Metaboflux

2.0

Generated by Doxygen 1.7.4

Wed Apr 27 2011 16:40:50

Contents

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 2

File Index

2.1 File List

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

Chapter 3

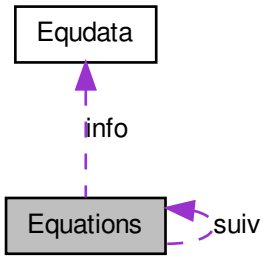
Data Structure Documentation

3.1 Equations Struct Reference

Function pointers used for operations.

`#include <equations.h>`

Collaboration diagram for Equations:



Data Fields

- Equtype type
- Equdata info

3.1.1 Detailed Description

Function pointers used for operations.

Definition at line 68 of file equations.h.

3.1.2 Field Documentation

3.1.2.1 Equdata Equations::info

Equation Data

Definition at line 71 of file equations.h.

Referenced by Equations_print().

3.1.2.2 Equtype Equations::type

Operation constants

Definition at line 70 of file equations.h.

Referenced by Equations_print(), and Equations_resultat().

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

- metaboflux/include/equations.h

3.2 Equdata Union Reference

Equation data.

`#include <equations.h>`

Data Fields

- double data
- char * var

3.2.1 Detailed Description

Equation data.

Definition at line 57 of file equations.h.

3.2.2 Field Documentation

3.2.2.1 double Equdata::data

Constant

Definition at line 60 of file equations.h.

Referenced by Equations_print().

3.2.2.2 char* Equdata::var

Variable

Definition at line 61 of file equations.h.

Referenced by Equations_print().

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

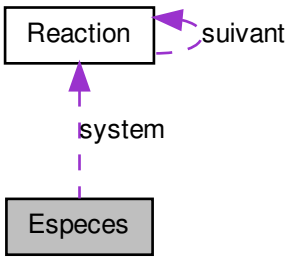
- metaboflux/include/equations.h

3.3 Especes Struct Reference

Molecules data.

`#include <especes.h>`

Collaboration diagram for Especes:



Data Fields

- double quantite
- const char * id
- pReaction system

3.3.1 Detailed Description

Molecules data.

Definition at line 45 of file especes.h.

3.3.2 Field Documentation

3.3.2.1 const char* Especes::id

Molecule id

Definition at line 47 of file especes.h.

Referenced by Especes_save().

3.3.2.2 double Especes::quantite

Molecule quantity

Definition at line 46 of file especes.h.

Referenced by Especes_alloc(), Especes_getQuantite(), Especes_save(), Especes_scoreSpecies(), and Especes_setQuantite().

3.3.2.3 pReaction Especes::system

Reaction where the molecule is implicated

Definition at line 48 of file especes.h.

Referenced by Especes_alloc(), Especes_allocReactions(), Especes_freeReactions(), Especes_getNbreactions(), Especes_print(), SBML_EstimationReaction(), SBML_reactChoice(), and SBML_simulate().

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

- metaboflux/include/especes.h

3.4 ListArguments Struct Reference

List of arguments.

Data Fields

- int activity
- int debug
- int port

- int group
- char ** files_path

3.4.1 Detailed Description

List of arguments.

Definition at line 65 of file MetaBoFlux.c.

3.4.2 Field Documentation

3.4.2.1 int ListArguments::activity

Activity chosen

Definition at line 67 of file MetaBoFlux.c.

Referenced by alloc_arguments(), argument_analysis(), check_arguments(), and main().

3.4.2.2 int ListArguments::debug

Debug chosen

Definition at line 68 of file MetaBoFlux.c.

Referenced by alloc_arguments(), argument_analysis(), and main().

3.4.2.3 char** ListArguments::files_path

Path to files path : 0.SBML 1.PAR 2.OUT 3.RATIO_FILE 4.LOG 5.OUTPUT

Definition at line 71 of file MetaBoFlux.c.

Referenced by alloc_arguments(), argument_analysis(), check_arguments(), check_common(), free_arguments(), and main().

3.4.2.4 int ListArguments::group

Group chosen

Definition at line 70 of file MetaBoFlux.c.

Referenced by alloc_arguments(), argument_analysis(), and main().

3.4.2.5 int ListArguments::port

Port chosen

Definition at line 69 of file MetaBoFlux.c.

Referenced by alloc_arguments(), argument_analysis(), and main().

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

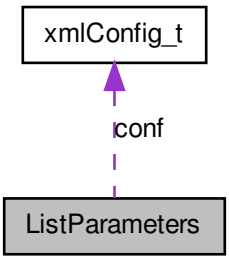
- metaboflux/src/MetaBoFlux.c

3.5 ListParameters Struct Reference

Global list parameters.

`#include <data_parameters.h>`

Collaboration diagram for ListParameters:



Data Fields

- Model_t * model
- xmlConfig_t * conf
- char **** equation
- int nb_parameters
- int interest_parameters
- int nb_couples
- int nb_equations
- int nb_reaction
- int * noeud
- int * parameters
- char ** banned
- int nb_banned
- int nb_triesMod
- int nb_triesSa

3.5.1 Detailed Description

Global list parameters.

Definition at line 35 of file data_parameters.h.

3.5.2 Field Documentation

3.5.2.1 char** ListParameters::banned

Names of banned molecules : ex ATP, ADP ...

Definition at line 47 of file data_parameters.h.

Referenced by Data_allocParameters(), Data_desallocation(), Min_my_f(), Mod_compute_modeling(), Recuit_energyFunction(), and Sd_compute_simulation().

3.5.2.2 xmlConfig_t* ListParameters::conf

Parameter file

Definition at line 38 of file data_parameters.h.

Referenced by Data_allocEquations(), Data_allocParameters(), Data_desallocation(), Data_parameters(), Data_scoreAlloc(), Data_simParameters(), Min_my_f(), Mod_compute_modeling(), Mpi_master(), Mpi_sizeResultTab(), Mpi_slave(), Mpi_writer(), Recuit_compute_recuit(), Recuit_printParametre(), and Sd_compute_standard_deviation().

3.5.2.3 char**** ListParameters::equation

Equations text

Definition at line 39 of file data_parameters.h.

Referenced by Data_allocEquations(), Data_desallocation(), and Data_equationsInit().

3.5.2.4 int ListParameters::interest_parameters

Number of interest parameters

Definition at line 41 of file data_parameters.h.

Referenced by Data_allocParameters(), Min_compute_minimization(), Recuit_compute_recuit(), Recuit_metricDistance(), Recuit_printParametre(), and Recuit_takeStep().

3.5.2.5 Model_t* ListParameters::model

SBML file

Definition at line 37 of file data_parameters.h.

Referenced by Data_desallocation(), Data_parameters(), Data_scoreAlloc(), Data_simParameters(), Min_my_f(), Mod_compute_modeling(), Mpi_sizeResultTab(), Mpi_writer(), Recuit_energyFunction(), and Sd_compute_simulation().

3.5.2.6 int ListParameters::nb_banned

Number of banned molecules

Definition at line 48 of file data_parameters.h.

Referenced by Data_allocParameters(), Min_my_f(), Mod_compute_modeling(), Recuit_energyFunction(), and Sd_compute_simulation().

3.5.2.7 int ListParameters::nb_couples

Number of couples

Definition at line 42 of file data_parameters.h.

Referenced by Data_allocParameters(), Data_updateTab(), Min_compute_minimization(), Min_my_f(), Recuit_defParametre(), and Recuit_takeStep().

3.5.2.8 int ListParameters::nb_equations

Number of equations

Definition at line 43 of file data_parameters.h.

Referenced by Data_allocEquations(), Data_desallocation(), Data_equationsAlloc(), Data_equationsInit(), Min_my_f(), Mod_compute_modeling(), Recuit_energyFunction(), and Sd_compute_simulation().

3.5.2.9 int ListParameters::nb_parameters

Number of parameters

Definition at line 40 of file data_parameters.h.

Referenced by Data_allocParameters(), Data_allocSimParameters(), Data_simParameters(), Min_compute_minimization(), Min_getTampon(), Min_my_f(), Min_score_print_mean(), Mod_compute_modeling(), Mod_score_print_mean(), Mpi_master(), Mpi_sizeResultTab(), Mpi_slave(), Mpi_writeSimFile(), Recuit_printPosition(), and Sd_compute_standard_deviation().

3.5.2.10 int ListParameters::nb_reaction

Number of reaction

Definition at line 44 of file data_parameters.h.

3.5.2.11 int ListParameters::nb_triesMod

Number of tries for modelling

Definition at line 49 of file data_parameters.h.

Referenced by Data_allocParameters().

3.5.2.12 int ListParameters::nb_triesSa

Number of tries for simulated annealing and minimization

Definition at line 50 of file data_parameters.h.

Referenced by Data_allocParameters(), and Recuit_energyFunction().

3.5.2.13 int* ListParameters::noeud

Number of nodes in an equation

Definition at line 45 of file data_parameters.h.

Referenced by Data_allocEquations(), Data_desallocation(), and Data_equationsInit().

3.5.2.14 int* ListParameters::parameters

Details on the parameters of reactions

Definition at line 46 of file data_parameters.h.

Referenced by Data_allocParameters(), Data_desallocation(), Data_updateTab(), Min_compute_minimization(), Min_my_f(), Recuit_defParametre(), and Recuit_takeStep().

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

- metaboflux/include/data_parameters.h

3.6 option Struct Reference

List of flag for getopt.

3.6.1 Detailed Description

List of flag for getopt.

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

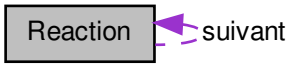
- metaboflux/src/MetaBoFlux.c

3.7 Reaction Struct Reference

Reactions data.

`#include <especes.h>`

Collaboration diagram for Reaction:



Data Fields

- Reaction_t * link
- struct Reaction * suivant

3.7.1 Detailed Description

Reactions data.

Definition at line 34 of file especes.h.

3.7.2 Field Documentation

3.7.2.1 Reaction_t* Reaction::link

Reaction id

Definition at line 36 of file especes.h.

Referenced by Especes_allocReactions(), Especes_print(), SBML_EstimationReaction(), SBML_reactChoice(), and SBML_simulate().

3.7.2.2 struct Reaction* Reaction::suivant

Next reaction implicated for the molecule

Definition at line 38 of file especes.h.

Referenced by Especes_allocReactions(), Especes_freeReactions(), Especes_getNbreactions(), Especes_print(), SBML_EstimationReaction(), and SBML_reactChoice().

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

- metaboflux/include/especes.h

3.8 Score Struct Reference

Structure containing all information from simulations.

`#include <simulation.h>`

Data Fields

- char ** name
- double * quantite
- char ** reaction
- int nb_reaction
- int tailleReactions
- int tailleSpecies
- int taille
- char ** species
- int nb_species
- int * species_amount
- int * species_weight

3.8.1 Detailed Description

Structure containing all information from simulations.

Definition at line 51 of file simulation.h.

3.8.2 Field Documentation

3.8.2.1 char** Score::name

Molecules id

Definition at line 53 of file simulation.h.

Referenced by Data_allocScore(), Data_desallocSim(), Data_scoreFree(), Min_my_f(), Min_score_print_mean(), Mod_compute_modeling(), Mod_score_print_mean(), Recuit_energyFunction(), SBML_score(), SBML_score_add(), and Sd_compute_simulation().

3.8.2.2 int Score::nb_reaction

Number of reaction

Definition at line 56 of file simulation.h.

Referenced by Data_allocScore(), Data_desallocSim(), Data_scoreFree(), SBML_score(), and SBML_setReactions().

3.8.2.3 int Score::nb_species

Number of molecules

Definition at line 61 of file simulation.h.

Referenced by Data_allocScore(), Min_my_f(), Min_score_print_mean(), Mod_compute_modeling(), Mod_score_print_mean(), Recuit_energyFunction(), and Sd_compute_simulation().

3.8.2.4 double* Score::quantite

Quantity of molecules

Definition at line 54 of file simulation.h.

