoolingPrivacy, 294
 find
 CArgs, 63
 findAllHomozygousRuns
 Plink, 795
 findAllIndividuals
 IDHelper, 449
 findBestSplit
 TreeClassification, 1041
 TreeProbability, 1049
 TreeRegression, 1058
 TreeSurvival, 1066
 findBestSplitMaxstat
 TreeRegression, 1058
 TreeSurvival, 1067
 findBestSplitValueAUC
 TreeSurvival, 1067
 findBestSplitValueLargeQ
 TreeClassification, 1041
 TreeProbability, 1049
 TreeRegression, 1058
 findBestSplitValueLogRank
 TreeSurvival, 1067
 findBestSplitValueLogRankUnordered
 TreeSurvival, 1068
 findBestSplitValueSmallQ
 TreeClassification, 1041
 TreeProbability, 1049
 TreeRegression, 1058
 findBestSplitValueUnordered
 TreeClassification, 1041
 TreeProbability, 1049
 TreeRegression, 1059
 findHomoRuns
 Plink, 795
 findHomoWindow
 Plink, 795
 findIBSRuns
 Plink, 795
 findMissRuns
 Plink, 796
 findSegments
 Plink, 796
 FindSnps
 DcVar, 218
 findUniqueIndividual
 IDHelper, 449
 finish
 Segment, 987
 finishWindow
 HaploPhase, 414
 finished
 HWindow, 436
 Perm, 763
 firstWord
 NList, 545
 fisher
 fisher.cpp, 1238
 fisher.h, 1243
 fisher.cpp
 alogam, 1233
 dwrk, 1232
 f10act, 1234
 f11act, 1234
 f2xact, 1234
 f3xact, 1235
 f4xact, 1236
 f5xact, 1236

f6xact, 1236
f7xact, 1237
f8xact, 1237
f9xact, 1237
fexact, 1237
fisher, 1238
gammmds, 1238
i_int, 1232
i_real, 1232
isort, 1238
iwork, 1238
iwrk, 1232
max, 1233
min, 1233
prterr, 1239
rwrk, 1233
SINT_MAX, 1233
fisher.h
 DOUBLE_DIGITS, 1240
 DOUBLE_EPS, 1240
 DOUBLE_XMAX, 1241
 DOUBLE_XMIN, 1241
 fexact, 1243
 fisher, 1243
 M_PI, 1241
 PI, 1241
 R_EXT_BOOLEAN_H_, 1241
 R_EXT_CONSTANTS_H_, 1241
 R_EXT_MEMORY_H_, 1242
 R_alloc, 1243
 R_gc, 1243
 Rboolean, 1242
 S_alloc, 1244
 S_realloc, 1244
 SINGLE_BASE, 1242
 SINGLE_EPS, 1242
 SINGLE_XMAX, 1242
 SINGLE_XMIN, 1242
 vmaxget, 1244
 vmaxset, 1244
fisher_test
 par, 633
fitConverged
 Model, 526
fitLDSetTest
 Set, 992
fitLM
 LinearModel, 501
 LogisticModel, 513
 Model, 526
fitModel
 stats.cpp, 1482
 stats.h, 1501
fitNumIterations

 Model, 526
 fitStepwiseModel
 Set, 992
 fitUnivariateLM
 LinearModel, 501
 LogisticModel, 513
 Model, 526
 fix_content
 Rdouble, 887
 Rinteger, 949
 Rlist, 954
 Rsymbol, 977
 Rvector, 982
 fix_lambda
 par, 634
 fix_prev
 par, 634
fixdcpy
 Rsrv.h, 1438
fixed_prev
 par, 635
flag
 Individual, 461
 par, 635
FlattenModules
 InteractionNetwork, 475
FlattenPvals
 DcVar, 219
flip_file
 par, 635
flip_scan
 par, 635
flip_scan_threshold
 par, 635
flip_scan_verbose
 par, 635
flip_strand
 par, 636
flip_subset
 par, 636
flip_subset_file
 par, 636
flipStrand
 Plink, 796
floatvec_t
 plink.h, 1392
fmatrix_t
 plink.h, 1392
force_initial_cluster
 par, 636
force_span
 par, 636
Forest, 312
 ~Forest, 316

alpha, 325
 case_weights, 326
 computePermutationImportance, 316
 computePredictionError, 316
 computePredictionErrorInternal, 316
 computeTreePermutationImportanceInThread, 316
 condition_variable, 326
 DISALLOW_COPY_AND_ASSIGN, 317
 data, 326
 dependent_varID, 326
 deterministic_varIDs, 326
 Forest, 315
 getChildNodeIDs, 317
 getDependentVarId, 317
 getInbagCounts, 317
 getIsOrderedVariable, 317
 getMinNodeSize, 318
 getMtry, 318
 getNumIndependentVariables, 318
 getNumTrees, 318
 getOverallPredictionError, 318
 getPredictionValues, 319
 getPredictions, 318
 getSplitValues, 319
 getSplitVarIDs, 319
 getVariableImportance, 319
 grow, 319
 growInternal, 319
 growTreesInThread, 320
 holdout, 326
 importance_mode, 327
 init, 320
 initCpp, 320
 initInternal, 321
 initR, 321
 is_ordered_variable, 327
 keep_inbag, 327
 loadFromFile, 322
 loadFromFileInternal, 322
 memory_mode, 327
 memory_saving_splitting, 327
 min_node_size, 327
 minprop, 328
 mtry, 328
 mutex, 328
 no_split_variables, 328
 num_independent_variables, 328
 num_samples, 328
 num_threads, 329
 num_trees, 329
 num_variations, 329
 output_prefix, 329
 overall_prediction_error, 329
 predict, 322
 predict_all, 329
 predictInternal, 323
 predictTreesInThread, 323
 prediction_mode, 330
 prediction_type, 330
 predictions, 330
 progress, 330
 random_number_generator, 330
 run, 323
 sample_fraction, 330
 sample_with_replacement, 331
 saveToFile, 323
 saveToFileInternal, 323
 seed, 331
 setAlwaysSplitVariables, 324
 setSplitWeightVector, 324
 setVerboseOutput, 324
 showProgress, 324
 split_select_varIDs, 331
 split_select_weights, 331
 splitrule, 331
 thread_ranges, 331
 trees, 332
 variable_importance, 332
 verbose_out, 332
 writeConfusionFile, 324
 writeImportanceFile, 325
 writeOutput, 325
 writeOutputInternal, 325
 writePredictionFile, 325
 forest
 RandomForest, 865
 ForestClassification, 333
 ~ForestClassification, 335
 class_values, 338
 classification_table, 338
 computePredictionErrorInternal, 335
 DISALLOW_COPY_AND_ASSIGN, 335
 ForestClassification, 335
 getClassValues, 336
 growInternal, 336
 initInternal, 336
 loadForest, 336
 loadFromFileInternal, 336
 predictInternal, 337
 response_classIDs, 338
 saveToFileInternal, 337
 writeConfusionFile, 337
 writeOutputInternal, 337
 writePredictionFile, 337
 ForestProbability, 339
 ~ForestProbability, 341
 class_values, 344
 classification_table, 344

computePredictionErrorInternal, 341
DISALLOW_COPY_AND_ASSIGN, 341
ForestProbability, 341
getClassValues, 342
getTerminalClassCounts, 342
growInternal, 342
initInternal, 342
loadForest, 342
loadFromFileInternal, 343
predictInternal, 343
response_classIDs, 344
saveToFileInternal, 343
writeConfusionFile, 343
writeOutputInternal, 344
writePredictionFile, 344
ForestRegression, 345
~ForestRegression, 348
computePredictionErrorInternal, 348
DISALLOW_COPY_AND_ASSIGN, 348
ForestRegression, 348
growInternal, 348
initInternal, 349
loadForest, 349
loadFromFileInternal, 349
predictInternal, 349
saveToFileInternal, 350
writeConfusionFile, 350
writeOutputInternal, 350
writePredictionFile, 350
ForestSurvival, 351
~ForestSurvival, 354
computePredictionErrorInternal, 354
DISALLOW_COPY_AND_ASSIGN, 354
ForestSurvival, 354
getChf, 355
getStatusVarId, 355
getUniqueTimepoints, 355
growInternal, 355
initInternal, 355
loadForest, 356
loadFromFileInternal, 356
predictInternal, 356
response_timepointIDs, 357
saveToFileInternal, 356
status_varID, 358
unique_timepoints, 358
writeConfusionFile, 357
writeOutputInternal, 357
writePredictionFile, 357
founder
 Individual, 461
fpser
 cdflib.h, 1126
 dcdflib.cpp, 1187
fraction
 ArgumentHandler, 37
 par, 636
freeEnergyScores
 EvaporativeCooling, 284
freq
 HaploPhase, 402
 Locus, 510
 Segment, 987
freqs
 Variant, 1075
fringe
 par, 637
from_string
 helper.h, 1306
from_window
 par, 637
fromPriorLog
 CArgs, 63
fromScript
 CArgs, 63
ftnstop
 cdflib.h, 1126
 dcdflib.cpp, 1187
ftol
 dcdflib.cpp, 1164
full_model_assoc
 par, 637
fullGenotype, 359
 a1, 360
 a2, 360
 fullGenotype, 359
 operator<, 360
fullModelAssoc
 Plink, 796
func
 gzomanip2, 395
function
 LinearModel, 501
fuzzy_homo
 par, 637
G
 CentralityRanker, 71
g
 MultiLocusGenotype, 539
gAA
 SimParametersQT, 1005
gAB
 SimParametersQT, 1005
gBB
 SimParametersQT, 1005
GET_DT
 Rsrv.h, 1438

GET_STRING
 webcheck.cpp, 1568

GET_XT
 Rsrv.h, 1438

GSLRandomBase, 361
 ~GSLRandomBase, 363
 GSLRandomBase, 362, 363
 nextRandVal, 363
 rStatePtr_, 363
 state, 363

GSLRandomFlat, 364
 ~GSLRandomFlat, 366
 GSLRandomFlat, 366
 lower_, 366
 nextRandVal, 366
 upper_, 366

GVariant, 367
 allele1, 368
 allele2, 368
 dosage1, 368
 dosage2, 368
 GVariant, 368
 missing, 369

gainFile
 CentralityRanker, 71

gainIntPvals
 Regain, 897

gainMatrixFilename
 par, 637

gam1
 cdflib.h, 1126
 dcdflib.cpp, 1187

gaminv
 cdflib.h, 1126
 dcdflib.cpp, 1187

gamln
 cdflib.h, 1127
 dcdflib.cpp, 1188

gamln1
 cdflib.h, 1127
 dcdflib.cpp, 1188

gamma
 CentralityRanker, 72

gammaVector
 CentralityRanker, 72

gammmds
 fisher.cpp, 1238

gammln
 stats.cpp, 1482
 stats.h, 1501

gammp
 stats.h, 1502

GaussEliminationSolver
 CentralityRanker, 69

gaussj
 linear.cpp, 1366

gcf
 stats.h, 1502

Gdiag
 CentralityRanker, 72

gene2segment
 Plink, 827

geneDrop
 Perm, 763

geneExprNames
 DcVar, 226

geneExprSubjects
 DcVar, 226

geneList
 Plink, 827

geneNames
 DgeData, 243

genedrop
 Perm, 767

genepi.cpp
 bartlett, 1254
 betaln, 1254
 bico, 1254
 C, 1254
 CCA_caseonly, 1255
 CCA_logit, 1255
 calcGENEPMeanVariance, 1255
 factln, 1255
 I, 1256
 pillai, 1256
 transposeMatrix, 1256
 vector_tld, 1254

generateMDS
 Plink, 796

GenerateRandomUniformProbabilities
 EvaporativeCoolingPrivacy, 291

geno
 Perm, 767

geno2matrix
 helper.cpp, 1285
 helper.h, 1306

geno_qual_file
 par, 637

geno_qual_max
 par, 638

geno_qual_min
 par, 638

genoGroup
 HaploWindow, 430

genoerr.cpp
 LOG, 1257

genome_2sets
 par, 638

genome_groups
 par, 638

genome_only_check_rels
 par, 638

genome_output
 par, 638

genome_output_full
 par, 639

genome_output_minimal
 par, 639

genome_setlist1
 par, 639

genome_setlist2
 par, 639

genome_test
 par, 639

genome_test_min_snp
 par, 639

genome_test_threshold
 par, 640

genotype
 helper.cpp, 1285
 helper.h, 1306
 probabilisticGenotype, 856

genotypeCounts
 BirdseedData, 56
 Dataset, 190

genotypeMatrix
 DcVar, 226

genotypeSubjects
 DcVar, 226

genotypeToFile
 helper.cpp, 1285
 helper.h, 1307

genotyped
 probabilisticGenotype, 856

genotypes
 HaploWindow, 430

genotypingRate
 helper.cpp, 1286
 helper.h, 1307

genreI
 cfamily.cpp, 1132
 cfamily.h, 1134

get
 Data, 115
 DataChar, 124
 DataDouble, 129
 DataFloat, 134

get_bits
 insilico, 19

GetAdjacencyMatrix
 InteractionNetwork, 475

GetAlgorithmMode

 ReliefSeqController, 928

GetAlgorithmType
 EvaporativeCooling, 277

getAllValues
 Data, 115

GetAlleleCounts
 BirdseedData, 51

GetAlternatePhenotypesFilename
 Dataset, 150

GetAttribute
 Dataset, 150
 DatasetInstance, 204
 PlinkInternalsDatasetInstance, 847

GetAttributeAlleles
 Dataset, 151

GetAttributeIndexFromName
 Dataset, 151

GetAttributeMAF
 Dataset, 151

GetAttributeMutationType
 Dataset, 152

GetAttributeNames
 Dataset, 152

GetAttributeRowCol
 Dataset, 153

GetAttributeTiTvCounts
 Dataset, 153

GetAttributeValues
 Dataset, 153, 154

GetAttributes
 DatasetInstance, 204

getChf
 ForestSurvival, 355
 TreeSurvival, 1068

getChildNodeIDs
 Forest, 317
 Tree, 1027

getChromosomeCode
 helper.cpp, 1286
 helper.h, 1307

getChromosomeMarkerRange
 helper.cpp, 1286
 helper.h, 1307

getChromosomeRange
 helper.cpp, 1286
 helper.h, 1307

GetClass
 DatasetInstance, 204

GetClassColumn
 Dataset, 154

GetClassIndexes
 Dataset, 155

GetClassProbability
 Dataset, 155

GetClassValues
 Dataset, 155
getClassValues
 ForestClassification, 336
 ForestProbability, 342
GetClassificationError
 AttributeRanker, 44
 RandomForest, 864
getCoefs
 LinearModel, 501
 LogisticModel, 514
 Model, 526
GetConfigValue
 Insilico.cpp, 1333
 Insilico.h, 1349
GetConnectivityMatrix
 InteractionNetwork, 475
GetCountsFilename
 DgeData, 240
GetCvSetType
 DatasetInstance, 205
GetData
 ExpressionDataSimulator, 303
GetDatasetPtr
 DatasetInstance, 205
getDependentVarId
 Forest, 317
GetDistanceMetrics
 Dataset, 157
getDouble
 helper.cpp, 1286
 helper.h, 1308
GetECScores
 EvaporativeCooling, 277
GetFileAttributeNames
 Dataset, 157
GetFileBasename
 Insilico.cpp, 1333
 Insilico.h, 1349
GetFileExtension
 Insilico.cpp, 1334
 Insilico.h, 1350
GetFileNumericsNames
 Dataset, 157
GetFrequencyCounts
 ChiSquared, 87
GetGeneCountsSum
 DgeData, 240
GetGeneMinMax
 DgeData, 241
GetGeneNames
 DgeData, 241
GetGenotypeCounts
 BirdseedData, 51
GetId
 DatasetInstance, 205
getInbagCounts
 Forest, 317
 Tree, 1028
getIndex
 Data, 115
GetInfluenceFactorD
 DatasetInstance, 205
GetInstance
 Dataset, 157
GetInstanceIds
 Dataset, 158
GetInstanceIndexForID
 Dataset, 158
getInt
 helper.cpp, 1287
 helper.h, 1308
GetIntForGenotype
 Dataset, 158
GetInteractionScores
 EvaporativeCooling, 277
getIsOrderedVariable
 Forest, 317
GetJukesCantorDistance
 Dataset, 159
GetKeptRemoved
 EvaporativeCoolingPrivacy, 291
GetKimuraDistance
 Dataset, 159
GetKmax
 ReliefSeqController, 928
 ReliefF, 910
getLnLk
 LogisticModel, 514
GetLocalCis
 DcVar, 219
 EpistasisEQtl, 265
getLongUnsignedInt
 helper.cpp, 1287
 helper.h, 1308
GetMAFs
 Dataset, 159
GetMainEffectScores
 EvaporativeCooling, 277
GetMajorAlleleFrequency
 BirdseedData, 51
GetMajorMinorAlleles
 BirdseedData, 52
getMarkerChromosome
 helper.cpp, 1287
 helper.h, 1308
getMarkerNumber
 helper.cpp, 1287

helper.h, 1308
GetMatchingIds
 Insilico.cpp, 1334
 Insilico.h, 1350
getMaxNumUniqueValues
 Data, 116
GetMeanForNumeric
 Dataset, 159
GetMinMaxForContinuousPhenotype
 Dataset, 161
GetMinMaxForNumeric
 Dataset, 161
getMinNodeSize
 Forest, 318
getMissing
 Model, 527
GetMissingValues
 BirdseedData, 52
GetModules
 InteractionNetwork, 475
getMtry
 Forest, 318
GetNNearestInstances
 DatasetInstance, 205, 206
GetNewmanModules
 InteractionNetwork, 476
GetNodeNames
 InteractionNetwork, 476
GetNormalizationFactors
 DgeData, 241
GetNormalizeFlag
 AttributeRanker, 44
getNP
 Model, 527
getNumCols
 Data, 116
GetNumGenes
 DgeData, 241
getNumIndependentVariables
 Forest, 318
getNumRows
 Data, 116
GetNumSNPs
 BirdseedData, 52
GetNumSamples
 DgeData, 241
getNumSamplesOob
 Tree, 1028
GetNumSubjects
 BirdseedData, 52
getNumTrees
 Forest, 318
getNumUniqueDataValues
 Data, 116
GetNumeric
 Dataset, 161
 DatasetInstance, 207
 PlinkInternalsDatasetInstance, 848
getNumericCaseControl
 stats.cpp, 1483
 stats.h, 1502
GetNumericIndexFromName
 Dataset, 162
GetNumericMatrix
 Dataset, 162
GetNumericRowCol
 Dataset, 162
GetNumericValues
 Dataset, 163, 164
GetNumerics
 DatasetInstance, 207
GetNumericsFilename
 Dataset, 163
GetNumericsNames
 Dataset, 163
getOobSampleIDs
 Tree, 1028
getOptions
 Options, 548
getOutputFilename
 options.h, 1384
 parse.cpp, 1387
getOverallPredictionError
 Forest, 318
getPVals
 LinearModel, 501
 LogisticModel, 514
 Model, 527
getPValue
 LinearModel, 502
 LogisticModel, 514
GetPredictedValueTau
 DatasetInstance, 207
getPrediction
 TreeClassification, 1042
 TreeProbability, 1050
 TreeRegression, 1059
 TreeSurvival, 1068
getPredictionTerminalNodeID
 TreeClassification, 1042
 TreeProbability, 1050
 TreeRegression, 1059
 TreeSurvival, 1069
getPredictionValues
 Forest, 319
getPredictions
 Forest, 318
GetProbabilityValueGivenClass

Dataset, 164
GetRadius
 DcVar, 219
 EpistasisEQtl, 265
GetRandomInstance
 Dataset, 165
getRawMatrix
 Regain, 892
GetResultsByVariable
 CentralityRanker, 69
GetSNPNames
 BirdseedData, 53
GetSampleCounts
 DgeData, 242
GetSampleNames
 DgeData, 242
GetSamplePhenotype
 BirdseedData, 53
 DgeData, 242
GetScores
 AttributeRanker, 45
 ReliefFSeq, 923
 ReliefSeqController, 929
getSE
 LinearModel, 502
 LogisticModel, 514
 Model, 527
GetSequencingDepthForSample
 DgeData, 242
GetSimpleSNPValue
 PlinkInternalsDatasetInstance, 848
getSimpleSNPValue
 Model, 527
 Plink, 797
GetSnpsFilename
 Dataset, 165
GetSnpsForTFs
 DcVar, 219
 EpistasisEQtl, 266
GetSnpsForTranscript
 DcVar, 219
 EpistasisEQtl, 266
getSplitValues
 Forest, 319
 Tree, 1028
getSplitVarIds
 Forest, 319
 Tree, 1028
getStatistic
 Model, 527
getStatusVarId
 ForestSurvival, 355
getSubHaplotypeName
 HaploPhase, 402
GetSubjectCallConfidences
 BirdseedData, 53
GetSubjectGenotypes
 BirdseedData, 53
GetSubjectLabels
 BirdseedData, 54
GetSubjectNames
 BirdseedData, 54
GetTFlInfo
 DcVar, 220
 EpistasisEQtl, 266
GetTFRadius
 DcVar, 220
 EpistasisEQtl, 266
getTerminalClassCounts
 ForestProbability, 342
 TreeProbability, 1050
GetTF
 DcVar, 220
 EpistasisEQtl, 266
getUniqueDataValue
 Data, 116
getUniqueTimepoints
 ForestSurvival, 355
getValue
 OptionSet, 551
getValues
 OptionSet, 551
getVar
 LinearModel, 502
 LogisticModel, 515
 Model, 528
getVariableID
 Data, 117
getVariableImportance
 Forest, 319
GetVariableNames
 Dataset, 165
getVariableNames
 Data, 117
getWindowRange
 helper.cpp, 1287
 helper.h, 1309
getttag_file
 par, 640
getttag_kb
 par, 640
getttag_listall
 par, 640
getttag_mode
 par, 640
getttag_mode1
 par, 640
getttag_mode2

par, 641
getttag_r2
 par, 641
gfamfile
 par, 641
glm_dominant
 par, 641
glm_no_auto_sex_effect
 par, 641
glm_recessive
 par, 641
glm_sex_effect
 par, 642
glm_user_parameters
 par, 642
glm_user_test
 par, 642
glmAssoc
 Plink, 797
glmCNVBurdenModel
 Plink, 797
glmHaplotypeTest
 Plink, 797
globals.h
 DEFAULT_ALPHA, 1263
 DEFAULT_IMPORTANCE_MODE, 1264
 DEFAULT_MIN_NODE_SIZE_CLASSIFICATION,
 1264
 DEFAULT_MIN_NODE_SIZE_PROBABILITY, 1264
 DEFAULT_MIN_NODE_SIZE_REGRESSION, 1264
 DEFAULT_MIN_NODE_SIZE_SURVIVAL, 1264
 DEFAULT_MINPROP, 1264
 DEFAULT_NUM_THREADS, 1265
 DEFAULT_NUM_TREE, 1265
 DEFAULT_PREDICTIONTYPE, 1265
 DEFAULT_SPLITRULE, 1265
 DISALLOW_COPY_AND_ASSIGN, 1261
 ImportanceMode, 1262
 M_PI, 1261
 MAX_IMP_MODE, 1265
 MAX_MEM_MODE, 1266
 mask, 1265
 MemoryMode, 1262
 offset, 1266
 PredictionType, 1262
 Q_THRESHOLD, 1266
 STATUS_INTERVAL, 1266
 SplitRule, 1263
 TreeType, 1263
 uint, 1262
gmapfile
 par, 642
gmulti_output
 par, 642
goodModels
 EpistasisEQtl, 270
gplink
 par, 642
gprint
 trio.cpp, 1528
grat1
 cdflib.h, 1127
 dcdflib.cpp, 1188
gratio
 cdflib.h, 1127
 dcdflib.cpp, 1188
greport
 par, 643
greport.cpp
 PP, 1267
greport_display_empty
 par, 643
greport_gene_list
 par, 643
greport_results
 par, 643
greport_subset
 par, 643
greport_subset_file
 par, 643
grid
 par, 644
group
 ChapModel, 78
 Pool, 852
 Range, 869
groupAvgLink
 cluster.cpp, 1140
groupGenome
 Plink, 797
groupNames
 Range, 869
groupSegmentsSpanning
 Plink, 798
grow
 Forest, 319
 Tree, 1028
growInternal
 Forest, 319
 ForestClassification, 336
 ForestProbability, 342
 ForestRegression, 348
 ForestSurvival, 355
growTreesInThread
 Forest, 320
gser
 stats.h, 1502
gset1

Plink, 828
 gset2
 Plink, 828
 gsumln
 cdflib.h, 1128
 dcdflib.cpp, 1189
 Gtrace
 CentralityRanker, 72
 gval
 RCount, 881
 gvar
 Individual, 461
 par, 644
 Plink, 828
 gvar.cpp
 analyseModel, 1270
 compareModels, 1270
 gvar_full_report
 par, 644
 gvar_include_all_variants
 par, 644
 gvar_to_standard
 par, 644
 gvar_verbose_association
 par, 644
 gvar_write
 par, 645
 gvarfile
 par, 645
 gwin
 RCount, 881
 gzfilebuf, 369
 ~gzfilebuf, 373
 attach, 373
 buffer, 378
 buffer_size, 378
 close, 374
 disable_buffer, 374
 enable_buffer, 374
 file, 379
 gzfilebuf, 373
 io_mode, 379
 is_open, 374
 open, 375
 open_mode, 375
 overflow, 375
 own_buffer, 379
 own_fd, 379
 setbuf, 376
 setcompression, 376
 showmany, 377
 sync, 377
 underflow, 378
 gzifstream, 380
 attach, 384
 close, 385
 gzifstream, 383, 384
 is_open, 385
 open, 385
 rdbuf, 386
 sb, 386
 gzofstream, 386
 attach, 390
 close, 391
 gzofstream, 389, 390
 is_open, 391
 open, 391
 rdbuf, 392
 sb, 392
 gzomanip2
 func, 395
 gzomanip2, 394
 operator<<, 394
 val1, 395
 val2, 395
 gzomanip2< T1, T2 >, 392

H

HAS_ATTR
 Rsrv.h, 1438
 HFRQ
 HaploPhase, 416
 HIGH
 stats.cpp, 1479
 IMPUTE
 HaploPhase, 416
 PHASE
 HaploPhase, 417
 HTEST
 HaploPhase, 417
 HWD_filter_on_all
 par, 653
 HWD_limit
 par, 653
 HWD_report
 par, 653
 HWD_standard
 par, 653
 HWD_test
 par, 654
 HWindow, 435
 finished, 436
 HWindow, 436
 hetCount, 437
 homCount, 437
 leftHomozyg, 437
 leftMissing, 437

misCount, 437
P, 437
person, 438
recount, 436
rightHomozyg, 438
rightMissing, 438
set, 436
shift, 436
start, 438
stop, 438
valid, 438
half
 dcdflib.cpp, 1164
hap
 HaploPhase, 414
 HaploWindow, 431
hap1
 HaploPhase, 414
 HaploWindow, 431
hap2
 HaploPhase, 415
 HaploWindow, 431
hap_max_nf_phases
 par, 645
hap_min_phase_prob
 par, 645
hap_missing_geno
 par, 645
hap_post_prob
 par, 645
hap_specific_snps
 par, 646
hap_specific_snps_list
 par, 646
hapi
 HaploPhase, 415
haplo
 HaploWindow, 431
 Plink, 828
haplo_plem_follow
 par, 646
haplo_plem_follow_fid
 par, 646
haplo_plem_follow_iid
 par, 646
haplo_plem_follow_ind
 par, 646
haplo_plem_iter
 par, 647
haplo_plem_likelihood_iter
 par, 647
haplo_plem_meta_iter
 par, 647
haplo_plem_meta_likelihood_iter
 par, 647
 haplo_plem_meta_prune_haplotype
 par, 647
 haplo_plem_meta_prune_phase
 par, 647
 haplo_plem_meta_tol
 par, 648
 haplo_plem_meta_window
 par, 648
 haplo_plem_nonzero_threshold
 par, 648
 haplo_plem_original_overlap
 par, 648
 haplo_plem_overlap
 par, 648
 haplo_plem_verbose
 par, 648
 haplo_plem_window
 par, 649
 haplo_plem_window_prune_phase
 par, 649
 haplo_plem_window_tol
 par, 649
 haplo_plem_zero_threshold
 par, 649
haploGroup
 ChapModel, 78
HaploPhase, 396
 actual_map, 412
 actual_nw, 412
 ambig, 413
 calculateDp, 413
 calculateEmpiricalVariance, 399
 calculateHaplotypeFrequencies, 400
 calculateHaplotypePhases, 400
 case_freq, 413
 cnt_f, 413
 consistentNonfounderMalePhaseGivenXGenotypes,
 400
 consistentNonfounderPhaseGivenGenotypes, 400
 consistentNonfounderPhaseGivenParents, 400
 control_freq, 413
 current, 413
 dosage, 401
 downcoding, 414
 dprime, 401
 empiricalVariance, 414
 enumerateAllPhases, 401
 enumerateHaplotypes, 401
 enumerateNonfounderPhase, 401
 enumeratePhasedWindows, 402
 f, 414
 finishWindow, 414
 freq, 402

getSubHaplotypeName, 402
HFRQ, 416
HIMPUTE, 416
HPHASE, 417
HTEST, 417
hap, 414
hap1, 414
hap2, 415
hapi, 415
HaploPhase, 399
haploid, 415
haploid_np, 415
haploid_ph_freq, 415
haploid_ph_hap1, 415
haplotypeName, 402
haplotypicCC, 403
haplotypicQTL, 403
haplotypicTDT, 403
haplotypicWeightedCC, 403
haplotypicWeightedTDT, 403
hapmap, 416
hapmapb, 416
hname, 416
homozyg, 416
imputeAllHaplotypes, 404
imputeGenotype, 404
imputeHaplotypes, 404
imputeThisHaplotype, 404
include, 417
includeIndividuals, 404
mainImputation, 405
makeSetFromMap, 405
makeSlidingWindow, 405
makeSubHaplotypeSet, 405
makeTestSet, 405
makeWaplotype, 406
nHap, 406
name, 406
new_map, 417
new_one, 417
new_pred_allele, 417
new_pred_locus, 418
new_pred_weighted_allele, 418
new_two, 418
nh, 418
nonfounders, 418
np, 418
ns, 419
nsh, 419
nt, 419
nw, 419
odds, 419
P, 419
p1, 420
p2, 420
performAlternEM, 406
performEM_original, 406
performHaplotypeTests, 407
ph_freq, 420
ph_hap1, 420
ph_hap2, 420
phaseAllHaplotypes, 407
phaseAndScoreNonfounder, 407
phasemap, 420
pp, 421
prunePhase, 407
pvalue, 421
queryGenotype, 407
queryThisGenotype, 408
ratio, 421
readTagFile, 408
reference_only, 421
reportHaplotypeFrequencies, 408
reportPhase, 408
reportPhaseWideFormat, 408
reset, 409
resolveWithKids, 409
result, 421
returnHaplotypeSet, 409
returnHaplotypes, 409
rsq, 409
rsq_internal, 410
S, 421
scoreTransmissions, 410
setSpecificSNPs, 410
setTestHaplotype, 411
sets, 422
startWindow, 422
subhaplotypes, 422
test_hap, 422
testHaplotypeFreq, 411
testSet, 422
trackSharedHaplotypes, 411
trackThisSegment, 411
trackedIBS, 422
trackedN, 423
trans, 423
transmissionCount, 411
transmissionTotal, 423
transmissionX2, 423
transmissionX, 423
untrans, 423
updateForImputation, 412
useEmpiricalVariance, 424
VPHASE, 424
validateNonfounder, 412
validN, 424
verboseDisplayWindows, 412

