



# MDEA

MDEA

0.1

Generated by Doxygen 1.8.9.1

Mon Jun 20 2016 09:31:59



# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Génération de modèles diversifiés à l'aide d'algorithmes génétiques</b> | <b>1</b>  |
| <b>2</b> | <b>Deprecated List</b>   | <b>3</b>  |
| <b>3</b> | <b>Hierarchical Index</b>  | <b>5</b>  |
| 3.1      | Class Hierarchy . . . . .  | 5         |
| <b>4</b> | <b>Class Index</b>   | <b>7</b>  |
| 4.1      | Class List . . . . .   | 7         |
| <b>5</b> | <b>File Index</b>  | <b>9</b>  |
| 5.1      | File List . . . . .  | 9         |
| <b>6</b> | <b>Class Documentation</b>   | <b>11</b> |
| 6.1      | Chromosom Class Reference . . . . .  | 11        |
| 6.1.1    | Detailed Description . . . . .   | 12        |
| 6.1.2    | Constructor & Destructor Documentation . . . . .                           | 12        |
| 6.1.2.1  | Chromosom . . . . .  | 12        |
| 6.1.2.2  | Chromosom . . . . .  | 12        |
| 6.1.2.3  | ~Chromosom . . . . .   | 12        |
| 6.1.3    | Member Function Documentation . . . . .                                    | 12        |
| 6.1.3.1  | getModels . . . . .  | 12        |
| 6.1.3.2  | getNbModels . . . . .  | 12        |
| 6.1.3.3  | isValid . . . . .  | 12        |
| 6.1.3.4  | mutate . . . . .   | 13        |
| 6.1.3.5  | setModels . . . . .  | 14        |
| 6.1.3.6  | setNbModels . . . . .  | 14        |
| 6.2      | CommandLine Class Reference . . . . .                                      | 14        |
| 6.2.1    | Detailed Description . . . . .   | 14        |
| 6.2.2    | Constructor & Destructor Documentation . . . . .                           | 15        |
| 6.2.2.1  | CommandLine . . . . .  | 15        |
| 6.2.3    | Member Function Documentation . . . . .                                    | 16        |
| 6.2.3.1  | addOption . . . . .  | 16        |

|          |  |    |
|----------|--|----|
| 6.2.3.2  | <a href="#">addOption</a>                                    | 16 |
| 6.2.3.3  | <a href="#">parse</a>  | 16 |
| 6.2.3.4  | <a href="#">printUsage</a>                                   | 16 |
| 6.3      | <a href="#">Directory Class Reference</a>                    | 16 |
| 6.3.1    | <a href="#">Detailed Description</a>                         | 17 |
| 6.3.2    | <a href="#">Constructor &amp; Destructor Documentation</a>   | 17 |
| 6.3.2.1  | <a href="#">Directory</a>                                    | 17 |
| 6.3.2.2  | <a href="#">~Directory</a>                                   | 17 |
| 6.3.3    | <a href="#">Member Function Documentation</a>                | 17 |
| 6.3.3.1  | <a href="#">getFiles</a>                                     | 17 |
| 6.3.3.2  | <a href="#">getNbFiles</a>                                   | 17 |
| 6.4      | <a href="#">GACHromosom Class Reference</a>                  | 18 |
| 6.4.1    | <a href="#">Detailed Description</a>                         | 19 |
| 6.4.2    | <a href="#">Constructor &amp; Destructor Documentation</a>   | 19 |
| 6.4.2.1  | <a href="#">GACHromosom</a>                                  | 19 |
| 6.4.2.2  | <a href="#">GACHromosom</a>                                  | 19 |
| 6.4.2.3  | <a href="#">GACHromosom</a>                                  | 19 |
| 6.4.2.4  | <a href="#">~GACHromosom</a>                                 | 19 |
| 6.4.3    | <a href="#">Member Function Documentation</a>                | 19 |
| 6.4.3.1  | <a href="#">avgEvaluator</a>                                 | 19 |
| 6.4.3.2  | <a href="#">clone</a>  | 20 |
| 6.4.3.3  | <a href="#">compare</a>                                      | 20 |
| 6.4.3.4  | <a href="#">copy</a>   | 20 |
| 6.4.3.5  | <a href="#">evaluate</a>                                     | 20 |
| 6.4.3.6  | <a href="#">evaluate</a>                                     | 20 |
| 6.4.3.7  | <a href="#">init</a>   | 21 |
| 6.4.3.8  | <a href="#">mutate</a>                                       | 21 |
| 6.4.3.9  | <a href="#">onePointCrossover</a>                            | 21 |
| 6.4.3.10 | <a href="#">score</a>  | 21 |
| 6.4.3.11 | <a href="#">score</a>  | 22 |
| 6.4.3.12 | <a href="#">setModels</a>                                    | 22 |
| 6.4.3.13 | <a href="#">to_string</a>                                    | 22 |
| 6.4.3.14 | <a href="#">write</a>  | 22 |
| 6.5      | <a href="#">Interval&lt; T &gt; Class Template Reference</a> | 22 |
| 6.5.1    | <a href="#">Detailed Description</a>                         | 23 |
| 6.5.2    | <a href="#">Constructor &amp; Destructor Documentation</a>   | 23 |
| 6.5.2.1  | <a href="#">Interval</a>                                     | 23 |
| 6.5.2.2  | <a href="#">Interval</a>                                     | 23 |
| 6.5.2.3  | <a href="#">~Interval</a>                                    | 23 |
| 6.5.3    | <a href="#">Member Function Documentation</a>                | 23 |

|         |  |    |
|---------|--|----|
| 6.5.3.1 | <a href="#">include</a>  | 23 |
| 6.5.3.2 | <a href="#">lbound</a>   | 23 |
| 6.5.3.3 | <a href="#">to_string</a>  | 24 |
| 6.5.3.4 | <a href="#">ubound</a>   | 24 |
| 6.6     | <a href="#">IntervalVector&lt; T &gt; Class Template Reference</a> | 24 |
| 6.6.1   | <a href="#">Detailed Description</a>                               | 24 |
| 6.6.2   | <a href="#">Constructor &amp; Destructor Documentation</a>         | 24 |
| 6.6.2.1 | <a href="#">IntervalVector</a>                                     | 24 |
| 6.6.2.2 | <a href="#">IntervalVector</a>                                     | 25 |
| 6.6.2.3 | <a href="#">~IntervalVector</a>                                    | 25 |
| 6.6.3   | <a href="#">Member Function Documentation</a>                      | 25 |
| 6.6.3.1 | <a href="#">add</a>  | 25 |
| 6.6.3.2 | <a href="#">add</a>  | 25 |
| 6.6.3.3 | <a href="#">add</a>  | 25 |
| 6.6.3.4 | <a href="#">include</a>  | 25 |
| 6.6.3.5 | <a href="#">lbound</a>   | 26 |
| 6.6.3.6 | <a href="#">to_string</a>  | 26 |
| 6.6.3.7 | <a href="#">ubound</a>   | 26 |
| 6.7     | <a href="#">Logger Class Reference</a>                             | 26 |
| 6.7.1   | <a href="#">Detailed Description</a>                               | 27 |
| 6.7.2   | <a href="#">Constructor &amp; Destructor Documentation</a>         | 27 |
| 6.7.2.1 | <a href="#">Logger</a>   | 27 |
| 6.7.2.2 | <a href="#">~Logger</a>  | 27 |
| 6.7.3   | <a href="#">Member Function Documentation</a>                      | 27 |
| 6.7.3.1 | <a href="#">operator&lt;&lt;</a>                                   | 27 |
| 6.7.3.2 | <a href="#">operator&lt;&lt;</a>                                   | 27 |
| 6.7.3.3 | <a href="#">operator&lt;&lt;</a>                                   | 27 |
| 6.7.3.4 | <a href="#">write</a>  | 28 |
| 6.8     | <a href="#">Matrix Class Reference</a>                             | 28 |
| 6.8.1   | <a href="#">Detailed Description</a>                               | 28 |
| 6.8.2   | <a href="#">Constructor &amp; Destructor Documentation</a>         | 29 |
| 6.8.2.1 | <a href="#">Matrix</a>   | 29 |
| 6.8.2.2 | <a href="#">~Matrix</a>  | 30 |
| 6.8.3   | <a href="#">Member Function Documentation</a>                      | 30 |
| 6.8.3.1 | <a href="#">average</a>  | 30 |
| 6.8.3.2 | <a href="#">isPositive</a>   | 30 |
| 6.8.3.3 | <a href="#">lineAverage</a>  | 30 |
| 6.8.3.4 | <a href="#">min</a>  | 30 |
| 6.8.3.5 | <a href="#">operator"</a>  | 30 |
| 6.8.3.6 | <a href="#">operator-</a>  | 31 |

|          |  |    |
|----------|--|----|
| 6.8.3.7  | operator>                              | 32 |
| 6.8.3.8  | operator>=                             | 32 |
| 6.8.3.9  | operator[]                             | 32 |
| 6.8.3.10 | set                                    | 32 |
| 6.8.3.11 | size                                   | 33 |
| 6.8.3.12 | to_csv_string                          | 33 |
| 6.8.3.13 | to_string                              | 33 |
| 6.9      | Model Class Reference                  | 33 |
| 6.9.1    | Detailed Description                   | 34 |
| 6.9.2    | Constructor & Destructor Documentation | 34 |
| 6.9.2.1  | Model                                  | 34 |
| 6.9.2.2  | Model                                  | 34 |
| 6.9.2.3  | Model                                  | 34 |
| 6.9.2.4  | ~Model                                 | 34 |
| 6.9.3    | Member Function Documentation          | 35 |
| 6.9.3.1  | evaluate                               | 35 |
| 6.9.3.2  | evaluateExtern                         | 36 |
| 6.9.3.3  | generateDotFile                        | 36 |
| 6.9.3.4  | getDomains                             | 36 |
| 6.9.3.5  | getVal                                 | 36 |
| 6.9.3.6  | isValid                                | 36 |
| 6.9.3.7  | operator!=                             | 36 |
| 6.9.3.8  | operator==                             | 37 |
| 6.9.3.9  | setDomains                             | 37 |
| 6.9.3.10 | setVal                                 | 37 |
| 6.9.3.11 | to_string                              | 37 |
| 6.10     | NSGAI Class Reference                  | 37 |
| 6.10.1   | Detailed Description                   | 38 |
| 6.10.2   | Member Enumeration Documentation       | 38 |
| 6.10.2.1 | Fitness                                | 38 |
| 6.10.3   | Constructor & Destructor Documentation | 38 |
| 6.10.3.1 | NSGAI                                  | 38 |
| 6.10.3.2 | ~NSGAI                                 | 38 |
| 6.10.4   | Member Function Documentation          | 39 |
| 6.10.4.1 | crowdingDistanceAssignment             | 39 |
| 6.10.4.2 | dominate                               | 39 |
| 6.10.4.3 | fastNonDominatedSort                   | 39 |
| 6.10.4.4 | operator++                             | 39 |
| 6.10.4.5 | step                                   | 39 |
| 6.10.5   | Member Data Documentation              | 39 |

