

# OpenCLonalg

0.0.1

Generated by Doxygen 1.8.6

Tue Jun 7 2016 14:55:12



# Contents

<b>1</b>	<b>Hierarchical Index</b>	<b>1</b>
1.1	Class Hierarchy . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List . . . . .	5
<b>4</b>	<b>Class Documentation</b>	<b>7</b>
4.1	BaseClonalg Class Reference . . . . .	7
4.1.1	Constructor & Destructor Documentation . . . . .	8
4.1.1.1	BaseClonalg . . . . .	8
4.1.1.2	~BaseClonalg . . . . .	8
4.1.2	Member Function Documentation . . . . .	8
4.1.2.1	BinaryToDecimal . . . . .	8
4.1.2.2	CompareIndividuals . . . . .	8
4.1.2.3	GetWord . . . . .	8
4.1.2.4	HipermutationRate . . . . .	8
4.1.2.5	PrintPop . . . . .	8
4.1.2.6	Search . . . . .	8
4.1.2.7	Sort . . . . .	8
4.1.3	Member Data Documentation . . . . .	8
4.1.3.1	m_bitsperdimension . . . . .	8
4.1.3.2	m_bitsperparameter . . . . .	8
4.1.3.3	m_cloningfactor . . . . .	8
4.1.3.4	m_cromLen . . . . .	8
4.1.3.5	m_dimensions . . . . .	8
4.1.3.6	m_fitness . . . . .	9
4.1.3.7	m_fitnessNorm . . . . .	9
4.1.3.8	m_generations . . . . .	9
4.1.3.9	m_lowerlim . . . . .	9

4.1.3.10	<a href="#">m_mutationfactor</a>	9
4.1.3.11	<a href="#">m_numclones</a>	9
4.1.3.12	<a href="#">m_objective</a>	9
4.1.3.13	<a href="#">m_optimizationProblem</a>	9
4.1.3.14	<a href="#">m_pop</a>	9
4.1.3.15	<a href="#">m_popsiz</a>	9
4.1.3.16	<a href="#">m_randominsertion</a>	9
4.1.3.17	<a href="#">m_realLen</a>	9
4.1.3.18	<a href="#">m_statistics</a>	9
4.1.3.19	<a href="#">m_upperlim</a>	9
4.2	<a href="#">Clonalg Class Reference</a>	9
4.2.1	<a href="#">Constructor &amp; Destructor Documentation</a>	10
4.2.1.1	<a href="#">Clonalg</a>	10
4.2.1.2	<a href="#">~Clonalg</a>	10
4.2.2	<a href="#">Member Function Documentation</a>	10
4.2.2.1	<a href="#">CalculateAffinity</a>	10
4.2.2.2	<a href="#">CloneAndHypermutate</a>	10
4.2.2.3	<a href="#">Decode</a>	10
4.2.2.4	<a href="#">Evaluate</a>	10
4.2.2.5	<a href="#">EvaluatePop</a>	10
4.2.2.6	<a href="#">FindBestAndWorst</a>	10
4.2.2.7	<a href="#">InitPopulation</a>	10
4.2.2.8	<a href="#">Mutate</a>	10
4.2.2.9	<a href="#">RandomInsertion</a>	10
4.2.2.10	<a href="#">Search</a>	10
4.2.2.11	<a href="#">Statistics</a>	11
4.2.3	<a href="#">Member Data Documentation</a>	11
4.2.3.1	<a href="#">m_aux_binary</a>	11
4.2.3.2	<a href="#">m_bestClone</a>	11
4.2.3.3	<a href="#">m_clone</a>	11
4.2.3.4	<a href="#">m_v</a>	11
4.3	<a href="#">ClonalgCL Class Reference</a>	11
4.3.1	<a href="#">Detailed Description</a>	12
4.3.2	<a href="#">Constructor &amp; Destructor Documentation</a>	13
4.3.2.1	<a href="#">ClonalgCL</a>	13
4.3.2.2	<a href="#">~ClonalgCL</a>	13
4.3.3	<a href="#">Member Function Documentation</a>	13
4.3.3.1	<a href="#">CloneAndHypermutate</a>	13
4.3.3.2	<a href="#">EvaluatePop</a>	13
4.3.3.3	<a href="#">FindBestAndWorst</a>	13

4.3.3.4	InitPopulation	13
4.3.3.5	LoadParameters	13
4.3.3.6	LoadSeeds	13
4.3.3.7	NormalizeAffinity	13
4.3.3.8	OpenCLInit	14
4.3.3.9	RandomInsertion	14
4.3.3.10	Search	14
4.3.3.11	Statistics	14
4.3.4	Member Data Documentation	14
4.3.4.1	affinityBuffer	14
4.3.4.2	affinityNormBuffer	14
4.3.4.3	clUtils	14
4.3.4.4	evaluationKernel	15
4.3.4.5	hipermutationKernel	15
4.3.4.6	host_Seeds	15
4.3.4.7	initKernel	15
4.3.4.8	m_cpu_count	15
4.3.4.9	m_cpu_queues	15
4.3.4.10	m_gpu_count	15
4.3.4.11	m_gpu_queues	15
4.3.4.12	m_gpu_ratio	15
4.3.4.13	m_pop_size_per_queue	15
4.3.4.14	m_stats	15
4.3.4.15	m_workGroupSize_hipermutation	15
4.3.4.16	normalizeAffinityKernel	16
4.3.4.17	parametersBuffer	16
4.3.4.18	popBuffer	16
4.3.4.19	randomInsertionKernel	16
4.3.4.20	seedBuffer	16
4.3.4.21	statisticsBuffer	16
4.3.4.22	statisticsKernel	16
4.4	ElipsoidalObjectiveFunction Class Reference	16
4.4.1	Constructor & Destructor Documentation	17
4.4.1.1	ElipsoidalObjectiveFunction	17
4.4.1.2	~ElipsoidalObjectiveFunction	17
4.4.2	Member Function Documentation	17
4.4.2.1	Evaluate	17
4.4.2.2	EvaluateFloat	17
4.4.2.3	getFormula	17
4.4.2.4	getName	17

4.4.2.5	getSumLimit	17
4.5	KernelParameters Struct Reference	17
4.5.1	Member Data Documentation	18
4.5.1.1	BITS_PER_DIMENSION	18
4.5.1.2	CHROM_LEN	18
4.5.1.3	CLONING_FACTOR	18
4.5.1.4	DIMENSIONS	18
4.5.1.5	LOWER_LIM	18
4.5.1.6	MUTATION_FACTOR	18
4.5.1.7	NCLON	18
4.5.1.8	POP_SIZE	18
4.5.1.9	REAL_LEN	18
4.5.1.10	UPPER_LIM	18
4.6	ObjectiveFunction Class Reference	18
4.6.1	Constructor & Destructor Documentation	19
4.6.1.1	ObjectiveFunction	19
4.6.1.2	~ObjectiveFunction	19
4.6.2	Member Function Documentation	19
4.6.2.1	Evaluate	19
4.6.2.2	EvaluateFloat	19
4.6.2.3	getFormula	19
4.6.2.4	getName	19
4.6.2.5	getSumLimit	19
4.6.3	Member Data Documentation	19
4.6.3.1	m_vectorLength	19
4.7	OneMaxProblem Class Reference	19
4.7.1	Constructor & Destructor Documentation	20
4.7.1.1	OneMaxProblem	20
4.7.1.2	~OneMaxProblem	20
4.7.2	Member Function Documentation	20
4.7.2.1	Evaluate	20
4.7.2.2	EvaluateFloat	20
4.7.2.3	getFormula	20
4.7.2.4	getName	20
4.7.2.5	getSumLimit	20
4.8	OpenCLUtills Class Reference	20
4.8.1	Detailed Description	21
4.8.2	Constructor & Destructor Documentation	21
4.8.2.1	OpenCLUtills	21
4.8.2.2	~OpenCLUtills	21

4.8.3	Member Function Documentation	21
4.8.3.1	CreateProgramFromBinary	21
4.8.3.2	CreateProgramFromSource	21
4.8.3.3	GetBinarySize	21
4.8.3.4	getElapsedTime	21
4.8.3.5	SaveProgramBinary	21
4.8.3.6	ShowDeviceInfo	21
4.9	ProblemFactory Class Reference	21
4.9.1	Constructor & Destructor Documentation	22
4.9.1.1	ProblemFactory	22
4.9.1.2	~ProblemFactory	22
4.9.2	Member Function Documentation	22
4.9.2.1	CreateProblem	22
4.10	RosenbrockObjectiveFunction Class Reference	22
4.10.1	Constructor & Destructor Documentation	22
4.10.1.1	RosenbrockObjectiveFunction	22
4.10.1.2	~RosenbrockObjectiveFunction	22
4.10.2	Member Function Documentation	22
4.10.2.1	Evaluate	23
4.10.2.2	EvaluateFloat	23
4.10.2.3	getFormula	23
4.10.2.4	getName	23
4.10.2.5	getSumLimit	23
4.11	t_parameters Struct Reference	23
4.11.1	Member Data Documentation	23
4.11.1.1	bitsPerDimension	23
4.11.1.2	cpus	23
4.11.1.3	generations	23
4.11.1.4	gpuRatio	24
4.11.1.5	gpus	24
4.11.1.6	lowerLim	24
4.11.1.7	mutationFactor	24
4.11.1.8	nclones	24
4.11.1.9	optimizationProblem	24
4.11.1.10	parallel	24
4.11.1.11	popsiz	24
4.11.1.12	problemDimension	24
4.11.1.13	randomInsertion	24
4.11.1.14	upperLim	24
4.12	t_stats Struct Reference	24

