Package 'rMVP'

January 16, 2019

Type Package

Title A Memory-efficient, Visualization-enhanced, and Parallel-accelerated Tool For GWAS

Version 1.2.0

Date 2018-12-12

Author Lilin Yin, Haohao Zhang, Zhiwu Zhang, Xinyun Li, Shuhong Zhao, and Xiaolei Liu

Maintainer Xiaolei Liu <xiaoleiliu@mail.hzau.edu.cn>

Description a Memory efficient, Visualization enhanced, and Parallel accelerated Tool For GWAS

License Apache License | file LICENSE

SystemRequirements C++11

BugReports https://github.com/XiaoleiLiuBio/MVP/issues

Imports utils, stats, methods, graphics, grDevices

Depends R (>= 3.5), Rcpp, RcppProgress, BH, bigmemory, biganalytics, parallel, MASS

LinkingTo Rcpp, RcppProgress, BH, bigmemory

NeedsCompilation yes

biocViews GenomeWideAssociation, Visualization, Genetics, Preprocessing, QualityControl, Software

Suggests knitr, testthat, rmarkdown

VignetteBuilder knitr

RoxygenNote 6.1.1

R topics documented:

MVP *MVP, Memory-efficient, Visualization-enhanced, Parallel-accelerated*

Description

Object 1: To perform GWAS using General Linear Model (GLM), Mixed Linear Model (MLM), and FarmCPU model Object 2: To calculate kinship among individuals using Varaden method Object 3: Estimate variance components using EMMA, FaST-LMM, and HE regression Object 4: Generate high-quality figures

Usage

```
MVP(phe, geno, map, K = NULL, nPC.GLM = NULL, nPC.MLM = NULL,
  nPC.FarmCPU = NULL, perc = 1, CV.GLM = NULL, CV.MLM = NULL,
  CV.FarmCPU = NULL, REML = NULL, priority = "speed",
  ncpus = detectCores(logical = FALSE), vc.method = "EMMA",
  method = "MLM", maxLine = 1000, memo = NULL, P = NULL,
  method.sub = "reward", method.sub.final = "reward",
  method.bin = "static", bin.size = c(5e+05, 5e+06, 5e+07),
  bin.selection = seq(10, 100, 10), Prior = NULL, maxLoop = 10,
  threshold.output = 1, iteration.output = FALSE, p.threshold = NA,
  QTN.threshold = NULL, bound = NULL, outward = FALSE,
  permutation.threshold = FALSE, permutation.rep = 100, bar = TRUE,
  col = c("dodgerblue4", "olivedrab4", "violetred", "darkgoldenrod1",
  "purple4"), plot.type = "b", file.output = TRUE, file = "jpg",
  dpi = 300, threshold = 0.05, Ncluster = 1, signal.cex = 0.8,
  box = FALSE)
```

Arguments

phe	phenotype, n * 2 matrix, n is sample size
geno	Genotype in bigmatrix format; m * n, m is marker size, n is sample size
map	SNP map information, SNP name, Chr, Pos
K	Kinship, Covariance matrix(n * n) for random effects, must be positive semi-definite
nPC.GLM	number of PCs added as fixed effects in GLM
nPC.MLM	number of PCs added as fixed effects in MLM
nPC.FarmCPU	number of PCs added as fixed effects in FarmCPU
perc	percentage of total SNPs selected for PCA
CV.GLM	covariates added in GLM
CV.MLM	covariates added in MLM
CV.FarmCPU	covariates added in FarmCPU
REML	a list contains ve and vg
priority	speed or memory
ncpus	number of cpus used for parallel
vc.method	methods for estimating variance component("EMMA" or "GEMMA")
method	the GWAS model, "GLM", "MLM", and "FarmCPU", models can be selected simutaneously, i.e. c("GLM", "MLM", "FarmCPU")
maxLine	when the priority is 'memory', users can change this parameter to limit the memory
memo	a marker added on output file name
P	a start p value for each SNP
method.sub, method.sub.final	method.sub.final method used in substitution process
	method used in substitution process, five options: 'penalty', 'reward', 'mean', 'median', or 'onsite'
method.bin	EMMA or FaSTLMM
bin.size	window size in genome
bin.selection	a vector, how many windows selected
Prior	four columns, SNP name, Chr, Pos, P
maxLoop	maximum number of iterations
threshold.output	output GWAS results only for SNPs with p value lower than the threshold.output
iteration.output	whether to output results for FarmCPU iterations
p.threshold	if all p values in the 1st iteration are bigger than p.threshold, FarmCPU stops
QTN.threshold	Only SNPs have a more significant p value than QTN.threshold have chance to be selected as pseudo QTNs
bound	maximum number of SNPs selected as pseudo QTNs for each iteration
outward	the direction of circular Manhattan plot
permutation.threshold	if use a permutation cutoff or not (bonferroni cutoff)

permutation.rep	number of permutation replicates
bar	if TRUE, the progress bar will be drawn on the terminal
col	for color of points in each chromosome on manhattan plot
plot.type	"b" (both Manhattan plot and qq plot will be draw) or "q" (qq plot only)
file.output	whether to output files or not
file	figure formats, "jpg", "tiff"
dpi	resolution for output figures
threshold	a cutoff line on manhattan plot, 0.05/marker size
Ncluster	number of colors used for drawing PC 1 and PC 2
signal.cex	point size on output figures
box	logical, if TRUE, the box frame will be added in output figure