|          |  |           |
|----------|--|-----------|
| 6.10.5.1 | fitness                                | 39        |
| 6.10.5.2 | gen                                    | 40        |
| 6.11     | Population Class Reference             | 40        |
| 6.11.1   | Detailed Description                   | 40        |
| 6.11.2   | Constructor & Destructor Documentation | 40        |
| 6.11.2.1 | Population                             | 40        |
| 6.11.2.2 | Population                             | 40        |
| 6.11.3   | Member Function Documentation          | 40        |
| 6.11.3.1 | best                                   | 40        |
| 6.11.3.2 | evaluate                               | 41        |
| 6.11.3.3 | worst                                  | 41        |
| 6.12     | Statistics Class Reference             | 41        |
| 6.12.1   | Detailed Description                   | 42        |
| 6.12.2   | Member Function Documentation          | 42        |
| 6.12.2.1 | getFirstQuartile                       | 42        |
| 6.12.2.2 | getMax                                 | 42        |
| 6.12.2.3 | getMed                                 | 42        |
| 6.12.2.4 | getMin                                 | 42        |
| 6.12.2.5 | getThirdQuartile                       | 42        |
| 6.12.2.6 | update                                 | 42        |
| 6.12.3   | Member Data Documentation              | 43        |
| 6.12.3.1 | outfile                                | 43        |
| <b>7</b> | <b>File Documentation</b>              | <b>45</b> |
| 7.1      | model/Chromosom.h File Reference       | 45        |
| 7.1.1    | Detailed Description                   | 45        |
| 7.2      | model/GAChromosom.h File Reference     | 45        |
| 7.2.1    | Detailed Description                   | 46        |
| 7.2.2    | Typedef Documentation                  | 46        |
| 7.2.2.1  | MatrixPtr                              | 46        |
| 7.3      | model/Matrix.h File Reference          | 46        |
| 7.3.1    | Detailed Description                   | 47        |
| 7.3.2    | Function Documentation                 | 47        |
| 7.3.2.1  | operator<<                             | 47        |
| 7.3.2.2  | operator<<                             | 47        |
| 7.4      | model/Model.h File Reference           | 48        |
| 7.4.1    | Detailed Description                   | 48        |
| 7.4.2    | Typedef Documentation                  | 48        |
| 7.4.2.1  | Domains                                | 48        |
| 7.4.2.2  | DomainsPtr                             | 48        |

|              |                                       |           |
|--------------|---------------------------------------|-----------|
| 7.4.2.3      | Genes                                 | 49        |
| 7.4.2.4      | GenesPtr                              | 49        |
| 7.4.3        | Function Documentation                | 49        |
| 7.4.3.1      | operator<<                            | 49        |
| 7.5          | model/NSGAll.h File Reference         | 49        |
| 7.5.1        | Detailed Description                  | 49        |
| 7.6          | model/Population.h File Reference     | 50        |
| 7.6.1        | Detailed Description                  | 50        |
| 7.6.2        | Function Documentation                | 50        |
| 7.6.2.1      | operator<<                            | 50        |
| 7.7          | model/Statistics.h File Reference     | 51        |
| 7.7.1        | Detailed Description                  | 51        |
| 7.7.2        | Function Documentation                | 51        |
| 7.7.2.1      | operator<<                            | 51        |
| 7.8          | utils/CommandLine.h File Reference    | 51        |
| 7.8.1        | Detailed Description                  | 52        |
| 7.9          | utils/Directory.h File Reference      | 52        |
| 7.9.1        | Detailed Description                  | 52        |
| 7.10         | utils/Interval.h File Reference       | 53        |
| 7.10.1       | Detailed Description                  | 53        |
| 7.11         | utils/IntervalVector.h File Reference | 53        |
| 7.11.1       | Detailed Description                  | 53        |
| 7.12         | utils/Logger.h File Reference         | 54        |
| 7.12.1       | Detailed Description                  | 54        |
| <b>Index</b> |                                       | <b>55</b> |



# Chapter 1

## Génération de modèles diversifiés à l'aide d'algorithmes génétiques

Ce projet est un projet de recherche réalisé au Laboratoire d'Informatique, de Robotique et Microélectronique de Montpellier (UMR CNRS et Université de Montpellier). Un résumé est donné ci-dessous.

### Résumé/Abstract

#### Français

La génération de modèles à partir de méta-modèles est une solution apportée par l'ingénierie des modèles pour tester les transformations de modèles ou pour la validation de méta-modèles. Il existe dans la littérature un certain nombre de techniques combinatoires pour l'instanciation de méta-modèle, mais peu se sont intéressées au problème de la diversité. L'utilisation des algorithmes génétiques est une des plus prometteuses en terme de diversité. Durant nos travaux, nous avons expérimentés deux approches, basées sur l'algorithme évolutionnaire NSGA-II, pour la génération de nouveaux modèles, plus diversifiés. Les résultats ainsi obtenus mettent en évidence une réelle augmentation de la distance entre modèles, permettant de fournir des bancs d'essais plus réalistes et plus complets que ceux initiaux.

#### English

Generation of models that conform to a meta-model is one of the solutions from model driven engineering for models transformations testing or meta-models validation. Some combinatorial techniques have been explored in previous works, but few have focused on problem of diversity. Amongst previous works, genetic algorithms usage seems to be one of the most promising for diversity generation. In this work, we experiment two approach, based on the evolutionary algorithm NSGA-II, to generate more diverse models. These approaches provide results with an increased distance, enabling creation of more realistic and complete benchmark.

### Spécification

L'objectif de cette application est de prendre en entrée un certain nombre de modèles préexistants et d'en générer des nouveaux, plus diversifiés.

Syntaxe : “ mdea -in <models directory>=”> [-cx <inter|intra>] [-dist <cosine|hamming|centrality>] [-fitness <min|avg|minavg|minavgs|dist>] [-g <number of=” generations>=”>] [-m <percentage of=” mutation=” chance>=”>] [-nb <number of=” model=” in=” a=” single=” chromosom>=”>] [-out <output directory>=”>] “

Les fichiers de définition de modèles doivent être définis en suivant la syntaxe suivante : “ Vecteur (entiers séparés par des espaces) Emplacement du fichier XCSP Emplacement du fichier .grimm Emplacement du méta-modèle Racine du méta-modèle “

Generated on Mon Jun 20 2016 09:31:59 for MDEA by Doxygen

## Chapter 2

# Deprecated List

Member [Model::setDomains](#) (DomainsPtr d)

Parameters

|     |  |
|-----|--|
| $v$ | The new value of <a href="#">Model</a> |
|-----|--|

Member [Model::setVal](#) (GenesPtr v)

Parameters

|     |  |
|-----|--|
| $v$ | The new value of <a href="#">Model</a> |
|-----|--|



## Chapter 3

# Hierarchical Index

### 3.1 Class Hierarchy

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

|                               |    |
|-------------------------------|----|
| Chromosom . . . . .           | 11 |
| GChromosom . . . . .          | 18 |
| CommandLine . . . . .         | 14 |
| Directory . . . . .           | 16 |
| GAGenome                      |    |
| GChromosom . . . . .          | 18 |
| GAPopulation                  |    |
| Population . . . . .          | 40 |
| GASimpleGA                    |    |
| NSGAI . . . . .               | 37 |
| Interval< T > . . . . .       | 22 |
| IntervalVector< T > . . . . . | 24 |
| Logger . . . . .              | 26 |
| Matrix . . . . .              | 28 |
| Model . . . . .               | 33 |
| Statistics . . . . .          | 41 |



## Chapter 4

# Class Index

### 4.1 Class List

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

|   |   |    |
|---|---|----|
| <a href="#">Chromosom</a>                 | Class that represent a genome in our problem . . . . .                    | 11 |
| <a href="#">CommandLine</a>               | Class for command line options parsing . . . . .                          | 14 |
| <a href="#">Directory</a>                 | This class provide methods for directory manipulation . . . . .           | 16 |
| <a href="#">GAChromosom</a>               | Class that implement a genome for our problem . . . . .                   | 18 |
| <a href="#">Interval&lt; T &gt;</a>       | An interval representation . . . . .                                      | 22 |
| <a href="#">IntervalVector&lt; T &gt;</a> | A set of disjoint intervals . . . . .                                     | 24 |
| <a href="#">Logger</a>                    | Class that allow to log information in a file . . . . .                   | 26 |
| <a href="#">Matrix</a>                    | Class that implement a square matrix . . . . .                            | 28 |
| <a href="#">Model</a>                     | Class that represent a model . . . . .                                    | 33 |
| <a href="#">NSGAI</a>                     | Implementation of NSGA-II algorithm . . . . .                             | 37 |
| <a href="#">Population</a>                | This class manage all chromosomes (especially the initial ones) . . . . . | 40 |
| <a href="#">Statistics</a>                | A class to store genetic algorithm stats . . . . .                        | 41 |