4.12.1	Member Data Documentation	24
4.12.1.1	affinityBest	24
4.12.1.2	affinityWorst	24
4.12.1.3	indexBest	24
4.12.1.4	indexWorst	24
4.13	TimeUtils Class Reference	25
4.13.1	Member Function Documentation	25
4.13.1.1	getRealTime	25
<b>5</b>	<b>File Documentation</b>	<b>27</b>
5.1	src/BaseClonalg.cpp File Reference	27
5.2	src/BaseClonalg.h File Reference	27
5.3	src/Clonalg.cpp File Reference	27
5.4	src/Clonalg.h File Reference	27
5.5	src/ClonalgCL.cpp File Reference	28
5.6	src/ClonalgCL.h File Reference	28
5.6.1	Macro Definition Documentation	28
5.6.1.1	__CL_ENABLE_EXCEPTIONS	28
5.6.1.2	__NO_STD_VECTOR	28
5.7	src/ElipsoidalObjectiveFunction.cpp File Reference	28
5.8	src/ElipsoidalObjectiveFunction.h File Reference	28
5.9	src/kernels/kernel.cpp File Reference	29
5.9.1	Function Documentation	29
5.9.1.1	BinaryToDecimal	29
5.9.1.2	cloneAndHypermutation	29
5.9.1.3	EvaluateIndividual_local	29
5.9.1.4	Evaluation	29
5.9.1.5	initPopulation	29
5.9.1.6	NormalizeAffinity	29
5.9.1.7	randomInsertion	29
5.9.1.8	StatisticsReduction1	29
5.10	src/kernels/utls.cpp File Reference	30
5.10.1	Macro Definition Documentation	30
5.10.1.1	BIT_CHECK	30
5.10.1.2	BIT_CLEAR	30
5.10.1.3	BIT_FLIP	30
5.10.1.4	BIT_SET	30
5.10.1.5	CONST1	30
5.10.1.6	CONST2	30
5.10.1.7	MASK	30



5.10.2	Function Documentation	30
5.10.2.1	rand	30
5.10.2.2	u_rand	30
5.11	src/main.cpp File Reference	30
5.11.1	Function Documentation	31
5.11.1.1	GetArgs	31
5.11.1.2	main	31
5.12	src/ObjectiveFunction.cpp File Reference	31
5.13	src/ObjectiveFunction.h File Reference	31
5.14	src/OneMaxProblem.cpp File Reference	31
5.15	src/OneMaxProblem.h File Reference	31
5.16	src/opencLUtills.c File Reference	31
5.16.1	Function Documentation	32
5.16.1.1	clCompilaPrograma	32
5.16.1.2	clObtemDispositivos	32
5.16.1.3	clObtemPlataformas	32
5.17	src/OpenCLUtills.cpp File Reference	32
5.17.1	Macro Definition Documentation	32
5.17.1.1	__NO_STD_VECTOR	32
5.18	src/opencLUtills.h File Reference	32
5.18.1	Macro Definition Documentation	33
5.18.1.1	check_cl	33
5.18.1.2	log_arquivo	33
5.18.1.3	log_error	33
5.18.1.4	log_error_code	33
5.18.2	Function Documentation	33
5.18.2.1	clCompilaPrograma	33
5.18.2.2	clObtemDispositivos	33
5.18.2.3	clObtemPlataformas	33
5.19	src/OpenCLUtills.h File Reference	33
5.19.1	Macro Definition Documentation	33
5.19.1.1	__CL_ENABLE_EXCEPTIONS	33
5.19.1.2	__NO_STD_VECTOR	33
5.20	src/parameters.h File Reference	33
5.20.1	Macro Definition Documentation	33
5.20.1.1	BITS_PER_WORD	34
5.21	src/ProblemFactory.cpp File Reference	34
5.22	src/ProblemFactory.h File Reference	34
5.23	src/representation.h File Reference	34
5.23.1	Macro Definition Documentation	34

---

5.23.1.1	<a href="#">BIT_CHECK</a>	34
5.23.1.2	<a href="#">BIT_CLEAR</a>	34
5.23.1.3	<a href="#">BIT_FLIP</a>	34
5.23.1.4	<a href="#">BIT_SET</a>	34
5.23.2	<a href="#">Typedef Documentation</a>	35
5.23.2.1	<a href="#">KernelParameters</a>	35
5.24	<a href="#">src/RosenbrockObjectiveFunction.cpp File Reference</a>	35
5.25	<a href="#">src/RosenbrockObjectiveFunction.h File Reference</a>	35
5.26	<a href="#">src/TimeUtils.h File Reference</a>	35
<b>Index</b>		<b>36</b>

# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

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

BaseClonalg . . . . .	7
Clonalg . . . . .	9
ClonalgCL . . . . .	11
KernelParameters . . . . .	17
ObjectiveFunction . . . . .	18
ElipsoidalObjectiveFunction . . . . .	16
OneMaxProblem . . . . .	19
RosenbrockObjectiveFunction . . . . .	22
OpenCLUtills . . . . .	20
ProblemFactory . . . . .	21
t_parameters . . . . .	23
t_stats . . . . .	24
TimeUtills . . . . .	25



## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">BaseClonalg</a>	7
<a href="#">Clonalg</a>	9
<a href="#">ClonalgCL</a>	
Class for executing CLONALG algorithm in OpenCL devices	11
<a href="#">ElipsoidalObjectiveFunction</a>	16
<a href="#">KernelParameters</a>	17
<a href="#">ObjectiveFunction</a>	18
<a href="#">OneMaxProblem</a>	19
<a href="#">OpenCLUtills</a>	
Class for handling OpenCL common operations	20
<a href="#">ProblemFactory</a>	21
<a href="#">RosenbrockObjectiveFunction</a>	22
<a href="#">t_parameters</a>	23
<a href="#">t_stats</a>	24
<a href="#">TimeUtills</a>	25



## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

src/ <a href="#">BaseClonalg.cpp</a>	27
src/ <a href="#">BaseClonalg.h</a>	27
src/ <a href="#">Clonalg.cpp</a>	27
src/ <a href="#">Clonalg.h</a>	27
src/ <a href="#">ClonalgCL.cpp</a>	28
src/ <a href="#">ClonalgCL.h</a>	28
src/ <a href="#">ElipsoidalObjectiveFunction.cpp</a>	28
src/ <a href="#">ElipsoidalObjectiveFunction.h</a>	28
src/ <a href="#">main.cpp</a>	30
src/ <a href="#">ObjectiveFunction.cpp</a>	31
src/ <a href="#">ObjectiveFunction.h</a>	31
src/ <a href="#">OneMaxProblem.cpp</a>	31
src/ <a href="#">OneMaxProblem.h</a>	31
src/ <a href="#">openclUtils.c</a>	31
src/ <a href="#">OpenCLUtils.cpp</a>	32
src/ <a href="#">OpenCLUtils.h</a>	33
src/ <a href="#">openclUtils.h</a>	32
src/ <a href="#">parameters.h</a>	33
src/ <a href="#">ProblemFactory.cpp</a>	34
src/ <a href="#">ProblemFactory.h</a>	34
src/ <a href="#">representation.h</a>	34
src/ <a href="#">RosenbrockObjectiveFunction.cpp</a>	35
src/ <a href="#">RosenbrockObjectiveFunction.h</a>	35
src/ <a href="#">TimeUtils.h</a>	35
src/kernels/ <a href="#">kernel.cpp</a>	29
src/kernels/ <a href="#">utils.cpp</a>	30





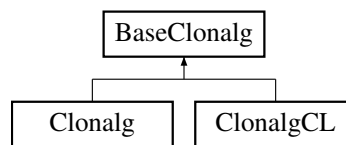
## Chapter 4

# Class Documentation

### 4.1 BaseClonalg Class Reference

```
#include <BaseClonalg.h>
```

Inheritance diagram for BaseClonalg:



#### Public Member Functions

- `BaseClonalg` (int generations, int popsize, int optimizationProblem, int dimensions, int bitsperdimension, float mutationfactor, float cloningfactor, int numclones, int randominsertion, float upperlim, float lowerlim)
- virtual float `Search` ()=0
- virtual `~BaseClonalg` ()

#### Protected Member Functions

- unsigned int `GetWord` (unsigned int \*pop, int id, int j)
- float `HipermutationRate` (float affinity)
- void `Sort` (unsigned int \*pop)
- int `BinaryToDecimal` (int \*binary, int begin, int end)
- void `PrintPop` ()

#### Static Protected Member Functions

- static int `CompareIndividuals` (const void \*a, const void \*b)

#### Protected Attributes

- int `m_generations`
- int `m_popsize`
- int `m_optimizationProblem`
- int `m_dimensions`

- int [m\\_bitsperdimension](#)
- int [m\\_bitsperparameter](#)
- float [m\\_mutationfactor](#)
- float [m\\_cloningfactor](#)
- int [m\\_numclones](#)
- int [m\\_randominsertion](#)
- float [m\\_upperlim](#)
- float [m\\_lowerlim](#)
- int [m\\_cromLen](#)
- int [m\\_realLen](#)
- unsigned \* [m\\_pop](#)
- float \* [m\\_fitness](#)
- float \* [m\\_fitnessNorm](#)
- [t\\_stats](#) [m\\_statistics](#)
- [ObjectiveFunction](#) \* [m\\_objective](#)

#### 4.1.1 Constructor & Destructor Documentation

4.1.1.1 **BaseClonalg::BaseClonalg ( int *generations*, int *popsiz*e, int *optimizationProblem*, int *dimensions*, int *bitsperdimension*, float *mutationfactor*, float *cloningfactor*, int *numclones*, int *randominsertion*, float *upperlim*, float *lowerlim* )**

4.1.1.2 **BaseClonalg::~~BaseClonalg ( )** [virtual]

#### 4.1.2 Member Function Documentation

4.1.2.1 **int BaseClonalg::BinaryToDecimal ( int \* *binary*, int *begin*, int *end* )** [protected]

4.1.2.2 **static int BaseClonalg::CompareIndividuals ( const void \* *a*, const void \* *b* )** [static], [protected]

4.1.2.3 **unsigned int BaseClonalg::GetWord ( unsigned int \* *pop*, int *id*, int *j* )** [inline], [protected]

4.1.2.4 **float BaseClonalg::HipermutationRate ( float *affinity* )** [protected]

4.1.2.5 **void BaseClonalg::PrintPop ( )** [protected]

4.1.2.6 **virtual float BaseClonalg::Search ( )** [pure virtual]

Implemented in [ClonalgCL](#), and [Clonalg](#).