WGT, 424
windows, 424
X, 424
HaploWindow, 425
~HaploWindow, 427
ambig, 430
converged, 430
enumerateGenogroups, 427
enumerateHaplotypes, 427
enumeratePhase, 427
expandGenogroups, 428
f, 430
genoGroup, 430
genotypes, 430
hap, 431
hap1, 431
hap2, 431
haplo, 431
HaploWindow, 427
haplotypeName, 428
hapmap, 431
iter, 431
left_passed, 432
leftStub, 432
leftStubFrequency, 428
nh, 432
np, 432
ns, 432
P, 432
performEM, 428
pp, 433
pruneGenogroups, 428
prunePhase, 429
reportPhase, 429
right_passed, 433
rightStub, 433
rightStubFrequency, 429
S, 433
sampleLogLikelihood, 433
setStubCodes, 429
start, 433
stop, 434
tallyUnambiguousCounts, 429
uc, 434
zero, 434
haplohelper.cpp
LOG, 1274
haploid
 HaploPhase, 415
 Model, 533
haploid_np
 HaploPhase, 415
haploid_ph_freq
 HaploPhase, 415
haploid_ph_hap1
 HaploPhase, 415
haplotype
 Model, 533
haplotypeName
 HaploPhase, 402
 HaploWindow, 428
haplotypesInSameGroup
 ChapModel, 77
haplotypicCC
 HaploPhase, 403
haplotypicQTL
 HaploPhase, 403
haplotypicTDT
 HaploPhase, 403
haplotypicWeightedCC
 HaploPhase, 403
haplotypicWeightedTDT
 HaploPhase, 403
haplindow.cpp
 verboseDisplayWindows2, 1276
hapmap
 HaploPhase, 416
 HaploWindow, 431
hapmappb
 HaploPhase, 416
hardyWeinbergCheck
 Plink, 798
has_nonfounders
 par, 649
has_snps
 Model, 533
HasAllelicInfo
 Dataset, 165
hasAllelicInfo
 Dataset, 190
HasAlternatePhenotypes
 Dataset, 166
hasAlternatePhenotypes
 Dataset, 190
HasContinuousPhenotypes
 Dataset, 166
hasContinuousPhenotypes
 Dataset, 191
hasExcludedSnps
 BirdseedData, 57
HasGenotypes
 Dataset, 166
hasGenotypes
 Dataset, 191
hasHeader
 IDFile, 445
hasIncludedSnps
 BirdseedData, 57

hasNormFactors
 DgeData, 244
 HasNumerics
 Dataset, 166, 167
 hasNumerics
 Dataset, 191
 HasPhenotypes
 BirdseedData, 54
 Dataset, 167
 hasPhenotypes
 BirdseedData, 57
 Dataset, 191
 hasSNPs
 Model, 528
 HasSubjectLabels
 BirdseedData, 54
 hasSubjectLabels
 BirdseedData, 57
 have_numerics
 par, 649
 have_snps
 par, 650
 head
 Rlist, 955
 Rmessage, 960
 helper.cpp
 affCoding, 1280
 calcSetCovarianceMatrix, 1280
 checkDuplicates, 1280
 checkFileExists, 1280, 1281
 chromosomeName, 1281
 commaParse, 1281
 compressed, 1281
 dbl2str, 1281
 dbl2str_fixed, 1282
 defineCowChromosomes, 1282
 defineDogChromosomes, 1282
 defineHorseChromosomes, 1282
 defineHumanChromosomes, 1282
 defineMouseChromosomes, 1283
 defineRiceChromosomes, 1283
 defineSheepChromosomes, 1283
 display, 1283, 1284
 displayLine, 1284
 doesFileExist, 1284
 error, 1284
 FPMIN, 1279
 filterRanges, 1285
 geno2matrix, 1285
 genotype, 1285
 genotypeToFile, 1285
 genotypingRate, 1286
 getChromosomeCode, 1286
 getChromosomeMarkerRange, 1286
 getChromosomeRange, 1286
 getDouble, 1286
 getInt, 1287
 getLongUnsignedInt, 1287
 getMarkerChromosome, 1287
 getMarkerNumber, 1287
 getWindowRange, 1287
 identicalSNPs, 1288
 int2str, 1288
 is_rare, 1288
 itoa, 1288
 LOG, 1297
 leftWindowEdge, 1288
 listPossibleHaplotypes, 1289
 longint2str, 1289
 MISSING1, 1279
 MISSING2, 1280
 makeLocusMap, 1289
 makePersonMap, 1289
 makeScaffold, 1289
 mapRanges2SNP, 1290
 mapRangesToSNPs, 1290
 mapSNPs2Range, 1290
 matrixGetNumericAll, 1290
 matrixGetNumericCaseControl, 1290
 modelComparisonPValue, 1291
 NoMem, 1291
 nvec_bool, 1291
 parse2int, 1291
 parse2str, 1291
 permute, 1292
 PP, 1297
 rangeIntersect, 1292
 readRange, 1292
 readString, 1292
 relType, 1293
 removeMissingPhenotypes, 1293
 rightWindowEdge, 1293
 SNPHWE, 1295
 searchAndReplace, 1293
 seeChromosome, 1293
 shutdown, 1294
 sizeMatrix, 1294
 sizeTable, 1294
 sizefMatrix, 1294
 smoother, 1294
 summaryBasics, 1295
 sw, 1295, 1296
 tokenizeLine, 1296
 two_locus_table, 1297
 vif_prune, 1297
 helper.h
 affCoding, 1301
 calcSetCovarianceMatrix, 1301

checkDuplicates, 1301
checkFileExists, 1301
chromosomeName, 1302
commaParse, 1302
compressed, 1302
dbl2str, 1302
dbl2str_fixed, 1302
defineCowChromosomes, 1303
defineDogChromosomes, 1303
defineHorseChromosomes, 1303
defineHumanChromosomes, 1303
defineMouseChromosomes, 1303
defineRiceChromosomes, 1304
defineSheepChromosomes, 1304
display, 1304, 1305
displayLine, 1305
doesFileExist, 1305
error, 1305
FDR_BH, 1305
filterRanges, 1306
from_string, 1306
geno2matrix, 1306
genotype, 1306
genotypeToFile, 1307
genotypingRate, 1307
getChromosomeCode, 1307
getChromosomeMarkerRange, 1307
getChromosomeRange, 1307
getDouble, 1308
getInt, 1308
getLongUnsignedInt, 1308
getMarkerChromosome, 1308
getMarkerNumber, 1308
getWindowRange, 1309
identicalSNPs, 1309
int2str, 1309
itoa, 1309
leftWindowEdge, 1309
listPossibleHaplotypes, 1310
longint2str, 1310
MAX, 1312
MIN, 1312
makeLocusMap, 1310
makePersonMap, 1310
makeScaffold, 1310
mapRanges2SNP, 1311
mapRangesToSNPs, 1311
mapSNPs2Range, 1311
matrix_t, 1300
matrixGetNumericAll, 1311
matrixGetNumericCaseControl, 1311
modelComparisonPValue, 1312
NoMem, 1312
nvec_bool, 1312
parse2int, 1313
parse2str, 1313
permute, 1313
rangeIntersect, 1313
readRange, 1314
readString, 1314
relType, 1314
removeMissingPhenotypes, 1314
rightWindowEdge, 1314
SIGN, 1315
SNPHWE, 1316
SQR, 1317
SWAP, 1318
searchAndReplace, 1315
seeChromosome, 1315
shutdown, 1315
sizeMatrix, 1316
sizeTable, 1316
sizefMatrix, 1315
smoother, 1316
summaryBasics, 1317
sw, 1317, 1318
tokenizeLine, 1318
two_locus_table, 1319
vector_t, 1300
vif_prune, 1319
hetCount
 HWindow, 437
hetOdds
 SimParameters, 1000
Histogram
 Statistics.h, 1471
Histogramlt
 Statistics.h, 1471
hln2pi
 dcdflib.cpp, 1164, 1165
hname
 HaploPhase, 416
holdError
 EvaporativeCoolingPrivacy, 295
holdout
 ArgumentHandler, 38
 EvaporativeCoolingPrivacy, 295
 Forest, 326
 par, 650
 Tree, 1031
holdoutErrors
 EvaporativeCoolingPrivacy, 295
holdoutImportance
 EvaporativeCoolingPrivacy, 295
holdoutInvImportance
 EvaporativeCoolingPrivacy, 295
homCount
 HWindow, 437

homOdds
 SimParameters, 1000

homo_miss_as_hom
 par, 650

homo_run
 par, 650

homo_run_consensus_match
 par, 650

homo_run_density
 par, 650

homo_run_gap
 par, 651

homo_run_het
 par, 651

homo_run_kb
 par, 651

homo_run_length_kb
 par, 651

homo_run_length_snps
 par, 651

homo_run_snps
 par, 651

homo_summary_allelic_match
 par, 652

homo_threshold
 par, 652

homo_windowAllowedHet
 par, 652

homo_windowAllowedMissing
 par, 652

homo_windowKB
 par, 652

homo_windowSize
 par, 652

homogeneous_clusters
 cluster.cpp, 1140

Homophily
 InteractionNetwork, 476

HomophilyResult
 InteractionNetwork.h, 1359

homozyg
 HaploPhase, 416

homozyg.cpp
 compareCNVs, 1320
 displayPoolVerbose, 1321
 PP, 1322
 segsIBDMatch, 1321
 segsIBDMatchCON, 1321
 segsMatch, 1321
 segsMatchCON, 1322
 segsOverlap, 1322
 summaryIndivSummaries, 1322

homozyg_verbose
 par, 653

homozygousSegmentPermutationTest
 Plink, 798

host
 Rconnection, 876

hotel
 par, 653

hotel.cpp
 calcHotelSetMeanVariance, 1323

hp
 clump_LD, 95

hubDegrees
 _RipmResult, 28

HuberWhite
 LinearModel, 502
 LogisticModel, 515
 Model, 528

hubs
 _RipmResult, 28

hypot
 Statistics.cpp, 1466
 Statistics.h, 1473

i
 genepi.cpp, 1256

i1
 Individual, 461

i2
 Individual, 461

i_int
 fisher.cpp, 1232

i_real
 fisher.cpp, 1232

iAllele
 plink.h, 1393

IBD_threshold
 par, 655

IBD_within
 par, 655

IDField, 439
 aliasList, 441
 attribute, 441
 eqid, 441
 equiv, 441
 IDField, 440
 joint, 441
 jointName, 442
 masterList, 442
 name, 442
 null, 442
 operator<, 440
 operator==, 440
 width, 442

IDFile, 443
 alias_delimit, 444

delimit, 444
equiv, 444
fields, 444
filename, 444
hasHeader, 445
IDFile, 444
injections, 445
joint, 445
missingValues, 445
uniqFieldCount, 445
IDGroup, 446
display, 447
file, 447
IDGroup, 446
resolved, 447
values, 447
IDHelper, 448
attribFields, 451
dict, 451
fieldMap, 452
fields, 452
files, 452
findAllIndividuals, 449
findUniqueIndividual, 449
iField, 452
idDump, 449
idHelp, 450
idListAlias, 450
idMatch, 450
idReplace, 450
idgroup, 452
idmap, 452
jointField, 453
jointMap, 453
jointOrder, 453
matchIndividual, 450
parseQuery, 450
setAlias, 451
setJointValues, 451
uniqueFields, 453
IDValue, 454
field, 456
IDValue, 455
jointValue, 456
operator!=, 455
operator<, 455
operator==, 455
singleMatch, 456
updateAlias, 456
value, 456
iField
 IDHelper, 452
iIndividual
 plink.h, 1393
iLocus
 plink.h, 1393
INT_MATRIX
 Regain, 897
INT_SIF
 Regain, 897
INVALID_ATTRIBUTE_VALUE
 Insilico.h, 1355
INVALID_DISCRETE_CLASS_VALUE
 Insilico.h, 1355
INVALID_DISTANCE
 Insilico.h, 1355
INVALID_INDEX
 Insilico.h, 1355
INVALID_INT_VALUE
 Insilico.h, 1355
INVALID_NUMERIC_CLASS_VALUE
 Insilico.h, 1356
INVALID_NUMERIC_VALUE
 Insilico.h, 1356
IP_ADDR
 lookup.cpp, 1370
 lookup2.cpp, 1372
 webcheck.cpp, 1568
IS_LARGE
 Rsrv.h, 1438
iSNP
 plink.h, 1394
IA
 CRandom, 109
ibd_file
 par, 654
ibd_file_list
 par, 654
ibd_read
 par, 654
ibd_read_list
 par, 654
ibd_read_minimal
 par, 654
ibs_2only
 par, 655
ibs_inner_run_length_kb
 par, 655
ibs_inner_run_length.snp
 par, 655
ibs_inter.snp_distance
 par, 655
ibs_join_kb
 par, 656
ibs_join.snp
 par, 656
ibs_run
 par, 656

ibs_run_0
 par, 656
ibs_run_density
 par, 656
ibs_run_length_kb
 par, 656
ibs_run_length_snps
 par, 657
ibs_run_missing
 par, 657
ibs_sharing_test
 par, 657
ibs_test
 par, 657
ibs_test_method2
 par, 657
ibs_test_min.snp
 par, 657
ibstest_gap
 par, 658
ID
 PlinkInternalsDatasetInstance, 849
id
 DatasetInstance, 213
idDump
 IDHelper, 449
idHelp
 IDHelper, 450
idListAlias
 IDHelper, 450
idMatch
 IDHelper, 450
idReplace
 IDHelper, 450
identicalSNPs
 helper.cpp, 1288
 helper.h, 1309
idgroup
 IDHelper, 452
idhelp
 par, 658
idhelp.cpp
 PP, 1324
idhelp_alias_update
 par, 658
idhelp_auto_alias
 par, 658
idhelp_command
 par, 658
idhelp_dictionary
 par, 658
idhelp_dump_from_dict
 par, 659
idhelp_dump_from_dict_cmd
 par, 659
idhelp_input
 par, 659
idhelp_list_aliases
 par, 659
idhelp_lookup
 par, 659
idhelp_lookup_string
 par, 659
idhelp_match
 par, 660
idhelp_match_string
 par, 660
idhelp_no_dict
 par, 660
idhelp_output_delimit
 par, 660
idhelp_replace
 par, 660
idhelp_replace_string
 par, 660
idhelp_subset
 par, 661
idhelp_subset_string
 par, 661
idmap
 IDHelper, 452
 Perm, 768
idum
 CRandom, 109
ignore_missing_sex
 par, 661
ignore_phenotypes
 par, 661
ignorePosition
 RCount, 879
iid
 Individual, 462
ikids
 Individual, 462
IM
 CRandom, 109
im
 Individual, 462
impmeasure
 ArgumentHandler, 38
 par, 661
importance_mode
 Forest, 327
 Tree, 1031
ImportanceMode
 globals.h, 1262
impute.cpp
 LOG, 1326

impute_sex
 par, 661
impute_tags
 par, 662
impute_verbose
 par, 662
imputeAllHaplotypes
 HaplоФase, 404
imputeGenotype
 HaplоФase, 404
imputeHaplotypes
 HaplоФase, 404
imputeThisHaplotype
 HaplоФase, 404
in_anal
 Plink, 828
inbag_counts
 Tree, 1032
inxix.cpp
 LOG, 1328
 main, 1328
 PDATE, 1328
 PREL, 1329
 PVERSION, 1329
 PP, 1329
inxixEnv
 InteractionNetwork, 483
inbreeding
 par, 662
inc_file
 par, 662
inc_read
 par, 662
inc_write
 par, 662
include
 Family, 308
 HaplоФase, 417
include_all_pairs
 par, 663
include_all_z1
 par, 663
include_cluster
 par, 663
include_cluster_filename
 par, 663
include_cluster_from_file
 par, 663
includeIndividuals
 HaplоФase, 404
includeSnps
 BirdseedData, 57
includeSnpsFilename
 BirdseedData, 58
Ind2SNP
 Plink, 798
ind_attrib_file
 par, 663
ind_attrib_filter
 par, 664
ind_attrib_value
 par, 664
index
 Pool, 852
index_data
 Data, 118
indexOf
 Rvector, 982
indexOfString
 Rstrings, 973
 Rvector, 982
Indices
 InteractionNetwork.h, 1360
indiv_report
 par, 664
indiv_report_fid
 par, 664
indiv_report_iid
 par, 664
indivPair, 466
 operator<, 467
 p1, 467
 p2, 467
indivSegmentGroup
 Plink, 828
indivSegmentSummary
 Plink, 798
indivSegmentSummaryCalc
 Plink, 799
Individual, 457
 aff, 459
 B, 459
 bcovar, 459
 bnumeric, 460
 clist, 460
 clistMissing, 460
 countMeioses, 459
 covar, 460
 family, 460
 fid, 460
 flag, 461
 founder, 461
 gvar, 461
 i1, 461
 i2, 461
 iid, 462
 ikids, 462
 im, 462

Individual, 459
 ip, 462
 kids, 462
 mat, 462
 missing, 463
 missing2, 463
 nlist, 463
 nlistMissing, 463
 one, 463
 pat, 463
 phenotype, 464
 plist, 464
 plistMissing, 464
 pm, 464
 pp, 464
 pperson, 464
 sex, 465
 sexcode, 465
 sol, 465
 T, 465
 two, 465
 W, 465
 wmlocus, 466
 individual
 PlinkInternalsDatasetInstance, 849
 inet_aton
 sisocks.h, 1455
 inf
 dcdflib.cpp, 1165, 1166
 ZInput, 1084
 infCount
 Regain, 897
 init
 Forest, 320
 Tree, 1029
 initCpp
 Forest, 320
 initInternal
 Forest, 321
 ForestClassification, 336
 ForestProbability, 342
 ForestRegression, 349
 ForestSurvival, 355
 Tree, 1029
 TreeClassification, 1042
 TreeProbability, 1050
 TreeRegression, 1059
 TreeSurvival, 1069
 initialiseGeneCountAssociation
 Plink, 799
 initialiseSetMapping
 Set, 992
 InitializeData
 AttributeRanker, 45
 RandomForest, 864
 initR
 Forest, 321
 initsocks
 sisocks.h, 1456
 injections
 IDFile, 445
 input.cpp
 LOG, 1330
 insilico, 17
 best_n, 18
 get_bits, 19
 join, 19
 split, 20–22
 split_if, 22, 23
 to_lower, 23
 to_upper, 24
 trim, 24, 25
 trim_left, 25
 trim_right, 25, 26
 zeroPadNumber, 26
 Insilico.cpp
 ChooseSnpsDatasetByType, 1332
 DetectClassType, 1332
 GetConfigValue, 1333
 GetFileBasename, 1333
 GetFileExtension, 1334
 GetMatchingIds, 1334
 LoadNumericIds, 1335
 LoadPhenoidls, 1335
 ProtectedLog, 1336
 scoresSortAsc, 1336
 scoresSortAscByName, 1336
 scoresSortDesc, 1336
 Timestamp, 1336
 Insilico.h
 AnalysisType, 1343
 AttributeLevel, 1340
 AttributeMutationType, 1343
 AttributeScores, 1340
 AttributeScoresClt, 1341
 AttributeScoresIt, 1341
 AttributeType, 1344
 COMMAND_LINE_ERROR, 1353
 ChooseSnpsDatasetByType, 1347
 ClassLevel, 1341
 ClassType, 1344
 ConfigMap, 1341
 CvSetType, 1344
 DATASET_LOAD_ERROR, 1353
 DEFAULT_FDR, 1354
 DEFAULT_PVALUE_THRESHOLD, 1354
 DEFAULT_PVALUE, 1354
 DEFAULT_ZVALUE, 1354

datasetTypeToExt, 1354
DetectClassType, 1348
DetermineRandomJungleTreeType, 1348
DistancePair, 1341
DistancePairs, 1342
DistancePairsIt, 1342
EcAlgorithmType, 1345
EcItAlgorithmType, 1345
EcMeAlgorithmType, 1345
GetConfigValue, 1349
GetFileBasename, 1349
GetFileExtension, 1350
GetMatchingIds, 1350
INVALID_ATTRIBUTE_VALUE, 1355
INVALID_DISCRETE_CLASS_VALUE, 1355
INVALID_DISTANCE, 1355
INVALID_INDEX, 1355
INVALID_INT_VALUE, 1355
INVALID_NUMERIC_CLASS_VALUE, 1356
INVALID_NUMERIC_VALUE, 1356
LoadNumericIds, 1351
LoadPhenoids, 1351
MIN_NUM_GENES, 1356
MIN_NUM_SUBJ_PER_GROUP, 1356
MISSING_ATTRIBUTE_VALUE, 1356
MISSING_DISCRETE_CLASS_VALUE, 1357
MISSING_NUMERIC_CLASS_VALUE, 1357
MISSING_NUMERIC_VALUE, 1357
matrixElement, 1342
NumericLevel, 1342
OutputDatasetType, 1346
PP, 1357
PrintVector, 1352
ProtectedLog, 1352
RandomJungleRunMode, 1346
RandomJungleTreeType, 1347
ScoreVarPair, 1342
scoresSortAsc, 1352
scoresSortAscByName, 1352
scoresSortDesc, 1353
Timestamp, 1353
ValueType, 1347
insilico::do_to_lower
 do_to_lower, 247, 248
 m_ctype, 248
 operator(), 248
insilico::do_to_lower< charT >, 246
insilico::do_to_upper
 do_to_upper, 250, 251
 m_ctype, 251
 operator(), 251
insilico::do_to_upper< charT >, 249
insilico::is_classified
 is_classified, 488, 489
 m_ctype, 489
 operator(), 489
insilico::is_classified< Type, charT >, 487
InstanceAttributeStats
 SNReliefF.h, 1459
InstanceAttributeStatsIt
 SNReliefF.h, 1459
InstanceHitMissStats
 SNReliefF.h, 1459
instancelds
 Dataset, 191
instanceldsToLoad
 Dataset, 192
instanceSNP_QT
 simul.cpp, 1451
instanceSNP
 simul.cpp, 1451
instances
 Dataset, 192
instancesMask
 Dataset, 192
instancesMaskPushed
 Dataset, 192
int2, 468
 int2, 468, 469
 operator<, 469
 operator==, 469
 p1, 469
 p2, 470
int2str
 helper.cpp, 1288
 helper.h, 1309
intArray
 Rinteger, 949
intAt
 Rinteger, 949
integerDosage
 Variant, 1076
integratedAttributes
 Regain, 897
inter_grid
 par, 664
interaction
 Model, 533
interaction_t
 plink.h, 1393
interactionAlgorithm
 EvaporativeCooling, 284
interactionEffect
 Regain, 892
InteractionNetwork, 470
 ~InteractionNetwork, 473
 AddModule, 473
 adjMatrix, 482

ApplyFisherTransform, 473
 ApplyPowerTransform, 474
 CheckIndices, 474
 CheckMergeResults, 474
 ComputeQ, 474
 connMatrix, 483
 connectivityThreshold, 483
 connectivityThresholdAbs, 483
 DebugMessage, 474
 debugMode, 483
 Deconvolve, 475
 degrees, 483
 FlattenModules, 475
 GetAdjacencyMatrix, 475
 GetConnectivityMatrix, 475
 GetModules, 475
 GetNewmanModules, 476
 GetNodeNames, 476
 Homophily, 476
 inbixEnv, 483
 InteractionNetwork, 472, 473
 maxMergeOrder, 484
 maxModuleSize, 484
 Merge, 476
 MergeSmallModules, 476
 minModuleSize, 484
 ModularityBestSplit, 477
 ModularityLeadingEigenvector, 477
 modules, 484
 networkFile, 484
 nodeNameIndex, 484
 nodeNames, 485
 numEdges, 485
 NumNodes, 477
 numNodes, 485
 PrepareConnectivityMatrix, 477
 PrintAdjacencyMatrix, 477
 PrintConnectivityMatrix, 478
 PrintModulesSummary, 478
 PrintSummary, 478
 Q, 485
 ReadBrainCorr1DFile, 478
 ReadCsvFile, 478
 ReadGainFile, 478
 ReadSifFile, 479
 RecursiveIndirectPathsModularity, 479
 ripM, 479
 ripmMatrix, 485
 ripmResult, 485
 SaveModules, 479
 SetBinaryThresholding, 479
 SetConnectivityThreshold, 480
 SetConnectivityThresholdAbs, 480
 SetConnectivityThresholding, 480
 SetDebugMode, 480
 SetModulesFromFile, 480
 ShowHomophily, 481
 ShowModuleIndices, 481
 ShowModuleSizes, 481
 ShowModules, 481
 startMergeOrder, 486
 SumMatrixPowerSeries, 481
 useBinaryThreshold, 486
 useConnectivityThreshold, 486
 WriteDelimitedFile, 482
 WriteSifFile, 482
 WriteToFile, 482
InteractionNetwork.h
 DEFAULT_CONNECTIVITY_THRESHOLD, 1361
 HomophilyResult, 1359
 Indices, 1360
 MODULARITY_THRESHOLD, 1361
 MatrixFileType, 1360
 ModularityResult, 1360
 ModuleIndices, 1360
 ModuleList, 1360
 NetworkMatrixType, 1361
 RpmResult, 1360
interactionScores
 EvaporativeCooling, 284
interactionVar_t
 plink.h, 1393
intersects
 cnv.cpp, 1143
 cnv.h, 1145
interval
 Perm, 768
intvec_t
 plink.h, 1393
inverse
 stats.cpp, 1483
 stats.h, 1502
inverse_chiprob
 stats.cpp, 1483
 stats.h, 1503
io_mode
 gzfilebuf, 379
ip
 Individual, 462
ipmpar
 cdflib.h, 1128
 ipmpar.h, 1362
ipmpar.h
 ipmpar, 1362
IQ
 CRandom, 109
iqtl_coord_file
 par, 665

iqtL_expression_file
 par, 665
iqtL_interaction_full
 par, 665
iqtL_local_cis
 par, 665
iqtL_pvalue
 par, 665
iqtL_radius
 par, 665
iqtL_tf_coord_file
 par, 666
iqtL_tf_radius
 par, 666

IR
 CRandom, 109

is_classified
 insilico::is_classified, 488, 489

is_complete
 Rmessage, 959

is_open
 gzfilebuf, 374
 gzifstream, 385
 gzofstream, 391

is_ordered_variable
 Forest, 327
 Tree, 1032

is_rare
 helper.cpp, 1288

isAncestorOf
 cfamily.cpp, 1133

isByteSexOk
 Rsrv.h, 1446

IsLoadableInstanceID
 Dataset, 167

isNested
 Chap, 75

isSet
 OptionSet, 551

isSexInModel
 Model, 528

IsSnpInTFs
 DcVar, 220
 EpistasisEQtl, 267

isValid
 Model, 528

isort
 fisher.cpp, 1238

itAlgorithmType
 EvaporativeCooling, 284

iter
 HaplWindow, 431

iteration
 EvaporativeCoolingPrivacy, 295

itoa
 helper.cpp, 1288
 helper.h, 1309

itop
 Rsrv.h, 1439

iv
 CRandom, 110

iwork
 fisher.cpp, 1238

iwrk
 fisher.cpp, 1232

iy
 CRandom, 110

join
 insilico, 19

joint
 IDField, 441
 IDFile, 445

jointField
 IDHelper, 453

jointMap
 IDHelper, 453

jointName
 IDField, 442

jointOrder
 IDHelper, 453

jointValue
 IDValue, 456

k
 AttributeRanker, 47
 par, 666

keep2SetsForGenome
 Plink, 799

keep_file
 par, 666

keep_inbag
 Forest, 327
 Tree, 1032

keep_indiv
 par, 666

keep_indiv_list
 par, 666

keepAttrs
 EvaporativeCoolingPrivacy, 296

keepIndividuals
 Plink, 799

keepSNPs
 Plink, 799, 800

KendallTau
 Statistics.cpp, 1466, 1467
 Statistics.h, 1473, 1474

kid
 Family, 308

kids
 Individual, 462
 klist
 Plink, 829
 kmap
 Plink, 829
 kname
 Plink, 829
 knownAA
 PairwiseLinkage, 559
 knownAB
 PairwiseLinkage, 559
 knownBA
 PairwiseLinkage, 560
 knownBB
 PairwiseLinkage, 560
 koptBegin
 par, 667
 ReliefSeqController, 932
 ReliefF, 915
 koptEnd
 par, 667
 ReliefSeqController, 932
 ReliefF, 915
 koptStep
 par, 667
 ReliefSeqController, 932
 ReliefF, 915