## Chapter 5

# File Index

### 5.1 File List

Here is a list of all documented files with brief descriptions:

|   |    |
|---|----|
| model/ <a href="#">Chromosom.h</a>                    |    |
| <a href="#">Chromosom</a> class header . . . . .      | 45 |
| model/ <a href="#">GAChromosom.h</a>                  |    |
| <a href="#">GAChromosom</a> class header . . . . .    | 45 |
| model/ <a href="#">Matrix.h</a>                       |    |
| <a href="#">Matrix</a> class header . . . . .         | 46 |
| model/ <a href="#">Model.h</a>                        |    |
| <a href="#">Model</a> class header . . . . .          | 48 |
| model/ <a href="#">NSGAI.h</a>                        |    |
| <a href="#">NSGAI</a> class header . . . . .          | 49 |
| model/ <a href="#">Population.h</a>                   |    |
| <a href="#">Population</a> class header . . . . .     | 50 |
| model/ <a href="#">Statistics.h</a>                   |    |
| <a href="#">Statistics</a> class header . . . . .     | 51 |
| utils/ <a href="#">CommandLine.h</a>                  |    |
| <a href="#">CommandLine</a> class header . . . . .    | 51 |
| utils/ <a href="#">Directory.h</a>                    |    |
| <a href="#">Directory</a> class header . . . . .      | 52 |
| utils/ <a href="#">Interval.h</a>                     |    |
| <a href="#">Interval</a> class header . . . . .       | 53 |
| utils/ <a href="#">IntervalVector.h</a>               |    |
| <a href="#">IntervalVector</a> class header . . . . . | 53 |
| utils/ <a href="#">Logger.h</a>                       |    |
| <a href="#">Logger</a> class header . . . . .         | 54 |



## Chapter 6

# Class Documentation

### 6.1 Chromosom Class Reference

Class that represent a genome in our problem.

```
#include <Chromosom.h>
```

Inheritance diagram for Chromosom:

#### Public Member Functions

- [Chromosom](#) ()
- [Chromosom](#) (const std::vector< std::string > &files)
- virtual [~Chromosom](#) () noexcept
- virtual const std::vector< [Model](#) > & [getModels](#) () const
- virtual void [setModels](#) (std::vector< [Model](#) > models)
- virtual int [mutate](#) (float pmut)
- virtual bool [isValid](#) () const

#### Static Public Member Functions

- static size\_t [setNbModels](#) (size\_t)
- static size\_t [getNbModels](#) ()

#### Public Attributes

- int **front**

#### Protected Attributes

- std::vector< [Model](#) > **models**

#### Static Protected Attributes

- static size\_t **nbModels** = 2

### 6.1.1 Detailed Description

Class that represent a genome in our problem.

This class is a genome representation of our problem.

#### Author

Florian Galinier

### 6.1.2 Constructor & Destructor Documentation

#### 6.1.2.1 Chromosom::Chromosom (

Create a new empty chromosom.

#### 6.1.2.2 Chromosom::Chromosom (const std::vector<std::string> & files )

Create a new chromosom with data from input files.

#### Parameters

|             |   |
|-------------|---|
| <i>file</i> | The file path that contains vector value for chromosom. |
|-------------|---|

#### 6.1.2.3 Chromosom::~Chromosom () [virtual],[noexcept]

Destructor for [Chromosom](#).

### 6.1.3 Member Function Documentation

#### 6.1.3.1 const std::vector<Model> & Chromosom::getModels() const [virtual]

Return vector of models

#### Returns

The vector of models

#### 6.1.3.2 size\_t Chromosom::getNbModels() [static]

Change actual value of [Chromosom](#)

#### Parameters

|          |  |
|----------|--|
| <i>v</i> | The new value of <a href="#">Chromosom</a> |
|----------|--|

#### 6.1.3.3 bool Chromosom::isValid() const [virtual]

Return true if the chromosom own only valid models.

#### Returns

true iff the chromosom is valid.

#### 6.1.3.4 int Chromosom::mutate(*pmut*) [virtual]

Mutate models with a chance of *pmut* percent.

**Parameters**

|             |                                   |
|-------------|-----------------------------------|
| <i>pmut</i> | The percentage chance of mutation |
|-------------|-----------------------------------|

**Returns**

The number of mutation

**6.1.3.5 void Chromosom::setModels([vector<Model>](#) *models*)** [virtual]

Change actual value of [Chromosom](#)

**Parameters**

|               |  |
|---------------|--|
| <i>models</i> | The new value of <a href="#">Chromosom</a> |
|---------------|--|

Reimplemented in [GAChromosom](#).

**6.1.3.6 size\_t Chromosom::setNbModels(*nb*)** [static]

Change actual value of [Chromosom](#)

**Parameters**

|          |  |
|----------|--|
| <i>v</i> | The new value of <a href="#">Chromosom</a> |
|----------|--|

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

- [model/Chromosom.h](#)
- [model/Chromosom.cpp](#)

## 6.2 CommandLine Class Reference

Class for command line options parsing.

```
#include <CommandLine.h>
```

**Public Member Functions**

- [CommandLine](#) (std::string app)
- void [addOption](#) (std::string option, std::string help, bool required=true)
- void [addOption](#) (std::pair< std::string, std::string > option, bool required=true)
- void [printUsage](#) () const
- std::map< std::string, std::string > [\\*parse](#) (int argc, char \* ~~argv~~ argv) const

### 6.2.1 Detailed Description

Class for command line options parsing.

This class allow usage of command line interface by parsing options.

**Author**

Florian Galinier

## 6.2.2 Constructor & Destructor Documentation

### 6.2.2.1 CommandLine::CommandLine(*string app*)

Constructor for new [CommandLine](#) parser.

**Parameters**

|            |                       |
|------------|-----------------------|
| <i>app</i> | The application name. |
|------------|-----------------------|

**6.2.3 Member Function Documentation****6.2.3.1 void CommandLine::addOption(*std::string option*, *std::string help*, *bool required* = true )**

Add an option to parse.

**Parameters**

|                 |  |
|-----------------|--|
| <i>option</i>   | The option value.  |
| <i>help</i>     | The help to display with this option.  |
| <i>required</i> | Specify if option is optional or required (true for required and by default) |

**6.2.3.2 void CommandLine::addOption(*std::pair<std::string, std::string>* option, *bool required* = true )**

Add option method (see [addOption\(std::string, std::string, bool\)](#)) with *std::pair*.

**Parameters**

|                 |  |
|-----------------|--|
| <i>option</i>   | <i>std::pair</i> composed of option value and help to display.               |
| <i>required</i> | Specify if option is optional or required (true for required and by default) |

**6.2.3.3 *std::map<std::string, std::string>* CommandLine::parse(*argc* *char\* \*argv*) const**

Parse command line options.

**Parameters**

|             |                                |
|-------------|--------------------------------|
| <i>argc</i> | The main <i>argc</i> argument. |
| <i>argv</i> | The main <i>argv</i> argument. |

**Returns**

Options passed in command.

**6.2.3.4 void CommandLine::printUsage(*const* *std::string* &msg)**

Print usage method. This method will exit program.

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

- [utils/CommandLine.h](#)
- [utils/CommandLine.cpp](#)

**6.3 Directory Class Reference**

This class provide methods for directory manipulation.

```
#include <Directory.h>
```



## Public Member Functions

- [Directory](#) (std::string path)
- virtual [~Directory](#) ()
- virtual size\_t [getNbFiles](#) (std::string ext="") const
- virtual std::shared\_ptr< std::vector< std::string > > [getFiles](#) (std::string ext="") const

### 6.3.1 Detailed Description

This class provide methods for directory manipulation.

This class provide methods for directory manipulation.

#### Author

Florian Galinier

### 6.3.2 Constructor & Destructor Documentation

#### 6.3.2.1 Directory::Directory (std::string path)

Default constructor with directory path.

##### Parameters

|             |                    |
|-------------|--------------------|
| <i>path</i> | The directory path |
|-------------|--------------------|

#### 6.3.2.2 Directory::Directory () [virtual]

[Directory](#) destructor

### 6.3.3 Member Function Documentation

#### 6.3.3.1 std::shared\_ptr< std::vector< std::string > > Directory::getFiles (std::string ext="" ) const [virtual]

Return a vector of all files in directory.

##### Parameters

|            |                          |
|------------|--------------------------|
| <i>ext</i> | Optional file extension. |
|------------|--------------------------|

##### Returns

A vector with files path.

#### 6.3.3.2 size\_t Directory::getNbFiles (std::string ext="" ) const [virtual]

Return the number of files in the directory.

##### Parameters

|            |                          |
|------------|--------------------------|
| <i>ext</i> | Optional file extension. |
|------------|--------------------------|

## Returns

The number of files in directory.

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

- [utils/Directory.h](#)
- [utils/Directory.cpp](#)

## 6.4 GACHromosom Class Reference

Class that implement a genome for our problem.

```
#include <GACHromosom.h>
```

Inheritance diagram for GACHromosom:

Collaboration diagram for GACHromosom:

## Public Types

- enum **Method** { **COSINE**, **HAMMING**, **CENTRALITY** }
- enum **Cross** { **INTRA**, **INTER** }

## Public Member Functions

- **GADefineIdentity** ("GACHromosom", 201)
- [GACHromosom](#) ()
- [GACHromosom](#) (const std::vector< std::string > &files)
- [GACHromosom](#) (const [GACHromosom](#) &g)
- virtual [~GACHromosom](#) () noexcept
- virtual [MatrixPtr](#) [score](#) ([MatrixPtr](#) m)
- virtual [MatrixPtr](#) [score](#) () const
- virtual [GAGenome](#) \*[clone](#) (CloneMethod method) const
- virtual void [copy](#) (const [GAGenome](#) &g)
- virtual void [setModels](#) (std::vector< [Model](#) > models)
- std::string [to\\_string](#) () const
- virtual int [write](#) (std::ostream &os) const
- virtual [MatrixPtr](#) [evaluate](#) ([GABoolean](#) gaFlag=gaFalse)

## Static Public Member Functions

- static float [compare](#) (const [GAGenome](#) &g1, const [GAGenome](#) &g2)
- static void [init](#) ([GAGenome](#) &g)
- static int [onePointIntraCrossover](#) (const [GAGenome](#) &dad, const [GAGenome](#) &mom, [GAGenome](#) \*[sis](#), [GAGenome](#) \*[bro](#))
- static int [onePointCrossover](#) (const [GAGenome](#) &dad, const [GAGenome](#) &mom, [GAGenome](#) \*[sis](#), [GAGenome](#) \*[bro](#))
- static float [avgEvaluator](#) ([GAGenome](#) &g)
- static [MatrixPtr](#) [evaluate](#) ([GAGenome](#) &g)
- static int [mutate](#) ([GAGenome](#) &g, float pmut)

## Static Public Attributes

- static enum GChromosom::Method **method** = GChromosom::Method::COSINE
- static enum GChromosom::Cross **crossover** = GChromosom::Cross::INTER

## Additional Inherited Members

### 6.4.1 Detailed Description

Class that implement a genome for our problem.