4.1.2.7 **void BaseClonalg::Sort ( unsigned int \* *pop* )** [protected]

#### 4.1.3 Member Data Documentation

4.1.3.1 **int BaseClonalg::m\_bitsperdimension** [protected]

4.1.3.2 **int BaseClonalg::m\_bitsperparameter** [protected]

4.1.3.3 **float BaseClonalg::m\_cloningfactor** [protected]

4.1.3.4 **int BaseClonalg::m\_cromLen** [protected]

4.1.3.5 **int BaseClonalg::m\_dimensions** [protected]

- 4.1.3.6 `float* BaseClonalg::m_fitness` [protected]
- 4.1.3.7 `float* BaseClonalg::m_fitnessNorm` [protected]
- 4.1.3.8 `int BaseClonalg::m_generations` [protected]
- 4.1.3.9 `float BaseClonalg::m_lowerlim` [protected]
- 4.1.3.10 `float BaseClonalg::m_mutationfactor` [protected]
- 4.1.3.11 `int BaseClonalg::m_numclones` [protected]
- 4.1.3.12 `ObjectiveFunction* BaseClonalg::m_objective` [protected]
- 4.1.3.13 `int BaseClonalg::m_optimizationProblem` [protected]
- 4.1.3.14 `unsigned* BaseClonalg::m_pop` [protected]
- 4.1.3.15 `int BaseClonalg::m_popsiz` [protected]
- 4.1.3.16 `int BaseClonalg::m_randominsertion` [protected]
- 4.1.3.17 `int BaseClonalg::m_realLen` [protected]
- 4.1.3.18 `t_stats BaseClonalg::m_statistics` [protected]
- 4.1.3.19 `float BaseClonalg::m_upperlim` [protected]

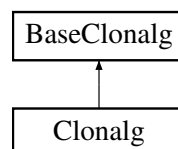
The documentation for this class was generated from the following files:

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

## 4.2 Clonalg Class Reference

```
#include <Clonalg.h>
```

Inheritance diagram for Clonalg:



### Public Member Functions

- [Clonalg](#) (int generations, int popsize, int optimizationProblem, int dimensions, int bitsperdimension, float mutationfactor, float cloningfactor, int numclones, int randominsertion, float upperlim, float lowerlim)
- float [Search](#) ()
- virtual [~Clonalg](#) ()

## Protected Member Functions

- void [InitPopulation](#) (unsigned \*\*pop, float \*\*fitness, float \*\*fitnessNorm)
- float [Evaluate](#) (unsigned \*individual)
- void [EvaluatePop](#) (unsigned \*pop)
- void [Mutate](#) (unsigned \*clone, float mutationRate)
- void [CloneAndHypermutate](#) (unsigned \*pop, float \*fitness, float \*fitnessNorm)
- void [RandomInsertion](#) (unsigned \*pop, float \*fitness)
- void [CalculateAffinity](#) (unsigned \*pop, float \*fitness, float \*fitnessNorm)
- void [FindBestAndWorst](#) ()
- void [Statistics](#) (unsigned \*pop, float \*fitness, int iterationNumber)
- void [Decode](#) (unsigned int \*pop, float \*v)

## Private Attributes

- unsigned \* [m\\_clone](#)
- unsigned \* [m\\_bestClone](#)
- float \* [m\\_v](#)
- int \* [m\\_aux\\_binary](#)

## Additional Inherited Members

### 4.2.1 Constructor & Destructor Documentation

4.2.1.1 `Clonalg::Clonalg ( int generations, int popsize, int optimizationProblem, int dimensions, int bitsperdimension, float mutationfactor, float cloningfactor, int numclones, int randominsertion, float upperlim, float lowerlim )`

4.2.1.2 `Clonalg::~~Clonalg ( )` [virtual]

### 4.2.2 Member Function Documentation

4.2.2.1 `void Clonalg::CalculateAffinity ( unsigned * pop, float * fitness, float * fitnessNorm )` [protected]

4.2.2.2 `void Clonalg::CloneAndHypermutate ( unsigned * pop, float * fitness, float * fitnessNorm )` [protected]

4.2.2.3 `void Clonalg::Decode ( unsigned int * pop, float * v )` [protected]

4.2.2.4 `float Clonalg::Evaluate ( unsigned * individual )` [protected]

4.2.2.5 `void Clonalg::EvaluatePop ( unsigned * pop )` [inline],[protected]

4.2.2.6 `void Clonalg::FindBestAndWorst ( )` [protected]

4.2.2.7 `void Clonalg::InitPopulation ( unsigned ** pop, float ** fitness, float ** fitnessNorm )` [protected]

4.2.2.8 `void Clonalg::Mutate ( unsigned * clone, float mutationRate )` [protected]

4.2.2.9 `void Clonalg::RandomInsertion ( unsigned * pop, float * fitness )` [protected]

4.2.2.10 `float Clonalg::Search ( )` [virtual]

Implements [BaseClonalg](#).

4.2.2.11 void Clonalg::Statistics ( unsigned \* *pop*, float \* *fitness*, int *iterationNumber* ) [protected]

### 4.2.3 Member Data Documentation

4.2.3.1 int\* Clonalg::m\_aux\_binary [private]

4.2.3.2 unsigned \* Clonalg::m\_bestClone [private]

4.2.3.3 unsigned\* Clonalg::m\_clone [private]

4.2.3.4 float\* Clonalg::m\_v [private]

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

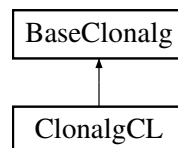
- src/Clonalg.h
- src/Clonalg.cpp

## 4.3 ClonalgCL Class Reference

Class for executing CLONALG algorithm in OpenCL devices.

```
#include <ClonalgCL.h>
```

Inheritance diagram for ClonalgCL:



### Public Member Functions

- [ClonalgCL](#) (int generations, int popsize, int optimizationProblem, int dimensions, int bitsperdimension, float mutationfactor, float cloningfactor, int numclones, int randominsertion, float upperlim, float lowerlim, int gpu-count, int cpucount, float gpuration)

*Default constructor.*

- virtual [~ClonalgCL](#) ()

*Destructor.*

- float [Search](#) ()

*Performs the optimization using the OpenCL devices.*

### Protected Member Functions

- void [OpenCLInit](#) ()  
*Create the command queues, contexts, kernels and memory buffers.*
- void [LoadSeeds](#) (int threadID)  
*Create and copy the seeds for the Park-miller pseudo-random generators to the devices.*
- void [LoadParameters](#) (int threadID)  
*Copy the parameters to the devices.*
- void [InitPopulation](#) (unsigned \*\*pop, float \*\*affinities, float \*\*affinitiesNorm, int threadID)  
*Calls a kernel that creates the initial population at the devices.*

- void [EvaluatePop](#) (unsigned \*pop, float \*affinities, int threadID)  
*Calls a kernel that evaluates a population of antibodies.*
- void [CloneAndHypermutate](#) (unsigned \*pop, float \*affinities, float \*affinitiesNorm, int threadID)  
*Calls a kernel that performs the cloning and hypermutation.*
- void [RandomInsertion](#) (unsigned \*pop, float \*affinities, int threadID)  
*Calls a kernel that replaces pRandom antibodies with new ones, randomly generated.*
- void [NormalizeAffinity](#) (unsigned \*pop, float \*affinities, float \*affinitiesNorm, int threadID)  
*Calls a kernel that normalize the affinities in [0,1].*
- void [FindBestAndWorst](#) (int threadID)  
*Calls a kernel that finds the best and worst affinity values.*
- void [Statistics](#) (unsigned \*pop, float \*affinities, int iterationNumber)  
*Prints the best affinity value found so far.*

## Protected Attributes

- Kernel \* [initKernel](#)
- Kernel \* [hipermutationKernel](#)
- Kernel \* [randomInsertionKernel](#)
- Kernel \* [statisticsKernel](#)
- Kernel \* [evaluationKernel](#)
- Kernel \* [normalizeAffinityKernel](#)
- Buffer \* [popBuffer](#)
- Buffer \* [seedBuffer](#)
- Buffer \* [statisticsBuffer](#)
- Buffer \* [affinityBuffer](#)
- Buffer \* [affinityNormBuffer](#)
- Buffer \* [parametersBuffer](#)
- int \* [m\\_pop\\_size\\_per\\_queue](#)
- int [host\\_Seeds](#) [64 \* 16384]
- CommandQueue \* [m\\_gpu\\_queues](#)
- CommandQueue \* [m\\_cpu\\_queues](#)
- int [m\\_gpu\\_count](#)
- int [m\\_cpu\\_count](#)
- float [m\\_gpu\\_ratio](#)
- [t\\_stats](#) \* [m\\_stats](#)
- [OpenCLUtils](#) [clUtils](#)

## Static Protected Attributes

- static const int [m\\_workGroupSize\\_hipermutation](#) = 128

## Additional Inherited Members

### 4.3.1 Detailed Description

Class for executing CLONALG algorithm in OpenCL devices.

This class performs OpenCL initialization and executes the CLONALG algorithm in the available OpenCL devices, using the OpenMP API to create a dedicated thread for each device.

### 4.3.2 Constructor & Destructor Documentation

4.3.2.1 **ClonalgCL::ClonalgCL** ( *int generations*, *int popsize*, *int optimizationProblem*, *int dimensions*, *int bitsperdimension*, *float mutationfactor*, *float cloningfactor*, *int numclones*, *int randominsertion*, *float upperlim*, *float lowerlim*, *int gpuccount*, *int cpuccount*, *float gpuration* )

Default constructor.

Receives the problem and hardware configurations; Creates the memory buffers; Compiles the kernels for each device; Creates the command queues.

4.3.2.2 **ClonalgCL::~~ClonalgCL** ( ) [virtual]

Destructor.

### 4.3.3 Member Function Documentation

4.3.3.1 **void ClonalgCL::CloneAndHypermutate** ( *unsigned \* pop*, *float \* affinities*, *float \* affinitiesNorm*, *int threadID* ) [protected]

Calls a kernel that performs the cloning and hypermutation.

4.3.3.2 **void ClonalgCL::EvaluatePop** ( *unsigned \* pop*, *float \* affinities*, *int threadID* ) [protected]

Calls a kernel that evaluates a population of antibodies.