Referenced by Data_allocScore(), Data_desallocSim(), Data_scoreFree(), Data_scoreInit(), Min_getTampon(), Min_my_f(), Min_score_print_mean(), Mod_compute_modeling(), Mod_score_print_mean(), Recuit_energyFunction(), SBML_score(), SBML_score_add(), SBML_score_mean(), and Sd_compute_simulation().

3.8.2.5 char** Score::reaction

Interest reaction

Definition at line 55 of file simulation.h.

Referenced by Data_allocScore(), Data_desallocSim(), Data_scoreFree(), SBML_score(), and SBML_setReactions().

3.8.2.6 char** Score::species

Interest molecules

Definition at line 60 of file simulation.h.

Referenced by Data_allocScore(), Data_desallocSim(), Min_my_f(), Min_score_print_mean(), Mod_compute_modeling(), Mod_score_print_mean(), Recuit_energyFunction(), and Sd_compute_simulation().

3.8.2.7 int* Score::species_amount

Quantity of the interest molecules

Definition at line 62 of file simulation.h.

Referenced by Data_allocScore(), Data_desallocSim(), Data_scoreFree(), Min_my_f(), Min_score_print_mean(), Mod_compute_modeling(), Mod_score_print_mean(), Recuit_energyFunction(), and Sd_compute_simulation().

3.8.2.8 int* Score::species_weight

Weight of the interest molecules

Definition at line 63 of file simulation.h.

Referenced by Data_allocScore(), Data_desallocSim(), Data_scoreFree(), Min_my_f(), Mod_compute_modeling(), Recuit_energyFunction(), and Sd_compute_simulation().

3.8.2.9 int Score::taille

Total of other two sizes

Definition at line 59 of file simulation.h.

Referenced by Data_allocScore(), Data_desallocSim(), Data_scoreFree(), Data_scoreInit(), Min_getTampon(), Min_my_f(), Min_score_print_mean(), Mod_compute_modeling(), Mod_score_print_mean(), Recuit_energyFunction(), SBML_score_add(), SBML_score_mean(), and Sd_compute_simulation().

3.8.2.10 int Score::tailleReactions

Number of reaction in the SBML file

Definition at line 57 of file simulation.h.

Referenced by Data_allocScore(), and SBML_compute_simulation().

3.8.2.11 int Score::tailleSpecies

Number of molecules in the SBML file

Definition at line 58 of file simulation.h.

Referenced by Data_allocScore(), Min_my_f(), Min_score_print_mean(), Mod_compute_modeling(), Mod_score_print_mean(), Recuit_energyFunction(), SBML_compute_simulation(), SBML_score_add(), and Sd_compute_simulation().

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

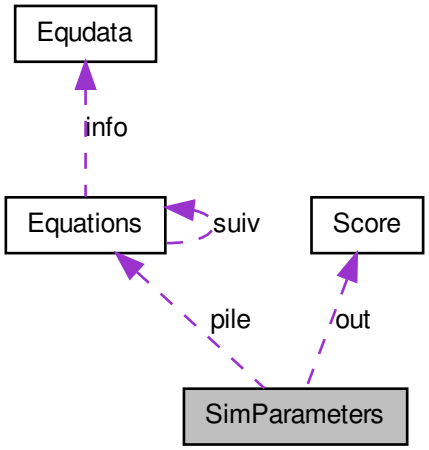
- metaboflux/include/simulation.h

3.9 SimParameters Struct Reference

List of parameters specific to each simulation.

`#include <data_parameters.h>`

Collaboration diagram for SimParameters:



Data Fields

- Equations ** pile
- pScore out
- gsl_rng * r
- double * y
- FILE * debugFile

3.9.1 Detailed Description

List of parameters specific to each simulation.

Definition at line 58 of file data_parameters.h.

3.9.2 Field Documentation

3.9.2.1 FILE* SimParameters::debugFile

Debug file : storage of intermediate values of the simulations

Definition at line 64 of file data_parameters.h.

Referenced by Data_desallocSim(), Data_simParameters(), Min_my_f(), Mod_compute_modeling(), Recuit_energyFunction(), and Sd_compute_simulation().

3.9.2.2 pScore SimParameters::out

Score

Definition at line 61 of file data_parameters.h.

Referenced by Data_desallocSim(), Data_simParameters(), Min_getTampon(), Min_my_f(), Min_score_print_mean(), Mod_compute_modeling(), Mod_score_print_mean(), Recuit_energyFunction(), and Sd_compute_simulation().

3.9.2.3 Equations** SimParameters::pile

Interpreted Equations

Definition at line 60 of file data_parameters.h.

Referenced by Data_simParameters(), Min_my_f(), Mod_compute_modeling(), Recuit_energyFunction(), and Sd_compute_simulation().

3.9.2.4 gsl_rng* SimParameters::r

Random number generator

Definition at line 62 of file data_parameters.h.

Referenced by Data_desallocSim(), Data_simParameters(), Min_my_f(), Mod_compute_modeling(), Recuit_compute_recuit(), Recuit_energyFunction(), and Sd_compute_simulation().

3.9.2.5 double* SimParameters::y

Table of reaction parameters

Definition at line 63 of file data_parameters.h.

Referenced by Data_copieTab(), Data_desallocSim(), Data_simParameters(), Min_compute_minimization(), Min_copieTab2(), Min_copieTab3(), Min_getTampon(), Min_my_f(), Min_score_print_mean(), Mod_compute_modeling(), Mod_score_print_mean(), Recuit_energyFunction(), Recuit_printPosition(), Sd_compute_simulation(), and Sd_compute_standard_deviation().

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

- metaboflux/include/data_parameters.h

3.10 TestReaction Struct Reference

Structure used to test reactions.

`#include <simulation.h>`

Data Fields

- Reaction_t ** tabReactions
- int * minStepTab

3.10.1 Detailed Description

Structure used to test reactions.

Definition at line 40 of file simulation.h.

3.10.2 Field Documentation

3.10.2.1 int* TestReaction::minStepTab

Stock le nombre de realisation possible pour cette reaction

Definition at line 43 of file simulation.h.

Referenced by SBML_allocTest(), SBML_EstimationReaction(), and SBML_freeTest().

3.10.2.2 Reaction_t** TestReaction::tabReactions

Stock temporairement une reaction

Definition at line 42 of file simulation.h.

Referenced by SBML_allocTest(), SBML_EstimationReaction(), and SBML_freeTest().

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

- metaboflux/include/simulation.h

3.11 xmlConfig_t Struct Reference

Structure containing all information from parameter.xml.

`#include <xml_parameter.h>`

Data Fields

- char * fichier
- xmlDocPtr doc
- xmlNodePtr racine
- xmlXPathContextPtr ctxt

3.11.1 Detailed Description

Structure containing all information from parameter.xml.

Definition at line 33 of file xml_parameter.h.

3.11.2 Field Documentation

3.11.2.1 xmlXPathContextPtr xmlConfig_t::ctxt

Xml ctxt

Definition at line 37 of file xml_parameter.h.

Referenced by Xml_freeConfig(), Xml_getBanned(), Xml_getCount(), Xml_getDoubleNumber(), Xml_getKineticLaw(), Xml_getKineticLawNodes(), Xml_getNumber(), Xml_getReactionsNames(), Xml_getReactionsNamesinNoeud(), Xml_getSpeciesFinalAmount(), Xml_getSpeciesWeight(), Xml_getString(), and Xml_loadConfig().

3.11.2.2 xmlDocPtr xmlConfig_t::doc

Xml doc

Definition at line 35 of file xml_parameter.h.

Referenced by Xml_freeConfig(), and Xml_loadConfig().

3.11.2.3 char* xmlConfig_t::fichier

Name of the parameter file

Definition at line 34 of file xml_parameter.h.

Referenced by Xml_freeConfig(), and Xml_loadConfig().

3.11.2.4 xmlNodePtr xmlConfig_t::racine

Xml root

Definition at line 36 of file xml_parameter.h.

Referenced by Xml_loadConfig().

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

- metaboflux/include/xml_parameter.h

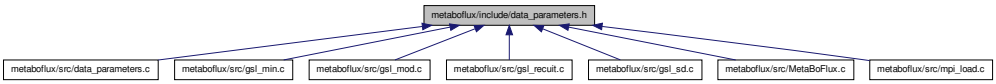
Chapter 4

File Documentation

4.1 metaboflux/include/data_parameters.h File Reference

Load data parameters.

This graph shows which files directly or indirectly include this file:



Data Structures

- struct ListParameters

  Global list parameters.
- struct SimParameters

  List of parameters specific to each simulation.

Functions

- xmlConfig_t * Data_initXML (char *)

  Load parameter file.
- Model_t * Data_initSBML (char *)

  Load SBML file.
- gsl_rng * Data_rngAlloc (gsl_rng *)

  Allocation of the random number generator.
- Equations ** Data_equationsAlloc (pListParameters, Equations **)

  Allocate the struct Equations.

- void Data_equationsInit (pListParameters, Equations **)

  Initialize equations.
- void Data_allocEquations (pListParameters a)

  Allocate the different elements related to he struct Equations.
- void Data_allocScore (pScore, xmlConfig_t *, Model_t *)

  Copy the partial table x in the full table y.
- pScore Data_scoreAlloc (pListParameters)

  Allocation of the struct Score.
- void Data_scoreInit (pScore)

  Initialize the struct Score.
- void Data_scoreFree (pScore)

  Free the struct Score.
- int Data_copieTab (pSimParameters, double *, int, int, int)

  Copy the partial table x in the full table y.
- void Data_updateTab (pListParameters, pSimParameters, double *)

  Copy table.
- void Data_allocSimParameters (double *, pListParameters)

  Allocation of tables needed for simulations.
- void Data_allocParameters (pListParameters)

  Copy the partial table x in the full table y.
- void Data_parameters (pListParameters, char **)

  Record parameters of the simulation.
- void Data_desallocation (pListParameters)

  Free memory allocated to the various objects of the program.
- void Data_simParameters (pListParameters, pSimParameters, char *, char *, int, int)

  Recording parameters specific to each simulation.
- void Data_desallocSim (pSimParameters)

  Free struct parameters specific to each simulation.

4.1.1 Detailed Description

Load data parameters. This file is part of MetaBoFlux (http://www.cbib.u-bordeaux2.fr/metabofl
Copyright (C) 2010 Amine Ghozlane from LaBRI and University of Bordeaux 1

MetaBoFlux is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

MetaBoFlux is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Author

{Amine Ghozlane}

Version

2.0

Date

10 novembre 2010

Definition in file data_parameters.h.

4.1.2 Function Documentation

4.1.2.1 void Data_allocEquations (pListParameters *a*)

Allocate the different elements related to he struct Equations.

Author

Amine Ghozlane

Parameters

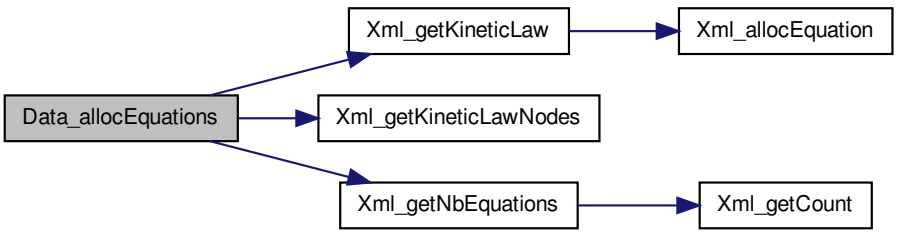
a	Global parameters : struct ListParameters

Definition at line 202 of file data_parameters.c.

References ListParameters::conf, ListParameters::equation, ListParameters::nb_equations, ListParameters::noeud, Xml_getKineticLaw(), Xml_getKineticLawNodes(), and Xml_getNbEquations().

Referenced by Data_parameters().

Here is the call graph for this function:



4.1.2.2 void Data_allocParameters (pListParameters *a*)

Copy the partial table x in the full table y.