I

Pair, 553
 Pair2, 555
 SetSortedSNP, 998
 SortedResult, 1016
 LDPair, 490
 dist, 491
 LDPair, 490
 Id, 491
 operator<, 491
 operator<<, 491
 s1, 492
 s2, 492

LOG

assoc.cpp, 1101
 binput.cpp, 1105
 cluster.cpp, 1141
 em.cpp, 1221
 epi.cpp, 1222
 genoerr.cpp, 1257
 haplohelper.cpp, 1274
 helper.cpp, 1297
 impute.cpp, 1326
 inbix.cpp, 1328
 input.cpp, 1330

linput.cpp, 1368
 metaem.cpp, 1376
 output.cpp, 1386
 phase.cpp, 1389
 prephap.cpp, 1399
 profile.cpp, 1401
 stats.cpp, 1496
 tinput.cpp, 1519

LOW

stats.cpp, 1480

label

ChapModel, 78
 Model, 533

lambda

par, 667

lapackf.cpp

eigen_lapack, 1363
 svd_lapack, 1363

lapackf.h

eigen_lapack, 1364
 svd_lapack, 1364

lastWord

NList, 545

ld

LDPair, 491

ld_SNP1

par, 668

ld_SNP1_file

par, 668

ld_SNP2

par, 668

ld_anchor

par, 667

ld_anchor_list

par, 667

ld_distance

clump_LD, 95

ldSet

Set, 994

left_passed

HaploWindow, 432

leftHomozyg

HWindow, 437

leftMissing

HWindow, 437

leftStub

HaploWindow, 432

leftStubFrequency

HaploWindow, 428

leftWindowEdge

helper.cpp, 1288
 helper.h, 1309

len

phdr, 772

Rexp, 943
Rmessage, 960
length
 Rdouble, 887
 Rexp, 941
 Rinteger, 949
 Rmessage, 959
levelCounts
 Dataset, 192
levelCountsByClass
 Dataset, 193
lfamfile
 par, 668
lfile_allele_count
 par, 668
lfile_input
 par, 668
lfit
 linear.cpp, 1366
lfreq
 SimParameters, 1001
 SimParametersQT, 1005
liability
 par, 669
linear.cpp
 covsrt, 1366
 gaussj, 1366
 lfit, 1366
linearHypothesis
 Model, 528
LinearModel, 496
 ~LinearModel, 499
 C, 503
 calculateAdjustedRSquared, 499
 calculateFTest, 500
 calculateMallowC, 500
 calculateRSquared, 500
 calculateRSS, 500
 chisq, 504
 displayResults, 500
 fitLM, 501
 fitUnivariateLM, 501
 function, 501
 getCoefs, 501
 getPVals, 501
 getPValue, 502
 getSE, 502
 getVar, 502
 HuberWhite, 502
 LinearModel, 499
 meanY, 504
 pruneY, 502
 RSS, 504
 reset, 503
se, 504
setDependent, 503
setVariance, 503
sig, 504
standardise, 503
u, 504
v, 505
varY, 505
w, 505
Y, 505
linkRelateds
 Plink, 800
linput.cpp
 LOG, 1368
list_by_allele
 par, 669
list_twolocus
 par, 669
listAllAncestors
 cfamily.cpp, 1133
listPossibleHaplotypes
 helper.cpp, 1289
 helper.h, 1310
lmarker
 SimParameters, 1001
 SimParametersQT, 1005
lnLk
 ChapModel, 79
load_gvar
 par, 669
LoadAlternatePhenotypes
 Dataset, 167
loadCovariate
 RCount, 879
LoadData
 BirdseedData, 54
 DgeData, 242
LoadDataset
 Dataset, 168, 169
LoadDatasetFromPlink
 PlinkInternalsDataset, 844
LoadDefaultTranscriptionFactorLUT
 DcVar, 220
 EpistasisEQtl, 267
loadDoubleVectorFromFile
 utility.cpp, 1537
 utility.h, 1554
loadForest
 ForestClassification, 336
 ForestProbability, 342
 ForestRegression, 349
 ForestSurvival, 356
loadFromDatasetMask
 DataDouble, 129

loadFromFile
 Data, 117
 Forest, 322
loadFromFileInternal
 Forest, 322
 ForestClassification, 336
 ForestProbability, 343
 ForestRegression, 349
 ForestSurvival, 356
loadFromFileOther
 Data, 117
loadFromFileWhitespace
 Data, 117
loadFromPlink
 DataDouble, 129
LoadInstanceFromInstancePtr
 DatasetInstance, 208
LoadInstanceFromVector
 DatasetInstance, 208
LoadNumericIds
 Insilico.cpp, 1335
 Insilico.h, 1351
LoadNumerics
 Dataset, 170
LoadOtherDatasetInstances
 Dataset, 170
loadPCACovariate
 RCount, 879
LoadPhenols
 Insilico.cpp, 1335
 Insilico.h, 1351
LoadPrivacySim
 Dataset, 171
LoadSnp
 Dataset, 171
localCis
 DcVar, 226
 EpistasisEQtl, 270
locked
 par, 669
Locus, 506
 allele1, 509
 allele2, 509
 bp, 509
 chr, 510
 copy, 508
 freq, 510
 Locus, 508
 name, 510
 nm, 510
 operator<, 508
 operator=, 509
 operator==, 509
 pos, 510
locus
 Plink, 829
lod
 DPrime, 255
 PairwiseLinkage, 560
logMatrixStats
 Regain, 892
logOutputOptions
 Regain, 893
login
 Rconnection, 874
logistic_model_iterations
 par, 669
logistic_tolerance_epsilon
 par, 670
LogisticModel, 511
 ~LogisticModel, 513
 chisq, 516
 displayResults, 513
 fitLM, 513
 fitUnivariateLM, 513
 getCoefs, 514
 getLnLk, 514
 getPVals, 514
 getPValue, 514
 getSE, 514
 getVar, 515
 HuberWhite, 515
 LogisticModel, 513
 p, 516
 pruneY, 515
 reset, 515
 setDependent, 515
 V, 516
 Y, 516
logrankScores
 utility.cpp, 1537
 utility.h, 1554
logrankScoresData
 utility.cpp, 1538
 utility.h, 1555
logscale
 par, 670
longint2str
 helper.cpp, 1289
 helper.h, 1310
lookup
 par, 670
 Plink, 800
lookup.cpp
 IP_ADDR, 1370
 PORT_NUM, 1371
lookup2
 par, 670

Plink, 800
lookup2.cpp
 convertPosition, 1372
 IP_ADDR, 1372
 PORT_NUM, 1372
lookup2_cmd
 par, 670
lookup_gene
 par, 670
lookup_gene_kb_window
 par, 671
lookup_gene_name
 par, 671
lookup_multiple_genes
 par, 671
lookup_single_snp
 par, 671
lookup_snp
 par, 671
lookup.snp_kb_window
 par, 671
lookup_to_file
 par, 672
loop_counter
 par, 672
loop_over
 par, 672
loop_over_filename
 par, 672
loop_over_label
 par, 672
lower_
 GSLRandomFlat, 366
lpedfile
 par, 672
lstart
 CInfo, 91
lstop
 CInfo, 91
ltqnorm
 stats.cpp, 1483
 stats.h, 1503
lubksb
 stats.cpp, 1483
 stats.h, 1503
ludcmp
 stats.cpp, 1484
 stats.h, 1503

m
 par, 673
 ReliefF, 915
m1
 par, 673
 Plink, 829
m2
 par, 673
 Plink, 829
M_PI
 fisher.h, 1241
 globals.h, 1261
m_ctype
 insilico::do_to_lower, 248
 insilico::do_to_upper, 251
 insilico::is_classified, 489
m_phenotype
 Plink, 830
m_pihat
 Plink, 830
MAX_CORR_PIHAT_PIHAT_G
 par, 677
MAX_GENO_MISSING
 par, 678
MAX_IMP_MODE
 globals.h, 1265
MAX_IND_MISSING
 par, 678
MAX_ITERATIONS
 EvaporativeCoolingPrivacy, 296
MAX_LINE_LENGTH
 par, 678
 zed.h, 1574
MAX_MEM_MODE
 globals.h, 1266
MAX_PIHAT
 par, 678
mAA
 Model, 534
mAB
 Model, 534
MAX
 helper.h, 1312
mBB
 Model, 534
mCount
 cfamily.cpp, 1133
MEBETAS
 Regain, 898
MENDEL_ind
 par, 679
MENDEL_report
 par, 679
MENDEL_snp
 par, 680
MENDEL_test
 par, 680
MIN_NUM_GENES
 Insilico.h, 1356

MIN_NUM_SUBJ_PER_GROUP
 Insilico.h, 1356

MIN_PIHAT
 par, 683

MISSING1
 helper.cpp, 1279

MISSING2
 helper.cpp, 1280

MISSING_ATTRIBUTE_VALUE
 Insilico.h, 1356

MISSING_DISCRETE_CLASS_VALUE
 Insilico.h, 1357

MISSING_NUMERIC_CLASS_VALUE
 Insilico.h, 1357

MISSING_NUMERIC_VALUE
 Insilico.h, 1357

MISSING
 filters.cpp, 1230

MIN
 helper.h, 1312

MODULARITY_THRESHOLD
 InteractionNetwork.h, 1361

mA
 Model, 533

main
 inbix.cpp, 1328

mainEffect
 Regain, 893

mainEffectComparator
 Regain, 893

MainExitException, 517
 exitMessage, 517

mainImputation
 HaploPhase, 405

mainStats
 RCount, 879

maineffectAlgorithm
 EvaporativeCooling, 285

maineffectScores
 EvaporativeCooling, 285

make2LTable
 output.cpp, 1386

make_blocks
 par, 673

make_founders
 par, 673

make_minor_allele
 par, 673

make_missing_parents
 par, 674

make_pheno
 par, 674

make_pheno_filename
 par, 674

make_pheno_present
 par, 674

make_pheno_value
 par, 674

make_set
 par, 674

make_set_border
 par, 675

make_set_collapse
 par, 675

make_set_collapse_label
 par, 675

make_set_complement
 par, 675

make_set_file
 par, 675

make_set_ignore_group
 par, 675

make_tags
 par, 676

makeFounders
 Plink, 800

makeLDSets
 Set, 993

makeLocusMap
 helper.cpp, 1289
 helper.h, 1310

makeMissingParents
 Plink, 801

makePersonMap
 helper.cpp, 1289
 helper.h, 1310

makePhenotype
 Plink, 801

makeScaffold
 helper.cpp, 1289
 helper.h, 1310

makeSetFromMap
 HaploPhase, 405

makeSlidingWindow
 HaploPhase, 405

makeSubHaplotypeSet
 HaploPhase, 405

makeTestSet
 HaploPhase, 405

makeWaplotype
 HaploPhase, 406

map3
 par, 676

MapPhenosToModel
 DcVar, 221

mapRanges2SNP
 helper.cpp, 1290
 helper.h, 1311

mapRangesToSNPs
 helper.cpp, 1290
 helper.h, 1311
mapSNPs2Range
 helper.cpp, 1290
 helper.h, 1311
MapSnpIndexToPlinkPhenos
 DcVar, 221
mapfile
 par, 676
mapfile_impute
 par, 676
mask
 globals.h, 1265
MaskGetAllVariableNames
 Dataset, 172
MaskGetAttributeIndices
 Dataset, 172
MaskGetAttributeMask
 Dataset, 172
MaskGetInstanceIds
 Dataset, 173
MaskGetInstanceIndices
 Dataset, 173
MaskGetInstanceMask
 Dataset, 173
MaskIncludeAllAttributes
 Dataset, 174
MaskIncludeAllInstances
 Dataset, 174
maskIsPushed
 Dataset, 193
MaskPopAll
 Dataset, 174
MaskPushAll
 Dataset, 175
MaskRemoveInstance
 Dataset, 175
MaskRemoveVariable
 Dataset, 175
MaskRemoveVariableType
 Dataset, 176
MaskSearchInstance
 Dataset, 176
MaskSearchVariableType
 Dataset, 177
MaskWriteNewDataset
 Dataset, 177
masked_conditioning_snps
 ChapModel, 79
master
 Rexp, 943
masterList
 IDField, 442
mat
 Family, 308
 Individual, 462
match
 Pool, 852
matchIndividual
 IDHelper, 450
matchCount
 Pool, 852
matches
 Alleles, 31
mating_tests
 par, 676
matrix
 par, 676
matrix_t
 helper.h, 1300
 plink.h, 1394
matrixAdd
 stats.cpp, 1484
 stats.h, 1503
matrixComputeCovariance
 stats.cpp, 1484
 stats.h, 1504
matrixComputeNodeDegrees
 stats.cpp, 1484
 stats.h, 1504
matrixConnectivityThreshold
 stats.cpp, 1485
 stats.h, 1504
matrixDivideScalar
 stats.cpp, 1485
 stats.h, 1504
matrixElement
 Insilico.h, 1342
matrixElementWiseMultiply
 stats.cpp, 1485
 stats.h, 1505
matrixExtractRowColIdx
 stats.cpp, 1485
 stats.h, 1505
MatrixFileType
 InteractionNetwork.h, 1360
matrixFill
 stats.cpp, 1486
 stats.h, 1505
matrixGetDiag
 stats.cpp, 1486
 stats.h, 1505
matrixGetNumericAll
 helper.cpp, 1290
 helper.h, 1311
matrixGetNumericCaseControl
 helper.cpp, 1290

helper.h, 1311
matrixGetTrace
 stats.cpp, 1486
 stats.h, 1506
matrixMultiplyScalar
 stats.cpp, 1486
 stats.h, 1506
matrixRead
 stats.cpp, 1486
 stats.h, 1506
matrixSetDiag
 stats.cpp, 1487
 stats.h, 1506
matrixSubtract
 stats.cpp, 1487
 stats.h, 1506
matrixSums
 stats.cpp, 1487
 stats.h, 1507
matrixTranspose
 stats.cpp, 1487
 stats.h, 1507
matrixWrite
 stats.cpp, 1488
 stats.h, 1507
max
 fisher.cpp, 1233
 Pool, 853
max_af
 par, 677
max_cluster_case
 par, 677
max_cluster_control
 par, 677
max_cluster_N
 par, 677
max_cluster_size
 par, 677
max_hf
 par, 678
max_neighbour
 par, 678
max_num_unique_values
 Data, 119
max_pvalue
 Perm, 764
maxInteraction
 Regain, 898
maxMainEffect
 Regain, 898
maxMergeOrder
 InteractionNetwork, 484
 par, 679
maxModuleSize
 InteractionNetwork, 484
 par, 679
maxcat
 NList, 545
maxdf
 dcdflib.cpp, 1166
maxit
 dcdflib.cpp, 1166
maxR
 Perm, 768
maxr2
 Plink, 830
maxstat
 utility.cpp, 1538
 utility.h, 1555
maxstatInData
 utility.cpp, 1539
 utility.h, 1557
maxstatPValueLau92
 utility.cpp, 1540
 utility.h, 1557
maxstatPValueLau94
 utility.cpp, 1540
 utility.h, 1558
maxstatPValueUnadjusted
 utility.cpp, 1541
 utility.h, 1559
mB
 Model, 534
mdist
 Plink, 830
mds_by_individual
 par, 679
meAlgorithmType
 EvaporativeCooling, 285
meanY
 LinearModel, 504
memmode
 ArgumentHandler, 38
 par, 679
memory_mode
 Forest, 327
memory_saving_splitting
 Forest, 327
 Tree, 1032
MemoryMode
 globals.h, 1262
Merge
 InteractionNetwork, 476
merge_bedfile
 par, 680
merge_bimfile
 par, 680
merge_binary

par, 680
merge_data
 par, 680
merge_famfile
 par, 681
merge_force_strand
 par, 681
merge_list
 par, 681
merge_list_filename
 par, 681
merge_mapfile
 par, 681
merge_mode
 par, 681
merge_p
 par, 682
merge_pedfile
 par, 682
mergeBinaryData
 Plink, 801
mergeData
 Plink, 801
mergeList
 Plink, 801
MergeSmallModules
 InteractionNetwork, 476
meta_analysis
 par, 682
meta_files
 par, 682
meta_large_phase
 par, 682
metaAnalysis
 Plink, 801
metaanal.cpp
 mymap_t, 1375
 PP, 1375
metaem.cpp
 LOG, 1376
min
 fisher.cpp, 1233
 Perm, 768
 Pool, 853
min_af
 par, 682
min_genotype_cell
 par, 683
min_hf
 par, 683
min_neighbour
 par, 683
min_node_size
 Forest, 327
Tree, 1032
minInteraction
 Regain, 898
minMainEffect
 Regain, 898
minMaxGeneCounts
 DgeData, 244
minMaxSampleCounts
 DgeData, 244
minModuleSize
 InteractionNetwork, 484
 par, 683
minNodeSize
 RandomForest, 866
minprop
 ArgumentHandler, 38
 Forest, 328
 par, 683
 Tree, 1032
misCount
 HWindow, 437
mishap.cpp
 PP, 1377
mishap_test
 par, 684
mishap_window
 par, 684
miss
 Model, 534
miss_run
 par, 684
miss_run_length
 par, 684
miss_run_length_kb
 par, 684
miss_run_level
 par, 684
missing
 GVariant, 369
 Individual, 463
 SimParameters, 1001
 SimParametersQT, 1005
missing2
 Individual, 463
missing_genotype
 par, 685
missing_genotype_explicit
 par, 685
missing_phenotype
 par, 685
missing_phenotype_explicit
 par, 685
missingGenotype
 Plink, 802

missingNumericValues
 Dataset, 193
 missingValues
 BirdseedData, 58
 Dataset, 193
 IDFile, 445
 mk_datfile
 par, 685
 mkBlks
 Plink, 802
 modComputeHomophily
 par, 685
 modConnectivityThreshold
 par, 686
 modEnableConnectivityThreshold
 par, 687
 modFisherTransform
 par, 687
 modFisherTransformCutoff
 par, 687
 modPowerTransform
 par, 687
 modPowerTransformExponent
 par, 687
 modUseBinaryThreshold
 par, 687
 mode
 ReliefFSeq, 924
 Model, 518
 ~Model, 521
 addAdditiveSNP, 522
 addCovariate, 522
 addDominanceSNP, 522
 addHaplotypeDosage, 522
 addInteraction, 522
 addNumeric, 522
 addSexEffect, 523
 addTypedInteraction, 523
 additive, 531
 all_valid, 531
 buildAdditive, 523
 buildCovariate, 523
 buildDesignMatrix, 523
 buildDominance, 524
 buildHaplotype, 524
 buildInteraction, 524
 buildIntercept, 524
 buildNumeric, 524
 buildQFAM, 525
 buildSex, 525
 buildTypedInteraction, 525
 checkVIF, 525
 clst, 532
 cluster, 532
 coef, 532
 converged, 532
 covariate, 532
 displayResults, 525
 dominance, 532
 fitConverged, 526
 fitLM, 526
 fitNumIterations, 526
 fitUnivariateLM, 526
 getCoefs, 526
 getMissing, 527
 getNP, 527
 getPVals, 527
 getSE, 527
 getSimpleSNPValue, 527
 getStatistic, 527
 getVar, 528
 haploid, 533
 haplotype, 533
 has_snps, 533
 hasSNPs, 528
 HuberWhite, 528
 interaction, 533
 isSexInModel, 528
 isValid, 528
 label, 533
 linearHypothesis, 528
 mAA, 534
 mAB, 534
 mBB, 534
 mA, 533
 mB, 534
 miss, 534
 Model, 521
 nc, 534
 nind, 535
 noCluster, 529
 np, 535
 numIterations, 535
 numeric, 535
 order, 535
 P, 535
 pruneY, 529
 S, 536
 setCluster, 529
 setDependent, 529
 setDominant, 529
 setHaploid, 530
 setMissing, 530
 setRecessive, 530
 setValid, 530
 setX, 530
 sex_effect, 536
 skip, 536

testParameter, 536
type, 536
typed_interaction, 536
valid, 537
validParameters, 531
X, 537
xchr, 537
yokeMissing, 531
Ysize, 531
model
 ChapModel, 79
 Plink, 830
model_perm_best
 par, 686
model_perm_dom
 par, 686
model_perm_gen
 par, 686
model_perm_rec
 par, 686
model_perm_trend
 par, 686
modelComparisonPValue
 helper.cpp, 1291
 helper.h, 1312
ModelTermType
 plink.h, 1394
ModularityBestSplit
 InteractionNetwork, 477
ModularityLeadingEigenvector
 InteractionNetwork, 477
ModularityResult
 InteractionNetwork.h, 1360
ModuleIndices
 InteractionNetwork.h, 1360
ModuleList
 InteractionNetwork.h, 1360
modules
 _RipmResult, 28
 InteractionNetwork, 484
mostFrequentClass
 utility.h, 1559
mostFrequentValue
 utility.cpp, 1541
 utility.h, 1560
mperm_rank
 par, 688
mperm_save_all
 par, 688
mperm_save_best
 par, 688
msg
 Rexp, 944
msqrt
 stats.cpp, 1488
 stats.h, 1507
mt
 FamilyTransmissions, 311
mtry
 ArgumentHandler, 38
 Forest, 328
 par, 688
 Tree, 1033
mu
 FamilyTransmissions, 311
MuDeltaAlphas
 ReliefFSeq, 923
mult_clst
 par, 688
mult_covar
 par, 688
mult_filter
 par, 689
mult_pheno
 par, 689
multMatrix
 stats.cpp, 1488
 stats.h, 1507
multcomp
 Plink, 802
multi.cpp
 EPS, 1379
multi_output
 par, 689
MultiLocusGenotype, 538
 count, 539
 g, 539
 operator<, 539
 operator==, 539
 reference, 539
 skip, 540
multiple.cpp
 pprint, 1380
multiple_phenotype_file
 par, 689
multiple_phenotypes
 par, 689
multtest
 par, 689
mutex
 Forest, 328
myID
 Rconnection.cpp, 1408
myfunction
 par, 690
mymap_t
 metaanal.cpp, 1375

N
 Perm, 768
 n
 CArgs, 65
 CNVIndivReport, 103
 CentralityRanker, 72
 Plink, 830
 n1
 ExpressionDataSimulator, 304
 n2
 ExpressionDataSimulator, 304
 NDIV
 CRandom, 110
 nHap
 HaploPhase, 406
 nInnerLoop
 DcVar, 227
 EpistasisEQtl, 270
 NList, 542
 delimit_char, 545
 deparseNumberList, 543
 deparseStringList, 544
 expandNumberList, 544
 firstWord, 545
 lastWord, 545
 maxcat, 545
 NList, 543
 negmode, 546
 range_char, 546
 range_string, 546
 setDelimiter, 544
 setRangeChar, 544
 tokenize, 545
 nOuterLoop
 DcVar, 227
 EpistasisEQtl, 270
 NTAB
 CRandom, 110
 NUM_MATRIX
 Regain, 899
 NUM_SIF
 Regain, 899
 nallele
 Variant, 1076
 nalleles
 RCount, 881
 name
 HaploPhase, 406
 IDField, 442
 Locus, 510
 ProxyResult, 859
 Range, 869
 Rsymbol, 978
 SimParameters, 1001
 SimParametersQT, 1006
 Transcript, 1020
 WMLocus, 1078
 name_pheno
 par, 690
 nanCount
 Regain, 898
 nc
 Model, 534
 ncoef
 dcdflib.cpp, 1166
 negmode
 NList, 546
 neighborInfluenceFactorDs
 DatasetInstance, 213
 NeighborStats
 SNReliefF.h, 1459
 neighborStats
 SNReliefF, 1015
 NeighborStatsClt
 SNReliefF.h, 1459
 NeighborStatslt
 SNReliefF.h, 1459
 Neighbour, 540
 dist, 541
 neighbour, 541
 operator<, 541
 neighbour
 Neighbour, 541
 nel
 Rstrings, 974
 networkFile
 InteractionNetwork, 484
 NetworkMatrixType
 InteractionNetwork.h, 1361
 new_map
 HaploPhase, 417
 new_one
 HaploPhase, 417
 new_parsed_Rexp
 Rconnection.cpp, 1407
 new_parsed_Rexp_from_Msg
 Rconnection.cpp, 1408
 new_pred_allele
 HaploPhase, 417
 new_pred_locus
 HaploPhase, 418
 new_pred_weighted_allele
 HaploPhase, 418
 new_two
 HaploPhase, 418
 next
 Rexp, 944
 nextRandVal

GSLRandomBase, 363
GSLRandomFlat, 366
nextSNP
 Perm, 764
ng
 Pool, 853
ngvar
 Plink, 831
nh
 HaploPhase, 418
 HaploWindow, 432
nhalf
 dcdflib.cpp, 1167
nind
 Model, 535
nk
 Plink, 831
nl
 Plink, 831
nl_all
 Plink, 831
nlist
 Individual, 463
nlistMissing
 Individual, 463
nlistname
 Plink, 831
nm
 Locus, 510
no_show_covar
 par, 690
no_split_variables
 Forest, 328
 Tree, 1033
noCluster
 Model, 529
NoMem
 helper.cpp, 1291
 helper.h, 1312
nodeNameIndex
 InteractionNetwork, 484
nodeNames
 InteractionNetwork, 485
nonfounderphasing.cpp
 displayFamTran, 1383
 PP, 1383
nonfounders
 HaploPhase, 418
norm
 DistanceMetrics.cpp, 1208
 DistanceMetrics.h, 1215
normFactors
 DgeData, 244
NormalizeScores
 AttributeRanker, 45
normalizeScores
 AttributeRanker, 47
normdist
 stats.cpp, 1488
 stats.h, 1508
normsFilename
 DgeData, 244
np
 HaploPhase, 418
 HaploWindow, 432
 Model, 535
 Plink, 831
npc
 RCount, 881
npheno
 Plink, 832
nrfthreads
 par, 690
ns
 HaploPhase, 419
 HaploWindow, 432
 Perm, 768
nsh
 HaploPhase, 419
nsnp
 SimParameters, 1001
 SimParametersQT, 1006
nsnps
 RCount, 881
nt
 HaploPhase, 419
nthreads
 ArgumentHandler, 38
ntree
 ArgumentHandler, 39
 par, 690
nudge
 par, 690
null
 Chap, 76
 IDField, 442
num_cols
 Data, 119
num_cols_no_sparse
 Data, 119
num_deaths
 TreeSurvival, 1070
num_independent_variables
 Forest, 328
num_rows
 Data, 119
num_rows_rounded
 Data, 119

num_samples
 Forest, 328
 Tree, 1033

num_samples_at_risk
 TreeSurvival, 1070

num_samples_oob
 Tree, 1033

num_threads
 Forest, 329

num_timepoints
 TreeSurvival, 1070

num_trees
 Forest, 329

num_variables
 Forest, 329

NumAttributes
 Dataset, 178
 DatasetInstance, 208
 PlinkInternalsDatasetInstance, 848

numAttributes
 Regain, 899

NumClasses
 Dataset, 178

numClasses
 ChiSquared, 89

numCombs
 DcVar, 227

numDiffFuncPtr
 Dataset, 193
 ReliefF, 916

numDiffMetricName
 Dataset, 194
 par, 691
 ReliefF, 916

numEdges
 InteractionNetwork, 485

NumInstances
 Dataset, 178

numInstances
 EvaporativeCoolingPrivacy, 296

numIterations
 Model, 535

NumLevels
 Dataset, 178

numLevels
 ChiSquared, 89

NumNodes
 InteractionNetwork, 477

numNodes
 InteractionNetwork, 485

NumNumerics
 Dataset, 179
 DatasetInstance, 208
 PlinkInternalsDatasetInstance, 849

numRFThreads
 EvaporativeCooling, 285

numSamplesLeftOfCutpoint
 utility.cpp, 1542
 utility.h, 1560

numSamplesLeftOfCutpointInData
 utility.cpp, 1542
 utility.h, 1560

numSig
 Set, 994

numSignalsInData
 EvaporativeCoolingPrivacy, 296

numTarget
 ReliefF, 916

numTargetAttributes
 EvaporativeCooling, 285
 ReliefSeqController, 933

numThreads
 ReliefSeqController, 933

numToRemoveNextIteration
 EvaporativeCooling, 286
 ReliefSeqController, 933

numToRemovePerIteration
 EvaporativeCooling, 286
 ReliefSeqController, 933

NumVariables
 Dataset, 179

numVariables
 EvaporativeCoolingPrivacy, 296

number_list_positive
 par, 691

number_list_string
 par, 691

numeric
 Model, 535

numeric_extract_file
 par, 691

numeric_file
 par, 691

numeric_filename
 par, 691

numeric_lowval_percentile
 par, 692

numeric_lowvar_percentile
 par, 692

NumericLevel
 Insilico.h, 1342

numericLowValueFilter
 stats.cpp, 1488
 stats.h, 1508

numericMeanCenter
 stats.cpp, 1489
 stats.h, 1508

numericStandardize

stats.cpp, 1489
stats.h, 1508
numericVarianceFilter
 stats.cpp, 1489
 stats.h, 1508
numerics
 DatasetInstance, 214
numericsFilename
 Dataset, 194
numericsIds
 Dataset, 194
numericsMask
 Dataset, 194
numericsMaskPushed
 Dataset, 195
numericsMinMax
 Dataset, 195
numericsNames
 Dataset, 195
numericsSums
 Dataset, 195
nvec_bool
 helper.cpp, 1291
 helper.h, 1312
nw
 HaploPhase, 419