Details

Build date: Aug 30, 2017 Last update: Dec 14, 2018

Value

a m * 2 matrix, the first column is the SNP effect, the second column is the P values Output: MVP.return$map - SNP map information, SNP name, Chr, Pos Output: MVP.return$glm.results - p-values obtained by GLM method Output: MVP.return$mlm.results - p-values obtained by MLM method Output: MVP.return$farmcpu.results - p-values obtained by FarmCPU method

Author(s)

Lilin Yin, Haohao Zhang, and Xiaolei Liu

Examples

```
phePath <- system.file("extdata", "mvp.phe", package = "rMVP")
phenotype <- read.table(phePath, header=TRUE)
print(dim(phenotype))
genoPath <- system.file("extdata", "mvp.geno.desc", package = "rMVP")
genotype <- attach.big.matrix(genoPath)
print(dim(genotype))
mapPath <- system.file("extdata", "mvp.map", package = "rMVP")
map <- read.table("mvp.map" , head = TRUE)
mvp <- MVP(phe=phenotype, geno=genotype, map=map,
  method=c("GLM", "MLM", "FarmCPU"), file.output=FALSE)
str(mvp)
```

MVP.Data *MVP.Data: To prepare data for MVP package Author: Xiaolei Liu, Lilin Yin and Haohao Zhang Build date: Aug 30, 2016 Last update: Sep 12, 2018*

Description

MVP.Data: To prepare data for MVP package Author: Xiaolei Liu, Lilin Yin and Haohao Zhang Build date: Aug 30, 2016 Last update: Sep 12, 2018

Usage

```
MVP.Data(fileMVP = NULL, fileVCF = NULL, fileHMP = NULL,
  fileBed = NULL, fileNum = NULL, fileMap = NULL, filePhe = NULL,
  fileInd = NULL, fileKin = TRUE, filePC = TRUE, out = "mvp",
  sep.num = "\t", auto_transpose = TRUE, sep.map = "\t",
  sep.phe = "\t", sep.kin = "\t", sep.pc = "\t",
  type.geno = "char", pheno_cols = NULL, SNP.impute = "Major",
  maxLine = 10000, priority = "speed", perc = 1, pcs.keep = 5,
  verbose = TRUE, ...)
```

Arguments

fileMVP	Genotype in MVP format
fileVCF	Genotype in VCF format
fileHMP	Genotype in hapmap format
fileBed	Genotype in PLINK binary format
fileNum	Genotype in numeric format; pure 0, 1, 2 matrix; m * n, m is marker size, n is sample size
fileMap	SNP map information, there are three columns, including SNP_ID, Chromosome, and Position
filePhe	Phenotype, the first column is taxa name, the subsequent columns are traits
fileInd	Individual name file
fileKin	Kinship that represents relationship among individuals, n * n matrix, n is sample size
filePC	Principal components, n*npc, n is sample size, npc is number of top columns of principal components
out	prefix of output file name
sep.num	seperator for numeric file.
auto_transpose	Whether to automatically transpose numeric genotypes, the default is True, which will identify the most one of the rows or columns as a marker, If set to False, the row represents the marker and the column represents the individual.
sep.map	seperator for map file.
sep.phe	seperator for phenotype file.
sep.kin	seperator for Kinship file.
sep.pc	seperator for PC file.

type.geno	type parameter in bigmemory, genotype data. The default is char, it is highly recommended *NOT* to modify this parameter.
pheno_cols	Extract which columns of the phenotype file (including individual IDs)
SNP.impute	"Left", "Middle", "Right", or NULL for skip impute.
maxLine	number of SNPs, only used for saving memory when calculate kinship matrix
priority	"speed" or "memory"
perc	Percentage of markers used to calculate PCA
pcs.keep	how many PCs to keep
verbose	whether to print detail.
...	Compatible with DEPRECATED parameters.

Value

NULL Output files: genotype.desc, genotype.bin: genotype file in bigmemory format phenotype.phe: ordered phenotype file, same taxa order with genotype file map.map: SNP information k.desc, k.bin: Kinship matrix in bigmemory format pc.desc, pc.bin: PC matrix in bigmemory format Requirement: fileHMP, fileBed, and fileNum can not input at the same time

Examples

```
bfilePath <- system.file("extdata", "02_bfile", "mvp", package = "rMVP")
MVP.Data(fileBed=bfilePath)
```

MVP.Data.Bfile2MVP *MVP.Data.Bfile2MVP: To transform plink binary data to MVP package Author: Haohao Zhang Build date: Sep 12, 2018*

Description

MVP.Data.Bfile2MVP: To transform plink binary data to MVP package Author: Haohao Zhang Build date: Sep 12, 2018

Usage

```
MVP.Data.Bfile2MVP(bfile, out = "mvp", maxLine = 10000,
  priority = "speed", type.geno = "char", threads = 0,
  verbose = TRUE)
```

Arguments

bfile	Genotype in binary format (.bed, .bim, .fam)
out	the name of output file
maxLine	the max number of line to write to big matrix for each loop
priority	'memory' or 'speed'
type.geno	the type of genotype elements
threads	number of thread for transforming
verbose	whether to print the reminder

Value

number of individuals and markers. Output files: genotype.desc, genotype.bin: genotype file in bigmemory format phenotype.phe: ordered phenotype file, same taxa order with genotype file map.map: SNP information

Examples

```
bfilePath <- system.file("extdata", "02_bfile", "mvp", package = "rMVP")
MVP.Data.Bfile2MVP(bfilePath)
```