Author

Amine Ghozlane

Parameters

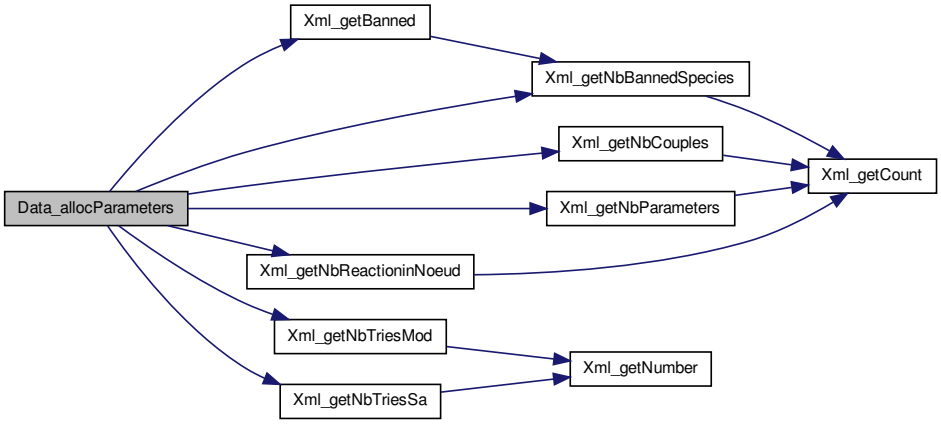
a	Global parameters : struct ListParameters

Definition at line 408 of file data_parameters.c.

References ListParameters::banned, ListParameters::conf, ListParameters::interest_parameters, ListParameters::nb_banned, ListParameters::nb_couples, ListParameters::nb_parameters, ListParameters::nb_triesMod, ListParameters::nb_triesSa, ListParameters::parameters, Xml_getBanned(), Xml_getNbBannedSpecies(), Xml_getNbCouples(), Xml_getNbParameters(), Xml_getNbReactioninNoeud(), Xml_getNbTriesMod(), and Xml_getNbTriesSa().

Referenced by Data_parameters().

Here is the call graph for this function:



4.1.2.3 void Data_allocScore (pScore *out,* xmlConfig_t * *conf,* Model_t * *model*)

Copy the partial table x in the full table y.

Author

Amine Ghozlane

Parameters

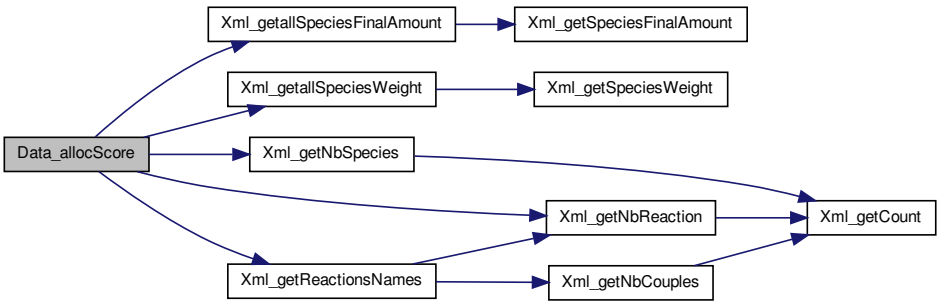
out	Struct Score
conf	Struct xmlConfig_t
model	Model of the SBML file

Definition at line 238 of file data_parameters.c.

References Score::name, Score::nb_reaction, Score::nb_species, Score::quantite, Score::reaction, Score::species, Score::species_amount, Score::species_weight, Score::taille, Score::tailleReactions, Score::tailleSpecies, Xml_getallSpeciesFinalAmount(), Xml_getallSpeciesWeight(), Xml_getNbReaction(), Xml_getNbSpecies(), and Xml_getReactionsNames().

Referenced by Data_scoreAlloc(), and Data_simParameters().

Here is the call graph for this function:



4.1.2.4 void Data_allocSimParameters (double * *y*, pListParameters *a*)

Allocation of tables needed for simulations.

void Data_allocSimParameters(double *y, pListParameters a)

Author

Amine Ghozlane

Parameters

y	table of reaction parameters
a	Global parameters : struct ListParameters

Definition at line 391 of file data_parameters.c.

References ListParameters::nb_parameters.

Referenced by Data_simParameters().

4.1.2.5 int Data_copieTab (pSimParameters *simulated*, double * *x*, int *a*, int *debut*, int *fin*)

Copy the partial table x in the full table y.

Author

Amine Ghozlane

Parameters

simulated	Simulation parameters : struct SimParameters
x	Short table of reaction parameters
a	Line
debut	Beginning
fin	End

Returns

Number of copied element

Definition at line 347 of file data_parameters.c.

References SimParameters::y.

Referenced by Data_updateTab().

4.1.2.6 void Data_desallocation (pListParameters *a*)

Free memory allocated to the various objects of the program.

Author

Amine Ghozlane

Parameters

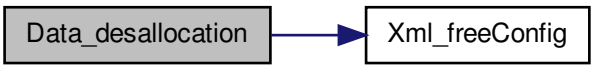
a	Global parameters : struct ListParameters

Definition at line 463 of file data_parameters.c.

References ListParameters::banned, ListParameters::conf, ListParameters::equation, ListParameters::model, ListParameters::nb_equations, ListParameters::noeud, ListParameters::parameters, and Xml_freeConfig().

Referenced by compute_mpi().

Here is the call graph for this function:



4.1.2.7 void Data_desallocSim (pSimParameters *sim*)

Free struct parameters specific to each simulation.

Author

Amine Ghozlane

Parameters

sim	Simulation parameters : struct SimParameters

Definition at line 549 of file data_parameters.c.

References SimParameters::debugFile, Score::name, Score::nb_reaction, SimParameters::out, Score::quantite, SimParameters::r, Score::reaction, Score::species, Score::species_amount, Score::species_weight, Score::taille, and SimParameters::y.

Referenced by Min_compute_minimization(), Mod_compute_modeling(), Recuit_compute_recuit(), and Sd_compute_standard_deviation().

4.1.2.8 Equations** Data_equationsAlloc (pListParameters *current*, Equations ** *pile*)

Allocate the struct Equations.

Author

Amine Ghozlane

Parameters

current	Current parameters : struct ListParameters
pile	Pile des equations : struct Equations

Returns

Allocated struct Equations

Definition at line 169 of file data_parameters.c.

References ListParameters::nb_equations.

Referenced by Min_my_f(), Mod_compute_modeling(), Recuit_energyFunction(), and Sd_compute_simulation().

4.1.2.9 void Data_equationsInit (pListParameters *current,* Equations ** *pile*)

Initialize equations.

Author

Amine Ghozlane

Parameters

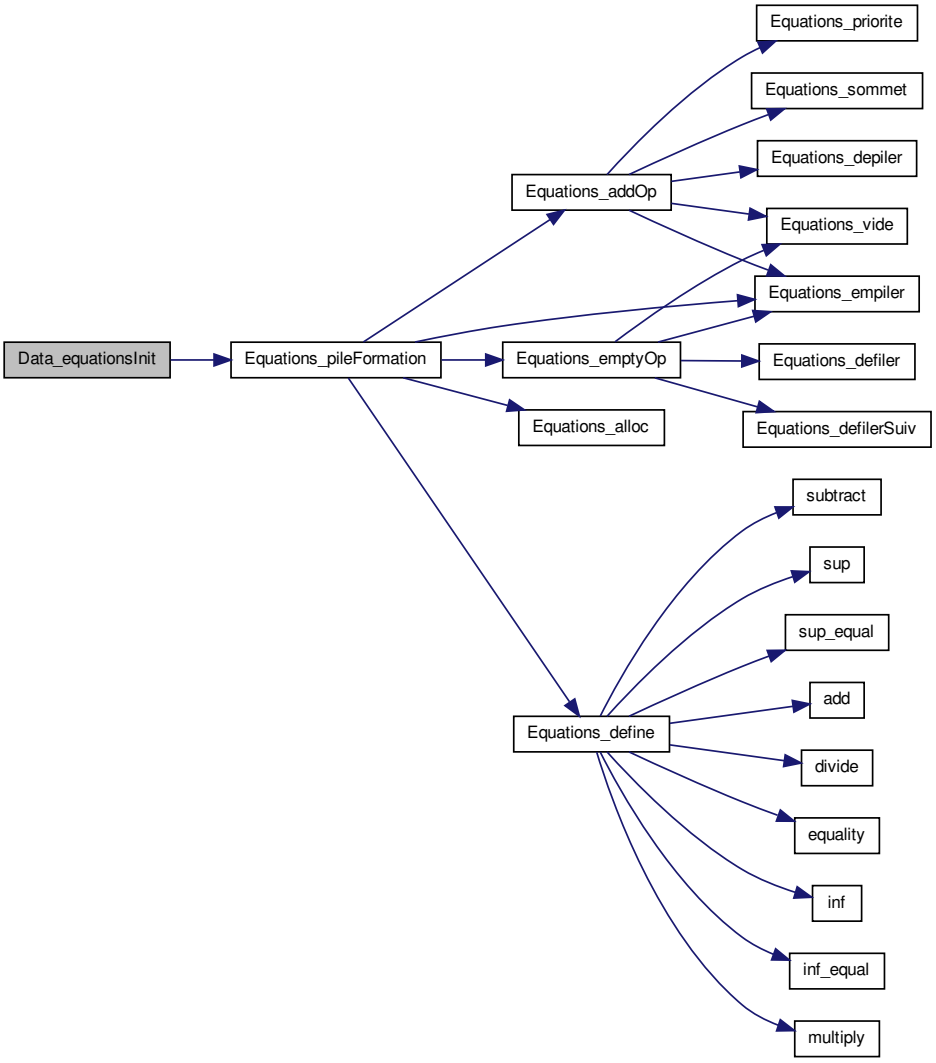
current	Current parameters : struct ListParameters
pile	Pile des equations : struct Equations

Definition at line 184 of file data_parameters.c.

References ListParameters::equation, Equations_pileFormation(), ListParameters::nb_equations, and ListParameters::noeud.

Referenced by Min_my_f(), Mod_compute_modeling(), Recuit_energyFunction(), and Sd_compute_simulation().

Here is the call graph for this function:



4.1.2.10 Model_t* Data_initSBML (char * *sbml_file*)

Load SBML file.

Author

Amine Ghozlane

Parameters

sbml_file	Name of the SBML file

Returns

Model of the SBML file

Definition at line 81 of file data_parameters.c.

Referenced by Data_parameters().

4.1.2.11 xmlConfig_t* Data_initXML (char * *xml_file*)

Load parameter file.

Author

Amine Ghozlane

Parameters

xml_file	Name of the parameter file

Returns

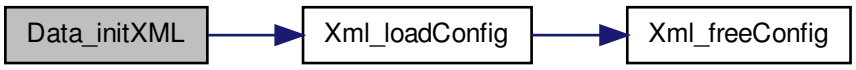
Model of the parameter file =:Struct xmlConfig_t

Definition at line 58 of file data_parameters.c.

References Xml_loadConfig().

Referenced by Data_parameters().

Here is the call graph for this function:



4.1.2.12 void Data_parameters (pListParameters *a,* char ** *files_path*)

Record parameters of the simulation.