OR_homog_test
 par, 693
OUTFILE
 Plink, 832
oblig_clusters_filename
 par, 692
oblig_missing
 par, 692
 Plink, 832
oblig_missing_filename
 par, 692
obligMissing
 Plink, 802
observedFreqTable
 ChiSquared, 89
odds
 HaploPhase, 419
 ProxyResult, 859
offset
 globals.h, 1266
one
 CSNP, 112
 dcdflib.cpp, 1167
 Individual, 463
oob_sampleIDs
 Tree, 1033
open
 gzfilebuf, 375
 gzifstream, 385
 gzofstream, 391
 ZInput, 1083
 ZOutput, 1088
open_mode
 gzfilebuf, 375
openBinaryFile
 Plink, 802
openFile
 Rconnection, 874
operator!=
 IDValue, 455
operator<
 Alleles, 31
 ClumpPair, 98
 ClumpResults, 100
 double2, 253
 FamilyTransmissions, 310
 fullGenotype, 360
 IDField, 440
 IDValue, 455
 indivPair, 467
 int2, 469
 LDPair, 491
 Locus, 508
 MultiLocusGenotype, 539
 Neighbour, 541
 Pair, 553
 Pair2, 555
 ProxyResult, 858
 Range, 868
 ResultTrio, 936
 Segment, 986
 SetSortedSNP, 998
 SortedResult, 1016
 sortedSegments, 1018
 Var, 1072
operator<<
 gzomanip2, 394
 LDPair, 491
 Rexp, 942
 ZOutput, 1088
 zfstream.h, 1577
operator()
 deref_less, 231
 deref_less_bcw, 232
 insilico::do_to_lower, 248
 insilico::do_to_upper, 251
 insilico::is_classified, 489
 Pair_cmp, 556
 SegmentSizeCmp, 989
 std::less< Locus * >, 493
 std::less< MultiLocusGenotype * >, 494

std::less< Pool * >, 495
 std::less< Segment * >, 496
operator=
 Family, 307
 Locus, 509
operator==
 Alleles, 31
 double2, 253
 IDField, 440
 IDValue, 455
 int2, 469
 Locus, 509
 MultiLocusGenotype, 539
 Range, 869
 Segment, 986
 Var, 1072
opt
 Options, 548
 par, 692
optimalTemperature
 EvaporativeCooling, 286
OptimizeTemperature
 EvaporativeCooling, 278
optimizeTemperature
 EvaporativeCooling, 286
option
 CArgs, 65
optionLabel
 CArgs, 66
OptionSet, 549
 display, 551
 getValue, 551
 getValues, 551
 isSet, 551
 val, 552
Options, 547
 ~Options, 548
 addOption, 548
 getOptions, 548
 opt, 548
options.h
 getOutputFilename, 1384
 setOptions, 1384
order
 Model, 535
 Perm, 769
 utility.h, 1561
orderInData
 utility.cpp, 1543
 utility.h, 1561
original
 CArgs, 66
 Plink, 832
originalOrder
 Perm, 764
os_print
 Rdouble, 887
 Rexp, 941
 Rinteger, 949
 Rlist, 954
 Rstring, 969
 Rstrings, 973
 Rsymbol, 978
 Rvector, 983
out_SNP_major
 par, 693
out_missing_genotype
 par, 693
out_missing_phenotype
 par, 693
outFilesPrefix
 ReliefSeqController, 933
outerLoopSnp
 EpistasisEQtl, 271
outf
 ZOutput, 1089
outlier_detection
 par, 693
outprefix
 ArgumentHandler, 39
output.cpp
 LOG, 1386
 make2LTable, 1386
output_bimbam_format
 Plink, 803
output_fastphase_format
 Plink, 803
output_file_name
 par, 693
output_pheno_perm
 par, 694
output_prefix
 Forest, 329
output_structure_format
 Plink, 803
outputArffFile
 Plink, 803
OutputDatasetType
 Insilico.h, 1346
outputDelimitedFile
 Plink, 803
outputFormat
 Regain, 899
outputGenericVariantFile
 Plink, 804
outputNumericExtract
 Plink, 804
outputNumericFiltered

Plink, 804
outputPermedPhenotypes
 Plink, 804
outputSetFile
 Plink, 804
outputSifToGain
 Plink, 805
outputThreshold
 Regain, 899
outputTransform
 Regain, 899
overall_prediction_error
 Forest, 329
overflow
 gzfilebuf, 375
own_buffer
 gzfilebuf, 379
own_fd
 gzfilebuf, 379

P

 Chap, 76
 clump_LD, 96
 HWindow, 437
 HaplPhase, 419
 HaplWindow, 432
 Model, 535
 Perm, 769
 RCount, 882

p

 ChapModel, 79
 ClumpResults, 100
 LogisticModel, 516
 Pair, 553
 Pair2, 555
 ResultTrio, 937
 SortedResult, 1016

p1

 double2, 253
 HaplPhase, 420
 indivPair, 467
 int2, 469
 Pair, 554
 Segment, 987

p2

 double2, 254
 HaplPhase, 420
 indivPair, 467
 int2, 470
 Pair, 554
 Segment, 987

PAR_LENGTH
 Rsrv.h, 1439

PAR_LEN

 Rsrv.h, 1439
PAR_TYPE
 Rsrv.h, 1439

PDATE
 inbix.cpp, 1328

PDUMP
 Perm, 769

PORT_NUM
 lookup.cpp, 1371
 lookup2.cpp, 1372
 webcheck.cpp, 1569

PREL
 inbix.cpp, 1329
 webcheck.cpp, 1569

PVERSION
 inbix.cpp, 1329
 sockets.cpp, 1461
 webcheck.cpp, 1569

pVals
 DcVar, 227

Pair, 552
 l, 553
 operator<, 553
 p, 553
 p1, 554
 p2, 554

pair1
 Plink, 832

Pair2, 554
 l, 555
 operator<, 555
 p, 555

pair2
 Plink, 832

Pair_cmp, 556
 operator(), 556

pairable_cluster
 cluster.cpp, 1140

pairid
 Plink, 833

PairwiseLinkage, 557
 a, 558
 b, 559
 calculateCI, 558
 calculateLD, 558
 dp, 559
 dp_lower, 559
 dp_upper, 559
 knownAA, 559
 knownAB, 559
 knownBA, 560
 knownBB, 560
 lod, 560
 PairwiseLinkage, 558

rsq, 560
 unknownDH, 560
 par, 561
 adaptive_alpha, 582
 adaptive_ci, 582
 adaptive_interval, 583
 adaptive_interval2, 583
 adaptive_max, 583
 adaptive_min, 583
 adaptive_perm, 583
 af_count, 583
 af_file, 584
 af_read, 584
 af_write, 584
 affpair, 584
 afni1dFile, 584
 afniNetwork, 584
 algorithm_verbose, 585
 algorithmMode, 585
 algorithmSeqMode, 585
 algorithmSeqS0, 585
 algorithmSnrMode, 585
 algorithmTstatMode, 585
 all_pheno, 586
 alpha, 586
 alwaysplitvars, 586
 analyse_cluster, 586
 annot_file, 586
 annot_filename, 586
 assoc_counts, 587
 assoc_glm, 587
 assoc_glm_without_main_snp, 587
 assoc_gxe, 587
 assoc_test, 587
 assoc_test_alt_perm, 587
 BATCH_SIZE, 588
 bitfilename, 588
 bitfilename_map, 588
 bmatch, 588
 bmatch_direction_filename, 588
 bmatch_filename, 588
 bmatch_usertype, 589
 boot, 589
 bound, 589
 breslowday, 589
 bt, 589
 built_families, 589
 CMH_test_1, 600
 CMH_test_2, 601
 CMH_test_ORD, 601
 calc_SNPSNP_LD, 590
 calcFst, 590
 caseweights, 590
 catvars, 590
 chap_add_grp_specifics, 590
 chap_drop_snps, 590
 chap_drop_snps_list, 591
 chap_entity, 591
 chap_haplotype_specific, 591
 chap_independent_effect, 591
 chap_model0, 591
 chap_model1, 591
 chap_sole_variant, 592
 chap_sole_variant_specific_allele_list, 592
 chap_sole_variant_specific_alleles, 592
 chap_specified_groups, 592
 chap_specified_snps, 592
 chap_test, 592
 check_sex, 593
 chr_code, 593
 chr_haploid, 593
 chr_map, 593
 chr_sex, 593
 chr_Y, 593
 ci_level, 594
 ci_zt, 594
 cli, 594
 clist, 594
 clist_filename, 594
 clist_number, 594
 clist_selection, 595
 clist_selection_name, 595
 clist_selection_number, 595
 clist_selection_string, 595
 clumpld, 595
 clumpld_annot, 595
 clumpld_annot_fields, 596
 clumpld_best, 596
 clumpld_column, 596
 clumpld_indep, 596
 clumpld_index1, 596
 clumpld_kb, 596
 clumpld_min, 597
 clumpld_only_show_replications, 597
 clumpld_only_show_replications_list, 597
 clumpld_p1, 597
 clumpld_p2, 597
 clumpld_r2, 597
 clumpld_range_annotate, 598
 clumpld_range_file, 598
 clumpld_results, 598
 clumpld_verbose, 598
 cluster, 598
 cluster_euclidean, 598
 cluster_group_avg, 599
 cluster_ibm_constraint, 599
 cluster_ibm_constraint_value, 599
 cluster_mds_dim, 599

cluster_missing, 599
cluster_on_mcc, 599
cluster_on_phenotype, 600
cluster_plot, 600
cluster_selcon, 600
cluster_selcon_file, 600
cm_map, 600
cnv_check_overlap, 601
cnv_col, 601
cnv_count, 601
cnv_count_baseline, 601
cnv_count_baseline_file, 602
cnv_defined_overlap, 602
cnv_del_only, 602
cnv_disrupt, 602
cnv_drop_no_segment, 602
cnv_dup_only, 602
cnv_en_model, 603
cnv_enrichment_test, 603
cnv_exclude, 603
cnv_freq_include, 603
cnv_freq_include_below, 603
cnv_freq_include_cnt, 603
cnv_freq_include_exact, 604
cnv_freq_include_exact_exclude, 604
cnv_freq_method2, 604
cnv_freq_method2_threshold, 604
cnv_glm, 604
cnv_indiv_perm, 604
cnv_intersect, 605
cnv_intersect_file, 605
cnv_intersect_subset, 605
cnv_intersect_subset_file, 605
cnv_intersect_writeback, 605
cnv_intersect_writeback_verbose, 605
cnv_list, 606
cnv_listname, 606
cnv_makemap, 606
cnv_max_kb, 606
cnv_max_score, 606
cnv_max_sites, 606
cnv_min_kb, 607
cnv_min_score, 607
cnv_min_sites, 607
cnv_overlap, 607
cnv_pos_perm, 607
cnv_region_border, 607
cnv_region_overlap, 608
cnv_type, 608
cnv_union_overlap, 608
cnv_unique, 608
cnv_weighted_gene_test, 608
cnv_write_freq, 608
cnv_writelist, 609
coding01, 609
compound_genotype_code, 609
compress_file, 609
compress_filename, 609
compress_genome, 609
conditioning_snp_name, 610
conditioning_snp_single, 610
conditioning_snps, 610
conditioning_snps_file, 610
covar_file, 610
covar_filename, 610
CP, 611
dcvar_chip_seq_file, 611
dcvar_gene_expression_file, 611
dcvar_genotypes_file, 611
dcvar_pfilter_type, 611
dcvar_pfilter_value, 611
dcvar_radius, 612
dcvar_resume.snp, 612
dcvar_snp_locations_file, 612
dcvar_var_model, 612
debug, 612
deconvolutionAlpha, 612
deconvolutionBeta, 613
deconvolutionControl, 613
depvarname, 613
dfam_sibs, 613
dfam_tdt, 613
dfam_unrelateds, 613
discordant_parents, 614
disp_r1, 614
disp_r2, 614
disp_r_window, 614
disp_r_window_kb, 614
disp_r_window_r2, 614
disp_r_window.snp, 615
display_ci, 615
display_cnv_track, 615
display_hap_freqs, 615
display_phase_probs, 615
display_phase_probs_wide, 615
display_segment_long, 616
distance_matrix, 616
distanceMatrixFilename, 616
do_coexpression_all, 616
do_coexpression_casecontrol, 616
do_covariance_matrix, 616
do_dcgain_abs, 617
do_dcvar, 617
do_dcvar_chipseq, 617
do_dcvar_pfilter, 617
do_deconvolution, 617
do_differential_coexpression, 617
do_differential_modularity, 618

do_dmgain_abs, 618
 do_ec, 618
 do_ec_privacy, 618
 do_iqtl, 618
 do_iqtl_tf, 618
 do_iterative_removal, 619
 do_iterative_removal_ec, 619
 do_iterative_write_scores, 619
 do_kopt, 619
 do_modularity, 619
 do_normalize_scores, 619
 do_not_load_snps, 620
 do_numeric_extract, 620
 do_numeric_lowval_filter, 620
 do_numeric_lowvar_filter, 620
 do_numeric_standardize, 620
 do_numeric_summary, 620
 do_randomforest, 621
 do_ranker_permutation, 621
 do_ranking, 621
 do_regain, 621
 do_regain_post, 621
 do_regain_pvalue_threshold, 621
 do_relieff, 622
 do_rfprobability, 622
 do_ripm, 622
 do_write_best_k, 622
 do_write_each_k_scores, 622
 done_global_pihat, 622
 dosage_assoc, 623
 dosage_file, 623
 dosage_hard_call, 623
 dosage_hard_call_thresh, 623
 dosage_hard_call_thresh2, 623
 dosage_hasMap, 623
 drop_sets, 624
 dummy, 624
 dummy_nind, 624
 dummy_nsnp, 624
 dump_clst, 624
 dump_covar, 624
 dump_covar_dummy_coding, 625
 dump_covar_with_phenotype, 625
 dump_gene, 625
 dump_genename, 625
 dumped, 625
 ecAlgorithmSteps, 625
 ecltAlgorithm, 626
 eclterNumToRemove, 626
 eclterPercentToRemove, 626
 ecMeAlgorithm, 626
 ecNumTarget, 626
 ecOptimizeTemp, 626
 ecPrivacyFinalTemp, 627
 ecPrivacyHoldoutFile, 627
 ecPrivacyPercentSignal, 627
 ecPrivacyStartTemp, 627
 ecPrivacyTau, 627
 ecPrivacyTestFile, 627
 ecPrivacyTrainFile, 628
 ecPrivacyUpdateFrequency, 628
 elf_baseline, 628
 elf_pcemode, 628
 elf_pcemode_2sided, 628
 epi_alpha1, 628
 epi_alpha2, 629
 epi_caseonly, 629
 epi_caseonly_kb_gap, 629
 epi_filter, 629
 epi_genebased, 629
 epi_quickscan, 629
 epistasis, 630
 epsilon, 630
 exclude_file, 630
 exclude_set, 630
 exportArff, 630
 exportDelimited, 630
 exportDelimiter, 631
 extract_before_exclude, 631
 extract_file, 631
 extract_set, 631
 FIX_IBD, 634
 FIX_p, 634
 FIXED_p, 634
 FIXED, 634
 famfile, 631
 fast_binary, 631
 fast_epistasis, 632
 fileroot, 632
 filter_cases, 632
 filter_controls, 632
 filter_females, 632
 filter_filename, 632
 filter_founders, 633
 filter_males, 633
 filter_nonfounders, 633
 filter_on_covar, 633
 filter_value, 633
 fisher_test, 633
 fix_lambda, 634
 fix_prev, 634
 fixed_prev, 635
 flag, 635
 flip_file, 635
 flip_scan, 635
 flip_scan_threshold, 635
 flip_scan_verbose, 635
 flip_strand, 636

flip_subset, 636
flip_subset_file, 636
force_initial_cluster, 636
force_span, 636
fraction, 636
fringe, 637
from_window, 637
full_model_assoc, 637
fuzzy_homo, 637
gainMatrixFilename, 637
geno_qual_file, 637
geno_qual_max, 638
geno_qual_min, 638
genome_2sets, 638
genome_groups, 638
genome_only_check_rels, 638
genome_output, 638
genome_output_full, 639
genome_output_minimal, 639
genome_setlist1, 639
genome_setlist2, 639
genome_test, 639
genome_test_min.snp, 639
genome_test_threshold, 640
gettag_file, 640
gettag_kb, 640
gettag_listall, 640
gettag_mode, 640
gettag_mode1, 640
gettag_mode2, 641
gettag_r2, 641
gfamfile, 641
glm_dominant, 641
glm_no_auto_sex_effect, 641
glm_recessive, 641
glm_sex_effect, 642
glm_user_parameters, 642
glm_user_test, 642
gmapfile, 642
gmulti_output, 642
gplink, 642
greport, 643
greport_display_empty, 643
greport_gene_list, 643
greport_results, 643
greport_subset, 643
greport_subset_file, 643
grid, 644
gvar, 644
gvar_full_report, 644
gvar_include_all_variants, 644
gvar_to_standard, 644
gvar_verbose_association, 644
gvar_write, 645
gvarfile, 645
HWD_filter_on_all, 653
HWD_limit, 653
HWD_report, 653
HWD_standard, 653
HWD_test, 654
hap_max_nf_phases, 645
hap_min_phase_prob, 645
hap_missing_geno, 645
hap_post_prob, 645
hap_specific_snps, 646
hap_specific_snps_list, 646
haplo_plem_follow, 646
haplo_plem_follow_fid, 646
haplo_plem_follow_id, 646
haplo_plem_follow_ind, 646
haplo_plem_iter, 647
haplo_plem_likelihood_iter, 647
haplo_plem_meta_iter, 647
haplo_plem_meta_likelihood_iter, 647
haplo_plem_meta_prune_haplotype, 647
haplo_plem_meta_prune_phase, 647
haplo_plem_meta_tol, 648
haplo_plem_meta_window, 648
haplo_plem_nonzero_threshold, 648
haplo_plem_original_overlap, 648
haplo_plem_overlap, 648
haplo_plem_verbose, 648
haplo_plem_window, 649
haplo_plem_window_prune_phase, 649
haplo_plem_window_tol, 649
haplo_plem_zero_threshold, 649
has_nonfounders, 649
have_numerics, 649
have_snps, 650
holdout, 650
homo_miss_as_hom, 650
homo_run, 650
homo_run_consensus_match, 650
homo_run_density, 650
homo_run_gap, 651
homo_run het, 651
homo_run_kb, 651
homo_run_length_kb, 651
homo_run_length_snps, 651
homo_run_snps, 651
homo_summary_allelic_match, 652
homo_threshold, 652
homo_windowAllowedHet, 652
homo_windowAllowedMissing, 652
homo_windowKB, 652
homo_windowSize, 652
homozyg_verbose, 653
hotel, 653

IBD_threshold, 655
 IBD_within, 655
 ibd_file, 654
 ibd_file_list, 654
 ibd_read, 654
 ibd_read_list, 654
 ibd_read_minimal, 654
 ibs_2only, 655
 ibs_inner_run_length_kb, 655
 ibs_inner_run_length_snp, 655
 ibs_inter.snp_distance, 655
 ibs_join_kb, 656
 ibs_join.snp, 656
 ibs_run, 656
 ibs_run_0, 656
 ibs_run_density, 656
 ibs_run_length_kb, 656
 ibs_run_length_snps, 657
 ibs_run_missing, 657
 ibs_sharing_test, 657
 ibs_test, 657
 ibs_test_method2, 657
 ibs_test_min.snp, 657
 ibstest_gap, 658
 idhelp, 658
 idhelp_alias_update, 658
 idhelp_auto_alias, 658
 idhelp_command, 658
 idhelp_dictionary, 658
 idhelp_dump_from_dict, 659
 idhelp_dump_from_dict_cmd, 659
 idhelp_input, 659
 idhelp_list_aliases, 659
 idhelp_lookup, 659
 idhelp_lookup_string, 659
 idhelp_match, 660
 idhelp_match_string, 660
 idhelp_no_dict, 660
 idhelp_output_delimit, 660
 idhelp_replace, 660
 idhelp_replace_string, 660
 idhelp_subset, 661
 idhelp_subset_string, 661
 ignore_missing_sex, 661
 ignore_phenotypes, 661
 impmeasure, 661
 impute_sex, 661
 impute_tags, 662
 impute_verbose, 662
 inbreeding, 662
 inc_file, 662
 inc_read, 662
 inc_write, 662
 include_all_pairs, 663
 include_all_z1, 663
 include_cluster, 663
 include_cluster_filename, 663
 include_cluster_from_file, 663
 ind_attrib_file, 663
 ind_attrib_filter, 664
 ind_attrib_value, 664
 indiv_report, 664
 indiv_report_fid, 664
 indiv_report_iid, 664
 inter_grid, 664
 iqtl_coord_file, 665
 iqtl_expression_file, 665
 iqtl_interaction_full, 665
 iqtl_local_cis, 665
 iqtl_pvalue, 665
 iqtl_radius, 665
 iqtl_tf_coord_file, 666
 iqtl_tf_radius, 666
 k, 666
 keep_file, 666
 keep_indiv, 666
 keep_indiv_list, 666
 koptBegin, 667
 koptEnd, 667
 koptStep, 667
 lambda, 667
 ld_SNP1, 668
 ld_SNP1_file, 668
 ld_SNP2, 668
 ld_anchor, 667
 ld_anchor_list, 667
 lfamfile, 668
 lfile_allele_count, 668
 lfile_input, 668
 liability, 669
 list_by_allele, 669
 list_twolocus, 669
 load_gvar, 669
 locked, 669
 logistic_model_iterations, 669
 logistic_tolerance_epsilon, 670
 logscale, 670
 lookup, 670
 lookup2, 670
 lookup2_cmd, 670
 lookup_gene, 670
 lookup_gene_kb_window, 671
 lookup_gene_name, 671
 lookup_multiple_genes, 671
 lookup_single_snp, 671
 lookup_snp, 671
 lookup.snp_kb_window, 671
 lookup_to_file, 672

loop_counter, 672
loop_over, 672
loop_over_filename, 672
loop_over_label, 672
lpedfile, 672
m, 673
m1, 673
m2, 673
MAX_CORR_PIHAT_PIHAT_G, 677
MAX_GENO_MISSING, 678
MAX_IND_MISSING, 678
MAX_LINE_LENGTH, 678
MAX_PIHAT, 678
MENDEL_ind, 679
MENDEL_report, 679
MENDEL_snp, 680
MENDEL_test, 680
MIN_PIHAT, 683
make_blocks, 673
make_founders, 673
make_minor_allele, 673
make_missing_parents, 674
make_pheno, 674
make_pheno_filename, 674
make_pheno_present, 674
make_pheno_value, 674
make_set, 674
make_set_border, 675
make_set_collapse, 675
make_set_collapse_label, 675
make_set_complement, 675
make_set_file, 675
make_set_ignore_group, 675
make_tags, 676
map3, 676
mapfile, 676
mapfile_impute, 676
mating_tests, 676
matrix, 676
max_af, 677
max_cluster_case, 677
max_cluster_control, 677
max_cluster_N, 677
max_cluster_size, 677
max_hf, 678
max_neighbour, 678
maxMergeOrder, 679
maxModuleSize, 679
mds_by_individual, 679
memmode, 679
merge_bedfile, 680
merge_bimfile, 680
merge_binary, 680
merge_data, 680
merge_famfile, 681
merge_force_strand, 681
merge_list, 681
merge_list_filename, 681
merge_mapfile, 681
merge_mode, 681
merge_p, 682
merge_pedfile, 682
meta_analysis, 682
meta_files, 682
meta_large_phase, 682
min_af, 682
min_genotype_cell, 683
min_hf, 683
min_neighbour, 683
minModuleSize, 683
minprop, 683
mishap_test, 684
mishap_window, 684
miss_run, 684
miss_run_length, 684
miss_run_length_kb, 684
miss_run_level, 684
missing_genotype, 685
missing_genotype_explicit, 685
missing_phenotype, 685
missing_phenotype_explicit, 685
mk_datfile, 685
modComputeHomophily, 685
modConnectivityThreshold, 686
modEnableConnectivityThreshold, 687
modFisherTransform, 687
modFisherTransformCutoff, 687
modPowerTransform, 687
modPowerTransformExponent, 687
modUseBinaryThreshold, 687
model_perm_best, 686
model_perm_dom, 686
model_perm_gen, 686
model_perm_rec, 686
model_perm_trend, 686
mperm_rank, 688
mperm_save_all, 688
mperm_save_best, 688
mtry, 688
mult_clst, 688
mult_covar, 688
mult_filter, 689
mult_pheno, 689
multi_output, 689
multiple_phenotype_file, 689
multiple_phenotypes, 689
multtest, 689
myfunction, 690

name_pheno, 690
no_show_covar, 690
nrftthreads, 690
ntree, 690
nudge, 690
numDiffMetricName, 691
number_list_positive, 691
number_list_string, 691
numeric_extract_file, 691
numeric_file, 691
numeric_filename, 691
numeric_lowval_percentile, 692
numeric_lowvar_percentile, 692
OR_homog_test, 693
oblig_clusters_filename, 692
oblig_missing, 692
oblig_missing_filename, 692
opt, 692
out_SNP_major, 693
out_missing_genotype, 693
out_missing_phenotype, 693
outlier_detection, 693
output_file_name, 693
output_pheno_perm, 694
parameter_list, 694
parent_of_origin, 694
ped_from_stdin, 694
ped_skip_fid, 694
ped_skip_parents, 694
ped_skip_pheno, 695
ped_skip_sex, 695
pedfile, 695
perm_POO_best, 696
perm_POO_mat, 696
perm_POO_pat, 696
perm_POO_poo, 697
perm_TDT_basic, 697
perm_TDT_parent, 697
perm_count, 695
perm_genedrop, 695
perm_genedrop_and_swap, 695
perm_genedrop_parents, 696
perm_genedrop_sibships, 696
perm_genedrop_unrel, 696
permute, 697
permute_within_sol, 697
pfilter, 697
pfvalue, 698
phase_hap_all, 698
phase_snps, 698
pheno_file, 698
pheno_filename, 698
pihat_filter, 698
plink, 699
plist, 699
plist_fid1, 699
plist_fid2, 699
plist_iid1, 699
plist_iid2, 699
plist_number, 700
plist_selection, 700
plist_selection_name, 700
plist_selection_number, 700
plist_selection_string, 700
pool_size_min, 700
position_window, 701
pp_maxfid, 701
pp_maxiid, 701
pp_maxsnp, 701
predall, 701
predict, 701
preserve_all_genotypes, 702
preserve_mendel_errors, 702
profile_sets, 702
proxy_CC, 703
proxy_TDT, 710
proxy_all, 702
proxy_all_list, 702
proxy_all_list_file, 702
proxy_assoc, 703
proxy_assoc.snp, 703
proxy_error, 703
proxy_exclude, 703
proxy_exclude_from_file, 703
proxy_exclude_list, 704
proxy_full_report, 704
proxy_geno, 704
proxy_glm, 704
proxy_impute, 704
proxy_impute_genotypic_concordance, 704
proxy_impute_preserve_genotyped, 705
proxy_impute_replace, 705
proxy_impute_threshold, 705
proxy_include_reference, 705
proxy_info_threshold, 705
proxy_kb, 705
proxy_kb_planA, 706
proxy_kb_planB, 706
proxy_leave_out, 706
proxy_list, 706
proxy_list_file, 706
proxy_list_proxies, 706
proxy_maf, 707
proxy_maxhap, 707
proxy_mhf, 707
proxy_planB_threshold, 707
proxy_r2, 707
proxy_r2_filter, 707