This class implement the GAGenome interface of GALib API and is a modelisation of our problem.

#### Author

Florian Galinier

### 6.4.2 Constructor & Destructor Documentation

#### 6.4.2.1 GChromosom::GChromosom (

Default constructor that create a new genome.

#### 6.4.2.2 GChromosom::GChromosom (const std::vector<std::string & files )

Create a new genome from files.

##### Parameters

|             |                          |
|-------------|--------------------------|
| <i>path</i> | The vector of files path |
|-------------|--------------------------|

#### 6.4.2.3 GChromosom::GChromosom (const GChromosom & g )

Create genome that is a copy of parameter

##### Parameters

|          |                       |
|----------|-----------------------|
| <i>g</i> | The chromosom to copy |
|----------|-----------------------|

#### 6.4.2.4 GChromosom::GChromosom ( [virtual], [noexcept]

[GChromosom](#) destructor

### 6.4.3 Member Function Documentation

#### 6.4.3.1 float GChromosom::avgEvaluation (const GAGenome & g) [static]

Give an average fitness value for the given [GChromosom](#) (greater is better)

##### Parameters

|          |   |
|----------|---|
| <i>g</i> | The <a href="#">GAChromosom</a> to evaluate |
|----------|---|

**Returns**

The fitness value of the [GAChromosom](#)

**6.4.3.2 GAGenome & GAChromosom::clone (const Method method) const [virtual]**

Return a new GAGenome that is a copy of actual one.

**Parameters**

|               |   |
|---------------|---|
| <i>method</i> | Clone method to use for copy (optional) |
|---------------|---|

**Returns**

A clone of actual [GAChromosom](#)

**6.4.3.3 float GAChromosom::compare (const GAGenome & g1, const GAGenome & g2) [static]**

Function that compare two [GAChromosom](#) and return a distance value.

**Parameters**

|           |                            |
|-----------|----------------------------|
| <i>g1</i> | First GAGenome to compare  |
| <i>g2</i> | Second GAGenome to compare |

**Returns**

A float that is equal to 0 if g1 and g2 are the same, -1 if an error occurred and another float greater than 0 that is a degree of difference if g1 and g2 can be compared.

**6.4.3.4 void GAChromosom::copy (const GAGenome & g) [virtual]**

Change actual [GAChromosom](#) by copying g values.

**Parameters**

|          |                  |
|----------|------------------|
| <i>g</i> | GAGenome to copy |
|----------|------------------|

**6.4.3.5 MatrixPtr GAChromosom::evaluate (const Boolean gaFlag = false) [virtual]**

Give a fitness value of the actual [GAChromosom](#)

**Parameters**

|               |                             |
|---------------|-----------------------------|
| <i>gaFlag</i> | Force or not the evaluation |
|---------------|-----------------------------|

**Returns**

The fitness value of the [GAChromosom](#)

**6.4.3.6 MatrixPtr GAChromosom::evaluate (const GAGenome & g) [static]**

Give a matrix of distance value for the given [GAChromosom](#) (greater is better)

**Parameters**

|          |   |
|----------|---|
| <i>g</i> | The <a href="#">GACHromosom</a> to evaluate |
|----------|---|

**Returns**

The fitness value of the [GACHromosom](#)

**6.4.3.7 void GACHromosom::init(GAGenome & g) [static]**

Initialize a new [GACHromosom](#).

**Parameters**

|          |   |
|----------|---|
| <i>g</i> | The <a href="#">GACHromosom</a> to initialize |
|----------|---|

**6.4.3.8 int GACHromosom::mutate(GAGenome & g, float pmut) [static]**

Mutate the given [GACHromosom](#).

**Parameters**

|             |   |
|-------------|---|
| <i>g</i>    | The <a href="#">GACHromosom</a> to mutate |
| <i>pmut</i> | The percentage chance of mutation         |

**Returns**

The number of mutation

**6.4.3.9 int GACHromosom::onePointCrossover(GAGenome & dad, GAGenome & mom, GAGenome & sis, GAGenome & bro) [static]**

Cross two [GACHromosom](#) and return two new one by tranlocation.

**Parameters**

|            |  |
|------------|--|
| <i>dad</i> | First <a href="#">GACHromosom</a> to cross |
| <i>mom</i> | Second GACHoromosom to cross               |
| <i>sis</i> | The first offspring crossed chromosom      |
| <i>bro</i> | The second offspring crossed chromosom     |

**Returns**

Number of new chromosom obtain by crossover

**6.4.3.10 MatrixPtr GACHromosom::score(MatrixPtr m) [virtual]**

Set the matrix score.

**Parameters**

|          |                       |
|----------|-----------------------|
| <i>m</i> | The new matrix score. |
|----------|-----------------------|

**Returns**

The new matrix score.

#### 6.4.3.11 MatrixPtr GChromosom::score() const [virtual]

Return the matrix score (if matrix is null, score is not yet set).

##### Returns

The matrix score.

#### 6.4.3.12 void GChromosom::setModels(vector<Model> &models) [virtual]

Change actual value of [GChromosom](#)

##### Parameters

|               |   |
|---------------|---|
| <i>models</i> | The new value of <a href="#">GChromosom</a> |
|---------------|---|

Reimplemented from [Chromosom](#).

#### 6.4.3.13 std::string GChromosom::to\_string() const

Convert chromosom to a string representation.

##### Returns

A string representation of current chromosom.

#### 6.4.3.14 int GChromosom::write(ostream &os) const [virtual]

Print [GChromosom](#) value in param stream.

##### Parameters

|           |                           |
|-----------|---------------------------|
| <i>os</i> | The stream where to write |
|-----------|---------------------------|

##### Returns

Undefined

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

- model/[GChromosom.h](#)
- model/[GChromosom.cpp](#)

## 6.5 Interval<T> Class Template Reference

An interval representation.

```
#include <Interval.h>
```

### Public Member Functions

- [Interval](#) (const T &min, const T &max)
- [Interval](#) (const std::string &bounds)
- virtual [~Interval](#) ()
- virtual const T & [lbound](#) () const
- virtual const T & [ubound](#) () const
- virtual bool [include](#) (const T &a) const
- virtual std::string [to\\_string](#) () const

### 6.5.1 Detailed Description

**template<typename>Tclass Interval<T>**

An interval representation.

This class is an interval representation. Every class that implement comparison operator can be used as template type.

#### Author

Florian Galinier

### 6.5.2 Constructor & Destructor Documentation

**6.5.2.1 template<typename>TInterval< T>::Interval ( const T & *min*, const T & *max* )**[\[inline\]](#)

Constructor for a new interval.

#### Parameters

|            |                         |
|------------|-------------------------|
| <i>min</i> | Lower bound of interval |
| <i>max</i> | Upper bound of interval |

**6.5.2.2 template<typename>TInterval< T>::Interval ( const std::string & *bounds* )**[\[inline\]](#)

Parse the string and construct new interval. String must be of type N or N..M

#### Parameters

|               |                     |
|---------------|---------------------|
| <i>bounds</i> | The string to parse |
|---------------|---------------------|

**6.5.2.3 template<typename>Tvirtual Interval< T>::~Interval ( )** [\[inline\]](#),[\[virtual\]](#)

Destructor for an [Interval](#)

### 6.5.3 Member Function Documentation

**6.5.3.1 template<typename>Tvirtual bool Interval< T>::include ( const T & *a* ) const**[\[inline\]](#),[\[virtual\]](#)

Test if a value is included in the interval.

#### Parameters

|          |                    |
|----------|--------------------|
| <i>a</i> | The value to test. |
|----------|--------------------|

#### Returns

true if *a* is included in the interval; false elsewhere.

**6.5.3.2 template<typename>Tvirtual const T& Interval< T>::lbound () const** [\[inline\]](#),[\[virtual\]](#)

Return lower bound of interval;

#### Returns

Lower bound of the interval.

### 6.5.3.3 `template<typename T> virtual std::string Interval<T>::to_string () const` [inline],[virtual]

Convert an interval vector to a string.

#### Returns

A string representation of the interval

### 6.5.3.4 `template<typename T> virtual const T& Interval<T>::ubound () const` [inline],[virtual]

Return upper bound bound of interval;

#### Returns

Upper bound of the interval.

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

- [utils/Interval.h](#)

## 6.6 IntervalVector<T> Class Template Reference

A set of disjoint intervals.

```
#include <IntervalVector.h>
```

### Public Member Functions

- [IntervalVector](#) ()
- [IntervalVector](#) (const [IntervalVector](#) &iv)
- virtual [~IntervalVector](#) ()
- virtual const T & [lbound](#) () const
- virtual const T & [ubound](#) () const
- virtual bool [include](#) (const T &a) const
- virtual int [add](#) (const [Interval](#)< T > &i)
- virtual int [add](#) (std::string bounds)
- virtual int [add](#) (const T &min, const T &max)
- virtual std::string [to\\_string](#) () const

### 6.6.1 Detailed Description

```
template<typename T> class IntervalVector<T>
```

A set of disjoint intervals.

This class provide a set of interval.

#### Author

Florian Galinier

### 6.6.2 Constructor & Destructor Documentation

#### 6.6.2.1 `template<typename T> IntervalVector< T>::IntervalVector ( )` [inline]

Constructor for a new interval vector.



**6.6.2.2** `template<typename T> IntervalVector< T >::IntervalVector ( const IntervalVector< T > & iv )`  
`[inline]`

Constructor for a new interval vector.