4.3.3.3 **void ClonalgCL::FindBestAndWorst** ( *int threadID* ) [protected]

Calls a kernel that finds the best and worst affinity values.

4.3.3.4 **void ClonalgCL::InitPopulation** ( *unsigned \*\* pop*, *float \*\* affinities*, *float \*\* affinitiesNorm*, *int threadID* ) [protected]

Calls a kernel that creates the initial population at the devices.

4.3.3.5 **void ClonalgCL::LoadParameters** ( *int threadID* ) [protected]

Copy the parameters to the devices.

4.3.3.6 **void ClonalgCL::LoadSeeds** ( *int threadID* ) [protected]

Create and copy the seeds for the Park-miller pseudo-random generators to the devices.

4.3.3.7 **void ClonalgCL::NormalizeAffinity** ( *unsigned \* pop*, *float \* affinities*, *float \* affinitiesNorm*, *int threadID* ) [protected]

Calls a kernel that normalize the affinities in [0,1].

#### 4.3.3.8 void ClonalgCL::OpenCLInit ( ) [protected]

Create the command queues, contexts, kernels and memory buffers.  
 Perform the necessary initialization steps to use the OpenCL devices:  
 Discovers the platforms;  
 Discovers the gpu and cpu devices available in each platform;  
 Creates a context for each device;  
 Creates a command queue for each device;  
 Compiles the kernels from source for all devices;  
 Creates the kernels, replicated for each device;  
 Creates the memory buffers.

#### 4.3.3.9 void ClonalgCL::RandomInsertion ( unsigned \* *pop*, float \* *affinities*, int *threadID* ) [protected]

Calls a kernel that replaces pRandom antibodies with new ones, randomly generated.

#### 4.3.3.10 float ClonalgCL::Search ( ) [virtual]

Performs the optimization using the OpenCL devices.

At first, declares a OpenMP parallel block with (m\_gpu\_count + m\_cpu\_count) threads. Then each thread loads the parameters, seeds and initializes their sub populations. The optimization goes on until m\_generations are executed. The only synchronization among the threads occurs in FindBestAndWorst, where is necessary to find the best and worst affinity values of the entire population.

##### Returns

The total computing time, without overheads.

Implements [BaseClonalg](#).

#### 4.3.3.11 void ClonalgCL::Statistics ( unsigned \* *pop*, float \* *affinities*, int *iterationNumber* ) [protected]

Prints the best affinity value found so far.

### 4.3.4 Member Data Documentation

#### 4.3.4.1 Buffer\* ClonalgCL::affinityBuffer [protected]

Buffer used to store the affinity values

#### 4.3.4.2 Buffer\* ClonalgCL::affinityNormBuffer [protected]

Buffer used to store the normalized affinity values

#### 4.3.4.3 OpenCLUtils ClonalgCL::clUtils [protected]

Used for compiling the kernels



**4.3.4.4 Kernel\* ClonalgCL::evaluationKernel** [protected]

Kernel used to evaluate a population

**4.3.4.5 Kernel\* ClonalgCL::hipermutationKernel** [protected]

Kernel used to clone and hypermutate

**4.3.4.6 int ClonalgCL::host\_Seeds[64 \* 16384]** [protected]

Seeds created by the host at the beginning of the process

**4.3.4.7 Kernel\* ClonalgCL::initKernel** [protected]

Kernel used to create the initial population

**4.3.4.8 int ClonalgCL::m\_cpu\_count** [protected]

Number of cpus being used

**4.3.4.9 CommandQueue\* ClonalgCL::m\_cpu\_queues** [protected]

Command queues for cpus

**4.3.4.10 int ClonalgCL::m\_gpu\_count** [protected]

Number of gpus being used

**4.3.4.11 CommandQueue\* ClonalgCL::m\_gpu\_queues** [protected]

Command queues for gpus

**4.3.4.12 float ClonalgCL::m\_gpu\_ratio** [protected]

GPU computing ratio (in respect to CPU)

**4.3.4.13 int\* ClonalgCL::m\_pop\_size\_per\_queue** [protected]

Stores how many individuals will be evolved within each sub population

**4.3.4.14 t\_stats\* ClonalgCL::m\_stats** [protected]

Used for printing statistics, such as the best solution found so far

**4.3.4.15 const int ClonalgCL::m\_workGroupSize\_hypermutation = 128** [static], [protected]

Number of work\_items to be used in the evaluation kernel

#### 4.3.4.16 Kernel\* ClonalgCL::normalizeAffinityKernel [protected]

Kernel used to compute each antibody's normalized affinity

#### 4.3.4.17 Buffer\* ClonalgCL::parametersBuffer [protected]

Buffer used to store the values of the parameters in constant memory

#### 4.3.4.18 Buffer\* ClonalgCL::popBuffer [protected]

Buffer for storing the population

#### 4.3.4.19 Kernel\* ClonalgCL::randomInsertionKernel [protected]

Kernel used to replace the worst antibodies with randomly generated ones

#### 4.3.4.20 Buffer\* ClonalgCL::seedBuffer [protected]

Buffer for storing the seeds

#### 4.3.4.21 Buffer\* ClonalgCL::statisticsBuffer [protected]

Buffer for storing the best and worst affinity values

#### 4.3.4.22 Kernel\* ClonalgCL::statisticsKernel [protected]

Kernel used to collect the best and worst affinity values

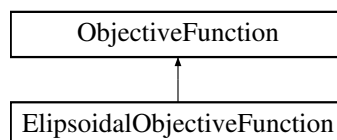
The documentation for this class was generated from the following files:

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

## 4.4 ElipsoidalObjectiveFunction Class Reference

```
#include <ElipsoidalObjectiveFunction.h>
```

Inheritance diagram for ElipsoidalObjectiveFunction:



### Public Member Functions

- [ElipsoidalObjectiveFunction](#) (int vectorLength)
- virtual [~ElipsoidalObjectiveFunction](#) ()
- float [Evaluate](#) (int \*v)
- float [EvaluateFloat](#) (float \*v)

- string [getName](#) ()
- string [getFormula](#) ()
- string [getSumLimit](#) ()

### Additional Inherited Members

#### 4.4.1 Constructor & Destructor Documentation

4.4.1.1 `ElipsoidalObjectiveFunction::ElipsoidalObjectiveFunction ( int vectorLength )`

4.4.1.2 `ElipsoidalObjectiveFunction::~~ElipsoidalObjectiveFunction ( )` [virtual]

#### 4.4.2 Member Function Documentation

4.4.2.1 `float ElipsoidalObjectiveFunction::Evaluate ( int * v )` [virtual]

Implements [ObjectiveFunction](#).

4.4.2.2 `float ElipsoidalObjectiveFunction::EvaluateFloat ( float * v )` [virtual]

Implements [ObjectiveFunction](#).

4.4.2.3 `string ElipsoidalObjectiveFunction::getFormula ( )` [virtual]

Implements [ObjectiveFunction](#).

4.4.2.4 `string ElipsoidalObjectiveFunction::getName ( )` [virtual]

Implements [ObjectiveFunction](#).

4.4.2.5 `string ElipsoidalObjectiveFunction::getSumLimit ( )` [virtual]

Implements [ObjectiveFunction](#).

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

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

## 4.5 KernelParameters Struct Reference

```
#include <representation.h>
```

### Public Attributes

- int [POP\\_SIZE](#)
- int [NCLON](#)
- int [CHROM\\_LEN](#)
- int [REAL\\_LEN](#)
- float [MUTATION\\_FACTOR](#)
- float [CLONING\\_FACTOR](#)

- int [BITS\\_PER\\_DIMENSION](#)
- int [DIMENSIONS](#)
- float [UPPER\\_LIM](#)
- float [LOWER\\_LIM](#)

#### 4.5.1 Member Data Documentation

4.5.1.1 int KernelParameters::BITS\_PER\_DIMENSION

4.5.1.2 int KernelParameters::CHROM\_LEN

4.5.1.3 float KernelParameters::CLONING\_FACTOR

4.5.1.4 int KernelParameters::DIMENSIONS

4.5.1.5 float KernelParameters::LOWER\_LIM

4.5.1.6 float KernelParameters::MUTATION\_FACTOR

4.5.1.7 int KernelParameters::NCLON

4.5.1.8 int KernelParameters::POP\_SIZE

4.5.1.9 int KernelParameters::REAL\_LEN

4.5.1.10 float KernelParameters::UPPER\_LIM

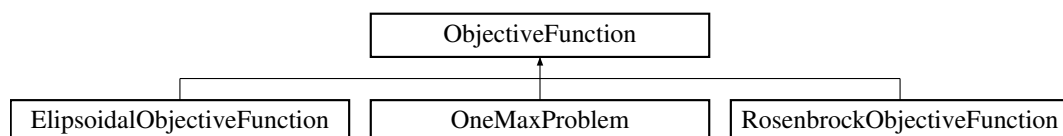
The documentation for this struct was generated from the following file:

- [src/representation.h](#)

## 4.6 ObjectiveFunction Class Reference

```
#include <ObjectiveFunction.h>
```

Inheritance diagram for ObjectiveFunction:



### Public Member Functions

- [ObjectiveFunction](#) (int vectorLength)
- virtual [~ObjectiveFunction](#) ()
- virtual float [Evaluate](#) (int \*v)=0
- virtual float [EvaluateFloat](#) (float \*v)=0
- virtual string [getName](#) ()=0
- virtual string [getFormula](#) ()=0
- virtual string [getSumLimit](#) ()=0

## Protected Attributes

- [int m\\_vectorLength](#)

### 4.6.1 Constructor & Destructor Documentation

4.6.1.1 `ObjectiveFunction::ObjectiveFunction ( int vectorLength )`

4.6.1.2 `ObjectiveFunction::~~ObjectiveFunction ( )` [virtual]

### 4.6.2 Member Function Documentation

4.6.2.1 `virtual float ObjectiveFunction::Evaluate ( int * v )` [pure virtual]

Implemented in [ElipsoidalObjectiveFunction](#), [OneMaxProblem](#), and [RosenbrockObjectiveFunction](#).