proxy_r2_filter_A_planA, 708
proxy_r2_filter_A_planB, 708
proxy_r2_filter_B_planA, 708
proxy_r2_filter_B_planB, 708
proxy_r2_filter_C_planA, 709
proxy_r2_filter_C_planB, 709
proxy_r2_filter_A, 708
proxy_r2_filter_B, 708
proxy_r2_filter_C, 709
proxy_record dosage, 709
proxy_reference_only, 709
proxy_snp_filter, 709
proxy.snp_filter_planA, 710
proxy.snp_filter_planB, 710
proxy_window, 710
proxy_window_planA, 710
proxy_window_planB, 710
prune_id, 711
prune_id_pairwise, 711
prune_id_pairwise_maf, 711
prune_id_r2, 711
prune_id_step, 711
prune_id_vif, 711
prune_id_win, 712
prune_r2_fixed, 712
prune_r2_fixed_list, 712
prune_r2_prefer, 712
prune_r2_prefer_list, 712
QFAM_adaptive, 712
QFAM_between, 713
QFAM_total, 713
QFAM_within1, 713
QFAM_within2, 713
QTDT_test, 714
qmatch, 713
qmatch_filename, 713
qmatch_threshold_filename, 714
qq_plot, 714
qt, 714
qt_means, 714
qt_with_covariates, 714
R_port, 715
R_script, 715
random_seed, 715
range_delimiter, 715
ranker_centrality_gamma, 715
ranker_input_file, 715
ranker_method, 716
ranker_save_data_file, 716
ranker_top_n, 716
rankerPermMethod, 716
rankerPermNum, 716
rankerPermThreshold, 716
rare_test, 717
rare_test_print_details, 717
rare_test_print_details_snp, 717
rare_test_score_range, 717
rare_test_score_range_file, 717
rare_test_score_range_threshold, 717
rare_test_score_results_file, 718
rare_test_summary_controls, 718
rare_test_weight1, 718
rarer_dist_threshold, 718
rarer_interval, 718
rarer_maf_threshold, 718
read_bitfile, 719
read_geno_qual, 719
read_ped, 719
read_segment_file, 719
read_segment_filename, 719
read_set, 719
read_snp_qual, 720
recode, 720
recode_12, 720
recode_1234, 720
recode_ACGT, 720
recode_AD_Aonly, 721
recode_AD_fixed, 721
recode_AD, 720
recode_HV, 722
recode_allele_coding, 721
recode_allele_coding_file, 721
recode_bim bam, 721
recode_delimit, 721
recode_fastphase, 722
recode_indelimit, 722
recode_long, 722
recode_long_ref, 722
recode_mutlist, 722
recode_structure, 723
recode_transpose, 723
recode_whap, 723
ref_file, 723
ref_file_name, 723
regainComponents, 723
regainCompress, 724
regainFailValue, 724
regainFdr, 724
regainFdrPrune, 724
regainFile, 724
regainLargeCoefPvalue, 724
regainLargeCoefTvalue, 725
regainMatrixFormat, 725
regainMatrixThreshold, 725
regainMatrixThresholdValue, 725
regainMatrixToSif, 725
regainMatrixTransform, 725
regainMaxBetaValue, 726

regainPureInteractions, 726
 regainPValueThreshold, 726
 regainSifFilter, 726
 regainSifThreshold, 726
 regainUseBetaValues, 726
 reliefMode, 727
 reliefSeqAlgorithmMode, 727
 reliefSeqAlgorithmS0, 727
 reliefSeqSnrMode, 728
 reliefSeqTstatMode, 728
 relieffilterNumToRemove, 727
 relieffilterPercentToRemove, 727
 reliefNumTarget, 727
 remove_before_keep, 728
 remove_file, 728
 remove_indiv, 728
 remove_indiv_list, 728
 remove_unaffected_pairs, 729
 replicates, 729
 report_missing, 729
 return_beta, 729
 rfreplace, 729
 Rmessage, 960
 run_R_chisq, 730
 run_R_nsnp, 730
 run_R_script, 730
 run_R_write_script, 730
 run_R_z, 730
 run_chr, 729
 run_end, 730
 run_start, 731
 SNP_major, 746
 savemem, 731
 score_impute_expected, 731
 score_qfile, 731
 score_qrange_file, 731
 score_risk, 731
 score_risk_file, 732
 score_risk_on_qrange, 732
 score_risk_ranges, 732
 score_risk_ranges_file, 732
 score_risk_ranges_min, 732
 score_test, 732
 SD, 733
 seg_test_region, 733
 seg_test_window, 733
 seg_test_window_bp, 733
 segment_current_focal_snp, 733
 segment_haplotype, 733
 segment_haplotype_fid1, 734
 segment_haplotype_fid2, 734
 segment_haplotype_iid1, 734
 segment_haplotype_iid2, 734
 segment_inter.snp_distance, 734
 segment_length, 734
 segment_m1, 735
 segment_m2, 735
 segment_minimal, 735
 segment_output, 735
 segment_output_started, 735
 segment_overlap, 735
 segment_silently_return_groups, 736
 segment_snp, 736
 segment_snp1, 736
 segment_snp2, 736
 segment_test_1sided, 736
 segment_test_fisher, 736
 segment_test_force_1sided, 737
 segment_test_ignore_discordant, 737
 segment_test_individual, 737
 segment_test_specific_segs, 737
 segment_threshold_finish, 737
 segment_threshold_start, 737
 segment_validate, 738
 segment_verbose, 738
 set_by_set, 738
 set_chisq_threshold, 738
 set_max, 738
 set_min, 738
 set_p2, 739
 set_r2, 739
 set_r2_phase, 739
 set_r2_read, 739
 set_r2_read_file, 739
 set_r2_val, 739
 set_r2_write, 740
 set_reference_allele, 740
 set_reference_allele_file, 740
 set_score, 740
 set_score_p, 740
 set_screen, 740
 set_screen_resultfile, 741
 set_step, 741
 set_step_in, 741
 set_table, 741
 set_test, 741
 setfile, 741
 sex_threshold_female, 742
 sex_threshold_male, 742
 show_impossible_IBD, 742
 sibTDT_test, 742
 sifFile, 742
 sifNetwork, 742
 sifToGain, 743
 silent, 743
 simple_interaction, 743
 simul, 743
 simul_file, 743

simul_haps, 743
simul_label, 744
simul_ncases, 744
simul_ncontrols, 744
simul_prevalence, 744
simul_qt, 744
simul_qt_var, 744
simul_tags, 745
singlepoint, 745
sliding_window, 745
sliding_window_size, 745
snp_attrib_file, 745
snp_attrib_filter, 745
snp_attrib_value, 746
snp_include_from_cl, 746
snp_include_range, 746
snp_qual_file, 746
snp_qual_max, 746
snp_qual_min, 747
snp_range_list, 747
snpDiffMetricName, 747
snpNearestNeighborMetricName, 747
sol_family, 747
species_cow, 747
species_dog, 748
species_horse, 748
species_mouse, 748
species_rice, 748
species_sheep, 748
splitrule, 748
splitweights, 749
standard_beta, 749
startMergeOrder, 749
statusvarname, 749
subsetfile, 749
summ_nonfounders, 749
summary_ibd_output, 750
TDT_test, 750
tagfile, 750
targetpartitionsize, 750
test_full_model, 750
test_hap_CC, 750
test_hap_GLM_omnibus, 751
test_hap_GLM, 751
test_hap_QTL, 751
test_hap_TDT, 751
test_hap_only, 751
test_list, 751
test_missing, 752
tfamfile, 752
tfile_input, 752
thin_param, 752
thin_snps, 752
thresholdType, 752
thresholdValue, 753
titvFilename, 753
to_window, 753
tpedfile, 753
treetype, 753
trend_only, 753
tucc, 754
twoDFmodel, 754
twoDFmodel_hethom, 754
twolocus_snp1, 754
twolocus_snp2, 754
uncompress_file, 754
update_allele_file, 755
update_alleles, 755
update_chr, 755
update_cm, 755
update_ids, 755
update_ids_file, 755
update_map, 756
update_mapfile, 756
update_name, 756
update_parents, 756
update_parents_file, 756
update_pheno, 756
update_pheno_file, 757
update_sex, 757
update_sex_file, 757
use_GC, 757
use_subset, 757
useAbs, 757
useWeighted, 758
verbose, 758
vif_threshold, 758
web_check, 758
weightByDistanceMethod, 758
weightByDistanceSigma, 758
weighted_mm, 759
window, 759
write_bitfile, 759
write dosage, 759
write_set, 759
write_snplist, 759
writeforest, 760
xchr_model, 760
zero_cluster, 760
zero_cluster_filename, 760
parameter_list
 par, 694
parent_of_origin
 par, 694
parents
 Family, 309
pars
 Rmessage, 960

parse
 Rexp, 941
 Rmessage, 959

parse.cpp
 getOutputFilename, 1387
 setOptions, 1387

parse2int
 helper.cpp, 1291
 helper.h, 1313

parse2str
 helper.cpp, 1291
 helper.h, 1313

parseOptions
 CArgs, 64

parseQuery
 IDHelper, 450

parseTrios
 Plink, 805

parsed
 CArgs, 66

pat
 Family, 309
 Individual, 463

pcMode
 RCount, 882

pca
 stats.cpp, 1489
 stats.h, 1509

ped_from_stdin
 par, 694

ped_skip_fid
 par, 694

ped_skip_parents
 par, 694

ped_skip_pheno
 par, 695

ped_skip_sex
 par, 695

pedfile
 par, 695

performAlternEM
 HaploPhase, 406

performEM_original
 HaploPhase, 406

performEM
 HaploWindow, 428

performHaplotypeTests
 HaploPhase, 407

performMisHapTests
 Plink, 805

performProxyTests
 Plink, 805

performPureInteraction
 Regain, 893

performed
 Perm, 769

Perm, 761
 adaptive, 767
 closeDUMP, 763
 count, 767
 current_reps, 763
 dropAlleles, 763
 dump_all, 767
 dump_best, 767
 finished, 763
 geneDrop, 763
 genedrop, 767
 geno, 767
 idmap, 768
 interval, 768
 max_pvalue, 764
 maxR, 768
 min, 768
 N, 768
 nextSNP, 764
 ns, 768
 order, 769
 originalOrder, 764
 P, 769
 PDUMP, 769
 performed, 769
 Perm, 762
 permuteInCluster, 764
 pheno, 769
 preGeneDrop, 764
 pvalue, 764
 R, 769
 rank, 765
 reorder, 770
 replicates, 770
 report, 765
 reps_done, 765
 s, 770
 setAdaptiveSetSNPs, 765
 setOriginalRanking, 765
 setPermClusters, 766
 setTests, 766
 .snp_test, 770
 t, 770
 test, 770
 update, 766
 updateSNP, 766
 zt, 771

perm_POO_best
 par, 696

perm_POO_mat
 par, 696

perm_POO_pat

par, 696
perm_POO_poo
 par, 697
perm_TDT_basic
 par, 697
perm_TDT_parent
 par, 697
perm_count
 par, 695
perm_genedrop
 par, 695
perm_genedrop_and_swap
 par, 695
perm_genedrop_parents
 par, 696
perm_genedrop_sibships
 par, 696
perm_genedrop_unrel
 par, 696
perm_segmentIndividualTest
 Plink, 805
perm_sharingIBSTest
 Plink, 806
perm_testGXE2
 Plink, 806
perm_testHotel
 Plink, 806
perm_testQTDT
 Plink, 806
perm_testTDT_POO
 Plink, 807
perm_testTDT
 Plink, 806
permTestRareDistribution
 Plink, 807
permutationIBSTest
 Plink, 807
Permute
 CentralityRanker, 69
permute
 helper.cpp, 1292
 helper.h, 1313
 par, 697
permute_within_sol
 par, 697
permuteAndPredictOobSamples
 Tree, 1029
permuteInCluster
 Perm, 764
person
 Cluster, 101
 HWindow, 438
pF
 stats.cpp, 1489
 stats.h, 1509
pfILTER
 par, 697
pfvalue
 par, 698
ph_freq
 HaploPhase, 420
ph_hap1
 HaploPhase, 420
ph_hap2
 HaploPhase, 420
phase.cpp
 LOG, 1389
phase_hap_all
 par, 698
phase_snps
 par, 698
phaseAllHaplotypes
 HaploPhase, 407
phaseAndScoreNonfounder
 HaploPhase, 407
phased
 probabilisticGenotype, 856
phased_genotype
 probabilisticGenotype, 856
phasemap
 HaploPhase, 420
phdr, 771
 cmd, 772
 dof, 772
 len, 772
 res, 772
pheno
 Perm, 769
pheno_file
 par, 698
pheno_filename
 par, 698
phenoLabel
 Plink, 833
phenosBase
 ExpressionDataSimulator, 304
phenosFilename
 BirdseedData, 58
 DgeData, 245
phenotype
 Individual, 464
 Plink, 833
phenotype_name
 Plink, 833
phenotypes
 BirdseedData, 58
 DgeData, 245
phenotypesIds

Dataset, 195
PI
 fisher.h, 1241
pihat
 Plink, 833
pihat_filter
 par, 698
pihat_G
 Plink, 833
pillai
 genepi.cpp, 1256
Plink, 773
 alleleRecoding, 780
 annotateFile, 780
 attribFilterInd, 780
 attribFilterSNP, 780
 buildCluster, 781
 buildT, 781
 calcAssociationWithBootstrap, 781
 calcAssociationWithPermutation, 781
 calcEpistasis, 781
 calcFlipScan, 782
 calcFst, 782
 calcGXE, 783
 calcGenomeIBD, 782
 calcGenomeIBM, 782
 calcGenomeIBS, 782
 calcHomog, 783
 calcHotel, 783
 calcInbreeding, 783
 calcInformative, 783
 calcLDStatistics, 784
 calcLocusIBD, 784
 calcMantelHaenszel_2x2xK, 784
 calcMantelHaenszel_IxJxK, 784
 calcMantelHaenszel_ORD, 784
 calcMH, 785
 calcMultiPoint, 785
 calcPairwiseLD, 785
 calcPhenotypes, 785
 calcQTDT, 785
 calcRegression, 786
 calcSinglePoint, 786
 calcStratifiedAlleleFreqs, 786
 calculateProfile, 786
 checkMendel, 787
 cleanUp, 787
 clistname, 824
 cnt_f, 824
 conditionalHaplotypeTest, 787
 conditioner, 824
 conditioner_mask, 824
 convertGenericVariantData, 787
 correlation2SNP, 787
countCNVPerRegion, 788
deleteIndividuals, 788
deleteIndividualsNumeric, 788
deleteSNPs, 788, 789
diff_concordant_overlap, 824
diff_nonmissing_overlap, 824
diff_overlap, 825
display_indivReport, 789
display_listByAllele, 789
display_pairList, 789
display_recoded_LONG, 790
display_recoded_MUTLIST, 790
display_recoded_PEDFILE_AD, 790
display_recoded_PEDFILE_transpose, 790
display_recoded_PEDFILE, 790
display_twolocus, 790
displayGMULTI, 791
displayGeneReport, 791
displayGenomePV, 791
displayGenomeWideInfo, 791
displayQTMeans, 791
displayRareRange, 792
displaySegmentsBED, 792
displaySegmentsLong, 792
doRegression, 792
driverSCREEPI, 792
dst, 825
dummyLoader, 793
E00, 825
E01, 825
E02, 825
E10, 825
E11, 826
E12, 826
E20, 826
E21, 826
E22, 826
elfBaseline, 793
epi1, 826
epi2, 827
expectedOverlap, 827
expectedOverlapBaseline, 827
extractExcludeSet, 793
family, 827
filterOnCase, 793
filterOnControl, 793
filterOnCovariate, 793
filterOnFemale, 794
filterOnFounder, 794
filterOnMale, 794
filterOnNonFounder, 794
filterQualGenotypes, 794
filterQualSNPs, 794
filterSNPs, 795

findAllHomozygousRuns, 795
findHomoRuns, 795
findHomoWindow, 795
findIBSRUNS, 795
findMissRuns, 796
findSegments, 796
flipStrand, 796
fullModelAssoc, 796
gene2segment, 827
geneList, 827
generateMDS, 796
getSimpleSNPValue, 797
glmAssoc, 797
glmCNVBurdenModel, 797
glmHaplotypeTest, 797
groupGenome, 797
groupSegmentsSpanning, 798
gset1, 828
gset2, 828
gvar, 828
haplo, 828
hardyWeinbergCheck, 798
homozygousSegmentPermutationTest, 798
in_anal, 828
Ind2SNP, 798
indivSegmentGroup, 828
indivSegmentSummary, 798
indivSegmentSummaryCalc, 799
initialiseGeneCountAssociation, 799
keep2SetsForGenome, 799
keepIndividuals, 799
keepSNPs, 799, 800
klist, 829
kmap, 829
kname, 829
linkRelateds, 800
locus, 829
lookup, 800
lookup2, 800
m1, 829
m2, 829
m_phenotype, 830
m_pihat, 830
makeFounders, 800
makeMissingParents, 801
makePhenotype, 801
maxr2, 830
mdist, 830
mergeBinaryData, 801
mergeData, 801
mergeList, 801
metaAnalysis, 801
missingGenotype, 802
mkBlks, 802
model, 830
multcomp, 802
n, 830
ngvar, 831
nk, 831
nl, 831
nl_all, 831
nlisname, 831
np, 831
npheno, 832
OUTFILE, 832
oblig_missing, 832
obligMissing, 802
openBinaryFile, 802
original, 832
output_bim bam_format, 803
output_fastphase_format, 803
output_structure_format, 803
outputArffFile, 803
outputDelimitedFile, 803
outputGenericVariantFile, 804
outputNumericExtract, 804
outputNumericFiltered, 804
outputPermedPhenotypes, 804
outputSetFile, 804
outputSifToGain, 805
pair1, 832
pair2, 832
pairid, 833
parseTrios, 805
performMisHapTests, 805
performProxyTests, 805
perm_segmentIndividualTest, 805
perm_sharingIBSTest, 806
perm_testGXE2, 806
perm_testHotel, 806
perm_testQTDT, 806
perm_testTDT_POO, 807
perm_testTDT, 806
permTestRareDistribution, 807
permutationIBSTest, 807
phenoLabel, 833
phenotype, 833
phenotype_name, 833
pihat, 833
pihat_G, 833
Plink, 780
plisname, 834
pos, 834
positionPermuteSegments, 807
pperm, 834
preCalcGenomeIBD, 807
preCalcMultiPoint, 808
preCalcPhenotypes, 808

preCalcRegression_PHENO, 808
preCalcRegression_PIHAT, 808
preCalcSinglePoint, 808
prettyPrintLengths, 808
prev_bt, 834
printLOG, 809
processCNVList, 809
processDosageFile, 809
processGVAR, 809
proxyLD, 834
proxyWrapper, 809
pruneLD, 809
pS, 834
pseudoCaseControl, 810
pv, 835
pvlBS0, 835
pvlBS2het, 835
readBinData, 810
readCNVList, 810
readClusterFile, 810
readConditioningList, 810
readCovListFile, 811
readCovariateFile, 810
readData, 811
readDataLongFormat, 811
readFamFile, 811
readGenericVariantData, 811
readHomozygSegmentFile, 811
readInformative, 812
readMapFile, 812
readMultiplePhenoFile, 812
readNumericFile, 812
readPhenoFile, 812
readSegmentFile, 813
readSegmentFileMinimal, 813
readSet, 813
readStdIn, 813
readTransposedData, 813
reconcileMerge, 814
related, 835
removeIndividuals, 814
runTestCNVwithGLM, 814
runTestCNVwithQT, 814
SNP2Ind, 817
SNP, 837
sample, 835
saved_IBDg, 835
scaffold, 836
scoreIndividuals, 814
segment, 836
segmentCount, 836
segmentCount2, 836
segmentCount2Baseline, 836
segmentIndividualTest, 815
segmentLength, 836
segmentPermutationTest, 815
setAssocSummary, 815
setFlagToCase, 815
setFlagToControl, 816
setFlags, 815
setMarkerRange, 816
setObligMissing, 816
setQtPhenoFromNumericIndex, 816
setReferenceAllele, 816
setTable, 816
setUpForCNVList, 817
setname, 837
sexCheck, 817
sharingIBSTest, 817
simulateSNPs, 817
simulateSNPs_QT, 817
skip_pair, 837
snpset, 837
summariseHomoRuns, 818
summaryIBDsegments, 818
summaryIBSsegments, 818
summaryIBD, 818
T00, 837
T01, 837
T02, 838
T10, 838
T11, 838
T12, 838
T20, 838
T21, 838
T22, 839
tagMode, 818
tcnt, 839
testAssoc, 819
testCNVwithGLM, 819
testCNVwithQT, 819
testGenomeIBDByCovariate, 819
testMiss, 820
testQAssoc, 820
testQAssocGXE2, 820
testRareDistribution, 820
testSibTDT, 820
testTDT_POO, 821
testTDT, 821
thinSNPs, 821
transcript, 839
updateAlleles, 821
updateFamFile, 822
updateMapFile, 822
v_phenotype, 839
v_pihat, 839
validateSegments, 822
warnings, 839

webcheck, 822
whap, 840
write_BITFILE, 822
write_clusters, 822
write_covariates, 823
write_snplist, 823
writeInformative, 823
writeSetFile, 823
ZOUTFILE, 840
zeroOnCluster, 823
Zlocus, 840
plink
 par, 699
plink.h
 boolmatrix_t, 1392
 boolvec_t, 1392
 floatvec_t, 1392
 fmatrix_t, 1392
 iAllele, 1393
 iIndividual, 1393
 iLocus, 1393
 iSNP, 1394
 interaction_t, 1393
 interactionVar_t, 1393
 intvec_t, 1393
 matrix_t, 1394
 ModelTermType, 1394
 rankedlist_t, 1394
 RegressionRankType, 1395
 table_t, 1394
 vector_t, 1394
PlinkInternalsDataset, 842
 ~PlinkInternalsDataset, 844
 LoadDatasetFromPlink, 844
 PlinkInternalsDataset, 844
 plinkInternalsPtr, 844
PlinkInternalsDatasetInstance, 845
 ~PlinkInternalsDatasetInstance, 847
 GetAttribute, 847
 GetNumeric, 848
 GetSimpleSNPValue, 848
 ID, 849
 individual, 849
 NumAttributes, 848
 NumNumerics, 849
 PlinkInternalsDatasetInstance, 847
 plinkInternalsPtr, 850
 Print, 849
plinkInternalsPtr
 PlinkInternalsDataset, 844
 PlinkInternalsDatasetInstance, 850
plist
 Individual, 464
 par, 699
plist_fid1
 par, 699
plist_fid2
 par, 699
plist_iid1
 par, 699
plist_iid2
 par, 699
plist_number
 par, 700
plist_selection
 par, 700
plist_selection_name
 par, 700
plist_selection_number
 par, 700
plist_selection_string
 par, 700
plistMissing
 Individual, 464
plistname
 Plink, 834
pm
 Individual, 464
polint
 stats.cpp, 1490
 stats.h, 1509
Pool, 851
 chr, 852
 group, 852
 index, 852
 match, 852
 matchcount, 852
 max, 853
 min, 853
 ng, 853
 segs, 853
 union_max, 853
 union_min, 853
pool_size_min
 par, 700
port
 Rconnection, 876
pos
 Locus, 510
 Plink, 834
 Rexp, 944
position
 CHIP_SEQ_INFO, 81
 SNP_INFO, 1010
position_window
 par, 701
positionPermuteSegments
 Plink, 807

possiblyRemove
 EvaporativeCoolingPrivacy, 296

PowerMethodSolver
 CentralityRanker, 70

PP
 annot.cpp, 1094
 blox.cpp, 1107
 cnv.cpp, 1144
 cnvqt.cpp, 1147
 Dataset.cpp, 1155
 dosage.cpp, 1217
 elf.cpp, 1220
 EvaporativeCooling, 286
 EvaporativeCoolingPrivacy, 297
 ExpressionDataSimulator, 305
 greport.cpp, 1267
 helper.cpp, 1297
 homozyg.cpp, 1322
 idhelp.cpp, 1324
 inbix.cpp, 1329
 Insilico.h, 1357
 metaanal.cpp, 1375
 mishap.cpp, 1377
 nonfounderphasing.cpp, 1383
 profile.cpp, 1401
 RandomForest, 866
 regain.cpp, 1414
 ReliefSeqController, 934
 ReliefF, 916
 sets.cpp, 1448
 setscreen.cpp, 1449
 stats.cpp, 1496
 trio.cpp, 1529
 zed.cpp, 1573

pp
 HaploPhase, 421
 HaploWindow, 433
 Individual, 464

pp_maxfid
 par, 701

pp_maxiid
 par, 701

pp_maxsnp
 par, 701

pperm
 Plink, 834

pperson
 Individual, 464

pprint
 multiple.cpp, 1380

preCalcGenomeIBD
 Plink, 807

preCalcMultiPoint
 Plink, 808

preCalcPhenotypes
 Plink, 808

preCalcRegression_PHENO
 Plink, 808

preCalcRegression_PIHAT
 Plink, 808

preCalcSinglePoint
 Plink, 808

PreComputeDistances
 ReliefF, 911

PreComputeNeighborGeneStats
 SNReliefF, 1014

preGeneDrop
 Perm, 764

predall
 ArgumentHandler, 39
 par, 701

Predict
 RandomForest, 864

predict
 ArgumentHandler, 39
 Forest, 322
 par, 701
 Tree, 1030

predict_all
 Forest, 329

predictInternal
 Forest, 323
 ForestClassification, 337
 ForestProbability, 343
 ForestRegression, 349
 ForestSurvival, 356

predictTreesInThread
 Forest, 323

predictedValueTau
 DatasetInstance, 214

prediction_mode
 Forest, 330

prediction_terminal_nodeIDs
 Tree, 1033

prediction_type
 Forest, 330

PredictionType
 globals.h, 1262

predictions
 Forest, 330

predictiontype
 ArgumentHandler, 39

PrepareConnectivityMatrix
 InteractionNetwork, 477

PrepareForAttribute
 ChiSquared, 87

prephap.cpp
 LOG, 1399

preserve_all_genotypes
 par, 702

preserve_mendel_errors
 par, 702

prettyPrintLengths
 Plink, 808

prev_bt
 Plink, 834

Print
 Dataset, 179

DatasetInstance, 209

PlinkInternalsDatasetInstance, 849

PrintAdjacencyMatrix
 InteractionNetwork, 477

PrintAllAttributeScores
 EvaporativeCooling, 278

PrintAllScoresTabular
 EvaporativeCooling, 278

PrintAlleleCounts
 BirdseedData, 55

PrintAttributeLevelsSeen
 Dataset, 179

PrintAttributeScores
 EvaporativeCooling, 278

ReliefSeqController, 929

ReliefF, 911

PrintBestKs
 ReliefSeqController, 929

 ReliefF, 911

PrintClassIndexInfo
 Dataset, 180

PrintConnectivityMatrix
 InteractionNetwork, 478

PrintDistancePairs
 DatasetInstance, 209

PrintHistogram
 Statistics.cpp, 1467

 Statistics.h, 1475

PrintInfo
 BirdseedData, 56

PrintInstanceAttributeStats
 SNReliefF, 1014

PrintInstancesMask
 ReliefF, 911

PrintInteractionAttributeScores
 EvaporativeCooling, 279

PrintKendallTaus
 EvaporativeCooling, 279

printLOG
 Plink, 809

PrintLevelCounts
 Dataset, 180

PrintMainEffectAttributeScores
 EvaporativeCooling, 279

PrintMaskStats
 Dataset, 180

PrintMissingValuesStats
 Dataset, 180

PrintModulesSummary
 InteractionNetwork, 478

PrintNeighborStats
 SNReliefF, 1015

PrintNumericsStats
 Dataset, 180

PrintSampleStats
 DgeData, 243

PrintScores
 AttributeRanker, 45

 Deseq, 236

 Edger, 259

 ReliefSeqController, 929

PrintScoresWithPValues
 ChiSquared, 88

PrintState
 DcVar, 221

 EpistasisEQtl, 267

 EvaporativeCoolingPrivacy, 292

PrintStats
 Dataset, 181

PrintStatsSimple
 Dataset, 181

PrintSummary
 InteractionNetwork, 478

PrintTables
 ChiSquared, 88

PrintVector
 Insilico.h, 1352

probOverlap
 cnv.cpp, 1143

probabilisticGenotype, 854

 AA, 855

 AB, 855

 BA, 855

 BB, 855

 calculated, 856

 genotype, 856

 genotyped, 856

 phased, 856

 phased_genotype, 856

 probabilisticGenotype, 855

probability
 ArgumentHandler, 39

processArguments
 ArgumentHandler, 36

processCNVList
 Plink, 809

processDosageFile
 Plink, 809

ProcessExclusionFile
 Dataset, 181
processGVAR
 Plink, 809
profile.cpp
 LOG, 1401
 PP, 1401
 scoreRanges, 1400
profile_sets
 par, 702
profileSNPs
 Set, 995
profileScore
 Set, 995
profileTestInitialise
 Set, 993
profileTestSNPInformation
 Set, 993
profileTestScore
 Set, 993
progress
 Forest, 330
ProtectedLog
 Insilico.cpp, 1336
 Insilico.h, 1352
proxy.cpp
 combinations_recursive, 1402
proxy_CC
 par, 703
proxy_TDT
 par, 710
proxy_all
 par, 702
proxy_all_list
 par, 702
proxy_all_list_file
 par, 702
proxy_assoc
 par, 703
proxy_assoc.snp
 par, 703
proxy_error
 par, 703
proxy_exclude
 par, 703
proxy_exclude_from_file
 par, 703
proxy_exclude_list
 par, 704
proxy_full_report
 par, 704
proxy_genotype
 par, 704
proxy_glm
 par, 704
proxy_impute
 par, 704
proxy_impute_genotypic_concordance
 par, 704
proxy_impute_preserve_genotyped
 par, 705
proxy_impute_replace
 par, 705
proxy_impute_threshold
 par, 705
proxy_include_reference
 par, 705
proxy_info_threshold
 par, 705
proxy_kb
 par, 705
proxy_kb_planA
 par, 706
proxy_kb_planB
 par, 706
proxy_leave_out
 par, 706
proxy_list
 par, 706
proxy_list_file
 par, 706
proxy_list_proxies
 par, 706
proxy_maf
 par, 707
proxy_maxhap
 par, 707
proxy_mhf
 par, 707
proxy_planB_threshold
 par, 707
proxy_r2
 par, 707
proxy_r2_filter
 par, 707
proxy_r2_filter_A_planA
 par, 708
proxy_r2_filter_A_planB
 par, 708
proxy_r2_filter_B_planA
 par, 708
proxy_r2_filter_B_planB
 par, 708
proxy_r2_filter_C_planA
 par, 709
proxy_r2_filter_C_planB
 par, 709
proxy_r2_filter_A

par, 708
proxy_r2_filter_B
 par, 708
proxy_r2_filter_C
 par, 709
proxy_record_usage
 par, 709
proxy_reference_only
 par, 709
proxy.snp_filter
 par, 709
proxy.snp_filter_planA
 par, 710
proxy.snp_filter_planB
 par, 710
proxy_window
 par, 710
proxy_window_planA
 par, 710
proxy_window_planB
 par, 710
proxyLD
 Plink, 834
ProxyResult, 857
 chisq, 858
 f, 858
 name, 859
 odds, 859
 operator<, 858
 ProxyResult, 858
 pvalue, 859
 r2, 859
proxyWrapper
 Plink, 809
prterr
 fisher.cpp, 1239
prune_id
 par, 711
prune_id_pairwise
 par, 711
prune_id_pairwise_maf
 par, 711
prune_id_r2
 par, 711
prune_id_step
 par, 711
prune_id_vif
 par, 711
prune_id_win
 par, 712
prune_r2_fixed
 par, 712
prune_r2_fixed_list
 par, 712
prune_r2_prefer
 par, 712
prune_r2_prefer_list
 par, 712
pruneGenogroups
 HaploWindow, 428
pruneLD
 Plink, 809
pruneMC
 Set, 993
prunePhase
 HaploPhase, 407
 HaploWindow, 429
pruneSets
 Set, 994
pruneY
 LinearModel, 502
 LogisticModel, 515
 Model, 529
pS
 Plink, 834
pseudoCaseControl
 Plink, 810
psi
 cdflib.h, 1128
 dcdflib.cpp, 1189
pstdnorm
 utility.cpp, 1543
 utility.h, 1562
pT
 stats.cpp, 1490
 stats.h, 1509
pt
 FamilyTransmissions, 311
ptod
 Rsrv.h, 1439
ptoI
 Rsrv.h, 1440
pu
 FamilyTransmissions, 311
pureInteractionEffect
 Regain, 893
pureInteractions
 Regain, 900
pv
 Plink, 835
pv_maxE_set
 Set, 995
pv_maxG_set
 Set, 995
pv_set
 Set, 995
pvIBSO
 Plink, 835

pvIBS2het
 Plink, 835
 pval_cutoff
 clump_LD, 96
 pvalComparator
 regain.cpp, 1413
 pvalComparatorAscending
 DcVar.cpp, 1196
 pvals
 clump_LD, 96
 RegressionRankResults, 903
 pvalue
 HaploPhase, 421
 Perm, 764
 ProxyResult, 859
 pythag
 stats.cpp, 1490
 stats.h, 1510