**6.6.2.3** `template<typename T> virtual IntervalVector< T >::~~IntervalVector ( )` `[inline],[virtual]`

Destructor for an [IntervalVector](#)

## 6.6.3 Member Function Documentation

**6.6.3.1** `template<typename T> virtual int IntervalVector< T >::add (const Interval< T > & i)` `[inline],[virtual]`

Add an interval to the vector

### Parameters

|          |                         |
|----------|-------------------------|
| <i>i</i> | The new interval to add |
|----------|-------------------------|

### Returns

1 if the interval intersect another existing one (there will be merged), 2 if the interval is already covered by an existing one, 0 elsewhere.

**6.6.3.2** `template<typename T> virtual int IntervalVector< T >::add (std::string bounds)` `[inline],[virtual]`

Add an interval to the vector

### Parameters

|               |                     |
|---------------|---------------------|
| <i>bounds</i> | The string to parse |
|---------------|---------------------|

### Returns

1 if the interval intersect another existing one (there will be merged), 0 elsewhere.

**6.6.3.3** `template<typename T> virtual int IntervalVector< T >::add (const T & min, const T & max)` `[inline],[virtual]`

Add an interval to the vector

### Parameters

|            |                             |
|------------|-----------------------------|
| <i>min</i> | Lower bound of new interval |
| <i>max</i> | Upper bound of new interval |

### Returns

1 if the interval intersect another existing one (there will be merged), 0 elsewhere.

**6.6.3.4** `template<typename T> virtual bool IntervalVector< T >::include (const T & a)` `const [inline],[virtual]`

Test if a value is included in one of the intervals.

**Parameters**

|                 |                    |
|-----------------|--------------------|
| <b><i>a</i></b> | The value to test. |
|-----------------|--------------------|

**Returns**

true if *a* is included in the interval vector; false elsewhere.

#### 6.6.3.5 `template<typename T> virtual const T& IntervalVector::lb() const [inline],[virtual]`

Return lower bound of all intervals;

**Returns**

Lower bound of the interval vector.

#### 6.6.3.6 `template<typename T> virtual std::string IntervalVector::to_string() const [inline],[virtual]`

Convert an interval to a string.

**Returns**

A string representation of the interval vector

#### 6.6.3.7 `template<typename T> virtual const T& IntervalVector::ub() const [inline],[virtual]`

Return upper bound bound of all intervals;

**Returns**

Upper bound of the interval vector.

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

- [utils/IntervalVector.h](#)

## 6.7 Logger Class Reference

Class that allow to log information in a file.

```
#include <Logger.h>
```

**Public Member Functions**

- [Logger](#) (std::string fileName, std::ios\_base::openmode mode=std::ios\_base::out)
- virtual [~Logger](#) () noexcept
- virtual void [write](#) (std::string s)
- template<typename T >  
  [Logger](#) & [operator<<](#) (const T &t)
- virtual [Logger](#) & [operator<<](#) (const std::string &s)
- virtual [Logger](#) & [operator<<](#) (const char \*c)

### 6.7.1 Detailed Description

Class that allow to log information in a file.

This class provide an interface to write informations in a file, like population state or rejected mutation rate.

#### Author

Florian Galinier

### 6.7.2 Constructor & Destructor Documentation

#### 6.7.2.1 `Logger::Logger( std::string fileName, std::ios_base::openmode mode = ios_base::out )`

Create a new log file

##### Parameters

|                 |                            |
|-----------------|----------------------------|
| <i>fileName</i> | name of the file to create |
| <i>mode</i>     | The mode to use to open    |

#### 6.7.2.2 `Logger::~Logger ()` `[virtual]`, `[noexcept]`

Destructor of [Logger](#)

### 6.7.3 Member Function Documentation

#### 6.7.3.1 `template<typename T> Logger& Logger::operator<< ( const T &t )` `[inline]`

Overloaded operator to use log as a stream

##### Parameters

|          |                     |
|----------|---------------------|
| <i>t</i> | the object to write |
|----------|---------------------|

##### Returns

A reference to current [Logger](#)

#### 6.7.3.2 `Logger & Logger::operator<< ( const std::string &s )` `[virtual]`

Specialization of operator<<

##### Parameters

|          |                     |
|----------|---------------------|
| <i>s</i> | the string to write |
|----------|---------------------|

##### Returns

A reference to current [Logger](#)

#### 6.7.3.3 `Logger & Logger::operator<< ( const char*c )` `[virtual]`

Specialization of operator<<

**Parameters**

|          |                     |
|----------|---------------------|
| <b>c</b> | the string to write |
|----------|---------------------|

**Returns**

A reference to current [Logger](#)

**6.7.3.4 void Logger::write(const string &s) [virtual]**

Write the given string in the log file

**Parameters**

|          |                     |
|----------|---------------------|
| <b>s</b> | The string to write |
|----------|---------------------|

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

- [utils/Logger.h](#)
- [utils/Logger.cpp](#)

## 6.8 Matrix Class Reference

Class that implement a square matrix.

```
#include <Matrix.h>
```

**Public Member Functions**

- [Matrix](#) (size\_t t)
- virtual [~Matrix](#) () noexcept
- virtual bool [operator!](#) () const
- virtual bool [isPositive](#) () const
- [Matrix](#) \*[operator-](#) (const [Matrix](#) &m2)
- virtual void [set](#) (size\_t i, size\_t j, double v)
- virtual bool [operator>=](#) (const [Matrix](#) &m2)
- virtual bool [operator>](#) (const [Matrix](#) &m2)
- const std::vector< double > & [operator\[\]](#) (size\_t i) const
- virtual double [average](#) () const
- virtual double [lineAverage](#) (int line) const
- virtual double [min](#) () const
- virtual std::string [to\\_csv\\_string](#) () const
- virtual std::string [to\\_string](#) () const
- size\_t [size](#) () const

### 6.8.1 Detailed Description

Class that implement a square matrix.

This class implement is used for store results of multi-dimensionnal score.

**Author**

Florian Galinier

## 6.8.2 Constructor & Destructor Documentation

### 6.8.2.1 Matrix::Matrix(size\_t t)

Create a new square matrix of size t.

**Parameters**

|          |                                 |
|----------|---------------------------------|
| <i>t</i> | The number of rows and columns. |
|----------|---------------------------------|

**6.8.2.2 Matrix::~Matrix ( )** [virtual],[noexcept]

Destructor for [Matrix](#) class.

**6.8.3 Member Function Documentation****6.8.3.1 double Matrix::average** ~~( )~~ const [virtual]

Return an average value of upper triangular matrix.

**Returns**

Average value of upper triangular matrix.

**6.8.3.2 bool Matrix::isPositive** ~~( )~~ const [virtual]

Return true if the matrix is positive, i.e. if all the matrix elements are greater or equal to 0.

**Returns**

true if matrix is positive.

**6.8.3.3 double Matrix::lineAverage** ~~(line )~~ const [virtual]

Return the average of a line.

**Parameters**

|             |                                    |
|-------------|------------------------------------|
| <i>line</i> | The line where to compute average. |
|-------------|------------------------------------|

**Returns**

The average of the line.

**6.8.3.4 double Matrix::min** ~~( )~~ const [virtual]

Return the minimum of upper triangular matrix.

**Returns**

Minimum of upper triangular matrix.

**6.8.3.5 bool Matrix::operator!** ~~( )~~ const [virtual]

Return true if and only if the matrix is null.

**Returns**

true if matrix is null, false elsewhere.

#### 6.8.3.6 `Matrix *Matrix::operator-(const Matrix & m2)`

Subtract two matrices and return the resulting matrix.

**Parameters**

|           |  |
|-----------|--|
| <i>m2</i> | The matrix to subtract to current one. |
|-----------|--|

**Returns**

A new pointer [Matrix](#) that is the result of subtraction.

**6.8.3.7 bool Matrix::operator<(const Matrix & m2) [virtual]**

Compare two matrices and return true only if current matrix is greater than m2.

**Parameters**

|           |                        |
|-----------|------------------------|
| <i>m2</i> | The matrix to compare. |
|-----------|------------------------|

**Returns**

true only if current matrix is greater than m2.

**6.8.3.8 bool Matrix::operator<=(const Matrix & m2) [virtual]**

Compare two matrices and return true only if current matrix is greater or equal than m2.

**Parameters**

|           |                        |
|-----------|------------------------|
| <i>m2</i> | The matrix to compare. |
|-----------|------------------------|

**Returns**

true only if current matrix is greater or equal than m2.

**6.8.3.9 const std::vector<double> & Matrix::operator[](size\_t i) const**

Accessor for line of [Matrix](#). Allow to access to matrix element by using [].

**Parameters**

|          |                     |
|----------|---------------------|
| <i>i</i> | The line to access. |
|----------|---------------------|

**Returns**

The ith line.

**6.8.3.10 void Matrix::set(size\_t i, size\_t j, double v) [inline], [virtual]**

Change element at position (i,j).

**Parameters**

|          |                        |
|----------|------------------------|
| <i>i</i> | The number of the row. |
|----------|------------------------|



|     |   |
|-----|---|
| $j$ | The number of the column.                         |
| $v$ | The new value of <a href="#">Matrix</a> at (i,j). |

#### 6.8.3.11 `size_t Matrix::size() const`

Return the size of matrix (height and width, both are equal).

##### Returns

The size of matrix.

#### 6.8.3.12 `std::string Matrix::to_csv_string() const [virtual]`

Return a CSV string representation of matrix.

##### Returns

CSV string representation of matrix.

#### 6.8.3.13 `std::string Matrix::to_string() const [virtual]`

Return a string representation of matrix.

##### Returns

string representation of matrix.