4.6.2.2 `virtual float ObjectiveFunction::EvaluateFloat ( float * v )` [pure virtual]

Implemented in [ElipsoidalObjectiveFunction](#), [OneMaxProblem](#), and [RosenbrockObjectiveFunction](#).

4.6.2.3 `virtual string ObjectiveFunction::getFormula ( )` [pure virtual]

Implemented in [ElipsoidalObjectiveFunction](#), [OneMaxProblem](#), and [RosenbrockObjectiveFunction](#).

4.6.2.4 `virtual string ObjectiveFunction::getName ( )` [pure virtual]

Implemented in [ElipsoidalObjectiveFunction](#), [OneMaxProblem](#), and [RosenbrockObjectiveFunction](#).

4.6.2.5 `virtual string ObjectiveFunction::getSumLimit ( )` [pure virtual]

Implemented in [ElipsoidalObjectiveFunction](#), [OneMaxProblem](#), and [RosenbrockObjectiveFunction](#).

### 4.6.3 Member Data Documentation

4.6.3.1 `int ObjectiveFunction::m_vectorLength` [protected]

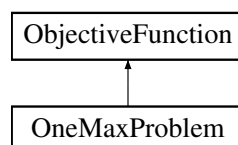
The documentation for this class was generated from the following files:

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

## 4.7 OneMaxProblem Class Reference

```
#include <OneMaxProblem.h>
```

Inheritance diagram for OneMaxProblem:



## Public Member Functions

- [OneMaxProblem](#) (int *vectorLength*)
- virtual [~OneMaxProblem](#) ()
- float [Evaluate](#) (int \**v*)
- float [EvaluateFloat](#) (float \**v*)
- string [getName](#) ()
- string [getFormula](#) ()
- string [getSumLimit](#) ()

## Additional Inherited Members

### 4.7.1 Constructor & Destructor Documentation

4.7.1.1 `OneMaxProblem::OneMaxProblem ( int vectorLength )`

4.7.1.2 `OneMaxProblem::~~OneMaxProblem ( ) [virtual]`

### 4.7.2 Member Function Documentation

4.7.2.1 `float OneMaxProblem::Evaluate ( int * v ) [virtual]`

Implements [ObjectiveFunction](#).

4.7.2.2 `float OneMaxProblem::EvaluateFloat ( float * v ) [virtual]`

Implements [ObjectiveFunction](#).

4.7.2.3 `string OneMaxProblem::getFormula ( ) [virtual]`

Implements [ObjectiveFunction](#).

4.7.2.4 `string OneMaxProblem::getName ( ) [virtual]`

Implements [ObjectiveFunction](#).

4.7.2.5 `string OneMaxProblem::getSumLimit ( ) [virtual]`

Implements [ObjectiveFunction](#).

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

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

## 4.8 OpenCLUtils Class Reference

Class for handling OpenCL common operations.

```
#include <OpenCLUtils.h>
```

## Public Member Functions

- [OpenCLUtils](#) ()
- virtual [~OpenCLUtils](#) ()
- Program [CreateProgramFromSource](#) (Context context, vector< Device > devices, std::string compilerOptions, const char \*fileName)
- Program [CreateProgramFromBinary](#) (Context context, vector< Device > devices, const char \*fileName)
- void [SaveProgramBinary](#) (Program program, vector< Device > devices, const char \*fileName)
- int [GetBinarySize](#) (const char \*filename, char \*&buffer)
- void [ShowDeviceInfo](#) (Device device)
- double [getElapsedTime](#) (Event evt)

### 4.8.1 Detailed Description

Class for handling OpenCL common operations.

This class is able to compile/create OpenCL kernels from source or binary, and to show the properties of the devices being used.

### 4.8.2 Constructor & Destructor Documentation

4.8.2.1 `OpenCLUtils::OpenCLUtils ( )`

4.8.2.2 `OpenCLUtils::~~OpenCLUtils ( )` `[virtual]`

### 4.8.3 Member Function Documentation

4.8.3.1 `Program OpenCLUtils::CreateProgramFromBinary ( Context context, vector< Device > devices, const char * fileName )`

4.8.3.2 `Program OpenCLUtils::CreateProgramFromSource ( Context context, vector< Device > devices, std::string compilerOptions, const char * fileName )`

4.8.3.3 `int OpenCLUtils::GetBinarySize ( const char * filename, char *& buffer )`

4.8.3.4 `double OpenCLUtils::getElapsedTime ( Event evt )`

4.8.3.5 `void OpenCLUtils::SaveProgramBinary ( Program program, vector< Device > devices, const char * fileName )`

4.8.3.6 `void OpenCLUtils::ShowDeviceInfo ( Device device )`

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

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

## 4.9 ProblemFactory Class Reference

```
#include <ProblemFactory.h>
```

## Public Member Functions

- [ProblemFactory](#) ()
- virtual [~ProblemFactory](#) ()

## Static Public Member Functions

- static [ObjectiveFunction](#) \* [CreateProblem](#) (int ID, int dimensions)

### 4.9.1 Constructor & Destructor Documentation

4.9.1.1 [ProblemFactory::ProblemFactory](#) ( )

4.9.1.2 [ProblemFactory::~~ProblemFactory](#) ( ) [virtual]

### 4.9.2 Member Function Documentation

4.9.2.1 [ObjectiveFunction](#) \* [ProblemFactory::CreateProblem](#) ( int ID, int *dimensions* ) [static]

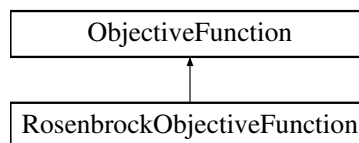
The documentation for this class was generated from the following files:

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

## 4.10 RosenbrockObjectiveFunction Class Reference

```
#include <RosenbrockObjectiveFunction.h>
```

Inheritance diagram for RosenbrockObjectiveFunction:



## Public Member Functions

- [RosenbrockObjectiveFunction](#) (int vectorLength)
- virtual [~RosenbrockObjectiveFunction](#) ( )
- float [Evaluate](#) (int \*v)
- float [EvaluateFloat](#) (float \*v)
- string [getName](#) ( )
- string [getFormula](#) ( )
- string [getSumLimit](#) ( )

## Additional Inherited Members

### 4.10.1 Constructor & Destructor Documentation

4.10.1.1 [RosenbrockObjectiveFunction::RosenbrockObjectiveFunction](#) ( int *vectorLength* )

4.10.1.2 [RosenbrockObjectiveFunction::~~RosenbrockObjectiveFunction](#) ( ) [virtual]

### 4.10.2 Member Function Documentation



4.10.2.1 float RosenbrockObjectiveFunction::Evaluate ( int \* v ) [virtual]

Implements [ObjectiveFunction](#).

4.10.2.2 float RosenbrockObjectiveFunction::EvaluateFloat ( float \* v ) [virtual]

Implements [ObjectiveFunction](#).

4.10.2.3 string RosenbrockObjectiveFunction::getFormula ( ) [virtual]

Implements [ObjectiveFunction](#).

4.10.2.4 string RosenbrockObjectiveFunction::getName ( ) [virtual]

Implements [ObjectiveFunction](#).

4.10.2.5 string RosenbrockObjectiveFunction::getSumLimit ( ) [virtual]

Implements [ObjectiveFunction](#).

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

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

## 4.11 t\_parameters Struct Reference

### Public Attributes

- int [generations](#)
- int [popsize](#)
- int [optimizationProblem](#)
- int [problemDimension](#)
- int [bitsPerDimension](#)
- int [nclones](#)
- int [randomInsertion](#)
- float [mutationFactor](#)
- float [upperLim](#)
- float [lowerLim](#)
- bool [parallel](#)
- int [gpus](#)
- int [cpus](#)
- float [gpuRatio](#)

### 4.11.1 Member Data Documentation

4.11.1.1 int t\_parameters::bitsPerDimension

4.11.1.2 int t\_parameters::cpus

4.11.1.3 int t\_parameters::generations

4.11.1.4 float t\_parameters::gpuRatio

4.11.1.5 int t\_parameters::gpus

4.11.1.6 float t\_parameters::lowerLim

4.11.1.7 float t\_parameters::mutationFactor

4.11.1.8 int t\_parameters::nclones

4.11.1.9 int t\_parameters::optimizationProblem

4.11.1.10 bool t\_parameters::parallel

4.11.1.11 int t\_parameters::popsize

4.11.1.12 int t\_parameters::problemDimension

4.11.1.13 int t\_parameters::randomInsertion

4.11.1.14 float t\_parameters::upperLim

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

- src/[main.cpp](#)

## 4.12 t\_stats Struct Reference

```
#include <representation.h>
```

### Public Attributes

- int [indexBest](#)
- int [indexWorst](#)
- float [affinityBest](#)
- float [affinityWorst](#)

### 4.12.1 Member Data Documentation

4.12.1.1 float t\_stats::affinityBest

4.12.1.2 float t\_stats::affinityWorst

4.12.1.3 int t\_stats::indexBest

4.12.1.4 int t\_stats::indexWorst

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

- src/[representation.h](#)

## 4.13 TimeUtils Class Reference

```
#include <TimeUtils.h>
```

### Static Public Member Functions

- static double [getRealTime](#) ()

#### 4.13.1 Member Function Documentation

4.13.1.1 static double TimeUtils::getRealTime ( ) [inline], [static]

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

- src/[TimeUtils.h](#)



## Chapter 5

# File Documentation

### 5.1 src/BaseClonalg.cpp File Reference

```
#include "BaseClonalg.h"
```

### 5.2 src/BaseClonalg.h File Reference

```
#include <algorithm>
#include <cmath>
#include <iostream>
#include "ProblemFactory.h"
#include "representation.h"
```

#### Classes

- class [BaseClonalg](#)

### 5.3 src/Clonalg.cpp File Reference

```
#include <stdlib.h>
#include <iostream>
#include "Clonalg.h"
```

### 5.4 src/Clonalg.h File Reference

```
#include "BaseClonalg.h"
```

#### Classes

- class [Clonalg](#)