`MVP.Data.Hapmap2MVP` *MVP.Data.Hapmap2MVP: To transform Hapmap data to MVP package Author: Haohao Zhang Build date: Sep 12, 2018*

Description

MVP.Data.Hapmap2MVP: To transform Hapmap data to MVP package Author: Haohao Zhang Build date: Sep 12, 2018

Usage

```
MVP.Data.Hapmap2MVP(hapmap_file, out = "mvp", type.geno = "char",
  verbose = TRUE)
```

Arguments

`hapmap_file`	Genotype in Hapmap format
`out`	the name of output file
`type.geno`	the type of genotype elements
`verbose`	whether to print the reminder

Value

number of individuals and markers. Output files: genotype.desc, genotype.bin: genotype file in bigmemory format phenotype.phe: ordered phenotype file, same taxa order with genotype file map.map: SNP information

Examples

```
hapmapPath <- system.file("extdata", "mvp.hmp", package = "rMVP")
MVP.Data.Hapmap2MVP(hapmapPath)
```

`MVP.Data.impute` *MVP.Data.impute: To impute the missing genotype Author: Haohao Zhang Build date: Sep 12, 2018*

Description

MVP.Data.impute: To impute the missing genotype Author: Haohao Zhang Build date: Sep 12, 2018

Usage

```
MVP.Data.impute(mvp_prefix, out = "mvp.imp", method = "Major",
  ncpus = NULL)
```

Arguments

`mvp_prefix`	the prefix of mvp file
`out`	the prefix of output file
`method`	'Major', 'Minor', "Middle"
`ncpus`	number of threads for imputing

Value

NULL Output files: imputed genotype file

Examples

```
mvpPath <- system.file("extdata", "mvp.geno.desc", package = "rMVP")
MVP.Data.impute(mvpPath)
```

`MVP.Data.Kin` *Kinship*

Description

Kinship

Usage

```
MVP.Data.Kin(fileKin = TRUE, mvp_prefix = "mvp", out = NULL,
  maxLine = 10000, priority = "speed", sep = "\t")
```

Arguments

`fileKin`	Kinship that represents relationship among individuals, n * n matrix, n is sample size
`mvp_prefix`	Prefix for mvp format files
`out`	prefix of output file name
`maxLine`	number of SNPs, only used for saving memory when calculate kinship matrix
`priority`	"speed" or "memory"
`sep`	seperator for Kinship file.

Examples

```
geno <- file.path(system.file(), "06_mvp-impute", "mvp.imp")
MVP.Data.Kin(TRUE, mvp_prefix=geno, out="myKin")
```

`MVP.Data.Map` *MVP.Data.Map: To check map file Author: Haohao Zhang Build date: Sep 12, 2018*

Description

MVP.Data.Map: To check map file Author: Haohao Zhang Build date: Sep 12, 2018

Usage

```
MVP.Data.Map(map, out = "mvp", cols = c(1, 2, 3), header = TRUE,
  sep = "\t")
```

Arguments

`map`	the name of map file or map object(data.frame or matrix)
`out`	the name of output file
`cols`	selected columns
`header`	whether the file contains header
`sep`	seperator of the file

Examples

```
mapPath <- system.file("extdata", "mvp.map", package = "rMVP")
MVP.Data.Map(mvp.map)
```

`MVP.Data.MVP2Bfile` *MVP.Data.MVP2Bfile: To transform MVP data to binary format Author: Haohao Zhang Build date: Sep 12, 2018*

Description

MVP.Data.MVP2Bfile: To transform MVP data to binary format Author: Haohao Zhang Build date: Sep 12, 2018

Usage

```
MVP.Data.MVP2Bfile(bigmat, map, pheno = NULL, out = "mvp.plink",
  verbose = TRUE)
```

Arguments

bigmat	Genotype in bigmatrix format (0,1,2)
map	the map file
pheno	the phenotype file
out	the name of output file
verbose	whether to print the reminder

Value

NULL Output files: .bed, .bim, .fam

Examples

```
bigmat <- as.big.matrix(matrix(1:6, 3, 2))
map <- matrix(c("rs1", "rs2", "rs3", 1, 1, 1, 10, 20, 30), 3, 3)
MVP.Data.MVP2Bfile(bigmat, map)
```

MVP.Data.Numeric2MVP *MVP.Data.Numeric2MVP: To transform Numeric data to MVP package Author: Haohao Zhang Build date: Sep 12, 2018*

Description

MVP.Data.Numeric2MVP: To transform Numeric data to MVP package Author: Haohao Zhang Build date: Sep 12, 2018

Usage

```
MVP.Data.Numeric2MVP(num_file, out = "mvp", maxLine = 10000,
  priority = "speed", row_names = FALSE, col_names = FALSE,
  type.geno = "char", auto_transpose = TRUE, verbose = TRUE)
```

Arguments

num_file	Genotype in Numeric format (0,1,2)
out	the name of output file
maxLine	the max number of line to write to big matrix for each loop
priority	'memory' or 'speed'
row_names	whether the numeric genotype has row names
col_names	whether the numeric genotype has column names
type.geno	the type of genotype elements
auto_transpose	whether to detecte the row and column
verbose	whether to print the reminder

Value

number of individuals and markers. Output files: genotype.desc, genotype.bin: genotype file in bigmemory format phenotype.phe: ordered phenotype file, same taxa order with genotype file map.map: SNP information

Examples

```
numericPath <- system.file("extdata", "mvp.num", package = "rMVP")
MVP.Data.Numeric2MVP(numericPath)
```

`MVP.Data.PC` *Principal component analysis*

Description

Principal component analysis

Usage

```
MVP.Data.PC(filePC = TRUE, mvp_prefix = "mvp", out = NULL,
  perc = 1, pcs.keep = 5, sep = "\t")
```

Arguments

`filePC`	Principal components, n*npc, n is sample size, npc is number of top columns of principal components
`mvp_prefix`	Prefix for mvp format files
`out`	prefix of output file name
`perc`	Percentage of markers used to calculate PCA
`pcs.keep`	how many PCs to keep
`sep`	seperator for PC file.