Author

Amine Ghozlane

Parameters

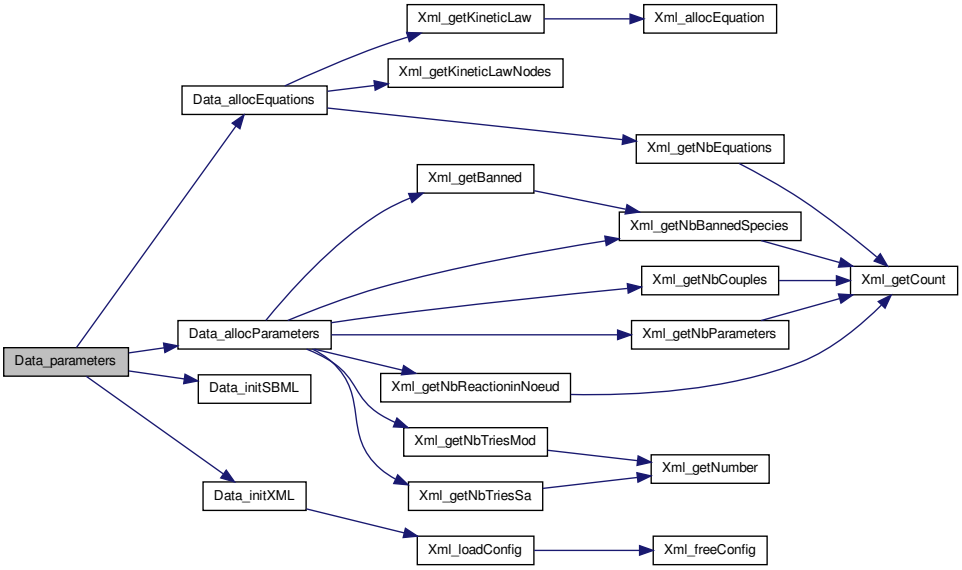
a	Global parameters : struct ListParameters
files_path	List of paths

Definition at line 442 of file data_parameters.c.

References ListParameters::conf, Data_allocEquations(), Data_allocParameters(), Data_initSBML(), Data_initXML(), and ListParameters::model.

Referenced by compute_mpi().

Here is the call graph for this function:



4.1.2.13 gsl_rng* Data_rngAlloc (gsl_rng * *r*)

Allocation of the random number generator.

Author

Amine Ghozlane

Parameters

r	Random number generator

Returns

Random number generator

Definition at line 135 of file data_parameters.c.

Referenced by Data_simParameters().

4.1.2.14 pScore Data_scoreAlloc (pListParameters *a*)

Allocation of the struct Score.

Author

Amine Ghozlane

Parameters

a	Global parameters : struct ListParameters

Returns

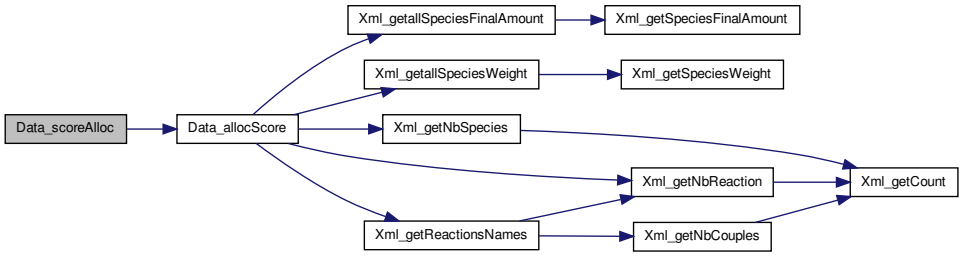
Allocated struct Score

Definition at line 274 of file data_parameters.c.

References ListParameters::conf, Data_allocScore(), and ListParameters::model.

Referenced by Min_compute_minimization(), Mod_compute_modeling(), and Recuit_-compute_recuit().

Here is the call graph for this function:



4.1.2.15 void Data_scoreFree (pScore *out*)

Free the struct Score.

Author

Amine Ghozlane

Parameters

out	Struct score

Definition at line 307 of file data_parameters.c.

References Score::name, Score::nb_reaction, Score::quantite, Score::reaction, Score::species_amount, Score::species_weight, and Score::taille.

Referenced by Min_compute_minimization(), Mod_compute_modeling(), and Recuit_compute_recuit().

4.1.2.16 void Data_scoreInit (pScore *out*)

Initialize the struct Score.

Author

Amine Ghozlane

Parameters

out	Empty struct Score

Definition at line 290 of file data_parameters.c.

References Score::quantite, and Score::taille.

Referenced by Min_my_f(), Recuit_energyFunction(), and Sd_compute_simulation().

4.1.2.17 void Data_simParameters (pListParameters *a*, pSimParameters *sim*, char * *out*, char * *texte*, int *number*, int *debug*)

Recording parameters specific to each simulation.

Author

Amine Ghozlane

Parameters

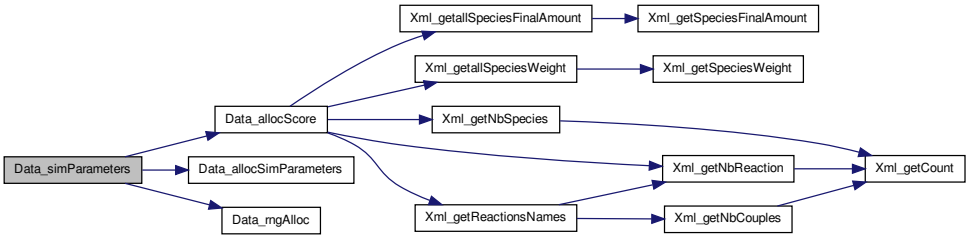
a	Global parameters : struct ListParameters
sim	Simulation parameters : struct SimParameters
out	Output repertory
texte	File name
number	Number of the debug file
debug	Determine debug

Definition at line 506 of file data_parameters.c.

References ListParameters::conf, Data_allocScore(), Data_allocSimParameters(), Data_rngAlloc(), SimParameters::debugFile, ListParameters::model, ListParameters::nb_parameters, SimParameters::out, SimParameters::pile, SimParameters::r, and SimParameters::y.

Referenced by Min_compute_minimization(), Mod_compute_modeling(), Recuit_compute_-recuit(), and Sd_compute_standard_deviation().

Here is the call graph for this function:



4.1.2.18 void Data_updateTab (pListParameters *current*, pSimParameters *simulated*, double * *x*)

Copy table.

Author

Amine Ghozlane

Parameters

current	Current parameters : struct ListParameters
simulated	Simulation parameters : struct SimParameters
x	Short table of reaction parameters

Definition at line 371 of file data_parameters.c.

References Data_copieTab(), ListParameters::nb_couples, and ListParameters::parameters.

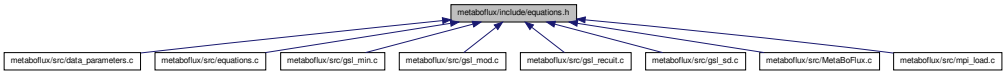
Referenced by Min_my_f(), Recuit_energyFunction(), and Recuit_printPosition().

Here is the call graph for this function:

4.2 metaboflux/include/equations.h File Reference

Processes an equation in MathML format.

This graph shows which files directly or indirectly include this file:



Data Structures

- union Equdata

  Equation data.
- struct Equations

  Function pointers used for operations.

Enumerations

- enum Equtype {

  constant, variable, addition, soustraction,

  multiplication, division, equal, superior,

  superior_equal, inferior, inferior_equal }

  Operation constants.

Functions

- double add (double, double)

  Add two numbers.
- double subtract (double, double)

  Subtraction two numbers.
- double multiply (double, double)

  Multiply two numbers.
- double divide (double, double)

  Divide two numbers.
- double equality (double, double)

  Test the equality of two numbers.
- double sup (double, double)

  Test the superiority between two numbers.
- double sup_equal (double, double)

  Test the superiority or equality between two numbers.

- double inf (double, double)

  Test the inferiority between two numbers.
- double inf_equal (double, double)

  Test the inferiority or equality between two numbers.
- pEquations Equations_alloc (void)

  Alloc memory and initialize the struct Equations.
- void Equations_define (pEquations, char ∗)

  Identify the mathematical operator used.
- int Equations_vide (pEquations)

  Test if the struct Equations is empty.
- pEquations Equations_empiler (pEquations, pEquations)

  Stack an element to the struct Equations.
- pEquations Equations_depiler (pEquations)

  Unstack the last element of the struct Equations.
- pEquations Equations_sommet (pEquations)

  Look for the last element of the struct Equations.
- pEquations Equations_defiler (pEquations)

  Look for the first element of the struct Equations.
- pEquations Equations_defilerSuiv (pEquations)

  Get the next element of the struct Equations.
- int Equations_priorite (pEquations)

  Give the priority of the operator. Multiplication and Division have higher priority than addition or subtration...
- void Equations_print (pEquations)

  Print all information on the element of the struct Equations.
- pEquations Equations_addOp (pEquations, pEquations, pEquations)

  Build the list of the Struct Equations result used to compute the equation.
- void Equations_emptyOp (pEquations, pEquations)

  Empty the Struct Equations op for the struct Equation result.
- pEquations Equations_pileFormation (char ∗∗∗, int)

  Build pile.
- double Equations_findSpecies (char ∗, char ∗∗, double ∗, int)

  Look for the quantity of a molecule in the list.
- double Equations_resultat (pEquations, pEquations, pEquations, char ∗∗, double ∗, int)

  Compute the result of two pile.
- double Equations_calcul (pEquations, char ∗∗, double ∗, int)

  Compute the score of the equation.
- double Equations_finalQuantite (int, char ∗∗, int ∗, int ∗, char ∗∗, double ∗, int)

  Compute the score of the quantity which means the difference between what is expected to what is simulated.

4.2.1 Detailed Description

Processes an equation in MathML format. This file is part of MetaBoFlux (http://www.cbib.u-bordeaux2.fr/meta

Author

{Amine Ghozlane}

Version

2.0

Date

15 janvier 2010

Definition in file equations.h.

4.2.2 Enumeration Type Documentation

4.2.2.1 enum Equtype

Operation constants.

Equtype is a set of predefined constants for the different operators.

Enumerator:

constant Constant
variable Variable : ex. transition_x ...
addition Addition
soustraction Substraction
multiplication Multiply
division Divide
equal Equality
superior Superiority
superior_equal Sup_equality
inferior Inferiority
inferior_equal Inf_equality

Definition at line 38 of file equations.h.

4.2.3 Function Documentation

4.2.3.1 double add (double *arg1*, double *arg2*)

Add two numbers.

Author

Amine Ghozlane

Parameters

arg1	Value 1
arg2	Value 2

Returns

Result

Definition at line 43 of file equations.c.

Referenced by Equations_define().

4.2.3.2 double divide (double *arg1*, double *arg2*)

Divide two numbers.

Author

Amine Ghozlane

Parameters

arg1	Value 1
arg2	Value 2

Returns

Result

Definition at line 82 of file equations.c.

Referenced by Equations_define().

4.2.3.3 double equality (double *arg1*, double *arg2*)

Test the equality of two numbers.

Author

Amine Ghozlane

Parameters

arg1	Value 1
arg2	Value 2

Returns

Result

Definition at line 95 of file equations.c.

Referenced by Equations_define().

4.2.3.4 pEquations Equations_addOp (pEquations *result,* pEquations *op,* pEquations *new*)

Build the list of the Struct Equations result used to compute the equation.

Author

Amine Ghozlane

Parameters

result	Struct Equations used for computation
op	Struct Equations used to store the operator
new	One element of the Struct Equations

Returns

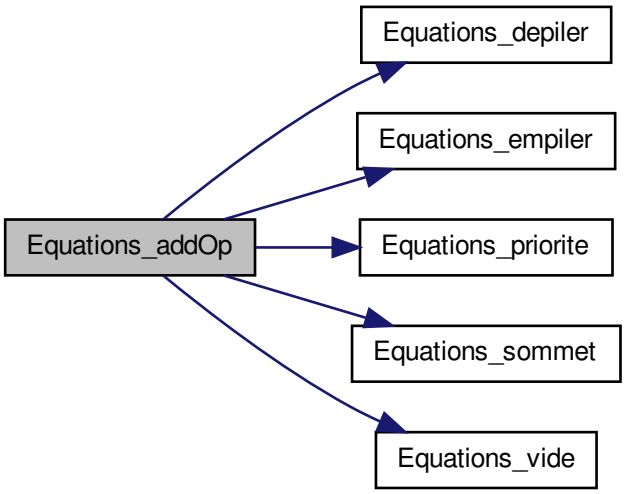
List of operator

Definition at line 383 of file equations.c.

References equal, Equations_depiler(), Equations_empiler(), Equations_priorite(), Equations_-sommet(), Equations_vide(), inferior, inferior_equal, superior, and superior_equal.

Referenced by Equations_pileFormation().