Q
 InteractionNetwork, 485

Q_EPS
 EvaporativeCoolingPrivacy, 297

Q_THRESHOLD
 globals.h, 1266

QFAM_adaptive
 par, 712

QFAM_between
 par, 713

QFAM_total
 par, 713

QFAM_within1
 par, 713

QFAM_within2
 par, 713

QTDT_test
 par, 714

qfam.cpp
 scoreBandW, 1403
 scoreBetween, 1403
 setCovariatesForSNP, 1403

qmatch
 par, 713

qmatch_filename
 par, 713

qmatch_threshold_filename
 par, 714

qq_plot
 par, 714

qromb
 stats.cpp, 1490
 stats.h, 1510

qsmall
 dcdflib.cpp, 1167, 1168

qt
 par, 714

qt_means
 par, 714

qt_with_covariates
 par, 714

qtired
 dcdflib.cpp, 1168

quantile
 stats.cpp, 1491
 stats.h, 1510

queryGenotype
 HaploPhase, 407

queryThisGenotype
 HaploPhase, 408

qxmon
 dcdflib.cpp, 1168

R

Perm, 769

r
 CentralityRanker, 72

r2
 ProxyResult, 859

r2_cutoff
 clump_LD, 96

r2pi
 dcdflib.cpp, 1168

R_EXT_BOOLEAN_H_
 fisher.h, 1241

R_EXT_CONSTANTS_H_
 fisher.h, 1241

R_EXT_MEMORY_H_
 fisher.h, 1242

R_alloc
 fisher.h, 1243

R_gc
 fisher.h, 1243

R_port
 par, 715

R_script
 par, 715

RANGER_VERSION
 version.h, 1567

RCount, 877
 acnt, 880
 acnt2, 880
 addSNP, 878
 aval, 880
 awin, 880
 displayWindow, 879
 domModel, 881
 gval, 881
 gwin, 881

ignorePosition, 879
loadCovariate, 879
loadPCACovariate, 879
mainStats, 879
nalleles, 881
npc, 881
nsnps, 881
P, 882
pcMode, 882
RCount, 878
rangeLookup, 882
removeSNP, 879
rval, 882
rwin, 882
setWindow, 880
ucnt, 882
ucnt2, 883
REGAIN_MATRIX
 Regain, 900
RESP_ERR
 Rsrv.h, 1440
RESP_OK
 Rsrv.h, 1440
RNMX
 CRandom, 110
RReliefF, 961
 ~RReliefF, 964
 ComputeAttributeScores, 964
 RReliefF, 964
RSRV_VER
 Rsrv.h, 1440
RSS
 LinearModel, 504
rStatePtr_
 GSLRandomBase, 363
radius
 DcVar, 227
 EpistasisEQtl, 271
rand
 CRandom, 108
randUniformProbs
 EvaporativeCoolingPrivacy, 297
randUniformValue
 EvaporativeCoolingPrivacy, 297
random_number_generator
 Forest, 330
 Tree, 1034
random_seed
 par, 715
RandomForest, 860
 ~RandomForest, 863
 ComputeScores, 864
 CreateDefaultForestForPheno, 864
 datasetFile, 865
 forest, 865
 GetClassificationError, 864
 InitializeData, 864
 minNodeSize, 866
 PP, 866
 Predict, 864
 RandomForest, 862, 863
 SaveForest, 865
 WriteScores, 865
randomForestPredictError
 EvaporativeCoolingPrivacy, 297
RandomJungleRunMode
 Insilico.h, 1346
RandomJungleTreeType
 Insilico.h, 1347
randomlySelect
 ReliefF, 917
Range, 866
 chr, 869
 group, 869
 groupNames, 869
 name, 869
 operator<, 868
 operator==, 869
 Range, 868
 start, 870
 stop, 870
range_char
 NList, 546
range_delimiter
 par, 715
range_string
 NList, 546
rangeIntersect
 helper.cpp, 1292
 helper.h, 1313
rangeLookup
 RCount, 882
rank
 Perm, 765
 utility.h, 1562
rankByRegression
 stats.cpp, 1491
 stats.h, 1510
rankedlist_t
 plink.h, 1394
ranker_centrality_gamma
 par, 715
ranker_input_file
 par, 715
ranker_method
 par, 716
ranker_save_data_file
 par, 716

ranker_top_n
 par, 716
 rankerPermMethod
 par, 716
 rankerPermNum
 par, 716
 rankerPermThreshold
 par, 716
 ranks
 CentralityRanker, 73
 rare_test
 par, 717
 rare_test_print_details
 par, 717
 rare_test_print_details.snp
 par, 717
 rare_test_score_range
 par, 717
 rare_test_score_range_file
 par, 717
 rare_test_score_range_threshold
 par, 717
 rare_test_score_results_file
 par, 718
 rare_test_summary_controls
 par, 718
 rare_test_weight1
 par, 718
 rarer_dist_threshold
 par, 718
 rarer_interval
 par, 718
 rarer_maf_threshold
 par, 718
 ratio
 HaplоТhase, 421
 Rboolean
 fisher.h, 1242
 rcomp
 cdflib.h, 1128
 dcdflib.cpp, 1189
 Rconnection, 870
 ~Rconnection, 872
 assign, 872
 auth, 876
 closeFile, 873
 connect, 873
 createFile, 873
 disconnect, 873
 eval, 873
 family, 876
 host, 876
 login, 874
 openFile, 874
 port, 876
 Rconnection, 872
 readFile, 874
 removeFile, 874
 request, 874, 875
 s, 876
 salt, 877
 shutdown, 875
 voidEval, 875
 writeFile, 875
 Rconnection.cpp
 AF_LOCAL, 1407
 myID, 1408
 new_parsed_Rexp, 1407
 new_parsed_Rexp_from_Msg, 1408
 Rconnection.h
 A_crypt, 1410
 A_plain, 1410
 A_required, 1410
 CERR_auth_unsupported, 1410
 CERR_connect_failed, 1410
 CERR_handshake_failed, 1411
 CERR_invalid_id, 1411
 CERR_io_error, 1411
 CERR malformed_packet, 1411
 CERR_not_connected, 1411
 CERR_not_supported, 1411
 CERR_out_of_mem, 1412
 CERR_peer_closed, 1412
 CERR_protocol_not_supp, 1412
 CERR_send_error, 1412
 Rsize_t, 1412
 rcount
 Rexp, 944
 rdbuf
 gzifstream, 386
 gzofstream, 392
 Rdouble, 883
 doubleArray, 887
 doubleAt, 887
 fix_content, 887
 length, 887
 os_print, 887
 Rdouble, 886
 read
 Rmessage, 959
 read_assoc_file
 clump_LD, 94
 read_bitfile
 par, 719
 read_geno_qual
 par, 719
 read_ped
 par, 719

read_segment_file
 par, 719
read_segment_filename
 par, 719
read_set
 par, 719
read.snp.qual
 par, 720
readBinData
 Plink, 810
ReadBrainCorr1DFile
 InteractionNetwork, 478
readCNVList
 Plink, 810
readChar
 ZInput, 1083
ReadCheckpoint
 DcVar, 221
ReadChipSeqFile
 DcVar, 221
readClusterFile
 Plink, 810
readConditioningList
 Plink, 810
readCovListFile
 Plink, 811
readCovariateFile
 Plink, 810
ReadCsvFile
 InteractionNetwork, 478
readData
 Plink, 811
readDataLongFormat
 Plink, 811
ReadDeseqScores
 Deseq, 236
ReadEdgerScores
 Edger, 259
readFamFile
 Plink, 811
readFile
 Rconnection, 874
ReadGainFile
 CentralityRanker, 70
 InteractionNetwork, 478
ReadGeneExpressionFile
 DcVar, 222
readGenericVariantData
 Plink, 811
ReadGenotypesFile
 DcVar, 222
readHomozygSegmentFile
 Plink, 811
readInformative
 Plink, 812
readLine
 ZInput, 1083
readMapFile
 Plink, 812
readMultiplePhenoFile
 Plink, 812
readNumericFile
 Plink, 812
readPhenoFile
 Plink, 812
readRange
 helper.cpp, 1292
 helper.h, 1314
readRegainFromFile
 Regain, 894
readSegmentFile
 Plink, 813
readSegmentFileMinimal
 Plink, 813
readSet
 Plink, 813
ReadSifFile
 InteractionNetwork, 479
ReadSnpLocationsFile
 DcVar, 222
readStdIn
 Plink, 813
readString
 helper.cpp, 1292
 helper.h, 1314
readTagFile
 HaploPhase, 408
ReadTranscriptCoordinates
 DcVar, 222
 EpistasisEQtl, 267
ReadTranscriptFactorCoordinates
 DcVar, 222
 EpistasisEQtl, 267
readTransposedData
 Plink, 813
readVector1D
 utility.h, 1563
readVector2D
 utility.h, 1563
realnum
 stats.cpp, 1491
 stats.h, 1510
recode
 par, 720
 recode_12
 par, 720
 recode_1234
 par, 720

recode_ACGT
 par, 720
 recode_AD_Aonly
 par, 721
 recode_AD_fixed
 par, 721
 recode_AD
 par, 720
 recode_HV
 par, 722
 recode_allele_coding
 par, 721
 recode_allele_coding_file
 par, 721
 recode_bim bam
 par, 721
 recode_delimit
 par, 721
 recode_fastphase
 par, 722
 recode_indelimit
 par, 722
 recode_long
 par, 722
 recode_long_ref
 par, 722
 recode_mutlist
 par, 722
 recode_structure
 par, 723
 recode_transpose
 par, 723
 recode_whap
 par, 723
 reconcileMerge
 Plink, 814
 recount
 HWindow, 436
 RecursiveIndirectPathsModularity
 InteractionNetwork, 479
 ref_file
 par, 723
 ref_file_name
 par, 723
 refAllele
 SNP_INFO, 1010
 reference
 MultiLocusGenotype, 539
 reference_only
 HaplоФase, 421
 Regain, 888
 ~Regain, 891
 addCovariates, 892
 attributeNames, 896
 BETAS, 896
 doFdrPrune, 896
 failureValue, 897
 failures, 896
 fdrPrune, 892
 gainIntPvals, 897
 getRawMatrix, 892
 INT_MATRIX, 897
 INT_SIF, 897
 infCount, 897
 integratedAttributes, 897
 interactionEffect, 892
 logMatrixStats, 892
 logOutputOptions, 893
 MEBETAS, 898
 mainEffect, 893
 mainEffectComparator, 893
 maxInteraction, 898
 maxMainEffect, 898
 minInteraction, 898
 minMainEffect, 898
 NUM_MATRIX, 899
 NUM_SIF, 899
 nanCount, 898
 numAttributes, 899
 outputFormat, 899
 outputThreshold, 899
 outputTransform, 899
 performPureInteraction, 893
 pureInteractionEffect, 893
 pureInteractions, 900
 REGAIN_MATRIX, 900
 readRegainFromFile, 894
 Regain, 891
 regainMatrix, 900
 regainPMatrix, 900
 run, 894
 SIF, 900
 SNP_MATRIX, 901
 SNP_SIF, 901
 setFailureValue, 894
 setOutputFormat, 894
 setOutputThreshold, 894
 setOutputTransform, 895
 sifThresh, 900
 updateStats, 895
 useOutputThreshold, 901
 warnings, 901
 writeComponents, 901
 writeCompressedFormat, 901
 writeRcomm, 895
 writeRegain, 895
 writeRegainToFile, 895
 writeRegainToSifFile, 896

regain.cpp
 PP, 1414
 pvalComparator, 1413
regain.h
 RegainOutputFormat, 1415
 RegainOutputTransform, 1415
regainComponents
 par, 723
regainCompress
 par, 724
regainFailValue
 par, 724
regainFdr
 par, 724
regainFdrPrune
 par, 724
regainFile
 par, 724
regainLargeCoefPvalue
 par, 724
regainLargeCoefTvalue
 par, 725
regainMatrix
 Regain, 900
regainMatrixFormat
 par, 725
regainMatrixThreshold
 par, 725
regainMatrixThresholdValue
 par, 725
regainMatrixToSif
 par, 725
regainMatrixTransform
 par, 725
regainMaxBetaValue
 par, 726
RegainOutputFormat
 regain.h, 1415
RegainOutputTransform
 regain.h, 1415
regainPMatrix
 Regain, 900
regainPureInteractions
 par, 726
regainPvalueThreshold
 par, 726
regainSifFilter
 par, 726
regainSifThreshold
 par, 726
regainUseBetaValues
 par, 726
RegressionRankResults, 902
 coefs, 903
 pvals, 903
 RegressionRankResults, 903
 stats, 903
 vars, 903
RegressionRankType
 plink.h, 1395
relType
 helper.cpp, 1293
 helper.h, 1314
related
 Plink, 835
ReliefF.cpp
 AttributeIndex, 1417
 AttributeIndexIt, 1417
 attributeSort, 1418
 scoreSort, 1418
 ScoresMap, 1417
 ScoresMapIt, 1418
 T, 1418
ReliefFSeq, 919
 ~ReliefFSeq, 922
 ComputeAttributeScores, 923
 GetScores, 923
 mode, 924
 MuDeltaAlphas, 923
 ReliefFSeq, 922
 s0, 924
 SigmaDeltaAlphas, 924
 snrMode, 924
 tstatMode, 925
reliefMode
 par, 727
reliefSeqAlgorithmMode
 par, 727
reliefSeqAlgorithmS0
 par, 727
ReliefSeqController, 925
 ~ReliefSeqController, 928
 algorithmMode, 931
 analysisType, 931
 bestKs, 931
 ComputeScores, 928
 ComputeScoresKopt, 928
 dataset, 932
 doRemovePercent, 932
 GetAlgorithmMode, 928
 GetKmax, 928
 GetScores, 929
 koptBegin, 932
 koptEnd, 932
 koptStep, 932
 numTargetAttributes, 933
 numThreads, 933
 numToRemoveNextIteration, 933

numToRemovePerIteration, 933
 outFilesPrefix, 933
 PP, 934
 PrintAttributeScores, 929
 PrintBestKs, 929
 PrintScores, 929
 ReliefSeqController, 927
 reliefseqAlgorithm, 934
 removePercentage, 934
 RemoveWorstAttributes, 930
 removedAttributes, 934
 RunReliefF, 930
 scores, 934
 SetKoptParameters, 930
 WriteAttributeScores, 930
 WriteBestKs, 931
 reliefSeqSnrMode
 par, 728
 reliefSeqTstatMode
 par, 728
 ReliefF, 904
 ~ReliefF, 908
 analysisType, 914
 bestKs, 914
 ComputeAttributeScores, 909
 ComputeAttributeScoresIteratively, 909
 ComputeAttributeScoresKopt, 910
 ComputeGRM, 910
 ComputeScores, 910
 ComputeWeightByDistanceFactors, 910
 distanceMatrix, 914
 doRemovePercent, 915
 GetKmax, 910
 koptBegin, 915
 koptEnd, 915
 koptStep, 915
 m, 915
 numDiffFuncPtr, 916
 numDiffMetricName, 916
 numTarget, 916
 PP, 916
 PreComputeDistances, 911
 PrintAttributeScores, 911
 PrintBestKs, 911
 PrintInstancesMask, 911
 randomlySelect, 917
 ReliefF, 908, 909
 removePerIteration, 917
 removePercentage, 917
 RemoveWorstAttributes, 912
 removedAttributes, 917
 ResetForNextIteration, 912
 SetKoptParameters, 912
 snpDiffFuncPtr, 917
 snpDiffMetricName, 918
 W, 918
 weightByDistanceMethod, 918
 weightByDistanceSigma, 918
 WriteAttributeScores, 912
 WriteBestKs, 914
 reliefffilterNumToRemove
 par, 727
 reliefffilterPercentToRemove
 par, 727
 reliefffilterNumTarget
 par, 727
 reliefseqAlgorithm
 ReliefSeqController, 934
 remove_before_keep
 par, 728
 remove_file
 par, 728
 remove_indiv
 par, 728
 remove_indiv_list
 par, 728
 remove_unaffected_pairs
 par, 729
 removeAttrs
 EvaporativeCoolingPrivacy, 297
 removeFile
 Rconnection, 874
 RemoveImportanceScore
 EvaporativeCoolingPrivacy, 292
 removeIndividuals
 Plink, 814
 removeMissingPhenotypes
 helper.cpp, 1293
 helper.h, 1314
 removePerIteration
 ReliefF, 917
 removePercentage
 ReliefSeqController, 934
 ReliefF, 917
 removeSNP
 RCount, 879
 RemoveWorstAttributes
 EvaporativeCooling, 280
 ReliefSeqController, 930
 ReliefF, 912
 removedAttributes
 ReliefSeqController, 934
 ReliefF, 917
 reorder
 Perm, 770
 replace
 ArgumentHandler, 40
 replicates

par, 729
Perm, 770
report
 Perm, 765
report_missing
 par, 729
reportHaplotypeFrequencies
 HaploPhase, 408
reportNumericSummaryStats
 stats.cpp, 1491
 stats.h, 1511
reportPhase
 HaploPhase, 408
 HaploWindow, 429
reportPhaseWideFormat
 HaploPhase, 408
reps_done
 Perm, 765
request
 Rconnection, 874, 875
res
 phdr, 772
reserveMemory
 Data, 118
 DataChar, 124
 DataDouble, 129
 DataFloat, 134
reset
 HaploPhase, 409
 LinearModel, 503
 LogisticModel, 515
 WMLocus, 1077
ResetForNextIteration
 AttributeRanker, 46
 ChiSquared, 88
 ReliefF, 912
ResetNearestNeighbors
 Dataset, 182
 DatasetInstance, 209
resolveWithKids
 HaploPhase, 409
resolved
 IDGroup, 447
response_classIDs
 ForestClassification, 338
 ForestProbability, 344
 TreeClassification, 1043
 TreeProbability, 1051
response_timepointIDs
 ForestSurvival, 357
 TreeSurvival, 1070
result
 HaploPhase, 421
ResultTrio, 935
annot, 936
f, 936
operator<, 936
p, 937
s, 937
ResultsLists
 EvaporativeCoolingPrivacy.h, 1228
resultsMatrixBetas
 EpistasisEQtl, 271
resultsMatrixPvals
 EpistasisEQtl, 271
return_beta
 par, 729
returnFullRangeList
 clumpld.cpp, 1137
returnHaplotypeSet
 HaploPhase, 409
returnHaplotypes
 HaploPhase, 409
Rexp, 937
 ~Rexp, 940
 attr, 943
 attribs, 943
 attribute, 940
 attributeNames, 941
 attrnames, 943
 data, 943
 len, 943
 length, 941
 master, 943
 msg, 944
 next, 944
 operator<<, 942
 os_print, 941
 parse, 941
 pos, 944
 rcount, 944
 Rexp, 939, 940
 set_master, 941
 storageSize, 942
 store, 942
 type, 944
rexp
 cdflib.h, 1128
 dcdflib.cpp, 1189
rfreplace
 par, 729
right_passed
 HaploWindow, 433
rightHomozyg
 HWindow, 438
rightMissing
 HWindow, 438
rightStub

HaploWindow, 433
 rightStubFrequency
 HaploWindow, 429
 rightWindowEdge
 helper.cpp, 1293
 helper.h, 1314
 Rinteger, 945
 fix_content, 949
 intArray, 949
 intAt, 949
 length, 949
 os_print, 949
 Rinteger, 948
 ripM
 InteractionNetwork, 479
 ripmMatrix
 InteractionNetwork, 485
 RipmResult
 InteractionNetwork.h, 1360
 ripmResult
 InteractionNetwork, 485
 Rlist, 950
 ~Rlist, 954
 entryByTagName, 954
 fix_content, 954
 head, 955
 os_print, 954
 Rlist, 953
 tag, 955
 tail, 955
 rlog
 cdflib.h, 1129
 dcdflib.cpp, 1189
 rlog1
 cdflib.h, 1129
 dcdflib.cpp, 1190
 Rmessage, 956
 ~Rmessage, 958
 command, 958
 complete, 960
 data, 960
 head, 960
 is_complete, 959
 len, 960
 length, 959
 par, 960
 pars, 960
 parse, 959
 read, 959
 Rmessage, 957, 958
 send, 959
 rnorm
 stats.cpp, 1491
 stats.h, 1511
 root_command
 CArgs, 66
 roundToNextMultiple
 utility.cpp, 1544
 utility.h, 1564
 Rsize_t
 Rconnection.h, 1412
 rsnum
 CHIP_SEQ_INFO, 81
 SNP_INFO, 1010
 rsq
 ChapModel, 79
 HaploPhase, 409
 PairwiseLinkage, 560
 rsq_internal
 HaploPhase, 410
 Rsrv.h
 BOOL_FALSE, 1427
 BOOL_NA, 1427
 BOOL_TRUE, 1427
 CMD_RESP, 1430
 CMD_SPECIAL_MASK, 1431
 CMD_STAT, 1432
 CMD_assignSEXP, 1428
 CMD_attachSession, 1428
 CMD_closeFile, 1428
 CMD_createFile, 1428
 CMD_detachSession, 1429
 CMD_detachedVoidEval, 1428
 CMD_eval, 1429
 CMD_login, 1429
 CMD_openFile, 1429
 CMD_readFile, 1429
 CMD_removeFile, 1430
 CMD_serAssign, 1430
 CMD_serEEval, 1430
 CMD_serEval, 1430
 CMD_setBufferSize, 1431
 CMD_setEncoding, 1431
 CMD_setSEXP, 1431
 CMD_shutdown, 1431
 CMD_voidEval, 1432
 CMD_writeFile, 1432
 DT_ARRAY, 1432
 DT_BYTESTREAM, 1433
 DT_CHAR, 1433
 DT_DOUBLE, 1433
 DT_INT, 1433
 DT_LARGE, 1433
 DT_SEXP, 1434
 DT_STRING, 1434
 default_Rsrv_port, 1432
 dtop, 1434
 ERR_IOerror, 1436

ERR_Rerror, 1437
ERR_accessDenied, 1434
ERR_auth_failed, 1434
ERR_conn_broken, 1435
ERR_data_overflow, 1435
ERR_detach_failed, 1435
ERR_inv_cmd, 1435
ERR_inv_par, 1436
ERR_notOpen, 1436
ERR_object_too_big, 1436
ERR_out_of_mem, 1436
ERR_session_busy, 1437
ERR_unknownCmd, 1437
ERR_unsupportedCmd, 1437
fixdcpy, 1438
GET_DT, 1438
GET_XT, 1438
HAS_ATTR, 1438
IS_LARGE, 1438
isByteSexOk, 1446
itop, 1439
PAR_LENGTH, 1439
PAR_LEN, 1439
PAR_TYPE, 1439
ptod, 1439
pto1, 1440
RESP_ERR, 1440
RESP_OK, 1440
RSRV_VER, 1440
SET_PAR, 1441
SET_STAT, 1441
XT_ARRAY_BOOL_UA, 1441
XT_ARRAY_BOOL, 1441
XT_ARRAY_CPLX, 1441
XT_ARRAY_DOUBLE, 1442
XT_ARRAY_INT, 1442
XT_ARRAY_STR, 1442
XT_BOOL, 1442
XT_CLOS, 1442
XT_DOUBLE, 1443
XT_HAS_ATTR, 1443
XT_INT, 1443
XT_LANG_NOTAG, 1443
XT_LANG_TAG, 1444
XT_LANG, 1443
XT_LARGE, 1444
XT_LIST_NOTAG, 1444
XT_LIST_TAG, 1444
XT_LIST, 1444
XT_NULL, 1445
XT_RAW, 1445
XT_S4, 1445
XT_STR, 1445
XT_SYMBOLNAME, 1445
XT_SYM, 1445
XT_UNKNOWN, 1446
XT_VECTOR_EXP, 1446
XT_VECTOR_STR, 1446
XT_VECTOR, 1446
Rstring, 965
os_print, 969
Rstring, 968
string, 969
Rstrings, 969
cont, 974
count, 973
decode, 973
indexOfString, 973
nel, 974
os_print, 973
Rstrings, 972
string, 973
stringAt, 973
strings, 974
Rsymbol, 975
fix_content, 977
name, 978
os_print, 978
Rsymbol, 977
symbolName, 978
Run
DcVar, 222
EpistasisEQtl, 268
run
Forest, 323
Regain, 894
run_R_chisq
par, 730
run_R_nsnp
par, 730
run_R_script
par, 730
run_R_write_script
par, 730
run_R_z
par, 730
run_chr
par, 729
run_end
par, 730
run_start
par, 731
RunEqtl
EpistasisEQtl, 268
RunlqlCisTrans
EpistasisEQtl, 268
RunlqlFull
EpistasisEQtl, 268

RunOMRFChipSeq
 DcVar, 223

RunOMRF
 DcVar, 223

RunPlink
 DcVar, 223

RunRandomForest
 EvaporativeCooling, 280

RunReliefF
 EvaporativeCooling, 280
 ReliefSeqController, 930

RunSnpDiagnosticTests
 Dataset, 182

runTestCNVwithGLM
 Plink, 814

runTestCNVwithQT
 Plink, 814

rval
 RCount, 882

Rvector, 979
 ~Rvector, 982
 byName, 982
 capacity, 983
 cont, 983
 count, 984
 fix_content, 982
 indexOf, 982
 indexOfString, 982
 os_print, 983
 Rvector, 981
 stringAt, 983
 strings, 983
 strs, 984

rwin
 RCount, 882

rwrk
 fisher.cpp, 1233

S
 HaploPhase, 421
 HaploWindow, 433
 Model, 536

s
 Perm, 770
 Rconnection, 876
 ResultTrio, 937
 sortedSegments, 1018

s0
 ReliefFSeq, 924

s1
 LDPair, 492

s2
 LDPair, 492

S_alloc
 fisher.h, 1244
 s_max
 Set, 995
 s_min
 Set, 996
 S_realloc
 fisher.h, 1244