## 5.5 src/ClonalgCL.cpp File Reference

```
#include "ClonalgCL.h"  
#include "TimeUtils.h"  
#include <cmath>  
#include <omp.h>
```

## 5.6 src/ClonalgCL.h File Reference

```
#include <iostream>  
#include <fstream>  
#include "CL/cl.h"  
#include "BaseClonalg.h"  
#include "OpenCLUtils.h"
```

### Classes

- class [ClonalgCL](#)  
*Class for executing CLONALG algorithm in OpenCL devices.*

### Macros

- `#define __NO_STD_VECTOR`
- `#define __CL_ENABLE_EXCEPTIONS`

### 5.6.1 Macro Definition Documentation

5.6.1.1 `#define __CL_ENABLE_EXCEPTIONS`

5.6.1.2 `#define __NO_STD_VECTOR`

## 5.7 src/ElipsoidalObjectiveFunction.cpp File Reference

```
#include "ElipsoidalObjectiveFunction.h"  
#include <iostream>
```

## 5.8 src/ElipsoidalObjectiveFunction.h File Reference

```
#include "ObjectiveFunction.h"
```

### Classes

- class [ElipsoidalObjectiveFunction](#)

## 5.9 src/kernels/kernel.cpp File Reference

```
#include "../src/parameters.h"
#include "../src/representation.h"
#include "../src/kernels/utils.cpp"
```

### Functions

- `__kernel void initPopulation ( __global unsigned *pop, __global int *D_seeds, __constant struct KernelParameters *parameters)`  
*BRIEF WOW. WOW EH NOIS.*
- `int BinaryToDecimal ( __local int *binary, int begin, int end, __local float *partialSum, int localID, int localSize)`
- `float EvaluateIndividual_local ( __local unsigned *individual, int lid, int localSize, __constant struct KernelParameters *parameters)`
- `__kernel void Evaluation ( __global unsigned *pop, __global float *fitness, __constant struct KernelParameters *parameters)`
- `__kernel void NormalizeAffinity ( __global float *fitness, __global float *fitnessNorm, __global t_stats *est, __constant struct KernelParameters *parameters)`
- `__kernel void cloneAndHypermutation ( __global unsigned *pop, __global float *fitness, __global float *fitnessNorm, __global int *D_seeds, __global t_stats *est, __constant struct KernelParameters *parameters)`
- `__kernel void StatisticsReduction1 ( __global unsigned *pop, __global float *fitness, __global t_stats *est, __global KernelParameters *parameters)`
- `__kernel void randomInsertion ( __global unsigned int *pop, __global float *fitness, __global t_stats *stats, __global int *D_seeds, __constant KernelParameters *p)`

### 5.9.1 Function Documentation

5.9.1.1 `int BinaryToDecimal ( __local int * binary, int begin, int end, __local float * partialSum, int localID, int localSize )`

5.9.1.2 `__kernel void cloneAndHypermutation ( __global unsigned * pop, __global float * fitness, __global float * fitnessNorm, __global int * D_seeds, __global t_stats * est, __constant struct KernelParameters * parameters )`

5.9.1.3 `float EvaluateIndividual_local ( __local unsigned * individual, int lid, int localSize, __constant struct KernelParameters * parameters )`

5.9.1.4 `__kernel void Evaluation ( __global unsigned * pop, __global float * fitness, __constant struct KernelParameters * parameters )`

5.9.1.5 `__kernel void initPopulation ( __global unsigned * pop, __global int * D_seeds, __constant struct KernelParameters * parameters )`

BRIEF WOW. WOW EH NOIS.

5.9.1.6 `__kernel void NormalizeAffinity ( __global float * fitness, __global float * fitnessNorm, __global t_stats * est, __constant struct KernelParameters * parameters )`

5.9.1.7 `__kernel void randomInsertion ( __global unsigned int * pop, __global float * fitness, __global t_stats * stats, __global int * D_seeds, __constant KernelParameters * p )`

5.9.1.8 `__kernel void StatisticsReduction1 ( __global unsigned * pop, __global float * fitness, __global t_stats * est, __global KernelParameters * parameters )`

## 5.10 src/kernels/utils.cpp File Reference

### Macros

- `#define` [CONST1](#) 2147483648.0
- `#define` [CONST2](#) 2147483647
- `#define` [MASK](#) 2147483648
- `#define` [BIT\\_SET](#)(a, b) ((a) |= (1<<(b)))
- `#define` [BIT\\_CLEAR](#)(a, b) ((a) &= ~(1<<(b)))
- `#define` [BIT\\_FLIP](#)(a, b) ((a) ^= (1<<(b)))
- `#define` [BIT\\_CHECK](#)(a, b) ((a) & (1<<(b)))

### Functions

- `int` [rand](#) (int \*seed)
- `float` [u\\_rand](#) (int \*seed)

#### 5.10.1 Macro Definition Documentation

5.10.1.1 `#define` [BIT\\_CHECK](#)( a, b ) ((a) & (1<<(b)))

5.10.1.2 `#define` [BIT\\_CLEAR](#)( a, b ) ((a) &= ~(1<<(b)))

5.10.1.3 `#define` [BIT\\_FLIP](#)( a, b ) ((a) ^= (1<<(b)))

5.10.1.4 `#define` [BIT\\_SET](#)( a, b ) ((a) |= (1<<(b)))

5.10.1.5 `#define` [CONST1](#) 2147483648.0

5.10.1.6 `#define` [CONST2](#) 2147483647

5.10.1.7 `#define` [MASK](#) 2147483648

#### 5.10.2 Function Documentation

5.10.2.1 `int` [rand](#) ( int \* *seed* )

5.10.2.2 `float` [u\\_rand](#) ( int \* *seed* )

## 5.11 src/main.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "Clonalg.h"
#include "ClonalgCL.h"
#include "TimeUtils.h"
```

### Classes

- `struct` [t\\_parameters](#)



## Functions

- void [GetArgs](#) (int argc, char \*\*argv, [t\\_parameters](#) &parameters)
- int [main](#) (int argc, char \*\*argv)

### 5.11.1 Function Documentation

5.11.1.1 void [GetArgs](#) ( int *argc*, char \*\* *argv*, [t\\_parameters](#) & *parameters* )

5.11.1.2 int [main](#) ( int *argc*, char \*\* *argv* )

## 5.12 src/ObjectiveFunction.cpp File Reference

```
#include "ObjectiveFunction.h"
```

## 5.13 src/ObjectiveFunction.h File Reference

```
#include <string>
#include "representation.h"
```

## Classes

- class [ObjectiveFunction](#)

## 5.14 src/OneMaxProblem.cpp File Reference

```
#include "OneMaxProblem.h"
#include <iostream>
```

## 5.15 src/OneMaxProblem.h File Reference

```
#include "ObjectiveFunction.h"
```

## Classes

- class [OneMaxProblem](#)

## 5.16 src/openglUtils.c File Reference

```
#include "openglUtils.h"
```

## Functions

- `cl_platform_id * clObtemPlataformas ()`
- `cl_device_id * clObtemDispositivos (cl_platform_id *plataformas)`
- `void clCompilaPrograma (cl_program *programa, cl_device_id *devices)`

### 5.16.1 Function Documentation

5.16.1.1 `void clCompilaPrograma ( cl_program * programa, cl_device_id * devices )`

5.16.1.2 `cl_device_id* clObtemDispositivos ( cl_platform_id * plataformas )`

5.16.1.3 `cl_platform_id* clObtemPlataformas ( )`

## 5.17 src/OpenCLUtils.cpp File Reference

```
#include "OpenCLUtils.h"
```

## Macros

- `#define __NO_STD_VECTOR`

### 5.17.1 Macro Definition Documentation

5.17.1.1 `#define __NO_STD_VECTOR`

## 5.18 src/oclUtils.h File Reference

```
#include <CL/cl.h>
#include <stdlib.h>
#include <stdio.h>
```

## Macros

- `#define log_arquivo() printf("Arquivo: %s, linha: %d\n", __FILE__, __LINE__)`
- `#define log_error(M) printf("%s\n", M)`
- `#define log_error_code(CODE) printf("Codigo: (%d)\n", CODE)`
- `#define check_cl(STATUS, M) if(STATUS!=CL_SUCCESS){ log_error(M); log_error_code(STATUS); log_arquivo(); exit(EXIT_FAILURE); }`

## Functions

- `cl_platform_id * clObtemPlataformas ()`
- `cl_device_id * clObtemDispositivos (cl_platform_id *plataformas)`
- `void clCompilaPrograma (cl_program *programa, cl_device_id *devices)`

### 5.18.1 Macro Definition Documentation

5.18.1.1 `#define check_cl( STATUS, M ) if(STATUS!=CL_SUCCESS){ log_error(M); log_error_code(STATUS); log_arquivo(); exit(EXIT_FAILURE); }`

5.18.1.2 `#define log_arquivo( ) printf("Arquivo: %s, linha: %d\n",__FILE__, __LINE__)`

5.18.1.3 `#define log_error( M ) printf("%s\n", M)`

5.18.1.4 `#define log_error_code( CODE ) printf("Codigo: (%d)\n", CODE)`

### 5.18.2 Function Documentation

5.18.2.1 `void clCompilaPrograma ( cl_program * programa, cl_device_id * devices )`

5.18.2.2 `cl_device_id* clObtemDispositivos ( cl_platform_id * plataformas )`

5.18.2.3 `cl_platform_id* clObtemPlataformas ( )`

## 5.19 src/OpenCLUtils.h File Reference

```
#include <iostream>
#include <fstream>
#include "CL/cl.h"
#include <string>
```

### Classes

- class [OpenCLUtils](#)  
*Class for handling OpenCL common operations.*

### Macros

- `#define __NO_STD_VECTOR`
- `#define __CL_ENABLE_EXCEPTIONS`

### 5.19.1 Macro Definition Documentation

5.19.1.1 `#define __CL_ENABLE_EXCEPTIONS`

5.19.1.2 `#define __NO_STD_VECTOR`

## 5.20 src/parameters.h File Reference

### Macros

- `#define BITS_PER_WORD 32`

### 5.20.1 Macro Definition Documentation

5.20.1.1 `#define BITS_PER_WORD 32`

## 5.21 `src/ProblemFactory.cpp` File Reference