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

- [model/Matrix.h](#)
- [model/Matrix.cpp](#)

## 6.9 Model Class Reference

Class that represent a model.

```
#include <Model.h>
```

### Public Member Functions

- [Model](#) ()
- [Model](#) (const [Model](#) &m)
- [Model](#) (std::string genesFile)
- virtual [~Model](#) () noexcept
- virtual void [setVal](#) ([GenesPtr](#) v)
- virtual [GenesPtr](#) [getVal](#) () const
- virtual void [setDomains](#) ([DomainsPtr](#) d)
- virtual [DomainsPtr](#) [getDomains](#) () const
- virtual bool [isValid](#) () const
- virtual bool [operator==](#) (const [Model](#) &b)
- virtual bool [operator!=](#) (const [Model](#) &b)
- virtual std::string [to\\_string](#) () const
- virtual std::string [generateDotFile](#) (int i)

## Static Public Member Functions

- static float [evaluateExtern](#) ([Model](#) &m1, [Model](#) &m2)
- static float [evaluate](#) (const [Model](#) &m1, const [Model](#) &m2)

## Protected Attributes

- [GenesPtr](#) **genes**
- [DomainsPtr](#) **domains**
- std::string **xcsp**
- std::string **mm**
- std::string **root**
- std::string **grimm**
- std::string **dot**
- bool **change**

### 6.9.1 Detailed Description

Class that represent a model.

This class is a gene vector representation of a model. Due to the fact that a model can be composed by more than a model, we create a class that encapsulate the idea of model.

#### Author

Florian Galinier

### 6.9.2 Constructor & Destructor Documentation

#### 6.9.2.1 [Model::Model](#) ( )

Create a new empty [Model](#).

#### 6.9.2.2 [Model::Model](#) (const [Model](#) & *m* )

Create a [Model](#) that is a copy of [Model](#) *m*.

##### Parameters

|          |                                    |
|----------|------------------------------------|
| <i>m</i> | The <a href="#">Model</a> to copy. |
|----------|------------------------------------|

#### 6.9.2.3 [Model::Model](#) (std::string *genesFile* )

Create a new model with data from input file.

##### Parameters

|                  |   |
|------------------|---|
| <i>genesFile</i> | The file path that contains vector value for model. |
|------------------|---|

#### 6.9.2.4 [Model::~Model](#) ( ) [virtual], [noexcept]

[Model](#) destructor.

### 6.9.3 Member Function Documentation

#### 6.9.3.1 `float Model::evaluate(const Model & m1, const Model & m2) [static]`

Take two models and return an evaluation of the distance between them.

**Parameters**

|           |                       |
|-----------|-----------------------|
| <i>m1</i> | First model to test.  |
| <i>m2</i> | Second model to test. |

**Returns**

The distance between the two models.

**6.9.3.2 float Model::evaluateExtModel (Model & m1, Model & m2) [static]**

Take two models and return an evaluation of the distance between them.

**Parameters**

|           |                       |
|-----------|-----------------------|
| <i>m1</i> | First model to test.  |
| <i>m2</i> | Second model to test. |

**Returns**

The distance between the two models.

**6.9.3.3 std::string Model::generateDotFile() [virtual]**

Generate a dot file from actual model.

**Returns**

The dot file path.

**6.9.3.4 DomainsPtr Model::getDomains() const [virtual]**

Return domains of genes in the [Model](#)

**Returns**

The variable domains of [Model](#)

**6.9.3.5 GenesPtr Model::getVal() const [virtual]**

Return actual value of [Model](#)

**Returns**

The current value of [Model](#)

**6.9.3.6 bool Model::isValid() const [virtual]**

Check if the [Model](#) is a valid model or not.

**Returns**

true if [Model](#) is valid, elsewhere false.

**6.9.3.7 bool Model::operator!=(const Model & b) [virtual]**

Compare two models and return true if models are different.

**Parameters**

|          |                        |
|----------|------------------------|
| <i>b</i> | Other model to compare |
|----------|------------------------|

**Returns**

true if this and *b* are not equal, false elsewhere.

**6.9.3.8 bool Model::operator==(const Model & *b*) [virtual]**

Compare two models and return true if models are equals.

**Parameters**

|          |                        |
|----------|------------------------|
| <i>b</i> | Other model to compare |
|----------|------------------------|

**Returns**

true if this and *b* are equal, false elsewhere.

**6.9.3.9 void Model::setDomainsPtr(*d*) [virtual]**

Change actual value of [Model](#)

**Deprecated Parameters**

|          |  |
|----------|--|
| <i>v</i> | The new value of <a href="#">Model</a> |
|----------|--|

**6.9.3.10 void Model::setValuesPtr(*v*) [virtual]**

Change actual value of [Model](#)

**Deprecated Parameters**

|          |  |
|----------|--|
| <i>v</i> | The new value of <a href="#">Model</a> |
|----------|--|

**6.9.3.11 std::string Model::to\_string() const [virtual]**

Return a string that represent current model.

**Returns**

A string that represent current model.

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

- [model/Model.h](#)
- [model/Model.cpp](#)

## 6.10 NSGAI Class Reference

Implementation of NSGA-II algorithm.

```
#include <NSGAI.h>
```

Inheritance diagram for NSGAI:

Collaboration diagram for NSGAI:

## Public Types

- enum [Fitness](#) { **MIN** = 0b0001, **AVG** = 0b0010, **DIST** = 0b0000, **MINAVG** = 0b0100 }  
Mask for fitness choice. Represent each fitness choice. Fitness can be used as mask with logical operator.

## Public Member Functions

- GADefineIdentity** ("NSGAI", 288)
- [NSGAI](#) (const GAPopulation &p, unsigned int pm=1)
- virtual [~NSGAI](#) () noexcept
- virtual bool [dominate](#) (MatrixPtr m1, MatrixPtr m2)
- virtual std::shared\_ptr< std::vector< std::vector< int > > > [fastNonDominatedSort](#) ()
- virtual std::map< size\_t, double > [crowdingDistanceAssignment](#) (std::vector< int > &f)
- virtual void [step](#) ()
- virtual [Statistics](#) & [extraStatistics](#) ()
- [NSGAI](#) & [operator++](#) ()

## Static Public Attributes

- static int [fitness](#) = NSGAI::Fitness::AVG
- static unsigned int [gen](#) = 0

### 6.10.1 Detailed Description

Implementation of NSGA-II algorithm.

This class implement the GAGeneticAlgorithm interface of GALib API in order to provide the NSGA-II algorithm.

#### Author

Florian Galinier

### 6.10.2 Member Enumeration Documentation

#### 6.10.2.1 enum NSGAI::Fitness

Mask for fitness choice. Represent each fitness choice. Fitness can be used as mask with logical operator.

#### Author

Florian Galinier

### 6.10.3 Constructor & Destructor Documentation

#### 6.10.3.1 NSGAI::NSGAI(const GAPopulation &p, unsigned int pm)

Constructor for new NSGA-II Algorithm

##### Parameters

|          |                        |
|----------|------------------------|
| <i>p</i> | The initial population |
|----------|------------------------|

#### 6.10.3.2 NSGAI::~NSGAI() [virtual], [noexcept]

Destructor

## 6.10.4 Member Function Documentation

### 6.10.4.1 `std::map<size_t, double> NSGAll::crowdingDistanceAssign(std::vector<int> & f)` [virtual]

Assign to each individual of a front a crowding distance value.

#### Parameters

|          |                                   |
|----------|-----------------------------------|
| <i>f</i> | A vector of individual in a front |
|----------|-----------------------------------|

#### Returns

A map that associate at each individual a crowding distance.

### 6.10.4.2 `bool NSGAll::dominate(MatrixPtr m1, MatrixPtr m2)` [virtual]

Compare two matrices of score and return true if the first one dominate the second one.

#### Parameters

|           |                           |
|-----------|---------------------------|
| <i>m1</i> | The first matrix to test. |
| <i>m2</i> | The second one.           |

#### Returns

true iff m1 dominates m2.

### 6.10.4.3 `std::shared_ptr<std::vector<std::vector<int>>> NSGAll::fastNonDominatedSort()` [virtual]

Sort population in Pareto front.

#### Returns

A vector of front composed by key of front's individual.

### 6.10.4.4 `NSGAll & NSGAll::operator++()`

Operator that create a new generation.

#### Returns

The NSGA-II algorithm with a new population.

### 6.10.4.5 `void NSGAll::step()` [virtual]

Create and evaluate a new generation.

## 6.10.5 Member Data Documentation

### 6.10.5.1 `int NSGAll::fitness = NSGAll::Fitness::AVG` [static]

Chooosed fitness

### 6.10.5.2 unsigned int NSGAll::gen [static]

Current generation.

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

- model/NSGAll.h
- model/NSGAll.cpp

## 6.11 Population Class Reference

This class manage all chromosomes (especially the initial ones).

```
#include <Population.h>
```

Inheritance diagram for Population:

Collaboration diagram for Population:

### Public Member Functions

- [Population](#) ()
- [Population](#) (std::string dirPath)
- [MatrixPtr evaluate](#) (GABoolean flag=gaFalse) const
- [GChromosom & best](#) (size\_t r, size\_t c)
- [GChromosom & worst](#) (size\_t r, size\_t c)

### 6.11.1 Detailed Description

This class manage all chromosomes (especially the initial ones).

This class provide methods for population management.

#### Author

Florian Galinier

### 6.11.2 Constructor & Destructor Documentation

#### 6.11.2.1 Population::Population ()

Default constructor.

#### 6.11.2.2 Population::Population (std::string dirPath )

Create a population from files given in directory.

#### Parameters

|                |   |
|----------------|---|
| <i>dirPath</i> | The directory where the models files are. |
|----------------|---|

### 6.11.3 Member Function Documentation

#### 6.11.3.1 GChromosom & Population::best (size\_t r, size\_t c )

Return the best element of population for the objectif in matrix score at m[r][c].



**Parameters**

|          |  |
|----------|--|
| <i>r</i> | <a href="#">Matrix</a> row of the objective.   |
| <i>c</i> | <a href="#">Matrix</a> column of the objective |

**Returns**

Best individual for objective score[r][c]

**6.11.3.2 MatrixPtr Population::evaluate([Boolean flag = False](#)) const**

Evaluate all individuals of population

**Parameters**

|              |                       |
|--------------|-----------------------|
| <i>Force</i> | or not the evaluation |
|--------------|-----------------------|

**6.11.3.3 GACHromosom & Population::worst([size\\_t rsize\\_t c](#))**

Return the worst element of population for the objectif in matrix score at m[r][c].