SAIN
 sisocks.h, 1456

SET_PAR
 Rsrv.h, 1441

SET_STAT
 Rsrv.h, 1441

SIGN
 helper.h, 1315

SINGLE_BASE
 fisher.h, 1242

SINGLE_EPS
 fisher.h, 1242

SINGLE_XMAX
 fisher.h, 1242

SINGLE_XMIN
 fisher.h, 1242

SINT_MAX
 fisher.cpp, 1233

SIF
 Regain, 900

SIInfo, 1007
 d, 1008
 SIInfo, 1007
 se, 1008

SMALLBUFSIZE
 fstream.cpp, 1575

SNP2Ind
 Plink, 817

SNP_EXTRACT_FIELD_IDX
 DcVar.h, 1199

SNP_INFO_LIST_IT
 DcVar.h, 1198

SNP_INFO_LIST
 DcVar.h, 1198

SNP_INFO, 1008
 chrom, 1009
 position, 1010
 refAllele, 1010
 rsnum, 1010

SNP_INPUT_TYPE
 DcVar.h, 1199

SNP_MATRIX
 Regain, 901

SNP_SIF
 Regain, 901

SNP_major
 par, 746

SNPHWE
 Dataset, 183
 helper.cpp, 1295
 helper.h, 1316
SNReliefF.h
 InstanceAttributeStats, 1459
 InstanceAttributeStatsIt, 1459
 InstanceHitMissStats, 1459
 NeighborStats, 1459
 NeighborStatsClt, 1459
 NeighborStatsIt, 1459
SNReliefF, 1011
 ~SNReliefF, 1013
 ComputeAttributeScores, 1013
 ComputeInstanceStats, 1014
 neighborStats, 1015
 PreComputeNeighborGeneStats, 1014
 PrintInstanceAttributeStats, 1014
 PrintNeighborStats, 1015
 SNReliefF, 1013
SNP
 Plink, 837
SQR
 helper.h, 1317
 stats.cpp, 1492
STATUS_INTERVAL
 globals.h, 1266
SWAP
 helper.h, 1318
SA
 sisocks.h, 1455
salt
 Rconnection, 877
sample
 Plink, 835
sample_fraction
 Forest, 330
 Tree, 1034
sample_with_replacement
 Forest, 331
 Tree, 1034
sampleIDs
 Tree, 1034
sampleLogLikelihood
 HaploWindow, 433
sampleNames
 DgeData, 245
sampleZeroes
 DgeData, 245
SaveForest
 RandomForest, 865
SaveModules
 InteractionNetwork, 479
saveToFile
 Forest, 323
 saveToFileInternal
 Forest, 323
 ForestClassification, 337
 ForestProbability, 343
 ForestRegression, 350
 ForestSurvival, 356
 saveVector1D
 utility.h, 1564, 1565
 saveVector2D
 utility.h, 1565
 saved_IBDg
 Plink, 835
 savemem
 ArgumentHandler, 40
 par, 731
 sb
 gzifstream, 386
 gzofstream, 392
 sccaffold
 Plink, 836
 scaledProbabilities
 EvaporativeCoolingPrivacy, 298
 score
 Segment, 987
 score_impute_expected
 par, 731
 score_qfile
 par, 731
 score_qrange_file
 par, 731
 score_risk
 par, 731
 score_risk_file
 par, 732
 score_risk_on_qrange
 par, 732
 score_risk_ranges
 par, 732
 score_risk_ranges_file
 par, 732
 score_risk_ranges_min
 par, 732
 score_test
 par, 732
 scoreBandW
 qfam.cpp, 1403
 scoreBetween
 qfam.cpp, 1403
 scoreIndividuals
 Plink, 814
 scoreNames
 AttributeRanker, 48
 scoreRanges

profile.cpp, 1400
scoreSort
 ReliefF.cpp, 1418
scoreTransmissions
 HaplоТасе, 410
ScoreVarPair
 Insilico.h, 1342
scores
 AttributeRanker, 48
 Deseq, 237
 Edger, 260
 ReliefSeqController, 934
ScoresMap
 ReliefF.cpp, 1417
ScoresMapIt
 ReliefF.cpp, 1418
scoresPvalues
 ChiSquared, 90
scoresSortAsc
 Insilico.cpp, 1336
 Insilico.h, 1352
scoresSortAscByName
 Insilico.cpp, 1336
 Insilico.h, 1352
scoresSortDesc
 Insilico.cpp, 1336
 Insilico.h, 1353
SD
 par, 733
se
 ChapModel, 79
 LinearModel, 504
 SInfo, 1008
searchAndReplace
 helper.cpp, 1293
 helper.h, 1315
second_pval_cutoff
 clump_LD, 96
seeChromosome
 helper.cpp, 1293
 helper.h, 1315
seed
 ArgumentHandler, 40
 Forest, 331
seg_test_region
 par, 733
seg_test_window
 par, 733
seg_test_window_bp
 par, 733
segCount
 CNVIndivReport, 104
Segment, 984
 baseline, 986
 count, 987
 finish, 987
 freq, 987
 operator<, 986
 operator==, 986
 p1, 987
 p2, 987
 score, 987
 Segment, 986
 sites, 988
 start, 988
 type, 988
 weightedBaseline, 988
 weightedCount, 988
segment
 Plink, 836
segment_current_focal_snp
 par, 733
segment_haplotrack
 par, 733
segment_haplotrack_fid1
 par, 734
segment_haplotrack_fid2
 par, 734
segment_haplotrack_iid1
 par, 734
segment_haplotrack_iid2
 par, 734
segment_inter_snp_distance
 par, 734
segment_length
 par, 734
segment_m1
 par, 735
segment_m2
 par, 735
segment_minimal
 par, 735
segment_output
 par, 735
segment_output_started
 par, 735
segment_overlap
 par, 735
segment_silently_return_groups
 par, 736
segment.snp
 par, 736
segment.snp1
 par, 736
segment.snp2
 par, 736
segment_test_1sided
 par, 736

segment_test_fisher
 par, 736
segment_test_force_1sided
 par, 737
segment_test_ignore_discordant
 par, 737
segment_test_individual
 par, 737
segment_test_specific_segs
 par, 737
segment_threshold_finish
 par, 737
segment_threshold_start
 par, 737
segment_validate
 par, 738
segment_verbose
 par, 738
segmentCount
 Plink, 836
segmentCount2
 Plink, 836
segmentCount2Baseline
 Plink, 836
segmentCountCaseControls
 cnv.cpp, 1143
 cnv.h, 1145
segmentIndividualTest
 Plink, 815
segmentLength
 Plink, 836
segmentPermutationTest
 Plink, 815
SegmentSizeCmp, 989
 operator(), 989
segss
 Pool, 853
segslBDMatch
 homozyg.cpp, 1321
segslBDMatchCON
 homozyg.cpp, 1321
segsmatch
 homozyg.cpp, 1321
segsmatchCON
 homozyg.cpp, 1322
segsoverlap
 homozyg.cpp, 1322
selcon_inds
 cluster.cpp, 1140
selectedSNPs
 Set, 996
SelfEntropy
 Statistics.cpp, 1468
 Statistics.h, 1475
send
 Rmessage, 959
Set, 990
 cumulativeSetSum_WITHLABELS, 991
 cumulativeSetSum_WITHOUTLABELS, 991
 cur, 994
 dropNotSet, 992
 empiricalSetPValues, 992
 fitLDSetTest, 992
 fitStepwiseModel, 992
 initialiseSetMapping, 992
 IdSet, 994
 makeLDSets, 993
 numSig, 994
 profileSNPs, 995
 profileScore, 995
 profileTestInitialise, 993
 profileTestSNPInformation, 993
 profileTestScore, 993
 pruneMC, 993
 pruneSets, 994
 pv_maxE_set, 995
 pv_maxG_set, 995
 pv_set, 995
 s_max, 995
 s_min, 996
 selectedSNPs, 996
 Set, 991
 setMapping, 996
 setsort, 996
 sizeSets, 994
 snpset, 996
 stat_set, 996
set
 Data, 118
 DataChar, 124
 DataDouble, 130
 DataFloat, 134
 Eigen, 262
 HWindow, 436
set_by_set
 par, 738
set_chisq_threshold
 par, 738
set_id
 clump_LD, 94
set_master
 Rexp, 941
set_max
 par, 738
set_min
 par, 738
set_p2
 par, 739

set_pval
 clump_LD, 94
 set_r2
 clump_LD, 94
 par, 739
 set_r2_phase
 par, 739
 set_r2_read
 par, 739
 set_r2_read_file
 par, 739
 set_r2_val
 par, 739
 set_r2_write
 par, 740
 set_reference_allele
 par, 740
 set_reference_allele_file
 par, 740
 set_score
 par, 740
 set_score_p
 par, 740
 set_screen
 par, 740
 set_screen_resultfile
 par, 741
 set_second_pval
 clump_LD, 94
 set_step
 par, 741
 set_step_in
 par, 741
 set_table
 par, 741
 set_test
 par, 741
 setAdaptiveSetSNPs
 Perm, 765
 setAlias
 IDHelper, 451
 setAlwaysSplitVariables
 Forest, 324
 setAssocSummary
 Plink, 815
 SetBinaryThresholding
 InteractionNetwork, 479
 SetClass
 DatasetInstance, 209
 setCluster
 Model, 529
 SetConnectivityThreshold
 InteractionNetwork, 480
 SetConnectivityThresholdAbs

 InteractionNetwork, 480
 SetConnectivityThresholding
 InteractionNetwork, 480
 setCovariatesForSNP
 qfam.cpp, 1403
 SetCvSetType
 DatasetInstance, 210
 SetDebugMode
 EpistasisEQtl, 268
 InteractionNetwork, 480
 setDelimiter
 NList, 544
 setDependent
 LinearModel, 503
 LogisticModel, 515
 Model, 529
 SetDistanceMetrics
 Dataset, 182
 SetDistanceSums
 DatasetInstance, 210
 setDominant
 Model, 529
 setFailureValue
 Regain, 894
 setFlagToCase
 Plink, 815
 setFlagToControl
 Plink, 816
 setFlags
 Plink, 815
 SetGammaVector
 CentralityRanker, 70
 SetGlobalGamma
 CentralityRanker, 70
 setHaploid
 Model, 530
 SetId
 DatasetInstance, 211
 setJointValues
 IDHelper, 451
 SetKoptParameters
 ReliefSeqController, 930
 ReliefF, 912
 SetLocalCis
 DcVar, 223
 EpistasisEQtl, 269
 setMapping
 Set, 996
 setMarkerRange
 Plink, 816
 setMissing
 Model, 530
 setModels
 Chap, 75

SetModulesFromFile
 InteractionNetwork, 480

SetNormalize
 AttributeRanker, 46

setObligMissing
 Plink, 816

setOptions
 options.h, 1384
 parse.cpp, 1387

setOriginalRanking
 Perm, 765

setOutputFormat
 Regain, 894

setOutputThreshold
 Regain, 894

setOutputTransform
 Regain, 895

setPermClusters
 Perm, 766

SetPredictedValueTau
 DatasetInstance, 211

setQtIPhenoFromNumericIndex
 Plink, 816

SetRadius
 DcVar, 223
 EpistasisEQtl, 269

setRangeChar
 NList, 544

setRecessive
 Model, 530

setReferenceAllele
 Plink, 816

setSNPList
 Chap, 75

SetSortedSNP, 997
 chisq, 998
 I, 998
 operator<, 998

setSpecificSNPs
 HaploPhase, 410

setSplitWeightVector
 Forest, 324

setStubCodes
 HaploWindow, 429

SetTFRadius
 DcVar, 224
 EpistasisEQtl, 269

setTable
 Plink, 816

setTestHaplotype
 HaploPhase, 411

setTests
 Perm, 766

SetTF

DcVar, 224
EpistasisEQtl, 269

setUpForCNVList
 Plink, 817

setUpQScoring
 dosage.cpp, 1217

setValid
 Model, 530

setVariance
 LinearModel, 503

setVerboseOutput
 Forest, 324

setWindow
 RCount, 880

setbuf
 gzfilebuf, 376

setcompression
 gzfilebuf, 376
 fstream.h, 1577

setfile
 par, 741

SetK
 AttributeRanker, 46

setname
 Plink, 837

sets
 HaploPhase, 422

sets.cpp
 PP, 1448

setscreen.cpp
 PP, 1449

setsort
 Set, 996

setX
 Model, 530

sex
 Individual, 465

sex_effect
 Model, 536

sex_threshold_female
 par, 742

sex_threshold_male
 par, 742

sexCheck
 Plink, 817

sexcode
 Individual, 465

sharingIBSTest
 Plink, 817

shift
 HWindow, 436

show_impossible_IBD
 par, 742

ShowHomophily

InteractionNetwork, 481
 ShowModuleIndices
 InteractionNetwork, 481
 ShowModuleSizes
 InteractionNetwork, 481
 ShowModules
 InteractionNetwork, 481
 showProgress
 Forest, 324
 showmanyC
 gzfilebuf, 377
 shuffleAndSplit
 utility.cpp, 1544
 utility.h, 1565
 shutdown
 helper.cpp, 1294
 helper.h, 1315
 Rconnection, 875
 sibTDT_test
 par, 742
 sibship
 Family, 309
 sifFile
 par, 742
 sifNetwork
 par, 742
 sifThresh
 Regain, 900
 sifToGain
 par, 743
 sig
 LinearModel, 504
 SigmaDeltaAlphas
 ReliefFSeq, 924
 signalNames
 EvaporativeCoolingPrivacy, 298
 silent
 par, 743
 SimParameters, 999
 dprime, 1000
 hetOdds, 1000
 homOdds, 1000
 lfreq, 1001
 lmarker, 1001
 missing, 1001
 name, 1001
 nsnp, 1001
 SimParameters, 1000
 ufreq, 1001
 umarker, 1002
 SimParametersQT, 1003
 dom, 1004
 dprime, 1004
 gAA, 1005
 gAB, 1005
 gBB, 1005
 lfreq, 1005
 lmarker, 1005
 missing, 1005
 name, 1006
 nsnp, 1006
 SimParametersQT, 1004
 ufreq, 1006
 umarker, 1006
 variance, 1006
 simple_interaction
 par, 743
 simul
 par, 743
 simul.cpp
 instanceSNP_QT, 1451
 instanceSNP, 1451
 simul_file
 par, 743
 simul_haps
 par, 743
 simul_label
 par, 744
 simul_ncases
 par, 744
 simul_ncontrols
 par, 744
 simul_prevalence
 par, 744
 simul_qt
 par, 744
 simul_qt_var
 par, 744
 simul_tags
 par, 745
 SimulateData
 ExpressionDataSimulator, 303
 simulateSNPs
 Plink, 817
 simulateSNPs_QT
 Plink, 817
 simulatedDataBase
 ExpressionDataSimulator, 305
 simulatedDataHoldout
 ExpressionDataSimulator, 305
 simulatedDataTest
 ExpressionDataSimulator, 305
 simulatedDataTrain
 ExpressionDataSimulator, 305
 singleMatch
 IDValue, 456
 singlepoint
 par, 745

singleton
 Family, 309

sisocks.h
 build_sin, 1456
 donesocks, 1453
 EACCES, 1453
 EADDRINUSE, 1453
 EAFNOSUPPORT, 1453
 EALREADY, 1453
 EBADF, 1453
 ECONNREFUSED, 1454
 EFAULT, 1454
 EINPROGRESS, 1454
 EINVAL, 1454
 EISCONN, 1454
 ENETUNREACH, 1454
 ENOTSOCK, 1455
 EOPNOTSUPP, 1455
 ETIMEDOUT, 1455
 EWOULDBLOCK, 1455
 inet_aton, 1455
 initsocks, 1456
 SAIN, 1456
 SA, 1455
 sockerrno, 1456
 windows, 1456

sites
 Segment, 988

sizeMatrix
 helper.cpp, 1294
 helper.h, 1316
 stats.h, 1511

sizeSets
 Set, 994

sizeTable
 helper.cpp, 1294
 helper.h, 1316
 stats.h, 1511

sizefMatrix
 helper.cpp, 1294
 helper.h, 1315

sizes
 _RipmResult, 28

skip
 Model, 536
 MultiLocusGenotype, 540

skip_pair
 Plink, 837

sliding_window
 par, 745

sliding_window_size
 par, 745

smoother
 helper.cpp, 1294

 helper.h, 1316

snp
 Alleles, 32
 ClumpPair, 98
 Var, 1072

snp_attrib_file
 par, 745

snp_attrib_filter
 par, 745

snp_attrib_value
 par, 746

snp_include_from_cl
 par, 746

snp_include_range
 par, 746

snp_qual_file
 par, 746

snp_qual_max
 par, 746

snp_qual_min
 par, 747

snp_range_list
 par, 747

snp_test
 Perm, 770

snpAlleleCounts
 BirdseedData, 58

snpDiffFuncPtr
 Dataset, 196
 ReliefF, 917

snpDiffMetricName
 Dataset, 197
 par, 747
 ReliefF, 918

snpGenotypes
 BirdseedData, 59

snpInputType
 DcVar, 227

snpLocations
 DcVar, 228

snpMajorAlleleFreq
 BirdseedData, 59

snpMajorMinorAlleles
 BirdseedData, 59

snpNames
 BirdseedData, 59
 DcVar, 228

snpNearestNeighborFuncPtr
 Dataset, 197

snpNearestNeighborMetricName
 Dataset, 197
 par, 747

snpinfoComparatorAscending
 DcVar.cpp, 1196

snps
 clump_LD, 96
 snpsFilename
 BirdseedData, 59
 Dataset, 198
 snpset
 Plink, 837
 Set, 996
 snrMode
 ReliefFSeq, 924
 sockerrno
 sisocks.h, 1456
 socketConnection
 sockets.cpp, 1460
 sockets.h, 1462
 sockets.cpp
 PVERSION, 1461
 socketConnection, 1460
 sockets.h
 socketConnection, 1462
 sol
 Individual, 465
 sol_family
 par, 747
 SolverMethod
 CentralityRanker.h, 1131
 sort
 Data, 118
 SortedResult, 1015
 chisq, 1016
 l, 1016
 operator<, 1016
 p, 1016
 sortedSegments, 1017
 operator<, 1018
 s, 1018
 sparse_data
 Data, 119
 spec_clusters
 cluster.cpp, 1141
 species_cow
 par, 747
 species_dog
 par, 748
 species_horse
 par, 748
 species_mouse
 par, 748
 species_rice
 par, 748
 species_sheep
 par, 748
 split
 insilico, 20–22
 split_if
 insilico, 22, 23
 split_select_varIDs
 Forest, 331
 Tree, 1034
 split_select_weights
 Forest, 331
 Tree, 1034
 split_values
 Tree, 1035
 split_varIDs
 Tree, 1035
 SplitExpressionCaseControl
 DcVar, 224
 splitNode
 Tree, 1030
 splitNodeInternal
 Tree, 1030
 TreeClassification, 1042
 TreeProbability, 1050
 TreeRegression, 1060
 TreeSurvival, 1069
 SplitRule
 globals.h, 1263
 splitString
 utility.cpp, 1545
 utility.h, 1566
 SplitType
 ExpressionDataSimulator.h, 1229
 splitlerule
 ArgumentHandler, 40
 Forest, 331
 par, 748
 Tree, 1035
 splitweights
 ArgumentHandler, 40
 par, 749
 spmpar
 cdflib.h, 1129
 dcdflib.cpp, 1190
 srand
 CRandom, 108
 src/ArgumentHandler.cpp, 1094
 src/ArgumentHandler.h, 1095
 src/ArmadilloFuncs.cpp, 1095
 src/ArmadilloFuncs.h, 1098
 src/AttributeRanker.cpp, 1102
 src/AttributeRanker.h, 1102
 src/BestN.h, 1103
 src/BirdseedData.cpp, 1105
 src/BirdseedData.h, 1105
 src/CentralityRanker.cpp, 1130
 src/CentralityRanker.h, 1130
 src/ChiSquared.cpp, 1135

src/ChiSquared.h, 1135
src/Data.cpp, 1149
src/Data.h, 1149
src/DataChar.cpp, 1150
src/DataChar.h, 1150
src/DataDouble.cpp, 1151
src/DataDouble.h, 1152
src/DataFloat.cpp, 1153
src/DataFloat.h, 1153
src/Dataset.cpp, 1154
src/Dataset.h, 1156
src/DatasetInstance.cpp, 1156
src/DatasetInstance.h, 1157
src/DcVar.cpp, 1195
src/DcVar.h, 1196
src/Deseq.cpp, 1200
src/Deseq.h, 1200
src/DgeData.cpp, 1202
src/DgeData.h, 1202
src/DistanceMetrics.cpp, 1203
src/DistanceMetrics.h, 1209
src/Edger.cpp, 1217
src/Edger.h, 1218
src/EpistasisEQtl.cpp, 1222
src/EpistasisEQtl.h, 1222
src/EvaporativeCooling.cpp, 1225
src/EvaporativeCooling.h, 1225
src/EvaporativeCoolingPrivacy.cpp, 1226
src/EvaporativeCoolingPrivacy.h, 1227
src/ExpressionDataSimulator.h, 1229
src/Forest.cpp, 1245
src/Forest.h, 1246
src/ForestClassification.cpp, 1247
src/ForestClassification.h, 1247
src/ForestProbability.cpp, 1248
src/ForestProbability.h, 1248
src/ForestRegression.cpp, 1249
src/ForestRegression.h, 1250
src/ForestSurvival.cpp, 1251
src/ForestSurvival.h, 1251
src/GSLRandomBase.h, 1268
src/GSLRandomFlat.h, 1268
src/Insilico.cpp, 1331
src/Insilico.h, 1337
src/InteractionNetwork.cpp, 1358
src/InteractionNetwork.h, 1358
src/PlinkInternalsDataset.cpp, 1395
src/PlinkInternalsDataset.h, 1396
src/PlinkInternalsDatasetInstance.cpp, 1396
src/PlinkInternalsDatasetInstance.h, 1397
src/RReliefF.cpp, 1423
src/RReliefF.h, 1423
src/RandomForest.cpp, 1405
src/RandomForest.h, 1405
src/Rconnection.cpp, 1406
src/Rconnection.h, 1408
src/ReliefF.cpp, 1416
src/ReliefF.h, 1419
src/ReliefFSeq.cpp, 1420
src/ReliefFSeq.h, 1420
src/ReliefFSeqController.cpp, 1421
src/ReliefFSeqController.h, 1422
src/Rsrv.h, 1424
src/SNReliefF.cpp, 1457
src/SNReliefF.h, 1457
src/Statistics.cpp, 1462
src/Statistics.h, 1469
src/StringUtils.h, 1515
src/Tree.cpp, 1520
src/Tree.h, 1520
src/TreeClassification.cpp, 1521
src/TreeClassification.h, 1521
src/TreeProbability.cpp, 1522
src/TreeProbability.h, 1523
src/TreeRegression.cpp, 1524
src/TreeRegression.h, 1525
src/TreeSurvival.cpp, 1526
src/TreeSurvival.h, 1526
src/annot.cpp, 1093
src/assoc.cpp, 1101
src/binput.cpp, 1104
src/blox.cpp, 1106
src/bmerge.cpp, 1108
src/cdflib.h, 1108
src/cfamily.cpp, 1132
src/cfamily.h, 1134
src/clumpld.cpp, 1136
src/clumpld.h, 1137
src/cluster.cpp, 1138
src/cnv.cpp, 1141
src/cnv.h, 1144
src/cnvqt.cpp, 1146
src/config.h, 1147
src/crandom.cpp, 1148
src/crandom.h, 1148
src/dcdflib.cpp, 1158
src/dcdflib.h, 1191
src/dfam.cpp, 1201
src/dosage.cpp, 1216
src/elf.cpp, 1218
src/em.cpp, 1220
src/epi.cpp, 1221
src/filters.cpp, 1230
src/fisher.cpp, 1231
src/fisher.h, 1239
src/flip.cpp, 1245
src/genedrop.cpp, 1252
src/genepi.cpp, 1253

src/genoerr.cpp, 1257
src/genogroup.cpp, 1258
src/genogroup.h, 1258
src/genome.cpp, 1259
src/glm.cpp, 1259
src/globals.h, 1260
src/greport.cpp, 1267
src/gvar.cpp, 1269
src/gvar.h, 1271
src/gxe.cpp, 1271
src/hapglm.cpp, 1272
src/haploCC.cpp, 1273
src/haploQTL.cpp, 1274
src/haploTDT.cpp, 1275
src/haplohelper.cpp, 1273
src/haplindow.cpp, 1275
src/haplindow.h, 1276
src/helper.cpp, 1277
src/helper.h, 1298
src/homozyg.cpp, 1319
src/hotel.cpp, 1323
src/idhelp.cpp, 1324
src/idhelp.h, 1325
src/impute.cpp, 1326
src/inbix.cpp, 1327
src/informative.cpp, 1329
src/input.cpp, 1330
src/ipmpar.h, 1362
src/lapackf.cpp, 1362
src/lapackf.h, 1364
src/legacy.cpp, 1364
src/linear.cpp, 1365
src/linear.h, 1367
src/linput.cpp, 1367
src/locus.cpp, 1368
src/logistic.cpp, 1369
src/logistic.h, 1369
src/lookup.cpp, 1370
src/lookup2.cpp, 1371
src/mds.cpp, 1373
src/merge.cpp, 1373
src/metaanal.cpp, 1374
src/metaem.cpp, 1375
src/mh.cpp, 1376
src/mishap.cpp, 1377
src/model.cpp, 1378
src/model.h, 1378
src/multi.cpp, 1379
src/multiple.cpp, 1380
src/nlist.cpp, 1381
src/nlist.h, 1381
src/nonfounderphasing.cpp, 1382
src/options.cpp, 1383
src/options.h, 1384
src/output.cpp, 1385
src/parse.cpp, 1386
src/pdriver.cpp, 1387
src/perm.cpp, 1388
src/perm.h, 1388
src/phase.cpp, 1389
src/phase.h, 1390
src/plink.cpp, 1390
src/plink.h, 1390
src/poo.cpp, 1398
src/prephap.cpp, 1399
src/profile.cpp, 1400
src/proxy.cpp, 1401
src/qfam.cpp, 1402
src/qualscores.cpp, 1404
src/r.cpp, 1404
src/regain.cpp, 1413
src/regain.h, 1414
src/segment.cpp, 1447
src/sets.cpp, 1447
src/sets.h, 1448
src/setscreen.cpp, 1449
src/sharing.cpp, 1450
src/simul.cpp, 1450
src/sisocks.h, 1451
src/sockets.cpp, 1460
src/sockets.h, 1461
src/stats.cpp, 1477
src/stats.h, 1497
src/step.cpp, 1515
src/tag.cpp, 1518
src/tdt.cpp, 1518
src/tinput.cpp, 1519
src/trio.cpp, 1527
src/utility.cpp, 1529
src/utility.h, 1546
src/version.h, 1567
src/webcheck.cpp, 1568
src/whap.cpp, 1570
src/whap.h, 1571
src/zed.cpp, 1572
src/zed.h, 1573
src/zfstream.cpp, 1575
src/zfstream.h, 1576
standard_beta
 par, 749
standardise
 LinearModel, 503
start
 HWindow, 438
 HaploWindow, 433
 Range, 870
 Segment, 988
startMergeOrder

InteractionNetwork, 486
par, 749
startTemp
 EvaporativeCoolingPrivacy, 298
startWindow
 HaploPhase, 422
stat_set
 Set, 996
state
 GSLRandomBase, 363
Statistics.cpp
 condentropy, 1464
 ConditionalEntropy, 1464
 ConstructAttributeCart, 1465
 DEBUG_E, 1464
 DEBUG_Z, 1464
 Entropy, 1465
 hypot, 1466
 KendallTau, 1466, 1467
 PrintHistogram, 1467
 SelfEntropy, 1468
 ZTransform, 1468
Statistics.h
 condentropy, 1471
 ConditionalEntropy, 1472
 ConstructAttributeCart, 1472
 Entropy, 1473
 Histogram, 1471
 Histogramlt, 1471
 hypot, 1473
 KendallTau, 1473, 1474
 PrintHistogram, 1475
 SelfEntropy, 1475
 VarStd, 1476
 VectorDouble, 1471
 VectorDoublelt, 1471
 ZTransform, 1476
stats
 RegressionRankResults, 903
stats.cpp
 a, 1495
 b, 1495
 betacf, 1480
 betai, 1480
 c, 1495
 chi2x2, 1480, 1481
 chiTable, 1481
 chiprobP, 1481
 d, 1496
 EV_tqli, 1482
 EV_tred2, 1482
 eigenvalues, 1481
 eigenvectors, 1481
 FPMIN, 1479
factorial, 1482
fitModel, 1482
gammln, 1482
getNumericCaseControl, 1483
HIGH, 1479
inverse, 1483
inverse_chiprob, 1483
LOG, 1496
LOW, 1480
ltqnorm, 1483
lubksb, 1483
ludcmp, 1484
matrixAdd, 1484
matrixComputeCovariance, 1484
matrixComputeNodeDegrees, 1484
matrixConnectivityThreshold, 1485
matrixDivideScalar, 1485
matrixElementWiseMultiply, 1485
matrixExtractRowColIdx, 1485
matrixFill, 1486
matrixGetDiag, 1486
matrixGetTrace, 1486
matrixMultiplyScalar, 1486
matrixRead, 1486
matrixSetDiag, 1487
matrixSubtract, 1487
matrixSums, 1487
matrixTranspose, 1487
matrixWrite, 1488
msqrt, 1488
multMatrix, 1488
normdist, 1488
numericLowValueFilter, 1488
numericMeanCenter, 1489
numericStandardize, 1489
numericVarianceFilter, 1489
pca, 1489
pF, 1489
polint, 1490
PP, 1496
pT, 1490
pythag, 1490
qromb, 1490
quantile, 1491
rankByRegression, 1491
realnum, 1491
reportNumericSummaryStats, 1491
rnorm, 1491
SQR, 1492
svbksb, 1492
svd, 1492
svd_inverse, 1492
svdcmp, 1492
svdvar, 1493

symTable, 1493
 tTest, 1494
 tqli, 1493
 trapzd, 1493
 tred2, 1494
 vec2diag, 1494
 vectorSummary, 1494
 zTest, 1494
stats.h
 betacf, 1499
 betai, 1499
 chi2x2, 1499, 1500
 chiTable, 1500
 chiprobP, 1500
 EV_tqli, 1501
 EV_tred2, 1501
 eigenvalues, 1500
 eigenvectors, 1500
 factorial, 1501
 fitModel, 1501
 gammln, 1501
 gammp, 1502
 gcf, 1502
 getNumericCaseControl, 1502
 gser, 1502
 inverse, 1502
 inverse_chiprob, 1503
 ltqnorm, 1503
 lubksb, 1503
 ludcmp, 1503
 matrixAdd, 1503
 matrixComputeCovariance, 1504
 matrixComputeNodeDegrees, 1504
 matrixConnectivityThreshold, 1504
 matrixDivideScalar, 1504
 matrixElementWiseMultiply, 1505
 matrixExtractRowColIdx, 1505
 matrixFill, 1505
 matrixGetDiag, 1505
 matrixGetTrace, 1506
 matrixMultiplyScalar, 1506
 matrixRead, 1506
 matrixSetDiag, 1506
 matrixSubtract, 1506
 matrixSums, 1507
 matrixTranspose, 1507
 matrixWrite, 1507
 msqrt, 1507
 multMatrix, 1507
 normdist, 1508
 numericLowValueFilter, 1508
 numericMeanCenter, 1508
 numericStandardize, 1508
 numericVarianceFilter, 1508
 pca, 1509
 pF, 1509
 polint, 1509
 pT, 1509
 pythag, 1510
 qromb, 1510
 quantile, 1510
 rankByRegression, 1510
 realnum, 1510
 reportNumericSummaryStats, 1511
 rnorm, 1511
 sizeMatrix, 1511
 sizeTable, 1511
 svbksb, 1511
 svd, 1512
 svd_inverse, 1512
 svdcmp, 1512
 svdvar, 1512
 symTable, 1513
 tTest, 1514
 tqli, 1513
 trapzd, 1513
 tred2, 1513
 vec2diag, 1514
 vectorSummary, 1514
 zTest, 1514
status_varID
 ForestSurvival, 358
 TreeSurvival, 1070
statusvarname
 ArgumentHandler, 40
 par, 749
std::less< Locus * >, 492
 operator(), 493
std::less< MultiLocusGenotype * >, 493
 operator(), 494
std::less< Pool * >, 494
 operator(), 495
std::less< Segment * >, 495
 operator(), 496
stop
 HWindow, 438
 HaploWindow, 434
 Range, 870
storageSize
 Rexp, 942
store
 Rexp, 942
string
 Rstring, 969
 Rstrings, 973
stringAt
 Rstrings, 973
Rvector, 983

strings
 Rstrings, 974
 Rvector, 983
strs
 Rvector, 984
stvaln
 cdflib.h, 1129
 dcdflib.cpp, 1190
sublds
 ExpressionDataSimulator, 305
subhaplotypes
 HaplоТhase, 422
subjectLabels
 BirdseedData, 60
subjectLabelsFilename
 BirdseedData, 60
subjectNames
 BirdseedData, 60
subsetfile
 par, 749
sumGeneCounts
 DgeData, 245
SumMatrixPowerSeries
 InteractionNetwork, 481
summ_nonfounders
 par, 749
summariseHomoRuns
 Plink, 818
summary_ibd_output
 par, 750
summaryBasics
 helper.cpp, 1295
 helper.h, 1317
summaryIBDsegments
 Plink, 818
summaryIBSsegments
 Plink, 818
summarylBD
 Plink, 818
summaryIndivSummaries
 homozyg.cpp, 1322
summedProbabilities
 EvaporativeCoolingPrivacy, 298
sums
 TreeRegression, 1060
svbksb
 stats.cpp, 1492
 stats.h, 1511
svd
 stats.cpp, 1492
 stats.h, 1512
svd_inverse
 stats.cpp, 1492
 stats.h, 1512
svd_lapack
 lapackf.cpp, 1363
 lapackf.h, 1364
svdcmp
 stats.cpp, 1492
 stats.h, 1512
svdvar
 stats.cpp, 1493
 stats.h, 1512
sw
 helper.cpp, 1295, 1296
 helper.h, 1317, 1318
swap
 Alleles, 31
SwapAttributes
 Dataset, 183
 DatasetInstance, 211
symTable
 stats.cpp, 1493
 stats.h, 1513
symbolName
 Rsymbol, 978
sync
 gzfilebuf, 377