Examples

```
geno <- file.path(system.file(), "06_mvp-impute", "mvp.imp")
MVP.Data.PC(TRUE, mvp_prefix=geno, out="myPC")
```

`MVP.Data.Pheno` *MVP.Data.Pheno: To clean up phenotype file Author: Haohao Zhang Build date: Sep 12, 2018*

Description

MVP.Data.Pheno: To clean up phenotype file Author: Haohao Zhang Build date: Sep 12, 2018

Usage

```
MVP.Data.Pheno(pheno_file, out = "mvp", cols = NULL, header = TRUE,
  sep = "\t", missing = c(NA, "NA", "-9", 9999))
```

Arguments

pheno_file	the name of phenotype file
out	the name of output file
cols	selected columns
header	whether the file contains header
sep	seperator of the file
missing	the missing value

Value

NULL Output files: cleaned phenotype file

Examples

```
phePath <- system.file("extdata", "mvp.phe", package = "rMVP")
MVP.Data.Pheno(phePath)
```

MVP.Data.QC *MVP.Data.QC: quality control of genotype Author: Haohao Zhang Build date: Sep 12, 2018*

Description

MVP.Data.QC: quality control of genotype Author: Haohao Zhang Build date: Sep 12, 2018

Usage

```
MVP.Data.QC(mvp_prefix, out = NULL, geno = 0.1, mind = 0.1,
  maf = 0.05, hwe = NULL, ncpus = NULL)
```

Arguments

mvp_prefix	the prefix of genotype file
out	the prefix of output file
geno	the threshold of calling rate of markers
mind	the threshold of calling rate of individuals
maf	the threshold of minor allel frequency
hwe	the threshold of hwe
ncpus	the number of threads for quality control

Value

NULL Output files: cleaned genotype file

Examples

```
geno <- file.path(system.file(), "06_mvp-impute", "mvp.imp")
MVP.Data.QC(geno, out="myKin")
```

`MVP.Data.VCF2MVP` *MVP.Data.VCF2MVP: To transform vcf data to MVP package Author: Haohao Zhang Build date: Sep 12, 2018*

Description

Accept the | or / separated markers, any variant sites that are not 0 or 1 will be considered NA.

Usage

```
MVP.Data.VCF2MVP(vcf_file, out = "mvp", maxLine = 10000,
  type.geno = "char", threads = 1, verbose = TRUE)
```

Arguments

`vcf_file`	Genotype in VCF format
`out`	the name of output file
`maxLine`	the max number of line to write to big matrix for each loop
`type.geno`	the type of genotype elements
`threads`	number of thread for transforming
`verbose`	whether to print the reminder

Value

number of individuals and markers. Output files: genotype.desc, genotype.bin: genotype file in bigmemory format phenotype.phe: ordered phenotype file, same taxa order with genotype file map.map: SNP information

Examples

```
vcfPath <- system.file("extdata", "mvp.vcf", package = "rMVP")
MVP.Data.VCF2MVP(vcfPath)
```

`MVP.EMMA.Vg.Ve` *Estimate variance components using EMMA*

Description

Build date: August 30, 2016 Last update: January 27, 2017

Usage

```
MVP.EMMA.Vg.Ve(y, X, K, ngrids = 100, llim = -10, ulim = 10,
  esp = 1e-10)
```

Arguments

y	phenotype, n * 2
X	covariate matrix, the first column is 1s
K	Kinship matrix
ngrids	parameters for estimating vg and ve
llim	parameters for estimating vg and ve
ulim	parameters for estimating vg and ve
esp	parameters for estimating vg and ve

Value

Output: REML - maximum log likelihood Output: delta - exp(root) Output: ve - residual variance Output: vg - genetic variance

Author(s)

EMMA (Kang et. al. Genetics, 2008), Modified only for speed up by Xiaolei Liu and Lilin Yin

Examples

```
phePath <- system.file("extdata", "mvp.phe", package = "rMVP")
phenotype <- read.table(phePath, header=TRUE)
print(dim(phenotype))
genoPath <- system.file("extdata", "mvp.geno.desc", package = "rMVP")
genotype <- attach.big.matrix(genoPath)
print(dim(genotype))
K <- MVP.K.VanRaden(genotype)
vc <- MVP.EMMA.Vg.Ve(y=phenotype[,2], X=matrix(1, nrow(phenotype)), K=K)
print(vc)
```

MVP.FarmCPU *Perform GWAS using FarmCPU method*

Description

Date build: Febuary 24, 2013 Last update: May 25, 2017 Requirement: Y, GD, and CV should have same taxa order. GD and GM should have the same order on SNPs

Usage

```
MVP.FarmCPU(phe, geno, map, CV = NULL, priority = "speed", P = NULL,
  method.sub = "reward", method.sub.final = "reward",
  method.bin = "EMMA", bin.size = c(5e+05, 5e+06, 5e+07),
  bin.selection = seq(10, 100, 10), memo = "MVP.FarmCPU",
  Prior = NULL, ncpus = 2, bar = TRUE, maxLoop = 10,
  threshold.output = 0.01, converge = 1, iteration.output = FALSE,
  p.threshold = NA, QTN.threshold = NULL, bound = NULL)
```