**Parameters**

|          |  |
|----------|--|
| <i>r</i> | <a href="#">Matrix</a> row of the objective.   |
| <i>c</i> | <a href="#">Matrix</a> column of the objective |

**Returns**

Worst individual for objective score[r][c]

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

- model/[Population.h](#)
- model/[Population.cpp](#)

## 6.12 Statistics Class Reference

A class to store genetic algorithm stats.

```
#include <Statistics.h>
```

**Public Member Functions**

- void [update](#) (GAPopulation &p)
- const std::vector< float > & [getMed](#) () const
- const std::vector< float > & [getFirstQuartile](#) () const
- const std::vector< float > & [getThirdQuartile](#) () const
- const std::vector< float > & [getMin](#) () const
- const std::vector< float > & [getMax](#) () const

**Static Public Attributes**

- static std::string [outfile](#) = ""

### 6.12.1 Detailed Description

A class to store genetic algorithm stats.

This class provides utility to record statistics of the evolutionary algorithm.

#### Author

Florian Galinier

### 6.12.2 Member Function Documentation

#### 6.12.2.1 `const std::vector<float> & Statistics::getFirstQuartile() const`

Getter for first quartiles (each entry is a generation)

##### Returns

A vector with all recorded first quartiles.

#### 6.12.2.2 `const std::vector<float> & Statistics::getMax() const`

Getter for maximums (each entry is a generation)

##### Returns

A vector with all recorded maximums.

#### 6.12.2.3 `const std::vector<float> & Statistics::getMed() const`

Getter for medians (each entry is a generation)

##### Returns

A vector with all recorded medians.

#### 6.12.2.4 `const std::vector<float> & Statistics::getMin() const`

Getter for minimums (each entry is a generation)

##### Returns

A vector with all recorded minimums.

#### 6.12.2.5 `const std::vector<float> & Statistics::getThirdQuartile() const`

Getter for third quartiles (each entry is a generation)

##### Returns

A vector with all recorded third quartiles.

#### 6.12.2.6 `void Statistics::updateAPopulation & p)`

Use population to update stats

**Parameters**

|     |   |
|-----|---|
| $p$ | <a href="#">Population</a> used for stats analysis. |
|-----|---|

**6.12.3 Member Data Documentation****6.12.3.1 `std::string Statistics::outfile` `#static`**

The name of the output file

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

- [model/Statistics.h](#)
- [model/Statistics.cpp](#)



# Chapter 7

## File Documentation

### 7.1 model/Chromosom.h File Reference

[Chromosom](#) class header.

```
#include <vector>
#include <memory>
#include <string>
#include "Model.h"
```

Include dependency graph for Chromosom.h: This graph shows which files directly or indirectly include this file:

#### Classes

- class [Chromosom](#)

*Class that represent a genome in our problem.*

#### 7.1.1 Detailed Description

[Chromosom](#) class header.

##### Author

Florian Galinier

##### Version

0.1

##### Date

06/04/16

Definition of our problem that will be inherited for the genetic algorithm.

### 7.2 model/GAChromosom.h File Reference

[GAChromosom](#) class header.

```
#include "Chromosom.h"
#include <ga/GAGenome.h>
#include <functional>
#include "utils/Logger.h"
#include "Matrix.h"
```

Include dependency graph for GACHromosom.h: This graph shows which files directly or indirectly include this file:

## Classes

- class [GACHromosom](#)  
*Class that implement a genome for our problem.*

## Typedefs

- typedef std::shared\_ptr< [Matrix](#) > [MatrixPtr](#)

## Functions

- [Logger](#) & **operator**<< ([Logger](#) &log, const [GACHromosom](#) &c)

### 7.2.1 Detailed Description

[GACHromosom](#) class header.

#### Author

Florian Galinier

#### Version

0.1

#### Date

06/04/16

Declaration of our own genome class.

### 7.2.2 Typedef Documentation

#### 7.2.2.1 typedef std::shared\_ptr<[Matrix](#)> [MatrixPtr](#)

Smart pointer to a [Matrix](#).

## 7.3 model/Matrix.h File Reference

[Matrix](#) class header.

```
#include <vector>
#include <iostream>
#include "utils/Logger.h"
```

Include dependency graph for Matrix.h: This graph shows which files directly or indirectly include this file:

## Classes

- class [Matrix](#)  
*Class that implement a square matrix.*

## Functions

- `std::ostream & operator<< (std::ostream &os, const Matrix &m)`
- `Logger & operator<< (Logger &log, const Matrix &m)`

### 7.3.1 Detailed Description

[Matrix](#) class header.

#### Author

Florian Galinier

#### Version

0.1

#### Date

27/04/16

Declaration of a square matrix class.

### 7.3.2 Function Documentation

#### 7.3.2.1 `std::ostream& operator<< ( std::ostream & os, const Matrix & m )`

`operator<<` overload for iostream.

##### Parameters

|           |                             |
|-----------|-----------------------------|
| <i>os</i> | The ostream where to write. |
| <i>m</i>  | The matrix to write.        |

##### Returns

The resulting ostream.

#### 7.3.2.2 `Logger& operator<< ( Logger & log, const Matrix & m )`

Specialization of [Logger](#) `operator<<` for [Matrix](#)

##### Parameters

|            |                           |
|------------|---------------------------|
| <i>log</i> | The logger where to write |
| <i>m</i>   | The matrix to write       |

##### Returns

The used logger

## 7.4 model/Model.h File Reference

[Model](#) class header.

```
#include <vector>
#include <memory>
#include <pugixml.hpp>
#include "utils/IntervalVector.h"
```

Include dependency graph for Model.h: This graph shows which files directly or indirectly include this file:

### Classes

- class [Model](#)  
*Class that represent a model.*

### Typedefs

- typedef std::vector< int > [Genes](#)
- typedef std::shared\_ptr< [Genes](#) > [GenesPtr](#)
- typedef std::vector< [IntervalVector](#)< int > > [Domains](#)
- typedef std::shared\_ptr< [Domains](#) > [DomainsPtr](#)

### Functions

- std::ostream & [operator<<](#) (std::ostream &os, const [Model](#) &m)

### 7.4.1 Detailed Description

[Model](#) class header.

#### Author

Florian Galinier

#### Version

0.1

#### Date

06/04/16

The representation of a model in our problem is an int vector. Due to the fact that a chromosom can be composed by more than one model vector, we choose to reify it in a [Model](#) class.

### 7.4.2 Typedef Documentation

#### 7.4.2.1 typedef std::vector<[IntervalVector](#)<int> > [Domains](#)

A vector of all domains. For all i in {0..[Genes](#).size()}, [Domains](#)[i] = domains of [Genes](#)[i]

#### 7.4.2.2 typedef std::shared\_ptr<[Domains](#)> [DomainsPtr](#)

Smart pointer to a domains vector.



### 7.4.2.3 typedef std::vector<int> Genes

In our problem, a gene vector is a vector of int. We create an alias for convenience.

### 7.4.2.4 typedef std::shared\_ptr<Genes> GenesPtr

Smart pointer to a genes vector.

## 7.4.3 Function Documentation

### 7.4.3.1 std::ostream& operator<<( std::ostream & os, const Model & m )

operator<< overload for ostream.

#### Parameters

|           |                             |
|-----------|-----------------------------|
| <i>os</i> | The ostream where to write. |
| <i>m</i>  | The model to write.         |

#### Returns

The resulting ostream.

## 7.5 model/NSGAII.h File Reference

[NSGAII](#) class header.

```
#include <ga/GASimpleGA.h>
#include <vector>
#include <map>
#include "Statistics.h"
Include dependency graph for NSGAII.h:
```

## Classes

- class [NSGAII](#)  
*Implementation of NSGA-II algorithm.*

### 7.5.1 Detailed Description

[NSGAII](#) class header.

#### Author

Florian Galinier

#### Version

0.1

#### Date

15/04/16

NSGA-II class implementation for the evolutionary algorithm.

## 7.6 model/Population.h File Reference

[Population](#) class header.

```
#include "GAChromosom.h"
#include "utils/Logger.h"
#include <ga/GAPopulation.h>
```

Include dependency graph for Population.h: This graph shows which files directly or indirectly include this file:

### Classes

- class [Population](#)

*This class manage all chromosomes (especially the initial ones).*

### Functions

- [Logger](#) & [operator<<](#) ([Logger](#) &log, const [Population](#) &p)

#### 7.6.1 Detailed Description

[Population](#) class header.

#### Author

Florian Galinier

#### Version

0.1

#### Date

06/04/16

[Population](#) that will be used in our genetic algorithm.

#### 7.6.2 Function Documentation

##### 7.6.2.1 [Logger](#)& [operator<<](#) ( [Logger](#) & *log*, const [Population](#) & *p* )

Specialization of [Logger](#) [operator<<](#) for [Population](#)

#### Parameters

|            |                           |
|------------|---------------------------|
| <i>log</i> | The logger where to write |
| <i>p</i>   | The population to write   |

#### Returns

The used logger

## 7.7 model/Statistics.h File Reference

[Statistics](#) class header.

```
#include "Population.h"
#include "utils/Logger.h"
```

Include dependency graph for Statistics.h: This graph shows which files directly or indirectly include this file:

### Classes

- class [Statistics](#)  
*A class to store genetic algorithm stats.*

### Functions

- [Logger](#) & [operator<<](#) ([Logger](#) &log, const [Statistics](#) &s)

#### 7.7.1 Detailed Description

[Statistics](#) class header.

##### Author

Florian Galinier

##### Version

0.1

##### Date

03/05/16

This statistics class implementation for the evolutionary algorithm.

#### 7.7.2 Function Documentation

##### 7.7.2.1 [Logger](#)& [operator<<](#) ( [Logger](#) & *log*, const [Statistics](#) & *s* )

Specialization of [Logger](#) [operator<<](#) for statistics

##### Parameters

|            |                           |
|------------|---------------------------|
| <i>log</i> | The logger where to write |
| <i>s</i>   | The statistics to write   |

##### Returns

The used logger

## 7.8 utils/CommandLine.h File Reference

[CommandLine](#) class header.

```
#include <map>
#include <string>
```

Include dependency graph for CommandLine.h:

## Classes

- class [CommandLine](#)  
*Class for command line options parsing.*

### 7.8.1 Detailed Description

[CommandLine](#) class header.