T

 DatasetInstance.cpp, 1157
 Individual, 465
 ReliefF.cpp, 1418

t

 Perm, 770

T00
 Plink, 837

T01
 Plink, 837

T02
 Plink, 838

t1
 CNVIndivReport, 104

T10
 Plink, 838

t10
 CNVIndivReport, 104

T11
 Plink, 838

t11
 CNVIndivReport, 104

T12
 Plink, 838

t12
 CNVIndivReport, 104

t2
 CNVIndivReport, 104

T20

Plink, 838
 T21
 Plink, 838
 T22
 Plink, 839
 t3
 CNVIndivReport, 105
 t4
 CNVIndivReport, 105
 t5
 CNVIndivReport, 105
 t6
 CNVIndivReport, 105
 t7
 CNVIndivReport, 105
 t8
 CNVIndivReport, 105
 t9
 CNVIndivReport, 106
 TDT_test
 par, 750
 TDT
 Family, 309
 tTest
 stats.cpp, 1494
 stats.h, 1514
 table_t
 plink.h, 1394
 tag
 Rlist, 955
 tagMode
 Plink, 818
 tagfile
 par, 750
 tail
 Rlist, 955
 tallyUnambiguousCounts
 HaploWindow, 429
 targetpartitionszie
 ArgumentHandler, 41
 par, 750
 tau
 EvaporativeCoolingPrivacy, 298
 tcnt
 Plink, 839
 temperatures
 EvaporativeCooling, 287
 tent4
 dcdflib.cpp, 1168
 terminal_class_counts
 TreeProbability, 1051
 test
 EvaporativeCoolingPrivacy, 298
 Perm, 770
 test_full_model
 par, 750
 test_hap
 HaploPhase, 422
 test_hap_CC
 par, 750
 test_hap_GLM_omnibus
 par, 751
 test_hap_GLM
 par, 751
 test_hap_QTL
 par, 751
 test_hap_TDT
 par, 751
 test_hap_only
 par, 751
 test_list
 par, 751
 test_missing
 par, 752
 testAssoc
 Plink, 819
 testCNVwithGLM
 Plink, 819
 testCNVwithQT
 Plink, 819
 testError
 EvaporativeCoolingPrivacy, 299
 testErrors
 EvaporativeCoolingPrivacy, 299
 testGenomeIBDByCovariate
 Plink, 819
 testHaplotypeFreq
 HaploPhase, 411
 testMiss
 Plink, 820
 testParameter
 Model, 536
 testQAssoc
 Plink, 820
 testQAssocGXE2
 Plink, 820
 testRareDistribution
 Plink, 820
 testSet
 HaploPhase, 422
 testSibTDT
 Plink, 820
 testTDT_POO
 Plink, 821
 testTDT
 Plink, 821
 tfMode
 DcVar, 228

EpistasisEQtl, 271
tfRadius
 DcVar, 228
 EpistasisEQtl, 271
tfTableLoaded
 DcVar, 228
 EpistasisEQtl, 272
tfamfile
 par, 752
tfile_input
 par, 752
thin_param
 par, 752
thin_snps
 par, 752
thinSNPs
 Plink, 821
thisTFSnpIndices
 DcVar, 228
 EpistasisEQtl, 272
thisTranscriptSnpIndices
 DcVar, 229
 EpistasisEQtl, 272
thread_ranges
 Forest, 331
threshold
 EvaporativeCoolingPrivacy, 299
thresholdType
 par, 752
thresholdValue
 par, 753
Timestamp
 Insilico.cpp, 1336
 Insilico.h, 1353
tinput.cpp
 LOG, 1519
titvFilename
 par, 753
to_lower
 insilico, 23
to_upper
 insilico, 24
to_window
 par, 753
tokenize
 NList, 545
tokenizeLine
 helper.cpp, 1296
 helper.h, 1318
 ZInput, 1083
tol
 dcdflib.cpp, 1169, 1170
tolerance
 EvaporativeCoolingPrivacy, 299
topN
 CentralityRanker, 73
totalRegionReads
 CHIP_SEQ_INFO, 81
totalTests
 DcVar, 229
tpedfile
 par, 753
tqli
 stats.cpp, 1493
 stats.h, 1513
trackSharedHaplotypes
 HaploPhase, 411
trackThisSegment
 HaploPhase, 411
trackedIBS
 HaploPhase, 422
trackedN
 HaploPhase, 423
train
 EvaporativeCoolingPrivacy, 299
trainError
 EvaporativeCoolingPrivacy, 299
trainErrors
 EvaporativeCoolingPrivacy, 300
trainImportance
 EvaporativeCoolingPrivacy, 300
trainInvlImportance
 EvaporativeCoolingPrivacy, 300
trans
 HaploPhase, 423
TranscriptFactorTable
 EpistasisEQtl.h, 1224
Transcript, 1019
 bpEnd, 1020
 bpStart, 1020
 chr, 1020
 name, 1020
 Transcript, 1020
transcript
 Plink, 839
transcriptFactorLUT
 DcVar, 229
 EpistasisEQtl, 272
TranscriptFactorTableClt
 EpistasisEQtl.h, 1224
transmissionCount
 HaploPhase, 411
transmissionTotal
 HaploPhase, 423
transmissionX2
 HaploPhase, 423
transmissionX
 HaploPhase, 423

transposeMatrix
 genepi.cpp, 1256
 trapzd
 stats.cpp, 1493
 stats.h, 1513
 tred2
 stats.cpp, 1494
 stats.h, 1513
 Tree, 1021
 ~Tree, 1024
 alpha, 1030
 appendToFile, 1025
 appendToFileInternal, 1025
 bootstrap, 1025
 bootstrapWeighted, 1025
 bootstrapWithoutReplacement, 1025
 bootstrapWithoutReplacementWeighted, 1026
 case_weights, 1030
 child_nodeIDs, 1031
 cleanUpInternal, 1026
 computePermutationImportance, 1026
 computePredictionAccuracyInternal, 1026
 createEmptyNode, 1026
 createEmptyNodeInternal, 1027
 createPossibleSplitVarSubset, 1027
 DISALLOW_COPY_AND_ASSIGN, 1027
 data, 1031
 dependent_varID, 1031
 deterministic_varIDs, 1031
 dropDownSamplePermuted, 1027
 getChildNodeIDs, 1027
 getInbagCounts, 1028
 getNumSamplesOob, 1028
 getOobSampleIDs, 1028
 getSplitValues, 1028
 getSplitVarIDs, 1028
 grow, 1028
 holdout, 1031
 importance_mode, 1031
 inbag_counts, 1032
 init, 1029
 initInternal, 1029
 is_ordered_variable, 1032
 keep_inbag, 1032
 memory_saving_splitting, 1032
 min_node_size, 1032
 minprop, 1032
 mtry, 1033
 no_split_variables, 1033
 num_samples, 1033
 num_samples_oob, 1033
 oob_sampleIDs, 1033
 permuteAndPredictOobSamples, 1029
 predict, 1030

prediction_terminal_nodeIDs, 1033
 random_number_generator, 1034
 sample_fraction, 1034
 sample_with_replacement, 1034
 sampleIDs, 1034
 split_select_varIDs, 1034
 split_select_weights, 1034
 split_values, 1035
 split_varIDs, 1035
 splitNode, 1030
 splitNodeInternal, 1030
 splitrule, 1035
 Tree, 1024
 variable_importance, 1035
 TreeClassification, 1036
 ~TreeClassification, 1039
 addGiniImportance, 1039
 appendToFileInternal, 1039
 class_values, 1043
 cleanUpInternal, 1039
 computePermutationImportanceInternal, 1040
 computePredictionAccuracyInternal, 1040
 counter, 1043
 counter_per_class, 1043
 createEmptyNodeInternal, 1040
 DISALLOW_COPY_AND_ASSIGN, 1040
 estimate, 1040
 findBestSplit, 1041
 findBestSplitValueLargeQ, 1041
 findBestSplitValueSmallQ, 1041
 findBestSplitValueUnordered, 1041
 getPrediction, 1042
 getPredictionTerminalNodeID, 1042
 initInternal, 1042
 response_classIDs, 1043
 splitNodeInternal, 1042
 TreeClassification, 1038
 TreeProbability, 1044
 ~TreeProbability, 1047
 addImpurityImportance, 1047
 addToTerminalNodes, 1047
 appendToFileInternal, 1047
 class_values, 1051
 cleanUpInternal, 1048
 computePermutationImportanceInternal, 1048
 computePredictionAccuracyInternal, 1048
 counter, 1051
 counter_per_class, 1051
 createEmptyNodeInternal, 1048
 DISALLOW_COPY_AND_ASSIGN, 1048
 findBestSplit, 1049
 findBestSplitValueLargeQ, 1049
 findBestSplitValueSmallQ, 1049
 findBestSplitValueUnordered, 1049

getPrediction, 1050
getPredictionTerminalNodeID, 1050
getTerminalClassCounts, 1050
initInternal, 1050
response_classIDs, 1051
splitNodeInternal, 1050
terminal_class_counts, 1051
TreeProbability, 1046
TreeRegression, 1052
 ~TreeRegression, 1056
 addImpurityImportance, 1056
 appendToFileInternal, 1056
 cleanUpInternal, 1056
 computePermutationImportanceInternal, 1057
 computePredictionAccuracyInternal, 1057
 computePredictionMSE, 1057
 counter, 1060
 createEmptyNodeInternal, 1057
 DISALLOW_COPY_AND_ASSIGN, 1057
 estimate, 1058
 findBestSplit, 1058
 findBestSplitMaxstat, 1058
 findBestSplitValueLargeQ, 1058
 findBestSplitValueSmallQ, 1058
 findBestSplitValueUnordered, 1059
 getPrediction, 1059
 getPredictionTerminalNodeID, 1059
 initInternal, 1059
 splitNodeInternal, 1060
 sums, 1060
 TreeRegression, 1055
TreeSurvival, 1061
 ~TreeSurvival, 1064
 appendToFileInternal, 1064
 chf, 1069
 cleanUpInternal, 1064
 computeAucSplit, 1065
 computeChildDeathCounts, 1065
 computeDeathCounts, 1065
 computePermutationImportanceInternal, 1065
 computePredictionAccuracyInternal, 1066
 computeSurvival, 1066
 createEmptyNodeInternal, 1066
 DISALLOW_COPY_AND_ASSIGN, 1066
 findBestSplit, 1066
 findBestSplitMaxstat, 1067
 findBestSplitValueAUC, 1067
 findBestSplitValueLogRank, 1067
 findBestSplitValueLogRankUnordered, 1068
 getChf, 1068
 getPrediction, 1068
 getPredictionTerminalNodeID, 1069
 initInternal, 1069
 num_deaths, 1070
 num_samples_at_risk, 1070
 num_timepoints, 1070
 response_timepointIDs, 1070
 splitNodeInternal, 1069
 status_varID, 1070
 TreeSurvival, 1064
 unique_timepoints, 1070
TreeType
 globals.h, 1263
trees
 Forest, 332
treetype
 ArgumentHandler, 41
 par, 753
trend_only
 par, 753
trim
 insilico, 24, 25
trim_left
 insilico, 25
trim_right
 insilico, 25, 26
trio.cpp
 addParent, 1528
 addPerson, 1528
 gprint, 1528
 PP, 1529
tstatMode
 ReliefFSeq, 925
tucc
 par, 754
two
 CSNP, 112
 Individual, 465
two_locus_table
 helper.cpp, 1297
 helper.h, 1319
twoDFmodel
 par, 754
twoDFmodel_hethom
 par, 754
twolocus_snp1
 par, 754
twolocus_snp2
 par, 754
type
 Model, 536
 Rexp, 944
 Segment, 988
typed_interaction
 Model, 536
u
 LinearModel, 504

uc
 HaploWindow, 434
 ucnt
 RCount, 882
 ucnt2
 RCount, 883
 ufreq
 SimParameters, 1001
 SimParametersQT, 1006
 uint
 globals.h, 1262
 uintToString
 utility.cpp, 1545
 utility.h, 1566
 umarker
 SimParameters, 1002
 SimParametersQT, 1006
 unbuffered
 ZInput, 1084
 ZOutput, 1088
 uncompress_file
 par, 754
 underflow
 gzfilebuf, 378
 union_max
 Pool, 853
 union_min
 Pool, 853
 uniqFieldCount
 IDFile, 445
 unique_data_values
 Data, 120
 unique_timepoints
 ForestSurvival, 358
 TreeSurvival, 1070
 uniqueFields
 IDHelper, 453
 unknownDH
 PairwiseLinkage, 560
 untrans
 HaploPhase, 423
 update
 EvaporativeCoolingPrivacy, 300
 Perm, 766
 update_allele_file
 par, 755
 update_alleles
 par, 755
 update_chr
 par, 755
 update_cm
 par, 755
 update_ids
 par, 755
 update_ids_file
 par, 755
 update_map
 par, 756
 update_mapfile
 par, 756
 update_name
 par, 756
 update_parents
 par, 756
 update_parents_file
 par, 756
 update_pheno
 par, 756
 update_pheno_file
 par, 757
 update_sex
 par, 757
 update_sex_file
 par, 757
 updateAlias
 IDValue, 456
 UpdateAllLevelCounts
 Dataset, 184
 updateAlleles
 Plink, 821
 updateFamFile
 Plink, 822
 updateForImputation
 HaploPhase, 412
 updateInterval
 EvaporativeCoolingPrivacy, 300
 UpdateLevelCounts
 Dataset, 184
 updateMapFile
 Plink, 822
 updateSNP
 Perm, 766
 updateStats
 Regain, 895
 UpdateTemperature
 EvaporativeCoolingPrivacy, 292
 upper_
 GSLRandomFlat, 366
 use_GC
 par, 757
 use_subset
 par, 757
 useAbs
 par, 757
 useBinaryThreshold
 InteractionNetwork, 486
 useConnectivityThreshold
 InteractionNetwork, 486

useEmpiricalVariance
 HaploPhase, 424

useOutputThreshold
 Regain, 901

useWeighted
 par, 758

UsingSimData
 EvaporativeCoolingPrivacy, 292

utility.cpp
 adjustPvalues, 1531
 beautifyTime, 1532
 checkPositiveIntegers, 1532
 checkUnorderedVariables, 1532
 computeConcordanceIndex, 1533
 drawWithoutReplacementFisherYates, 1533
 drawWithoutReplacementKnuth, 1534
 drawWithoutReplacementSimple, 1534
 drawWithoutReplacementSkip, 1535
 drawWithoutReplacementWeighted, 1535, 1536
 dstdnorm, 1536
 equalSplit, 1537
 loadDoubleVectorFromFile, 1537
 logrankScores, 1537
 logrankScoresData, 1538
 maxstat, 1538
 maxstatInData, 1539
 maxstatPValueLau92, 1540
 maxstatPValueLau94, 1540
 maxstatPValueUnadjusted, 1541
 mostFrequentValue, 1541
 numSamplesLeftOfCutpoint, 1542
 numSamplesLeftOfCutpointInData, 1542
 orderInData, 1543
 pstdnorm, 1543
 roundToNextMultiple, 1544
 shuffleAndSplit, 1544
 splitString, 1545
 uintToString, 1545

utility.h
 adjustPvalues, 1548
 beautifyTime, 1549
 checkPositiveIntegers, 1549
 checkUnorderedVariables, 1549
 computeConcordanceIndex, 1550
 drawWithoutReplacementKnuth, 1550
 drawWithoutReplacementSimple, 1551
 drawWithoutReplacementSkip, 1551
 drawWithoutReplacementWeighted, 1552
 dstdnorm, 1553
 equalSplit, 1553
 loadDoubleVectorFromFile, 1554
 logrankScores, 1554
 logrankScoresData, 1555
 maxstat, 1555

maxstatInData, 1557
maxstatPValueLau92, 1557
maxstatPValueLau94, 1558
maxstatPValueUnadjusted, 1559
mostFrequentClass, 1559
mostFrequentValue, 1560
numSamplesLeftOfCutpoint, 1560
numSamplesLeftOfCutpointInData, 1560
order, 1561
orderInData, 1561
pstdnorm, 1562
rank, 1562
readVector1D, 1563
readVector2D, 1563
roundToNextMultiple, 1564
saveVector1D, 1564, 1565
saveVector2D, 1565
shuffleAndSplit, 1565
splitString, 1566
uintToString, 1566

V
 LogisticModel, 516

v
 LinearModel, 505

v_phenotype
 Plink, 839

v_pihat
 Plink, 839

VPHASE
 HaploPhase, 424

val
 OptionSet, 552

val1
 gzomanip2, 395

val2
 gzomanip2, 395

valid
 HWindow, 438
 Model, 537

validParameters
 Model, 531

validateNonfounder
 HaploPhase, 412

validateSegments
 Plink, 822

validN
 HaploPhase, 424

value
 CArgs, 64
 IDValue, 456

value_double
 CArgs, 64

value_int

CArgs, 64
 value_lui
 CArgs, 65
 ValueType
 Insilico.h, 1347
 values
 IDGroup, 447
 Var, 1071
 a1, 1072
 operator<, 1072
 operator==, 1072
 snp, 1072
 VarStd
 Statistics.h, 1476
 varValue
 CArgs, 65
 variable_importance
 Forest, 332
 Tree, 1035
 variable_names
 Data, 120
 variableNames
 CentralityRanker, 73
 variance
 SimParametersQT, 1006
 Variant, 1073
 acode, 1075
 alleles, 1075
 allelicVariation, 1075
 copyNumberVariation, 1075
 freqs, 1075
 integerDosage, 1076
 nallele, 1076
 Variant, 1075
 vars
 RegressionRankResults, 903
 varY
 LinearModel, 505
 vec2diag
 stats.cpp, 1494
 stats.h, 1514
 vector_t
 helper.h, 1300
 plink.h, 1394
 vector_tld
 genepi.cpp, 1254
 VectorDouble
 Statistics.h, 1471
 VectorDoublelt
 Statistics.h, 1471
 vectorSummary
 stats.cpp, 1494
 stats.h, 1514
 verbose
 ArgumentHandler, 41
 par, 758
 verbose_out
 Forest, 332
 verboseDisplayWindows
 HaploPhase, 412
 verboseDisplayWindows2
 haplindow.cpp, 1276
 version.h
 RANGER_VERSION, 1567
 vif_prune
 helper.cpp, 1297
 helper.h, 1319
 vif_threshold
 par, 758
 vmaxget
 fisher.h, 1244
 vmaxset
 fisher.h, 1244
 voidEval
 Rconnection, 875

W

Individual, 465
 ReliefF, 918

w

LinearModel, 505

WGT

HaploPhase, 424

WMLocus, 1076

allele, 1077
 chr, 1078
 name, 1078
 reset, 1077
 WMLocus, 1077
 weight, 1078

warnings

Plink, 839
 Regain, 901

web_check

par, 758

webcheck

Plink, 822

webcheck.cpp

GET_STRING, 1568
 IP_ADDR, 1568
 PORT_NUM, 1569
 PREL, 1569
 PVERSION, 1569

weight

WMLocus, 1078

weightByDistanceMethod

par, 758
 ReliefF, 918

weightByDistanceSigma
 par, 758
 ReliefF, 918
weighted_count_intersects
 cnv.cpp, 1143
 cnv.h, 1146
weighted_mm
 par, 759
weightedBaseline
 Segment, 988
weightedCount
 Segment, 988
whap
 Plink, 840
whap.cpp
 ci, 1570
 displayHaploGroups, 1570
width
 IDField, 442
window
 par, 759
windows
 HaploPhase, 424
 sisocks.h, 1456
wmlocus
 Individual, 466
write
 ArgumentHandler, 41
 ZOutput, 1088
write_BITFILE
 Plink, 822
write_bitfile
 par, 759
write_clusters
 Plink, 822
write_covariates
 Plink, 823
write dosage
 par, 759
write_set
 par, 759
write_snplist
 par, 759
 Plink, 823
WriteAttributeScores
 EvaporativeCooling, 280
 ReliefSeqController, 930
 ReliefF, 912
WriteBestAttributes
 EvaporativeCoolingPrivacy, 292
WriteBestKs
 ReliefSeqController, 931
 ReliefF, 914
WriteCheckpoint
 DcVar, 224
WriteClassificationErrors
 EvaporativeCooling, 282
writeComponents
 Regain, 901
writeCompressedFormat
 Regain, 901
writeConfusionFile
 Forest, 324
 ForestClassification, 337
 ForestProbability, 343
 ForestRegression, 350
 ForestSurvival, 357
WriteDatasetInMayoFormat
 Deseq, 237
 Edger, 260
WriteDelimitedFile
 InteractionNetwork, 482
writeFile
 Rconnection, 875
writeImportanceFile
 Forest, 325
writeInformative
 Plink, 823
WriteLevelCounts
 Dataset, 185
writeLine
 ZOutput, 1088
WriteNewDataset
 Dataset, 185
WriteNewPlinkCovarDataset
 Dataset, 186
WriteNewPlinkPedDataset
 Dataset, 187
writeOutput
 Forest, 325
writeOutputInternal
 Forest, 325
 ForestClassification, 337
 ForestProbability, 344
 ForestRegression, 350
 ForestSurvival, 357
writePredictionFile
 Forest, 325
 ForestClassification, 337
 ForestProbability, 344
 ForestRegression, 350
 ForestSurvival, 357
writeRcomm
 Regain, 895
writeRegain
 Regain, 895
writeRegainToFile
 Regain, 895

writeRegainToSifFile
 Regain, 896

WriteResults
 DcVar, 224

WriteScores
 AttributeRanker, 46
 Deseq, 237
 Edger, 260
 RandomForest, 865

WriteScoresWithPValues
 ChiSquared, 88

writeSetFile
 Plink, 823

WriteSifFile
 InteractionNetwork, 482

WriteSnpTiTvInfo
 Dataset, 187

WriteTemperatures
 EvaporativeCooling, 282

WriteToConsole
 CentralityRanker, 70

WriteToFile
 CentralityRanker, 71
 InteractionNetwork, 482

writeforest
 par, 760

X
 HaploPhase, 424
 Model, 537

XT_ARRAY_BOOL_UA
 Rsrv.h, 1441

XT_ARRAY_BOOL
 Rsrv.h, 1441

XT_ARRAY_CPLX
 Rsrv.h, 1441

XT_ARRAY_DOUBLE
 Rsrv.h, 1442

XT_ARRAY_INT
 Rsrv.h, 1442

XT_ARRAY_STR
 Rsrv.h, 1442

XT_BOOL
 Rsrv.h, 1442

XT_CLOS
 Rsrv.h, 1442

XT_DOUBLE
 Rsrv.h, 1443

XT_HAS_ATTR
 Rsrv.h, 1443

XT_INT
 Rsrv.h, 1443

XT_LANG_NOTAG
 Rsrv.h, 1443

XT_LANG_TAG
 Rsrv.h, 1444

XT_LANG
 Rsrv.h, 1443

XT_LARGE
 Rsrv.h, 1444

XT_LIST_NOTAG
 Rsrv.h, 1444

XT_LIST_TAG
 Rsrv.h, 1444

XT_LIST
 Rsrv.h, 1444

XT_NULL
 Rsrv.h, 1445

XT_RAW
 Rsrv.h, 1445

XT_S4
 Rsrv.h, 1445

XT_STR
 Rsrv.h, 1445

XT_SYMNAME
 Rsrv.h, 1445

XT_SYM
 Rsrv.h, 1445

XT_UNKNOWN
 Rsrv.h, 1446

XT_VECTOR_EXP
 Rsrv.h, 1446

XT_VECTOR_STR
 Rsrv.h, 1446

XT_VECTOR
 Rsrv.h, 1446

xchr
 Model, 537

xchr_model
 par, 760

Xgamm
 cdflib.h, 1129
 dcdflib.cpp, 1190

Y
 LinearModel, 505
 LogisticModel, 516

yokeMissing
 Model, 531

Ysize
 Model, 531

Z, 1078
 Z, 1079
 z0, 1079
 z1, 1079
 z2, 1079

z
 Eigen, 262

z0
 Z, 1079
z00
 ZZ, 1091
z01
 ZZ, 1091
z02
 ZZ, 1091
z1
 Z, 1079
z10
 ZZ, 1091
z11
 ZZ, 1092
z12
 ZZ, 1092
z2
 Z, 1079
z20
 ZZ, 1092
z21
 ZZ, 1092
z22
 ZZ, 1092
ZInput, 1080
 buf, 1084
 close, 1082
 compressed, 1084
 endOfFile, 1083
 filename, 1084
 inf, 1084
 open, 1083
 readChar, 1083
 readLine, 1083
 tokenizeLine, 1083
 unbuffered, 1084
ZInput, 1082
 zinf, 1085
ZOUTFILE
 Plink, 840
ZOutput, 1085
 buf, 1089
 close, 1087
 compressed, 1089
 filename, 1089
 open, 1088
 operator<<, 1088
 outf, 1089
 unbuffered, 1088
 write, 1088
 writeLine, 1088
ZOutput, 1087
 zoutf, 1089
zTest
stats.cpp, 1494
stats.h, 1514
ZTransform
 Statistics.cpp, 1468
 Statistics.h, 1476
zVals
 DcVar, 229
zed.cpp
 fileCompress, 1572
 fileUncompress, 1573
 PP, 1573
zed.h
 fileCompress, 1574
 fileUncompress, 1574
 MAX_LINE_LENGTH, 1574
zero
 dcdflib.cpp, 1170, 1171
 HaploWindow, 434
zero_cluster
 par, 760
zero_cluster_filename
 par, 760
zeroOnCluster
 Plink, 823
zeroPadNumber
 insilico, 26
zfstream.cpp
 BIGBUFSIZE, 1575
 SMALLBUFSIZE, 1575
zfstream.h
 operator<<, 1577
 setcompression, 1577
zinf
 ZInput, 1085
Zlocus
 Plink, 840
zoutf
 ZOutput, 1089
zt
 Perm, 771
ZZ, 1090
 z00, 1091
 z01, 1091
 z02, 1091
 z10, 1091
 z11, 1092
 z12, 1092
 z20, 1092
 z21, 1092
 z22, 1092
 ZZ, 1091