Here is the call graph for this function:



4.2.3.5 pEquations Equations_alloc (void)

Alloc memory and initialize the struct Equations.

Author

Amine Ghozlane

Returns

Allocated struct Equations

Definition at line 158 of file equations.c.

Referenced by Equations_calcul(), and Equations_pileFormation().

4.2.3.6 double Equations_calcul (pEquations *liste*, char ** *name*, double * *quantite*, int *taille*)

Compute the score of the equation.

Author

Amine Ghozlane

Parameters

liste	pile of type Struct Equations
name	List of molecules
quantite	List of the quantity of the molecules
taille	Number of molecules in the list

Returns

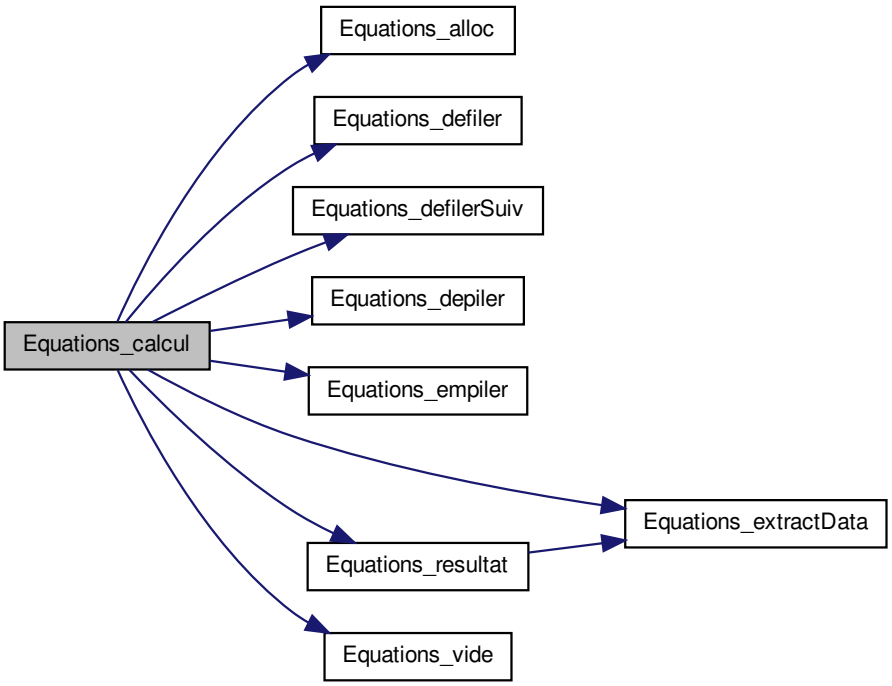
Result of the equation (First part of the energy)

Definition at line 614 of file equations.c.

References addition, constant, division, equal, Equations_alloc(), Equations_defiler(), Equations_defilerSuiv(), Equations_depiler(), Equations_empiler(), Equations_extractData(), Equations_resultat(), Equations_vide(), inferior, inferior_equal, multiplication, soustraction, superior, and superior_equal.

Referenced by Min_my_f(), Mod_compute_modeling(), Recuit_energyFunction(), and Sd_compute_simulation().

Here is the call graph for this function:



4.2.3.7 pEquations Equations_defiler (pEquations *liste*)

Look for the first element of the struct Equations.

Author

Amine Ghozlane

Parameters

liste	Struct Equations

Returns

Pointer on the first element of struct Equations

Definition at line 320 of file equations.c.

Referenced by Equations_calcul(), and Equations_emptyOp().

4.2.3.8 pEquations Equations_defilerSuiv (pEquations *liste*)

Get the next element of the struct Equations.

Author

Amine Ghozlane

Parameters

liste	Struct Equations

Returns

Pointer on the next element of struct Equations

Definition at line 334 of file equations.c.

Referenced by Equations_calcul(), and Equations_emptyOp().

4.2.3.9 void Equations_define (pEquations *new*, char * *operateur*)

Identify the mathematical operator used.

Author

Amine Ghozlane

Parameters

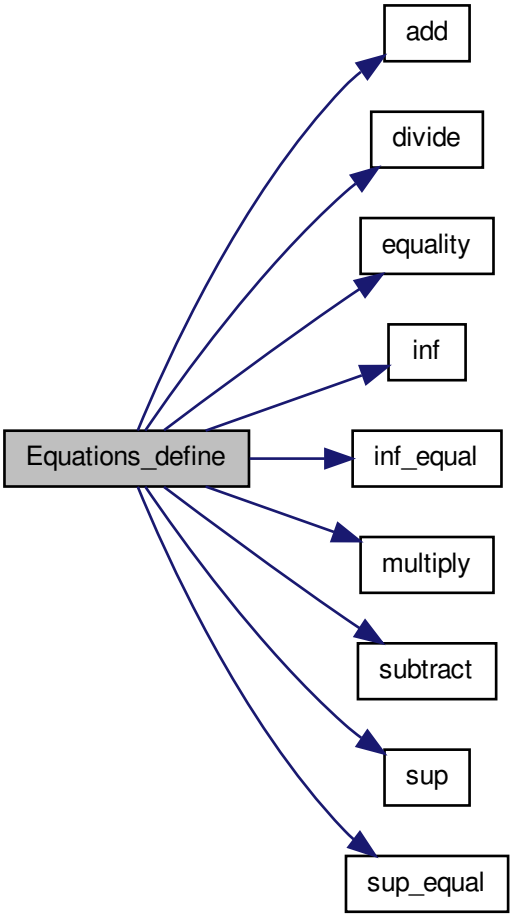
new	Struct Equations
operateur	Read operator

Definition at line 174 of file equations.c.

References add(), addition, divide(), division, equal, equality(), inf(), inf_equal(), inferior, inferior_equal, multiplication, multiply(), soustraction, subtract(), sup(), sup_equal(), superior, and superior_equal.

Referenced by Equations_pileFormation().

Here is the call graph for this function:



4.2.3.10 pEquations Equations_depiler (pEquations *liste*)

Unstack the last element of the struct Equations.

Author

Amine Ghozlane

Parameters

liste	Struct Equations.

Returns

Pointer on the "unstack" element of struct Equations.

Definition at line 282 of file equations.c.

Referenced by Equations_addOp(), and Equations_calcul().

4.2.3.11 pEquations Equations_empiler (pEquations *liste,* pEquations *new*)

Stack an element to the struct Equations.

Author

Amine Ghozlane

Parameters

liste	Struct Equations
new	New struct Equations element

Returns

Pointer on the last element of struct Equations

Definition at line 258 of file equations.c.

Referenced by Equations_addOp(), Equations_calcul(), Equations_emptyOp(), and Equations_-pileFormation().

4.2.3.12 void Equations_emptyOp (pEquations *result,* pEquations *op*)

Empty the Struct Equations op for the struct Equation result.

Author

Amine Ghozlane

Parameters

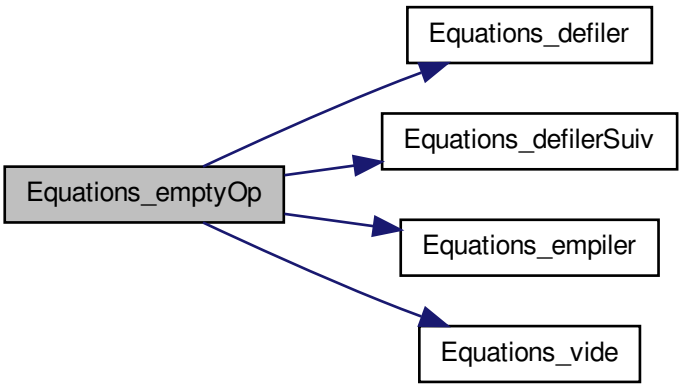
result	Struct Equations used for computation
op	Struct Equation used to store the operator

Definition at line 421 of file equations.c.

References Equations_defiler(), Equations_defilerSuiv(), Equations_empiler(), and Equations_vide().

Referenced by Equations_pileFormation().

Here is the call graph for this function:



4.2.3.13 double Equations_finalQuantite (int *file_nb_especes*, char ** *file_species*, int * *file_amount*, int * *file_weight*, char ** *sim_name*, double * *sim_quantite*, int *sim_taille*)

Compute the score of the quantity which means the difference between what is expected to what is simulated.

Author

Amine Ghozlane

Parameters

file_nb_especes	Number of species in the parameter file
file_species	List of species in the parameter file
file_amount	List of species expected quantity
file_weight	Weight defined for the species
sim_name	List of species simulated (sbml file)
sim_quantite	List of the quantity of the species simulated (sbml file)
sim_taille	Number of species simulated (sbml file)

Returns

Result of the difference (Second part of the Energy)

Definition at line 683 of file equations.c.

References Equations_findSpecies().

Referenced by Min_my_f(), Mod_compute_modeling(), Recuit_energyFunction(), and Sd_compute_simulation().

Here is the call graph for this function:

4.2.3.14 double Equations_findSpecies (char * *species*, char ** *name*, double * *quantite*, int *taille*)

Look for the quantity of a molecule in the list.

Author

Amine Ghozlane

Parameters

species	Name of a molecule
name	List of molecules
quantite	List of the quantity of the molecules
taille	Number of molecules in the list

Returns

Quantity of the molecule of interest

Definition at line 533 of file equations.c.

Referenced by Equations_finalQuantite(), Min_score_print_mean(), and Mod_score_print_mean().

4.2.3.15 pEquations Equations_pileFormation (char *** *equation*, int *nb_noeud*)

Build pile.

Author

Amine Ghozlane

Parameters

equation	Table with the equation in Mathml format
nb_noeud	Number of element inside the equation

Returns

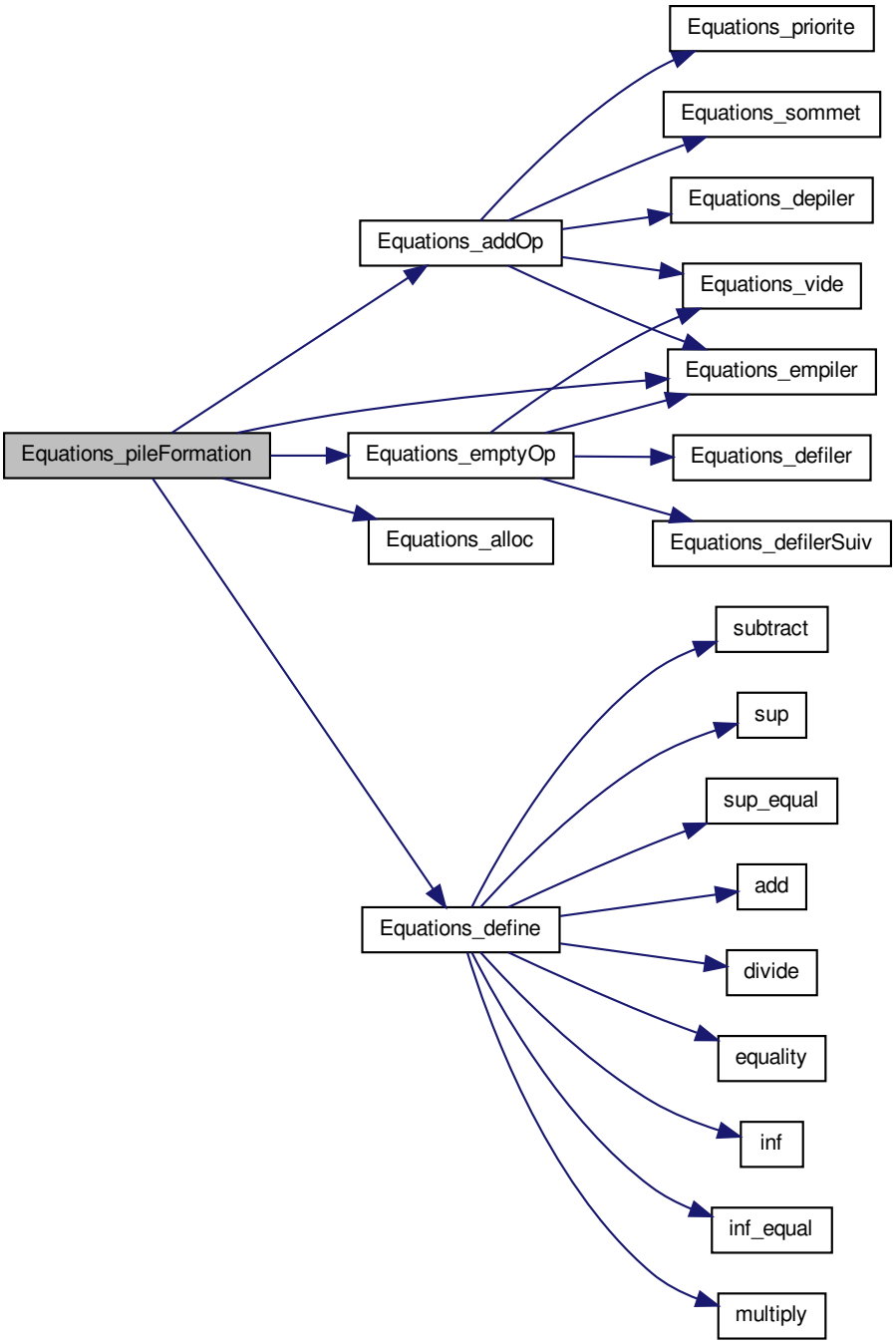
Struct Equation needed for compute the equation

Definition at line 441 of file equations.c.