```
#include "ProblemFactory.h"
```

## 5.22 `src/ProblemFactory.h` File Reference

```
#include "ObjectiveFunction.h"
#include "OneMaxProblem.h"
#include "ElipsoidalObjectiveFunction.h"
#include "RosenbrockObjectiveFunction.h"
```

### Classes

- class [ProblemFactory](#)

## 5.23 `src/representation.h` File Reference

```
#include "parameters.h"
```

### Classes

- struct [t\\_stats](#)
- struct [KernelParameters](#)

### Macros

- `#define BIT_SET(a, b) ((a) |= (1<<(b)))`
- `#define BIT_CLEAR(a, b) ((a) &= ~(1<<(b)))`
- `#define BIT_FLIP(a, b) ((a) ^= (1<<(b)))`
- `#define BIT_CHECK(a, b) ((a) & (1<<(b)))`

### Typedefs

- typedef struct [KernelParameters](#) [KernelParameters](#)

### 5.23.1 Macro Definition Documentation

5.23.1.1 `#define BIT_CHECK( a, b ) ((a) & (1<<(b)))`

5.23.1.2 `#define BIT_CLEAR( a, b ) ((a) &= ~(1<<(b)))`

5.23.1.3 `#define BIT_FLIP( a, b ) ((a) ^= (1<<(b)))`

5.23.1.4 `#define BIT_SET( a, b ) ((a) |= (1<<(b)))`

### 5.23.2 Typedef Documentation

#### 5.23.2.1 typedef struct KernelParameters KernelParameters

## 5.24 src/RosenbrockObjectiveFunction.cpp File Reference

```
#include "RosenbrockObjectiveFunction.h"
```

## 5.25 src/RosenbrockObjectiveFunction.h File Reference

```
#include "ObjectiveFunction.h"
```

### Classes

- class [RosenbrockObjectiveFunction](#)

## 5.26 src/TimeUtils.h File Reference