#### Author

Florian Galinier

#### Version

0.1

#### Date

12/04/16

Set of functions for command line options.

## 7.9 utils/Directory.h File Reference

[Directory](#) class header.

```
#include <dirent.h>
#include <vector>
#include <string>
#include <memory>
Include dependency graph for Directory.h:
```

## Classes

- class [Directory](#)  
*This class provide methods for directory manipulation.*

### 7.9.1 Detailed Description

[Directory](#) class header.

#### Author

Florian Galinier

#### Version

0.1

#### Date

06/04/16

Utils function for directory manipulation.

## 7.10 utils/Interval.h File Reference

[Interval](#) class header.

```
#include <stdexcept>
```

Include dependency graph for Interval.h: This graph shows which files directly or indirectly include this file:

### Classes

- class [Interval< T >](#)  
*An interval representation.*

### 7.10.1 Detailed Description

[Interval](#) class header.

#### Author

Florian Galinier

#### Version

0.1

#### Date

11/04/16

[Interval](#) manipulation.

## 7.11 utils/IntervalVector.h File Reference

[IntervalVector](#) class header.

```
#include <vector>
#include "Interval.h"
```

Include dependency graph for IntervalVector.h: This graph shows which files directly or indirectly include this file:

### Classes

- class [IntervalVector< T >](#)  
*A set of disjoint intervals.*

### 7.11.1 Detailed Description

[IntervalVector](#) class header.

#### Author

Florian Galinier

#### Version

0.1

**Date**

11/04/16

This class provide a vector of interval.

## 7.12 utils/Logger.h File Reference

[Logger](#) class header.

```
#include <string>
#include <fstream>
#include <ios>
```

Include dependency graph for Logger.h: This graph shows which files directly or indirectly include this file:

### Classes

- class [Logger](#)

*Class that allow to log information in a file.*

### 7.12.1 Detailed Description

[Logger](#) class header.

**Author**

Florian Galinier

**Version**

0.1

**Date**

29/04/16

A logger to write on a file.

# Index

- ~Chromosom
  - Chromosom, [12](#)
- ~Directory
  - Directory, [17](#)
- ~GACHromosom
  - GACHromosom, [19](#)
- ~Interval
  - Interval, [23](#)
- ~IntervalVector
  - IntervalVector, [25](#)
- ~Logger
  - Logger, [27](#)
- ~Matrix
  - Matrix, [30](#)
- ~Model
  - Model, [34](#)
- ~NSGAII
  - NSGAII, [38](#)
- add
  - IntervalVector, [25](#)
- addOption
  - CommandLine, [16](#)
- average
  - Matrix, [30](#)
- avgEvaluator
  - GACHromosom, [19](#)
- best
  - Population, [40](#)
- Chromosom, [11](#)
  - ~Chromosom, [12](#)
  - Chromosom, [12](#)
  - getModels, [12](#)
  - getNbModels, [12](#)
  - isValid, [12](#)
  - mutate, [12](#)
  - setModels, [14](#)
  - setNbModels, [14](#)
- clone
  - GACHromosom, [20](#)
- CommandLine, [14](#)
  - addOption, [16](#)
  - CommandLine, [15](#)
  - parse, [16](#)
  - printUsage, [16](#)
- compare
  - GACHromosom, [20](#)
- copy
  - GACHromosom, [20](#)
- crowdingDistanceAssignment
  - NSGAII, [39](#)
- Directory, [16](#)
  - ~Directory, [17](#)
  - Directory, [17](#)
  - getFiles, [17](#)
  - getNbFiles, [17](#)
- Domains
  - Model.h, [48](#)
- DomainsPtr
  - Model.h, [48](#)
- dominate
  - NSGAII, [39](#)
- evaluate
  - GACHromosom, [20](#)
  - Model, [35](#)
  - Population, [41](#)
- evaluateExtern
  - Model, [36](#)
- fastNonDominatedSort
  - NSGAII, [39](#)
- Fitness
  - NSGAII, [38](#)
- fitness
  - NSGAII, [39](#)
- GACHromosom, [18](#)
  - ~GACHromosom, [19](#)
  - avgEvaluator, [19](#)
  - clone, [20](#)
  - compare, [20](#)
  - copy, [20](#)
  - evaluate, [20](#)
  - GACHromosom, [19](#)
  - init, [21](#)
  - mutate, [21](#)
  - onePointCrossover, [21](#)
  - score, [21](#)
  - setModels, [22](#)
  - to\_string, [22](#)
  - write, [22](#)
- GACHromosom.h
  - MatrixPtr, [46](#)
- gen
  - NSGAII, [39](#)
- generateDotFile

- Model, 36
- Genes
  - Model.h, 48
- GenesPtr
  - Model.h, 49
- getDomains
  - Model, 36
- getFiles
  - Directory, 17
- getFirstQuartile
  - Statistics, 42
- getMax
  - Statistics, 42
- getMed
  - Statistics, 42
- getMin
  - Statistics, 42
- getModels
  - Chromosom, 12
- getNbFiles
  - Directory, 17
- getNbModels
  - Chromosom, 12
- getThirdQuartile
  - Statistics, 42
- getVal
  - Model, 36
- include
  - Interval, 23
  - IntervalVector, 25
- init
  - GACHromosom, 21
- Interval
  - ~Interval, 23
  - include, 23
  - Interval, 23
  - lbound, 23
  - to\_string, 23
  - ubound, 24
- Interval< T >, 22
- IntervalVector
  - ~IntervalVector, 25
  - add, 25
  - include, 25
  - IntervalVector, 24
  - lbound, 26
  - to\_string, 26
  - ubound, 26
- IntervalVector< T >, 24
- isPositive
  - Matrix, 30
- isValid
  - Chromosom, 12
  - Model, 36
- lbound
  - Interval, 23
  - IntervalVector, 26
- lineAverage
  - Matrix, 30
- Logger, 26
  - ~Logger, 27
  - Logger, 27
  - operator<<, 27
  - write, 28
- Matrix, 28
  - ~Matrix, 30
  - average, 30
  - isPositive, 30
  - lineAverage, 30
  - Matrix, 29
  - min, 30
  - operator!, 30
  - operator>, 32
  - operator>=, 32
  - operator-, 30
  - operator[], 32
  - set, 32
  - size, 33
  - to\_csv\_string, 33
  - to\_string, 33
- Matrix.h
  - operator<<, 47
- MatrixPtr
  - GACHromosom.h, 46
- min
  - Matrix, 30
- Model, 33
  - ~Model, 34
  - evaluate, 35
  - evaluateExtern, 36
  - generateDotFile, 36
  - getDomains, 36
  - getVal, 36
  - isValid, 36
  - Model, 34
  - operator!=, 36
  - operator==, 37
  - setDomains, 37
  - setVal, 37
  - to\_string, 37
- Model.h
  - Domains, 48
  - DomainsPtr, 48
  - Genes, 48
  - GenesPtr, 49
  - operator<<, 49
- model/Chromosom.h, 45
- model/GACHromosom.h, 45
- model/Matrix.h, 46
- model/Model.h, 48
- model/NSGAll.h, 49
- model/Population.h, 50
- model/Statistics.h, 51
- mutate
  - Chromosom, 12



- GAChromosom, [21](#)
- NSGAI, [37](#)
  - ~NSGAI, [38](#)
  - crowdingDistanceAssignment, [39](#)
  - dominate, [39](#)
  - fastNonDominatedSort, [39](#)
  - Fitness, [38](#)
  - fitness, [39](#)
  - gen, [39](#)
  - NSGAI, [38](#)
  - operator++, [39](#)
  - step, [39](#)
- onePointCrossover
  - GAChromosom, [21](#)
- operator!
  - Matrix, [30](#)
- operator!=
  - Model, [36](#)
- operator<<
  - Logger, [27](#)
  - Matrix.h, [47](#)
  - Model.h, [49](#)
  - Population.h, [50](#)
  - Statistics.h, [51](#)
- operator>
  - Matrix, [32](#)
- operator>=
  - Matrix, [32](#)
- operator++
  - NSGAI, [39](#)
- operator-
  - Matrix, [30](#)
- operator==
  - Model, [37](#)
- operator[]
  - Matrix, [32](#)
- outfile
  - Statistics, [43](#)
- parse
  - CommandLine, [16](#)
- Population, [40](#)
  - best, [40](#)
  - evaluate, [41](#)
  - Population, [40](#)
  - worst, [41](#)
- Population.h
  - operator<<, [50](#)
- printUsage
  - CommandLine, [16](#)
- score
  - GAChromosom, [21](#)
- set
  - Matrix, [32](#)
- setDomains
  - Model, [37](#)
- setModels
  - Chromosom, [14](#)
  - GAChromosom, [22](#)
- setNbModels
  - Chromosom, [14](#)
- setVal
  - Model, [37](#)
- size
  - Matrix, [33](#)
- Statistics, [41](#)
  - getFirstQuartile, [42](#)
  - getMax, [42](#)
  - getMed, [42](#)
  - getMin, [42](#)
  - getThirdQuartile, [42](#)
  - outfile, [43](#)
  - update, [42](#)
- Statistics.h
  - operator<<, [51](#)
- step
  - NSGAI, [39](#)
- to\_csv\_string
  - Matrix, [33](#)
- to\_string
  - GAChromosom, [22](#)
  - Interval, [23](#)
  - IntervalVector, [26](#)
  - Matrix, [33](#)
  - Model, [37](#)
- ubound
  - Interval, [24](#)
  - IntervalVector, [26](#)
- update
  - Statistics, [42](#)
- utils/CommandLine.h, [51](#)
- utils/Directory.h, [52](#)
- utils/Interval.h, [53](#)
- utils/IntervalVector.h, [53](#)
- utils/Logger.h, [54](#)
- worst
  - Population, [41](#)
- write
  - GAChromosom, [22](#)
  - Logger, [28](#)