Arguments

phe	phenotype, n by t matrix, n is sample size, t is number of phenotypes
geno	genotype, m by n matrix, m is marker size, n is sample size. This is Pure Genotype Data Matrix(GD). THERE IS NO COLUMN FOR TAXA.
map	SNP map information, m by 3 matrix, m is marker size, the three columns are SNP_ID, Chr, and Pos
CV	covariates, n by c matrix, n is sample size, c is number of covariates
priority	modes, two options: 'speed' or 'memory'
P	start p values for all SNPs
method.sub	method used in substitution process, five options: 'penalty', 'reward', 'mean', 'median', or 'onsite'
method.sub.final	method used in substitution process, five options: 'penalty', 'reward', 'mean', 'median', or 'onsite'
method.bin	method for selecting the most appropriate bins, two options: 'EMMA' or 'FaSTLMM'
bin.size	bin sizes for all iterations, a vector, the bin size is always from large to small
bin.selection	number of selected bins in each iteration, a vector
memo	a marker on output file name
Prior	prior information, four columns, which are SNP_ID, Chr, Pos, P-value
ncpus	number of threads used for parallele computation
bar	if TRUE, the progress bar will be drawn on the terminal
maxLoop	maximum number of iterations
threshold.output	only the GWAS results with p-values lower than threshold.output will be output
converge	a number, 0 to 1, if selected pseudo QTNs in the last and the second last iterations have a certain probality (the probability is converge) of overlap, the loop will stop
iteration.output	whether to output results of all iterations
p.threshold	if all p values generated in the first iteration are bigger than p.threshold, FarmCPU stops
QTN.threshold	in second and later iterations, only SNPs with lower p-values than QTN.threshold have chances to be selected as pseudo QTNs
bound	maximum number of SNPs selected as pseudo QTNs in each iteration

Value

a m by 4 results matrix, m is marker size, the four columns are SNP_ID, Chr, Pos, and p-value

Author(s)

Xiaolei Liu and Zhiwu Zhang

Examples

```
phePath <- system.file("extdata", "mvp.phe", package = "rMVP")
phenotype <- read.table(phePath, header=TRUE)
print(dim(phenotype))
genoPath <- system.file("extdata", "mvp.geno.desc", package = "rMVP")
genotype <- attach.big.matrix(genoPath)
print(dim(genotype))
mapPath <- system.file("extdata", "mvp.map", package = "rMVP")
map <- read.table("mvp.map" , head = TRUE)
farmcpu <- MVP.FarmCPU(phe=phenotype, geno=genotype, map=map, method.bin="static",
  ncpus=detectCores(logical = FALSE), maxLoop=3, P=NULL, method.sub="reward",
  method.sub.final="reward", bin.size=c(5e5,5e6,5e7), bin.selection=seq(10,100,10),
  Prior=NULL, p.threshold=NA, QTN.threshold=NULL, bound=NULL)
str(farmcpu)
```

MVP.FaSTLMM.LL *Evaluation of the maximum likelihood using FaST-LMM method*

Description

Last update: January 11, 2017

Usage

```
MVP.FaSTLMM.LL(pheno, snp.pool, X0 = NULL, ncpus = 2)
```

Arguments

pheno	a two-column phenotype matrix
snp.pool	matrix for pseudo QTNs
X0	covariates matrix
ncpus	number of threads used for parallel computation

Value

Output: beta - beta effect Output: delta - delta value Output: LL - log-likelihood Output: vg - genetic variance Output: ve - residual variance

Author(s)

Qishan Wang, Feng Tian and Zhiwu Zhang (Modified by Xiaolei Liu)

`MVP.GEMMA.Vg.Ve` *To estimate variance component using HE regression*

Description

Build date: Feb 2, 2017 Last update: Feb 2, 2017

Usage

```
MVP.GEMMA.Vg.Ve(y, X, K, rtol = 1e-06, atol = 1e-08, ctol = 1e-08)
```

Arguments

`y`	phenotype
`X`	genotype
`K`	kinship matrix
`rtol`	parameters for HE regression, no changes is recommended
`atol`	parameters for HE regression, no changes is recommended
`ctol`	parameters for HE regression, no changes is recommended

Value

vg, ve, and delta

Author(s)

Translated from C++(GEMMA, Xiang Zhou) to R by: Haohao Zhang

Examples

```
phePath <- system.file("extdata", "mvp.phe", package = "rMVP")
phenotype <- read.table(phePath, header=TRUE)
print(dim(phenotype))
genoPath <- system.file("extdata", "mvp.geno.desc", package = "rMVP")
genotype <- attach.big.matrix(genoPath)
print(dim(genotype))
K <- MVP.K.VanRaden(genotype)
vc <- MVP.GEMMA.Vg.Ve(y=phenotype[,2], X=matrix(1, nrow(phenotype)), K=K)
print(vc)
```

MVP.GLM *To perform GWAS with GLM and MLM model and get the P value of SNPs*

Description

Build date: Aug 30, 2016 Last update: May 25, 2017

Usage

```
MVP.GLM(phe, geno, CV = NULL, cpu = 2, priority = "speed",
  memo = "MVP.GLM", bar = TRUE)
```