References constant, Equations_addOp(), Equations_alloc(), Equations_define(), Equations_empiler(), Equations_emptyOp(), and variable.

Referenced by Data_equationsInit().

Here is the call graph for this function:



4.2.3.16 void Equations_print (pEquations *liste*)

Print all information on the element of the struct Equations.

Author

Amine Ghozlane

Parameters

liste	Struct Equations

Definition at line 359 of file equations.c.

References constant, Equdata::data, Equations::info, Equations::type, Equdata::var, and variable.

4.2.3.17 int Equations_priorite (pEquations *new*)

Give the priority of the operator. Multiplication and Division have higher priority than addition or subtration...

Author

Amine Ghozlane

Parameters

new	One element of the struct Equations

Returns

Priority of the operator

Definition at line 346 of file equations.c.

References addition, equal, inferior, inferior_equal, soustraction, superior, and superior_equal.

Referenced by Equations_addOp().

4.2.3.18 double Equations_resultat (pEquations *result,* pEquations *result1,* pEquations *eq,* char ∗∗ *name,* double ∗ *quantite,* int *taille*)

Compute the result of two pile.

Author

Amine Ghozlane

Parameters

result	Struct Equation result

result1	Struct Equation result after the operator
eq	Operator
name	List of molecules
quantite	List of the quantity of the molecules
taille	Number of molecules in the list

Returns

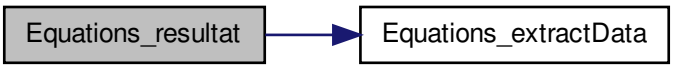
Result of the two pile

Definition at line 561 of file equations.c.

References equal, Equations_extractData(), inferior, inferior_equal, superior, superior_-equal, and Equations::type.

Referenced by Equations_calcul().

Here is the call graph for this function:



4.2.3.19 pEquations Equations_sommet (pEquations *liste*)

Look for the last element of the struct Equations.

Author

Amine Ghozlane

Parameters

liste	Struct Equations

Returns

Pointer on the last element of struct Equations

Definition at line 305 of file equations.c.

Referenced by Equations_addOp().

4.2.3.20 int Equations_vide (pEquations *liste*)

Test if the struct Equations is empty.

Author

Amine Ghozlane

Parameters

liste	Struct Equations

Returns

Number of elements

Definition at line 237 of file equations.c.

Referenced by Equations_addOp(), Equations_calcul(), and Equations_emptyOp().

4.2.3.21 double inf (double *arg1*, double *arg2*)

Test the inferiority between two numbers.

Author

Amine Ghozlane

Parameters

arg1	Value 1
arg2	Value 2

Returns

Result

Definition at line 134 of file equations.c.

Referenced by Equations_define().

4.2.3.22 double inf_equal (double *arg1*, double *arg2*)

Test the inferiority or equality between two numbers.

Author

Amine Ghozlane

Parameters

arg1	Value 1
arg2	Value 2

Returns

Result

Definition at line 147 of file equations.c.

Referenced by Equations_define().

4.2.3.23 double multiply (double *arg1*, double *arg2*)

Multiply two numbers.

Author

Amine Ghozlane

Parameters

arg1	Value 1
arg2	Value 2

Returns

Result

Definition at line 69 of file equations.c.

Referenced by Equations_define().

4.2.3.24 double subtract (double *arg1*, double *arg2*)

Subtraction two numbers.

Author

Amine Ghozlane

Parameters

arg1	Value 1
arg2	Value 2

Returns

Result

Definition at line 56 of file equations.c.

Referenced by Equations_define().

4.2.3.25 double sup (double *arg1*, double *arg2*)

Test the superiority between two numbers.

Author

Amine Ghozlane

Parameters

arg1	Value 1
arg2	Value 2

Returns

Result

Definition at line 108 of file equations.c.

Referenced by Equations_define().

4.2.3.26 double sup_equal (double *arg1*, double *arg2*)

Test the superiority or equality between two numbers.

Author

Amine Ghozlane

Parameters

arg1	Value 1
arg2	Value 2

Returns

Result

Definition at line 121 of file equations.c.

Referenced by Equations_define().

4.3 metaboflux/include/especes.h File Reference

Modelize a molecule.

This graph shows which files directly or indirectly include this file:

Data Structures

- struct Reaction

  Reactions data.
- struct Especes

  Molecules data.

Functions

- pEspeces Especes_alloc (int)

  Alloc memory and initialize the struct Especes.
- void Especes_allocReactions (pEspeces, int, Reaction_t ∗, double)

  Alloc memory and initialize the struct Reaction for each molecule.
- void Especes_free (pEspeces, int)

  Free the struct pEspeces.
- void Especes_freeReactions (Especes)

  Free the struct Reaction.
- void Especes_save (pEspeces, int, double, const char ∗)

  Save data on one molecule.
- void Especes_scoreSpecies (pEspeces, int, char ∗∗, double ∗)

  Save the quantite of all molecules for scoring.
- void Especes_print (pEspeces, int)

  Print data on each molecule.
- void Especes_print_2 (pEspeces molecules, int nbEspeces)

  Print Id and quantity on each molecule.
- int Especes_find (pEspeces, const char ∗, int)

  Find a molecule thank to it ID.
- void Especes_setQuantite (pEspeces, int, double)

  Change the quantity of a molecule.
- double Especes_getQuantite (pEspeces, int)

  Get the quanty of one molecule by its number.
- int Especes_getNbreactions (pEspeces molecules, int ref)

  Get number of reaction where one molecule is used (like a reactif)

4.3.1 Detailed Description

Modelize a molecule. This file is part of MetaBoFlux (http://www.cbib.u-bordeaux2.fr/metaboflu

MetaBoFlux is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Author

{Amine Ghozlane}

Version

2.0

Date

27 octobre 2009

Definition in file especes.h.

4.3.2 Function Documentation

4.3.2.1 pEspeces Especes_alloc (int *nbEspeces*)

Alloc memory and initialize the struct Especes.

Author

Amine Ghozlane

Parameters

nbEspeces	Number of molecules

Returns

Allocated struct Espece

Definition at line 40 of file especes.c.

References Especes::quantite, and Especes::system.

Referenced by SBML_compute_simulation(), and SBML_compute_simulation_mean().

4.3.2.2 void Especes_allocReactions (pEspeces *molecules*, int *Espece*, Reaction_t * *local*, double *reactRatio*)

Alloc memory and initialize the struct Reaction for each molecule.

Author

Amine Ghozlane

Parameters

molecules	List of molecules
Espece	Number of the selected molecule
local	Reaction sbml reference
reactRatio	Ratio of the reaction

Definition at line 64 of file especes.c.

References Reaction::link, Reaction::suivant, and Especes::system.

Referenced by SBML_setReactions().

4.3.2.3 int Especes_find (pEspeces *molecules*, const char * *especeId*, int *nbEspeces*)

Find a molecule thank to it ID.

Author

Amine Ghozlane

Parameters

molecules	List of molecules
especeId	SBML ID of the molecule
nbEspeces	Number of molecules

Returns

Number of a selected molecule

Definition at line 214 of file especes.c.

Referenced by SBML_checkQuantite(), SBML_reaction(), and SBML_setReactions().

4.3.2.4 void Especes_free (pEspeces *molecules*, int *nbEspeces*)

Free the struct pEspeces.

Author

Amine Ghozlane

Parameters

molecules	List of molecules
nbEspeces	Number of molecules

Definition at line 94 of file especes.c.

References Especes_freeReactions().

Referenced by SBML_compute_simulation(), and SBML_compute_simulation_mean().

Here is the call graph for this function:

4.3.2.5 void Especes_freeReactions (Especes *molecules*)

Free the struct Reaction.

Author

Amine Ghozlane

Parameters

molecules	List of molecules

Definition at line 109 of file especes.c.

References Reaction::suivant, and Especes::system.

Referenced by Especes_free().

4.3.2.6 int Especes_getNbreactions (pEspeces *molecules,* int *ref*)

Get number of reaction where one molecule is used (like a reactif)

Author

Amine Ghozlane

Parameters

molecules	List of molecules
ref	Number of a selected molecule

Returns

Number of reactions

Definition at line 265 of file especes.c.

References Reaction::suivant, and Especes::system.

Referenced by SBML_simulate().

4.3.2.7 double Especes_getQuantite (pEspeces *molecules*, int *Especes*)

Get the quanty of one molecule by its number.

Author

Amine Ghozlane

Parameters

molecules	List of molecules
Especes	Number of a selected molecule

Returns

Quantity of a selected molecule

Definition at line 252 of file especes.c.

References Especes::quantite.

Referenced by SBML_checkQuantite(), SBML_reaction(), and SBML_simulate().

4.3.2.8 void Especes_print (pEspeces *molecules*, int *nbEspeces*)

Print data on each molecule.

Author

Amine Ghozlane

Parameters

molecules	List of molecules
nbEspeces	Number of molecules

Definition at line 170 of file especes.c.

References Reaction::link, Reaction::suivant, and Especes::system.

4.3.2.9 void Especes_print_2 (pEspeces *molecules*, int *nbEspeces*)

Print Id and quantity on each molecule.

Author

Amine Ghozlane

Parameters

molecules	List of molecules
nbEspeces	Number of molecules

Definition at line 196 of file especes.c.

4.3.2.10 void Especes_save (pEspeces *molecules*, int *i*, double *quant*, const char * *dye*)

Save data on one molecule.

Author

Amine Ghozlane

Parameters

molecules	List of molecules
i	Number of the selected molecule
quant	Quantity of the molecule
dye	ID of the molecule

Definition at line 134 of file especes.c.

References Especes::id, and Especes::quantite.

Referenced by SBML_initEspeceAmounts().

4.3.2.11 void Especes_scoreSpecies (pEspeces *molecules*, int *nbEspeces*, char ** *name*, double * *quantite*)

Save the quantite of all molecules for scoring.

Author

Amine Ghozlane

Parameters

molecules	List of molecules
nbEspeces	Number of molecules
name	ID of the selected molecule
quantite	Table of molecule Quantity

Definition at line 149 of file especes.c.

References Especes::quantite.

Referenced by SBML_score().

4.3.2.12 void Especes_setQuantite (pEspeces *molecules*, int *Especes*, double *quant*)

Change the quantity of a molecule.

Author

Amine Ghozlane

Parameters

molecules	List of molecules
Especes	Number of a selected molecule
quant	Quantity of the molecule

Definition at line 239 of file especes.c.