```
#include <sys/time.h>
```

### Classes

- class [TimeUtils](#)

# Index

- ~BaseClonalg
  - BaseClonalg, [8](#)
- ~Clonalg
  - Clonalg, [10](#)
- ~ClonalgCL
  - ClonalgCL, [13](#)
- ~ElipsoidalObjectiveFunction
  - ElipsoidalObjectiveFunction, [17](#)
- ~ObjectiveFunction
  - ObjectiveFunction, [19](#)
- ~OneMaxProblem
  - OneMaxProblem, [20](#)
- ~OpenCLUtils
  - OpenCLUtils, [21](#)
- ~ProblemFactory
  - ProblemFactory, [22](#)
- ~RosenbrockObjectiveFunction
  - RosenbrockObjectiveFunction, [22](#)
- \_\_NO\_STD\_VECTOR
  - ClonalgCL.h, [28](#)
  - OpenCLUtils.cpp, [32](#)
  - OpenCLUtils.h, [33](#)
- affinityBest
  - t\_stats, [24](#)
- affinityBuffer
  - ClonalgCL, [14](#)
- affinityNormBuffer
  - ClonalgCL, [14](#)
- affinityWorst
  - t\_stats, [24](#)
- BIT\_CHECK
  - representation.h, [34](#)
  - utils.cpp, [30](#)
- BIT\_CLEAR
  - representation.h, [34](#)
  - utils.cpp, [30](#)
- BIT\_FLIP
  - representation.h, [34](#)
  - utils.cpp, [30](#)
- BIT\_SET
  - representation.h, [34](#)
  - utils.cpp, [30](#)
- BITS\_PER\_DIMENSION
  - KernelParameters, [18](#)
- BITS\_PER\_WORD
  - parameters.h, [33](#)
- BaseClonalg, [7](#)
  - ~BaseClonalg, [8](#)
- BaseClonalg, [8](#)
- BaseClonalg, [8](#)
- BinaryToDecimal, [8](#)
- CompareIndividuals, [8](#)
- GetWord, [8](#)
- HipermutationRate, [8](#)
- m\_bitsperdimension, [8](#)
- m\_bitsperparameter, [8](#)
- m\_cloningfactor, [8](#)
- m\_cromLen, [8](#)
- m\_dimensions, [8](#)
- m\_fitness, [8](#)
- m\_fitnessNorm, [9](#)
- m\_generations, [9](#)
- m\_lowerlim, [9](#)
- m\_mutationfactor, [9](#)
- m\_numclones, [9](#)
- m\_objective, [9](#)
- m\_optimizationProblem, [9](#)
- m\_pop, [9](#)
- m\_popsiz, [9](#)
- m\_randominsertion, [9](#)
- m\_reallLen, [9](#)
- m\_statistics, [9](#)
- m\_upperlim, [9](#)
- PrintPop, [8](#)
- Search, [8](#)
- Sort, [8](#)
- BinaryToDecimal
  - BaseClonalg, [8](#)
  - kernel.cpp, [29](#)
- bitsPerDimension
  - t\_parameters, [23](#)
- CHROM\_LEN
  - KernelParameters, [18](#)
- CLONING\_FACTOR
  - KernelParameters, [18](#)
- CONST1
  - utils.cpp, [30](#)
- CONST2
  - utils.cpp, [30](#)
- CalculateAffinity
  - Clonalg, [10](#)
- check\_cl
  - opencLUtils.h, [33](#)
- clCompilaPrograma
  - opencLUtils.c, [32](#)
  - opencLUtils.h, [33](#)
- clObtemDispositivos

- opencLUtills.c, [32](#)
  - opencLUtills.h, [33](#)
- clObtemPlataformas
  - opencLUtills.c, [32](#)
  - opencLUtills.h, [33](#)
- clUtils
  - ClonalgCL, [14](#)
- Clonalg, [9](#)
  - ~Clonalg, [10](#)
  - CalculateAffinity, [10](#)
  - Clonalg, [10](#)
  - CloneAndHypermutate, [10](#)
  - Decode, [10](#)
  - Evaluate, [10](#)
  - EvaluatePop, [10](#)
  - FindBestAndWorst, [10](#)
  - InitPopulation, [10](#)
  - m\_aux\_binary, [11](#)
  - m\_bestClone, [11](#)
  - m\_clone, [11](#)
  - m\_v, [11](#)
  - Mutate, [10](#)
  - RandomInsertion, [10](#)
  - Search, [10](#)
  - Statistics, [10](#)
- ClonalgCL, [11](#)
  - ~ClonalgCL, [13](#)
  - affinityBuffer, [14](#)
  - affinityNormBuffer, [14](#)
  - clUtils, [14](#)
  - ClonalgCL, [13](#)
  - ClonalgCL, [13](#)
  - CloneAndHypermutate, [13](#)
  - EvaluatePop, [13](#)
  - evaluationKernel, [14](#)
  - FindBestAndWorst, [13](#)
  - hipermutationKernel, [15](#)
  - host\_Seeds, [15](#)
  - initKernel, [15](#)
  - InitPopulation, [13](#)
  - LoadParameters, [13](#)
  - LoadSeeds, [13](#)
  - m\_cpu\_count, [15](#)
  - m\_cpu\_queues, [15](#)
  - m\_gpu\_count, [15](#)
  - m\_gpu\_queues, [15](#)
  - m\_gpu\_ratio, [15](#)
  - m\_pop\_size\_per\_queue, [15](#)
  - m\_stats, [15](#)
  - m\_workGroupSize\_hipermutation, [15](#)
  - NormalizeAffinity, [13](#)
  - normalizeAffinityKernel, [15](#)
  - OpenCLInit, [13](#)
  - parametersBuffer, [16](#)
  - popBuffer, [16](#)
  - RandomInsertion, [14](#)
  - randomInsertionKernel, [16](#)
  - Search, [14](#)
  - seedBuffer, [16](#)
  - Statistics, [14](#)
  - statisticsBuffer, [16](#)
  - statisticsKernel, [16](#)
- CloneAndHypermutate
  - Clonalg, [10](#)
  - ClonalgCL, [13](#)
- cloneAndHypermutation
  - kernel.cpp, [29](#)
- CompareIndividuals
  - BaseClonalg, [8](#)
- cpus
  - t\_parameters, [23](#)
- CreateProblem
  - ProblemFactory, [22](#)
- CreateProgramFromBinary
  - OpenCLUtills, [21](#)
- CreateProgramFromSource
  - OpenCLUtills, [21](#)
- DIMENSIONS
  - KernelParameters, [18](#)
- Decode
  - Clonalg, [10](#)
- ElipsoidalObjectiveFunction, [16](#)
  - ~ElipsoidalObjectiveFunction, [17](#)
  - ElipsoidalObjectiveFunction, [17](#)
  - ElipsoidalObjectiveFunction, [17](#)
  - Evaluate, [17](#)
  - EvaluateFloat, [17](#)
  - getFormula, [17](#)
  - getName, [17](#)
  - getSumLimit, [17](#)
- Evaluate
  - Clonalg, [10](#)
  - ElipsoidalObjectiveFunction, [17](#)
  - ObjectiveFunction, [19](#)
  - OneMaxProblem, [20](#)
  - RosenbrockObjectiveFunction, [22](#)
- EvaluateFloat
  - ElipsoidalObjectiveFunction, [17](#)
  - ObjectiveFunction, [19](#)
  - OneMaxProblem, [20](#)
  - RosenbrockObjectiveFunction, [23](#)
- EvaluateIndividual\_local
  - kernel.cpp, [29](#)
- EvaluatePop
  - Clonalg, [10](#)
  - ClonalgCL, [13](#)
- Evaluation
  - kernel.cpp, [29](#)
- evaluationKernel
  - ClonalgCL, [14](#)
- FindBestAndWorst
  - Clonalg, [10](#)
  - ClonalgCL, [13](#)

- generations
  - t\_parameters, 23
- GetArgs
  - main.cpp, 31
- GetBinarySize
  - OpenCLUtils, 21
- getElapsedTime
  - OpenCLUtils, 21
- getFormula
  - ElipsoidalObjectiveFunction, 17
  - ObjectiveFunction, 19
  - OneMaxProblem, 20
  - RosenbrockObjectiveFunction, 23
- getName
  - ElipsoidalObjectiveFunction, 17
  - ObjectiveFunction, 19
  - OneMaxProblem, 20
  - RosenbrockObjectiveFunction, 23
- getRealTime
  - TimeUtils, 25
- getSumLimit
  - ElipsoidalObjectiveFunction, 17
  - ObjectiveFunction, 19
  - OneMaxProblem, 20
  - RosenbrockObjectiveFunction, 23
- GetWord
  - BaseClonalg, 8
- gpuRatio
  - t\_parameters, 23
- gpus
  - t\_parameters, 24
- hipermutationKernel
  - ClonalgCL, 15
- HipermutationRate
  - BaseClonalg, 8
- host\_Seeds
  - ClonalgCL, 15
- indexBest
  - t\_stats, 24
- indexWorst
  - t\_stats, 24
- initKernel
  - ClonalgCL, 15
- InitPopulation
  - Clonalg, 10
  - ClonalgCL, 13
- initPopulation
  - kernel.cpp, 29
- kernel.cpp
  - BinaryToDecimal, 29
  - cloneAndHypermutation, 29
  - EvaluateIndividual\_local, 29
  - Evaluation, 29
  - initPopulation, 29
  - NormalizeAffinity, 29
  - randomInsertion, 29
  - StatisticsReduction1, 29
- KernelParameters, 17
  - BITS\_PER\_DIMENSION, 18
  - CHROM\_LEN, 18
  - CLONING\_FACTOR, 18
  - DIMENSIONS, 18
  - LOWER\_LIM, 18
  - MUTATION\_FACTOR, 18
  - NCLON, 18
  - POP\_SIZE, 18
  - REAL\_LEN, 18
  - representation.h, 35
  - UPPER\_LIM, 18
- LOWER\_LIM
  - KernelParameters, 18
- LoadParameters
  - ClonalgCL, 13
- LoadSeeds
  - ClonalgCL, 13
- log\_arquivo
  - opencLUtils.h, 33
- log\_error
  - opencLUtils.h, 33
- log\_error\_code
  - opencLUtils.h, 33
- lowerLim
  - t\_parameters, 24
- m\_aux\_binary
  - Clonalg, 11
- m\_bestClone
  - Clonalg, 11
- m\_bitsperdimension
  - BaseClonalg, 8
- m\_bitsperparameter
  - BaseClonalg, 8
- m\_clone
  - Clonalg, 11
- m\_cloningfactor
  - BaseClonalg, 8
- m\_cpu\_count
  - ClonalgCL, 15
- m\_cpu\_queues
  - ClonalgCL, 15
- m\_cromLen
  - BaseClonalg, 8
- m\_dimensions
  - BaseClonalg, 8
- m\_fitness
  - BaseClonalg, 8
- m\_fitnessNorm
  - BaseClonalg, 9
- m\_generations
  - BaseClonalg, 9
- m\_gpu\_count
  - ClonalgCL, 15
- m\_gpu\_queues
  - ClonalgCL, 15



- m\_gpu\_ratio
  - ClonalgCL, 15
- m\_lowerlim
  - BaseClonalg, 9
- m\_mutationfactor
  - BaseClonalg, 9
- m\_numclones
  - BaseClonalg, 9
- m\_objective
  - BaseClonalg, 9
- m\_optimizationProblem
  - BaseClonalg, 9
- m\_pop
  - BaseClonalg, 9
- m\_pop\_size\_per\_queue
  - ClonalgCL, 15
- m\_popsiz
  - BaseClonalg, 9
- m\_randominsertion
  - BaseClonalg, 9
- m\_realLen
  - BaseClonalg, 9
- m\_statistics
  - BaseClonalg, 9
- m\_stats
  - ClonalgCL, 15
- m\_upperlim
  - BaseClonalg, 9
- m\_v
  - Clonalg, 11
- m\_vectorLength
  - ObjectiveFunction, 19
- m\_workGroupSize\_hypermutation
  - ClonalgCL, 15
- MASK
  - utils.cpp, 30
- MUTATION\_FACTOR
  - KernelParameters, 18
- main
  - main.cpp, 31
- main.cpp
  - GetArgs, 31
  - main, 31
- Mutate
  - Clonalg, 10
- mutationFactor
  - t\_parameters, 24
- NCLON
  - KernelParameters, 18
- nclones
  - t\_parameters, 24
- NormalizeAffinity
  - ClonalgCL, 13
  - kernel.cpp, 29
- normalizeAffinityKernel
  - ClonalgCL, 15
- ObjectiveFunction, 18
  - ~ObjectiveFunction, 19
  - Evaluate, 19
  - EvaluateFloat, 19
  - getFormula, 19
  - getName, 19
  - getSumLimit, 19
  - m\_vectorLength, 19
  - ObjectiveFunction, 19
  - ObjectiveFunction, 19
- OneMaxProblem, 19
  - ~OneMaxProblem, 20
  - Evaluate, 20
  - EvaluateFloat, 20
  - getFormula, 20
  - getName, 20
  - getSumLimit, 20
  - OneMaxProblem, 20
  - OneMaxProblem, 20
- OpenCLInit
  - ClonalgCL, 13
- OpenCLUtils, 20
  - ~OpenCLUtils, 21
  - CreateProgramFromBinary, 21
  - CreateProgramFromSource, 21
  - GetBinarySize, 21
  - getElapsedTime, 21
  - OpenCLUtils, 21
  - OpenCLUtils, 21
  - SaveProgramBinary, 21
  - ShowDeviceInfo, 21
- opencLUtils.c
  - clCompilaPrograma, 32
  - clObtemDispositivos, 32
  - clObtemPlataformas, 32
- opencLUtils.h
  - check\_cl, 33
  - clCompilaPrograma, 33
  - clObtemDispositivos, 33
  - clObtemPlataformas, 33
  - log\_arquivo, 33
  - log\_error, 33
  - log\_error\_code, 33
- optimizationProblem
  - t\_parameters, 24
- POP\_SIZE
  - KernelParameters, 18
- parallel
  - t\_parameters, 24
- parameters.h
  - BITS\_PER\_WORD, 33
- parametersBuffer
  - ClonalgCL, 16
- popBuffer
  - ClonalgCL, 16
- popsiz
  - t\_parameters, 24
- PrintPop
  - BaseClonalg, 8

- problemDimension
  - t\_parameters, 24
- ProblemFactory, 21
  - ~ProblemFactory, 22
  - CreateProblem, 22
  - ProblemFactory, 22
  - ProblemFactory, 22
- REAL\_LEN
  - KernelParameters, 18
- rand
  - utils.cpp, 30
- RandomInsertion
  - Clonalg, 10
  - ClonalgCL, 14
- randomInsertion
  - kernel.cpp, 29
  - t\_parameters, 24
- randomInsertionKernel
  - ClonalgCL, 16
- representation.h
  - BIT\_CHECK, 34
  - BIT\_CLEAR, 34
  - BIT\_FLIP, 34
  - BIT\_SET, 34
  - KernelParameters, 35
- RosenbrockObjectiveFunction, 22
  - ~RosenbrockObjectiveFunction, 22
  - Evaluate, 22
  - EvaluateFloat, 23
  - getFormula, 23
  - getName, 23
  - getSumLimit, 23
  - RosenbrockObjectiveFunction, 22
  - RosenbrockObjectiveFunction, 22
- SaveProgramBinary
  - OpenCLUtils, 21
- Search
  - BaseClonalg, 8
  - Clonalg, 10
  - ClonalgCL, 14
- seedBuffer
  - ClonalgCL, 16
- ShowDeviceInfo
  - OpenCLUtils, 21
- Sort
  - BaseClonalg, 8
- src/BaseClonalg.cpp, 27
- src/BaseClonalg.h, 27
- src/Clonalg.cpp, 27
- src/Clonalg.h, 27
- src/ClonalgCL.cpp, 28
- src/ClonalgCL.h, 28
- src/ElipsoidalObjectiveFunction.cpp, 28
- src/ElipsoidalObjectiveFunction.h, 28
- src/ObjectiveFunction.cpp, 31
- src/ObjectiveFunction.h, 31
- src/OneMaxProblem.cpp, 31
- src/OneMaxProblem.h, 31
- src/OpenCLUtils.cpp, 32
- src/OpenCLUtils.h, 33
- src/ProblemFactory.cpp, 34
- src/ProblemFactory.h, 34
- src/RosenbrockObjectiveFunction.cpp, 35
- src/RosenbrockObjectiveFunction.h, 35
- src/TimeUtils.h, 35
- src/kernels/kernel.cpp, 29
- src/kernels/utils.cpp, 30
- src/main.cpp, 30
- src/opencLUtils.c, 31
- src/opencLUtils.h, 32
- src/parameters.h, 33
- src/representation.h, 34
- Statistics
  - Clonalg, 10
  - ClonalgCL, 14
- statisticsBuffer
  - ClonalgCL, 16
- statisticsKernel
  - ClonalgCL, 16
- StatisticsReduction1
  - kernel.cpp, 29
- t\_parameters, 23
  - bitsPerDimension, 23
  - cpus, 23
  - generations, 23
  - gpuRatio, 23
  - gpus, 24
  - lowerLim, 24
  - mutationFactor, 24
  - nclones, 24
  - optimizationProblem, 24
  - parallel, 24
  - popsiz, 24
  - problemDimension, 24
  - randomInsertion, 24
  - upperLim, 24
- t\_stats, 24
  - affinityBest, 24
  - affinityWorst, 24
  - indexBest, 24
  - indexWorst, 24
- TimeUtils, 25
  - getRealTime, 25
- u\_rand
  - utils.cpp, 30
- UPPER\_LIM
  - KernelParameters, 18
- upperLim
  - t\_parameters, 24
- utils.cpp
  - BIT\_CHECK, 30
  - BIT\_CLEAR, 30
  - BIT\_FLIP, 30
  - BIT\_SET, 30

CONST1, [30](#)  
CONST2, [30](#)  
MASK, [30](#)  
rand, [30](#)  
u\_rand, [30](#)