Arguments

phe	phenotype, n * 2 matrix
geno	Genotype in numeric format, pure 0, 1, 2 matrix; m * n, m is marker size, n is population size
CV	Covariance, design matrix(n * x) for the fixed effects
cpu	number of cpus used for parallel computation
priority	'memory' or 'speed'
memo	a marker on output file name
bar	whether to show the progress bar

Value

m * 2 matrix, the first column is the SNP effect, the second column is the P values

Author(s)

Lilin Yin and Xiaolei Liu

Examples

```
phePath <- system.file("extdata", "mvp.phe", package = "rMVP")
phenotype <- read.table(phePath, header=TRUE)
print(dim(phenotype))
genoPath <- system.file("extdata", "mvp.geno.desc", package = "rMVP")
genotype <- attach.big.matrix(genoPath)
print(dim(genotype))
glm <- MVP.GLM(phe=phenotype, geno=genotype)
str(glm)
```

MVP.Hist *Phenotype distribution histogram*

Description

Phenotype distribution histogram

Usage

```
MVP.Hist(phe, col = c("dodgerblue4", "olivedrab4", "violetred",
  "darkgoldenrod1", "purple4"), breakNum = 15, file.type = "pdf",
  dpi = 300)
```

Arguments

phe	phenotype data
col	The color vector of the histogram. If the number of colors is less than break.n, the color will be reused. If the number of colors is greater than break.n, only the previous break.n colors will be used.
breakNum	the number of cells for the histogram. The default value is 15.
file.type	A string or NULL is used to determine the type of output file. Can be "jpg", "pdf", "tiff". If it is NULL, it will use dev.new() to create a new graphics device in the current environment, which may be RStudioGD or the default device of the system.
dpi	The resolution of the image, specifying how many pixels per inch.

Examples

```
phePath <- system.file("extdata", "07_other", "mvp.phe", package = "rMVP")
phe <- read.table(phePath, header=TRUE)
MVP.Hist(phe)
```

MVP.K.VanRaden *Calculate Kinship matrix by VanRaden method*

Description

Build date: Dec 12, 2016 Last update: Dec 12, 2016

Usage

```
MVP.K.VanRaden(M, weight = NULL, priority = c("speed", "memory"),
  memo = NULL, SUM = NULL, maxLine = 1000)
```

Arguments

M	Genotype, m * n, m is marker size, n is population size
weight	vector, the weights for makers
priority	speed or memory
memo	add a character to the name of temporary files
SUM	the scaled value to kinship matrix
maxLine	when the priority is 'memory', users can change this parameter to limit the memory

Value

K, n * n matrix

Examples

```
genoPath <- system.file("extdata", "mvp.geno.desc", package = "rMVP")
genotype <- attach.big.matrix(genoPath)
print(dim(genotype))
K <- MVP.K.VanRaden(genotype)
```

MVP.MLM *To perform GWAS with GLM and MLM model and get the P value of SNPs*

Description

Build date: Aug 30, 2016 Last update: Aug 30, 2016

Usage

```
MVP.MLM(phe, geno, K = NULL, CV = NULL, REML = NULL,
  priority = "speed", cpu = 2, bar = TRUE, vc.method = "EMMA",
  maxLine = 1000, file.output = TRUE, memo = "MVP")
```

Arguments

phe	phenotype, n * 2 matrix
geno	genotype, m * n, m is marker size, n is population size
K	Kinship, Covariance matrix(n * n) for random effects; must be positive semi-definite
CV	covariates
REML	a list that contains ve and vg
priority	speed or memory
cpu	number of cpus used for parallel computation
bar	whether to show the progress bar
vc.method	the methods for estimating variance component("emma" or "gemma")
maxLine	when the priority is 'memory', users can change this parameter to limit the memory
file.output	whether to output files or not
memo	a marker on output file name

Value

results: a m * 2 matrix, the first column is the SNP effect, the second column is the P values

Author(s)

Lilin Yin and Xiaolei Liu

Examples

```
phePath <- system.file("extdata", "mvp.phe", package = "rMVP")
phenotype <- read.table(phePath, header=TRUE)
print(dim(phenotype))
genoPath <- system.file("extdata", "mvp.geno.desc", package = "rMVP")
genotype <- attach.big.matrix(genoPath)
print(dim(genotype))
K <- MVP.K.VanRaden(genotype)
mlm <- MVP.MLM(phe=phenotype, geno=genotype, K=K)
str(mlm)
```

MVP.PCA *Principal Component Analysis*

Description

Build date: Dec 14, 2016 Last update: Oct 29, 2018

Usage

```
MVP.PCA(M, perc = 1, pcs.keep = 5, memo = NULL)
```

Arguments

M	Genotype in numeric format, pure 0, 1, 2 matrix; m * n, m is marker size, n is population size
perc	percentage of total SNPs selected for calculate principal components
pcs.keep	maximum number of PCs for output
memo	a marker on output file name

Value

Output: PCs - a n * npc matrix of top number of PCs, n is population size and npc is @param pcs.keep

Author(s)

Xiaolei Liu, Lilin Yin and Haohao Zhang

Examples

```
genoPath <- system.file("extdata", "mvp.geno.desc", package = "rMVP")
genotype <- attach.big.matrix(genoPath)
print(dim(genotype))
pca <- MVP.PCA(M=genotype)
str(pca)
```

MVP.PCAplot *PCA Plot*

Description

PCA Plot

Usage

```
MVP.PCAplot(PCA, col = NULL, pch = NULL, class = NULL,
  legend.pos = "topright", Ncluster = 3, plot3D = TRUE,
  file = "pdf", dpi = 300, box = FALSE)
```