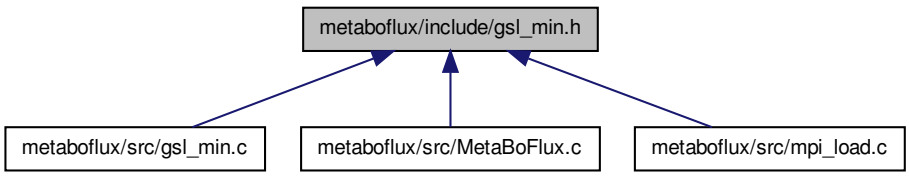
References Especes::quantite.

Referenced by SBML_reaction().

4.4 metaboflux/include/gsl_min.h File Reference

Compute minimization of scenarii.

This graph shows which files directly or indirectly include this file:



Functions

- int Min_copieTab2 (double *, int, int, int)

  Copy table of reaction parameters.
- int Min_copieTab3 (gsl_multimin_fminimizer *, int, int, int)

  Copy short table of parameter into the larger table.
- void Min_verifValue (gsl_vector *, double *, int, int)

  Checking reaction ratio parameters.
- double Min_my_f (const gsl_vector *, void *)

  Enter in the program for standard deviation.

- void Min_getTampon (double *, double)

  Enter in the program for standard deviation.
- void Min_score_print_mean (double *)

  Save the minimization result.
- void Min_compute_minimization (pListParameters, double *, double *, char **, int, int)

  Compute the minimization.

4.4.1 Detailed Description

Compute minimization of scenarii. This file is part of MetaBoFlux (http://www.cbib.u-bordeaux2.fr/metaboflu
Copyright (C) 2010 Amine Ghozlane from LaBRI and University of Bordeaux 1

MetaBoFlux is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

MetaBoFlux is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Author

{Amine Ghozlane}

Version

2.0

Date

9 novembre 2009

Definition in file gsl_min.h.

4.4.2 Function Documentation

4.4.2.1 void Min_compute_minimization (pListParameters *allone*, double * *fluxRatio*, double * *result_tab*, char ** *files_path*, int *number_arg*, int *debug*)

Compute the minimization.

Author

Amine Ghozlane

Parameters

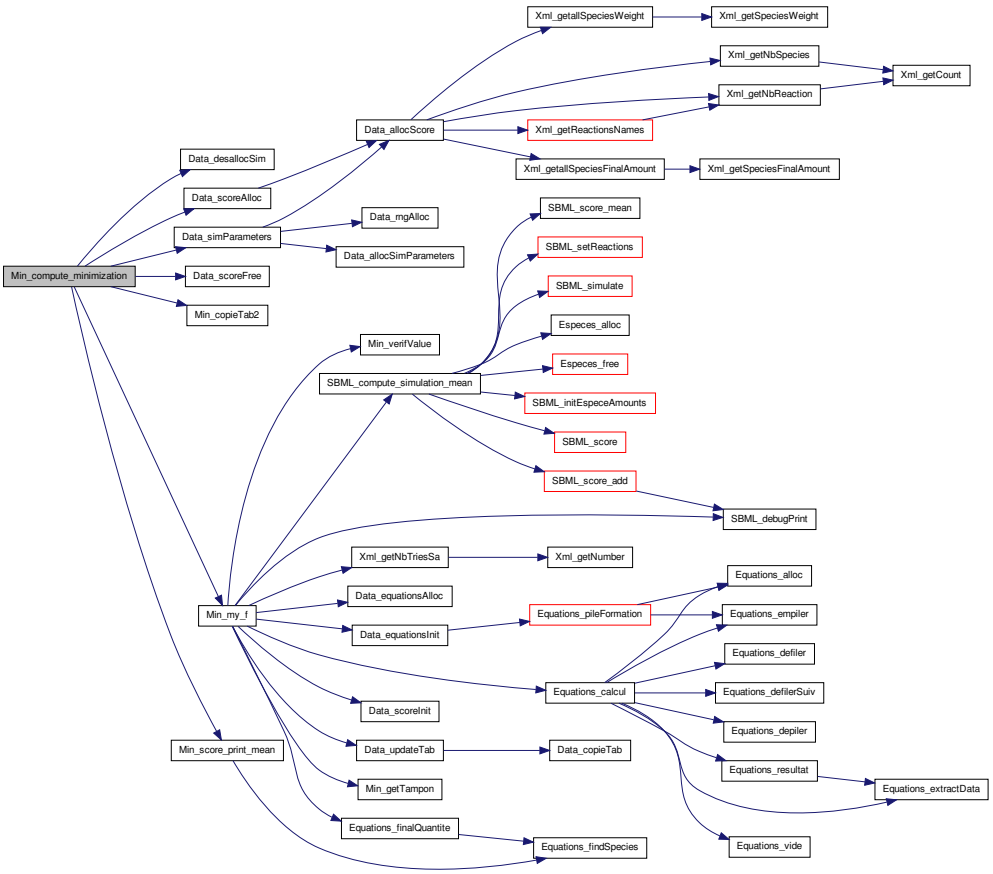
allone	struct ListParameters
fluxRatio	Ratio parameters
result_tab	Result table
files_path	List of paths
number_arg	Number of simulation
debug	Debug flag

Definition at line 274 of file gsl_min.c.

References Data_desallocSim(), Data_scoreAlloc(), Data_scoreFree(), Data_simParameters(), ListParameters::interest_parameters, Min_copieTab2(), Min_my_f(), Min_score_print_-mean(), ListParameters::nb_couples, ListParameters::nb_parameters, ListParameters::parameters, and SimParameters::y.

Referenced by Mpi_slave().

Here is the call graph for this function:



4.4.2.2 int Min_copieTab2 (double * x, int current, int debut, int fin)

Copy table of reaction parameters.

Author

Amine Ghozlane

Parameters

x	Short table of reaction parameters
current	Line
debut	Beginning
fin	End

Returns

Number of copied element

Definition at line 68 of file gsl_min.c.

References SimParameters::y.

Referenced by Min_compute_minimization().

4.4.2.3 int Min_copieTab3 (gsl_multimin_fminimizer * s, int current, int debut, int fin)

Copy short table of parameter into the larger table.

Author

Amine Ghozlane

Parameters

s	Minimizer parameter
current	Line
debut	Beginning
fin	End

Returns

Number of copied element

Definition at line 90 of file gsl_min.c.

References SimParameters::y.

4.4.2.4 void Min_getTampon (double * energie_temp, double energie)

Enter in the program for standard deviation.

Author

Amine Ghozlane

Parameters

energie_temp	Current energy
energie	Best energy

Definition at line 208 of file gsl_min.c.

References ListParameters::nb_parameters, SimParameters::out, Score::quantite, Score::taille, and SimParameters::y.

Referenced by Min_my_f().

4.4.2.5 double Min_my_f (const gsl_vector * *v*, void * *params*)

Enter in the program for standard deviation.

Author

Amine Ghozlane

Parameters

v	Vector of reaction parameters
params	Unused parameter define by GSL

Returns

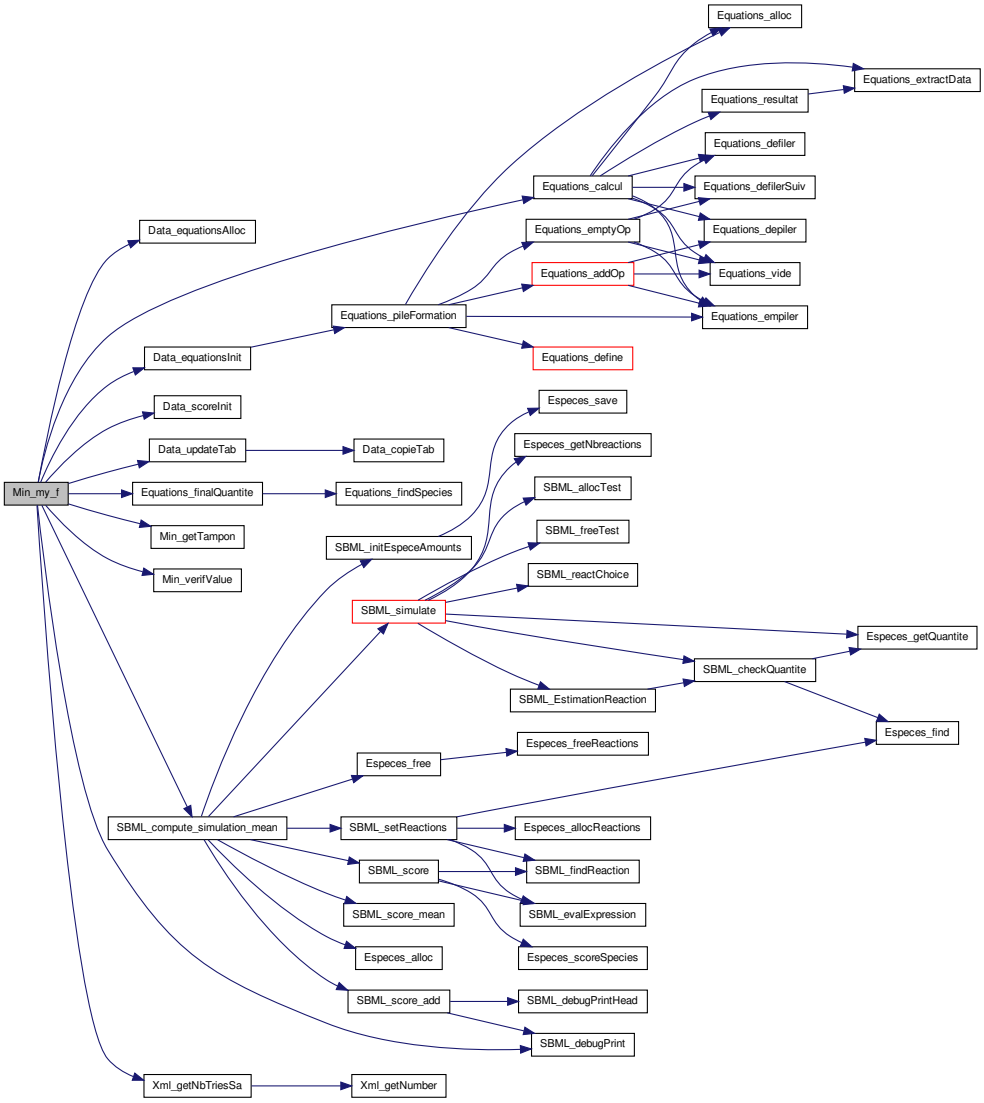
Energy value

Definition at line 149 of file gsl_min.c.

References ListParameters::banned, ListParameters::conf, Data_equationsAlloc(), Data_equationsInit(), Data_scoreInit(), Data_updateTab(), SimParameters::debugFile, Equations_calcul(), Equations_finalQuantite(), Min_getTampon(), Min_verifValue(), ListParameters::model, Score::name, ListParameters::nb_banned, ListParameters::nb_couples, ListParameters::nb_equations, ListParameters::nb_parameters, Score::nb_species, SimParameters::out, ListParameters::parameters, SimParameters::pile, Score::quantite, SimParameters::r, SBML_compute_simulation_mean(), SBML_debugPrint(), Score::species, Score::species_amount, Score::species_weight, Score::taille, Score::tailleSpecies, Xml_getNbTriesSa(), and SimParameters::y.

Referenced by Min_compute_minimization().

Here is the call graph for this function:



4.4.2.6 void Min_score_print_mean (double * *result_tab*)

Save the minimization result.

Author

Amine Ghozlane

Parameters

result_tab	Result table

Definition at line 230 of file gsl_min.c.

References Equations_findSpecies(), Score::name, ListParameters::nb_parameters, Score::nb_species, SimParameters::out, Score::quantite, Score::species, Score::species_amount, Score::taille, Score::tailleSpecies, and SimParameters::y.

Referenced by Min_compute_minimization().

Here is the call graph for this function:

4.4.2.7 void Min_verifValue (gsl_vector * *v*, double * *x*, int *debut*, int *max*)

Checking reaction ratio parameters.