Arguments

PCA	Principal component analysis result, 2-column matrix
col	colors for each cluster
pch	Either an integer specifying a symbol or a single character to be used as the default in plotting points. See points for possible values and their interpretation. Note that only integers and single-character strings can be set as a graphics parameter (and not NA nor NULL).
class	the class of all individuals, for example: "breed", "location"...
legend.pos	position of legend. default is "topright"
Ncluster	cluster number
plot3D	(DEPRECATED)if TRUE, plot PC figure in 3D format, it can be only used in windows and mac operation system, "rgl" package should be installed beforehead
file	Character. Options are jpg, pdf, and tiff
dpi	Number. Dots per inch for .jpg and .tiff files
box	Logical value. If TRUE, the border line of Manhattan plot will be added

Examples

```
PCPath <- system.file("extdata", "07_other", "mvp.imp.pc.desc", package = "rMVP")
pca <- attach.big.matrix(PCPath)[, 1:3]
MVP.PCAplot(PCA=pca, Ncluster=3, class=NULL, col=c("red", "green", "yellow"), file="jpg", pch=19)
```

MVP.Report *MVP.Report*

Description

MVP.Report

Usage

```
MVP.Report(MVP, col = c("#377EB8", "#4DAF4A", "#984EA3", "#FF7F00"),
  bin.size = 1e+06, bin.max = NULL, pch = 19, band = 1,
  cir.band = 0.5, H = 1.5, ylim = NULL, cex.axis = 1,
  plot.type = "b", multracks = FALSE, cex = c(0.5, 0.8, 1),
  r = 0.3, xlab = "Chromosome",
  ylab = expression(-log[10](italic(p))), xaxs = "i", yaxs = "r",
  outward = FALSE, threshold = NULL, threshold.col = "red",
  threshold.lwd = 1, threshold.lty = 2, amplify = TRUE,
  chr.labels = NULL, signal.cex = 0.8, signal.pch = 19,
  signal.col = "red", signal.line = NULL, cir.chr = TRUE,
  cir.chr.h = 1.3, chr.den.col = c("darkgreen", "yellow", "red"),
  cir.legend = TRUE, cir.legend.cex = 0.8, cir.legend.col = "grey45",
  LOG10 = TRUE, box = FALSE, conf.int.col = "grey",
  file.output = TRUE, file = "jpg", dpi = 300, memo = "")
```

Arguments

MVP	Data frame. Includes at least four columns. The first three columns are Marker ID, Chromosome ID, and Physical position. The fourth and following optional columns are P-values or effects of markers
col	Vector or matrix. If 'col' is a vector, multiple-group GWAS results will use the same color scheme for plotting points on different Chromosomes. The vector length can be shorter than the number of Chromosomes and the colors will be used circularly. If 'col' is a matrix, multiple-group GWAS results will be drawn with colors from different rows, NA is allowed in the color matrix, e.g. col = matrix(c("grey30", "grey60", NA, "red", "blue", "green", "orange", NA, NA), 3, 3, byrow=T)
bin.size	Number. Number of markers will be counted for each marker window and used for plotting marker density
bin.max	Number. The maximum marker density value used for plotting marker density. Windows with marker density higher than 'bin.max' will use the same color as 'bin.max'
pch	Number. Type of points, same as 'pch' in <plot> R function
band	Number. The space among chromosomes.
cir.band	Number. The space between circles when plotting Manhattan plot in circular manner
H	Number. The height for each circle when plotting multiple-group GWAS results using Manhattan plot in circular manner
ylim	Vector. The range of Y-axis when plotting Manhattan plot, same as "ylim" in <plot> R function
cex.axis	Number. The size of Chromosome ID and labels when plotting Manhattan plot in circular manner
plot.type	Character or Vector, options are "d", "c", "m", "q", and "b". If plot.type="d", marker density will be plotted; if plot.type="c", Manhattan plot in circular manner will be plotted; if plot.type="m", Manhattan plot in rectangular manner will be plotted; if plot.type="q", Q-Q plot will be plotted; if plot.type="b", both Manhattan plot and Q-Q plot will be plotted

multracks	Logical value. If FALSE, multiple-group GWAS results will be plotted on multiple tracks; if TRUE, multiple-group GWAS results will be plotted on a single track
cex	Number or Vector. The size of points. It is the same as "size" in <plot> R function. If given as a vector, the numbers are used to control the point size on Manhattan plot in circular manner, Manhattan plot in rectangular manner, and Q-Q plot, respectively
r	Number. The radius of the inside circle when plotting Manhattan plot in circular manner
xlab	Character. The label of X axis
ylab	Character. The label of Y axis
xaxs	Character. Options are "r", and "i". It is the same as "xaxs" in <plot> R function
yaxs	Character. Options are "r", and "i". It is the same as "yaxs" in <plot> R function
outward	Logical value. If TRUE, all points will be plotted from inside toward outside; otherwise, all points will be plotted from outside toward inside
threshold	Number or Vector. The cutoff line on Manhattan plot, e.g. Bonfferoni correction. More than one significant line can be added onto one figure. If threshold=0 or NULL, the threshold line will not be added
threshold.col	Character or Vector. The colors of threshold lines
threshold.lwd	Number or Vector. The widths of threshold lines
threshold.lty	Number or Vector. The type of threshold line
amplify	Logical value. If TRUE, the points that passed the threshold line will be highlighted
chr.labels	Vector. The labels for the Chromosome IDs on Manhattan plot in circular manner
signal.cex	Number. If "amplify" is TRUE, "signal.cex" is used to set the size of significant points
signal.pch	Number. If "amplify" is TRUE, users can set the type of significant points
signal.col	Character. If "amplify" is TRUE, "signal.col" is used to set the color of significant points, if "signal.col" is NULL, the colors of significant points will not be changed
signal.line	Number. The width of dotted lines used for marking significant points
cir.chr	Logical value. If TRUE, a band that represents marker density information will be added onto Manhattan plot in circular manner
cir.chr.h	Number. If "cir.chr=TRUE", it can be used to set the width of marker density band
chr.den.col	Character or Vector or NULL. The colors for plotting marker density band on Manhattan plot. If "chr.den.col=NULL", it will use the colors in parameter 'col'
cir.legend	Logical value. If TRUE, legends will be added on each circle of Manhattan plot in circular manner
cir.legend.cex	Number. The size of legend on Manhattan plot in circular manner
cir.legend.col	Character. The color of legends on Manhattan plot in circular manner
LOG10	Logical value. If TRUE, the p values of GWAS results will be scaled by log10
box	Logical value. If TRUE, the border line of Manhattan plot will be added

conf.int.col	Character. The color of confidence interval on QQ-plot
file.output	Logical value. If TRUE, the figures will be generated.
file	Character. Options are jpg, pdf, and tiff
dpi	Number. Dots per inch for .jpg and .tiff files
memo	Character. A text marker on output files

Examples

```
data(pig60K, package = "rMVP")
#MVP.Report(pig60K[,c(1:3, 5)], plot.type="m", threshold=0.05/nrow(pig60K))
#MVP.Report(pig60K, plot.type="c", threshold=0.05/nrow(pig60K))
```

MVP.Report.Density *SNP Density*

Description

SNP Density

Usage

```
MVP.Report.Density(Pmap, taxa, col = c("darkgreen", "yellow", "red"),
  dpi = 300, bin.size = 1e+06, bin.max = NULL, file.type = "jpg",
  file.output = TRUE)
```

Arguments

Pmap	P value Map
taxa	The identifier of the output file.
col	The color vector
dpi	Number. Dots per inch for .jpg and .tiff files
bin.size	the window size for counting SNP number
bin.max	maximum SNP number, for winows, which has more SNPs than bin.max, will be painted in same color
file.type	format of output figure
file.output	Whether to output the file

Examples

```
data(pig60K, package = "rMVP")
MVP.Report.Density(pig60K, "mvp")
```

`MVP.Report.QQplot` *QQ Plot*

Description

QQ Plot

Usage

```
MVP.Report.QQplot(P.values, taxa_name, col = c("blue"), cex = 0.5,
  threshold = NULL, amplify = TRUE, signal.col = "red",
  signal.pch = 19, signal.cex = 0.8, conf.int = TRUE, cex.axis = 1,
  conf.int.col = "grey", threshold.col = "red", file.type = "jpg",
  memo = "MVP", box = TRUE, dpi = 300)
```

Arguments

`P.values`	P values
`taxa_name`	The identifier of the phenotype will be used to generate a portion of the image file name. If the title parameter is NULL, it will also be part of the title.
`col`	default color is "blue"
`cex`	A numerical value giving the amount by which plotting text and symbols should be magnified relative to the default. This starts as 1 when a device is opened, and is reset when the layout is changed, e.g. by setting mfrow. see `par`.
`threshold`	Number or Vector. The cutoff line on Manhattan plot, e.g. Bonfferoni correction. More than one significant line can be added onto one figure. If threshold=0 or NULL, the threshold line will not be added.
`amplify`	Logical value. If TRUE, the points that passed the threshold line will be highlighted
`signal.col`	Character. If "amplify" is TRUE, "signal.col" is used to set the color of significant points, if "signal.col" is NULL, the colors of significant points will not be changed
`signal.pch`	Number. If "amplify" is TRUE, users can set the type of significant points
`signal.cex`	Number. If "amplify" is TRUE, "signal.cex" is used to set the size of significant points
`conf.int`	Whether to draw a confidence interval
`cex.axis`	a number, controls the size of numbers of X-axis and the size of labels of circle plot.
`conf.int.col`	a character, the color of the confidence interval on QQ-plot.
`threshold.col`	Character or Vector. The colors of threshold lines
`file.type`	A string or NULL is used to determine the type of output file. Can be "jpg", "pdf", "tiff". If it is NULL, it will use `dev.new()` to create a new graphics device in the current environment, which may be RStudioGD or the default device of the system.
`memo`	the prefix of the output image file.
`box`	A Boolean value that controls whether to draw a box around QQplot.
`dpi`	a number, the picture element for .jpg and .tiff files. The default is 300.

Examples

```
data(pig60K, package = "rMVP")
MVP.Report(pig60K,plot.type="q",conf.int.col=NULL,box=TRUE,file="jpg",memo="",dpi=300)
```

MVP.Version *Print MVP Banner*

Description

Build date: Aug 30, 2017 Last update: Dec 12, 2018

Usage

```
MVP.Version(width = 60)
```

Arguments

width	the width of the message

Author(s)

Lilin Yin, Haohao Zhang, and Xiaolei Liu

Examples

```
MVP.version()
```

---

pig60K *Genotyped by pig 60k chip*

Description

This dataset gives the results of Genome-wide association study of 3 traits, individuals were genotyped by pig 60K chip.

Usage

```
data(pig60K)
```

Format

A dataframe containing 3 traits' Pvalue

Index