Author

Amine Ghozlane

Parameters

v	Vector of reaction parameters
x	Short table of reaction parameters
debut	Beginning
max	End

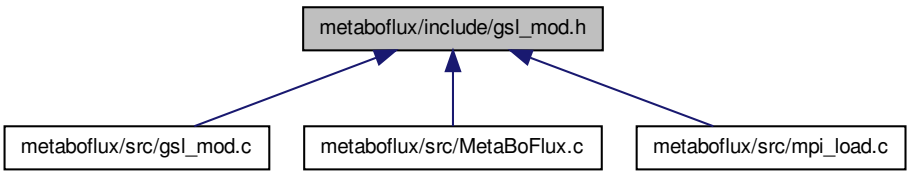
Definition at line 114 of file gsl_min.c.

Referenced by Min_my_f().

4.5 metaboflux/include/gsl_mod.h File Reference

Compute the modeling of scenarii.

This graph shows which files directly or indirectly include this file:



Functions

- void Mod_score_print_mean (pListParameters, pSimParameters, double *, double, int, int)

  Print the mean score.
- void Mod_compute_modeling (pListParameters, double *, double *, char **, int, int, int)

  Compute the modeling.

4.5.1 Detailed Description

Compute the modeling of scenarii. This file is part of MetaBoFlux (http://www.cbib.u-bordeaux2.fr/metaboflu
Copyright (C) 2010 Amine Ghozlane from LaBRI and University of Bordeaux 1

MetaBoFlux is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

MetaBoFlux is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Author

{Amine Ghozlane}

Version

2.0

Date

9 novembre 2009

Definition in file gsl_mod.h.

4.5.2 Function Documentation

4.5.2.1 void Mod_compute_modeling (pListParameters *a*, double * *fluxRatio*, double * *result_tab*, char ** *files_path*, int *number_group*, int *group*, int *debug*)

Compute the modeling.

Author

Amine Ghozlane

Parameters

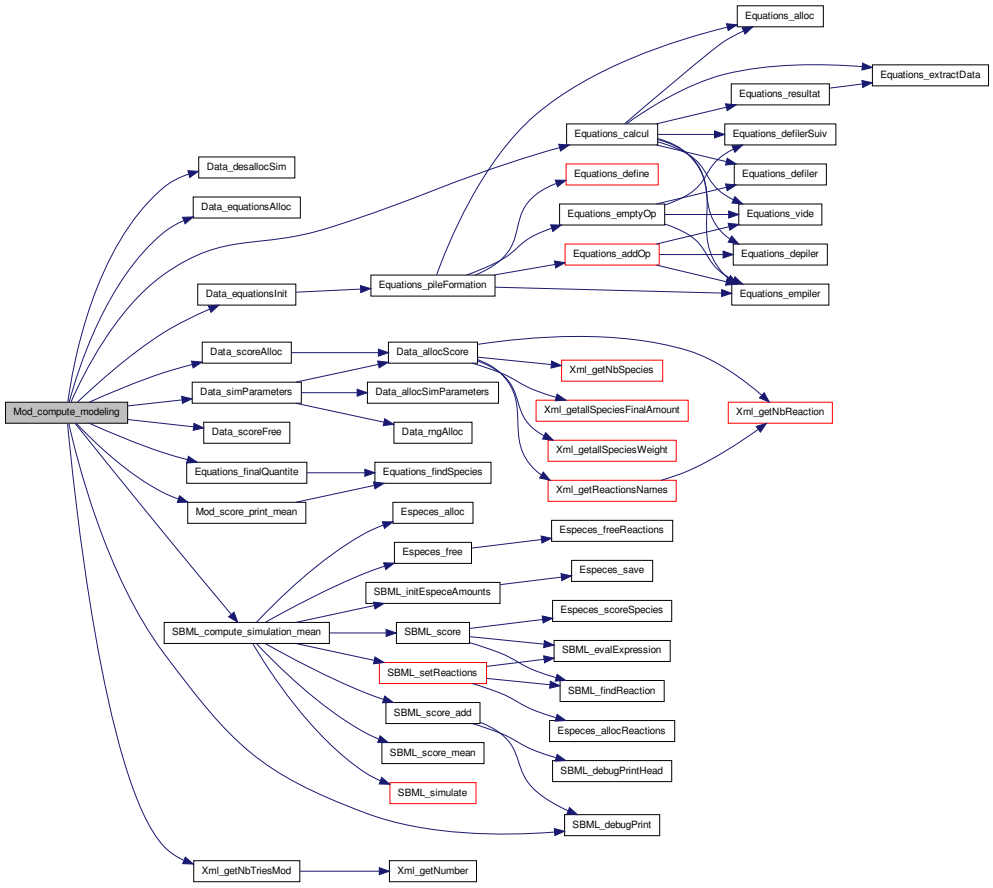
a	struct ListParameters
fluxRatio	Ratio parameters
result_tab	Result table
files_path	List of paths
number_group	Number of the group
group	Group flag
debug	Debug flag

Definition at line 109 of file gsl_mod.c.

References ListParameters::banned, ListParameters::conf, Data_desallocSim(), Data_equationsAlloc(), Data_equationsInit(), Data_scoreAlloc(), Data_scoreFree(), Data_simParameters(), SimParameters::debugFile, Equations_calcul(), Equations_finalQuantite(), Mod_score_print_mean(), ListParameters::model, Score::name, ListParameters::nb_banned, ListParameters::nb_equations, ListParameters::nb_parameters, Score::nb_species, SimParameters::out, SimParameters::pile, Score::quantite, SimParameters::r, SBML_compute_simulation_mean(), SBML_debugPrint(), Score::species, Score::species_amount, Score::species_weight, Score::taille, Score::tailleSpecies, Xml_getNbTriesMod(), and SimParameters::y.

Referenced by Mpi_slave().

Here is the call graph for this function:



4.5.2.2 void Mod_score_print_mean (pListParameters *a*, pSimParameters *simu*, double * *result_tab*, double *energie*, int *number_group*, int *group*)

Print the mean score.

Author

Amine Ghozlane

Parameters

a	Global parameters
simu	Simulation parameters
result_tab	Result table
energie	Energy of the simulation

number_group	Number of simulation
group	Number of the group (if a group is simulated)

Definition at line 58 of file gsl_mod.c.

References Equations_findSpecies(), Score::name, ListParameters::nb_parameters, Score::nb_species, SimParameters::out, Score::quantite, Score::species, Score::species_amount, Score::taille, Score::tailleSpecies, and SimParameters::y.

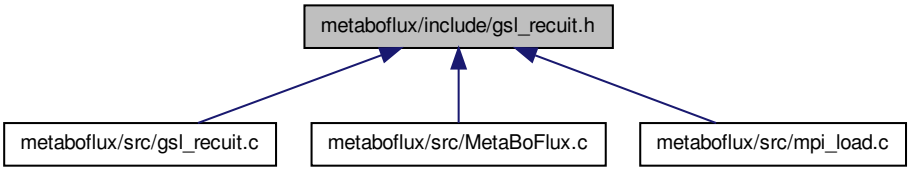
Referenced by Mod_compute_modeling().

Here is the call graph for this function:

4.6 metaboflux/include/gsl_recuit.h File Reference

Compute the simulated annealing.

This graph shows which files directly or indirectly include this file:



Functions

- double Recuit_energyFunction (void *)

  Simulate the petri net and compute the energy.
- double Recuit_metricDistance (void *, void *)

Compute the distance between the two table.

- void Recuit_takeStep (const gsl_rng *, void *, double)

  Take a step through the solution space TSP.
- void Recuit_printPosition (void *)

  Print the table of reaction parameters.
- FILE * Recuit_redirectionFlux (xmlConfig_t *, FILE *, char **, int)

  Redirect stdout flux.
- void Recuit_verifParameters (double *, const gsl_rng *, int, int)

  Choice of parameters and verification of their value.
- void Recuit_verifParameters_2 (double *, double *, const gsl_rng *, int, int, double)

  Choice of parameters and verification of their value.
- void Recuit_defParametre (double *, const gsl_rng *)

  Definition of the initial parameters of the simulation.
- void Recuit_printParametre (double *)

  Print initial parameters.
- void Recuit_compute_recuit (char **, int, int, pListParameters, gsl_siman_params_t)

  Compute the simulated annealing.

4.6.1 Detailed Description

Compute the simulated annealing. This file is part of MetaBoFlux (http://www.cbib.u-bordeaux2.fr/metaboflu

Author

{Amine Ghozlane}

Version

2.0

Date

9 novembre 2009

Definition in file gsl_recuit.h.

4.6.2 Function Documentation

4.6.2.1 void Recuit_compute_recuit (char ** *files_path*, int *debug*, int *number*, pListParameters *allone*, gsl_siman_params_t *params*)

Compute the simulated annealing.

Author

Amine Ghozlane

Parameters

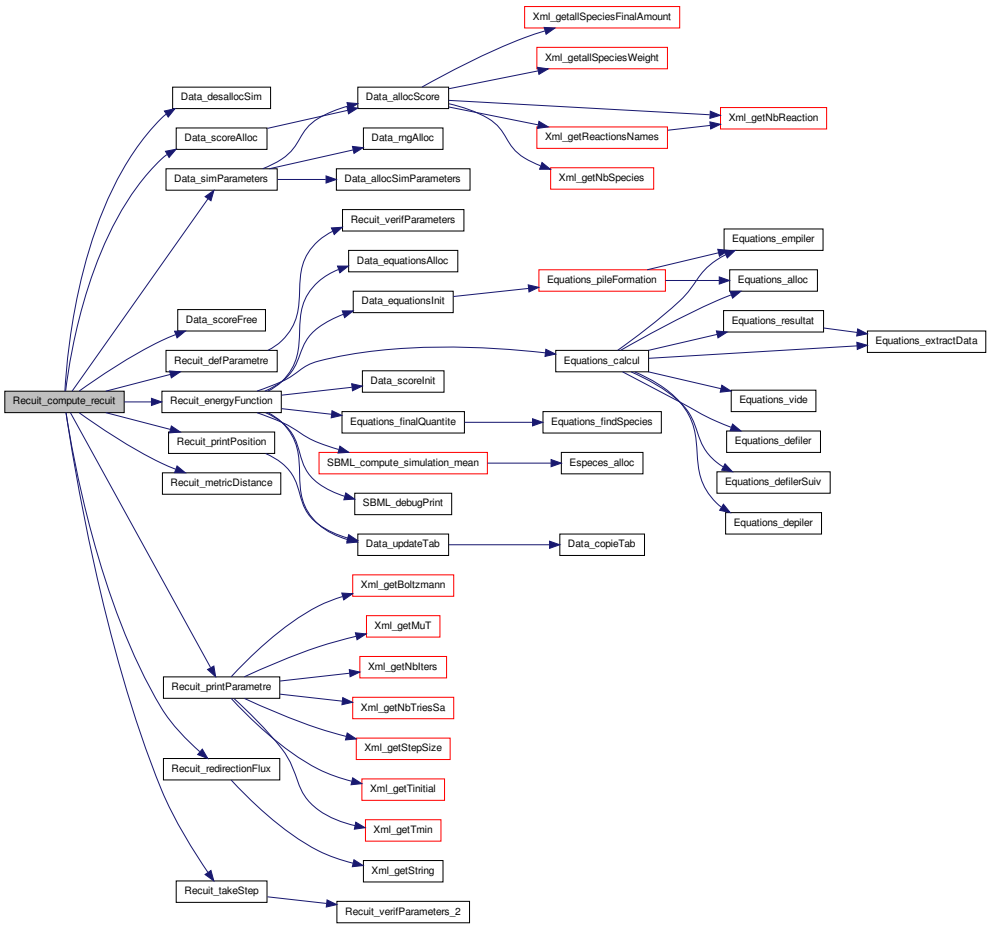
files_path	List of paths
debug	Debug flag
number	Number of the simulation
allone	Global parameters : struct ListParameters
params	Gsl parameters

Definition at line 304 of file gsl_recuit.c.

References ListParameters::conf, Data_desallocSim(), Data_scoreAlloc(), Data_scoreFree(), Data_simParameters(), ListParameters::interest_parameters, SimParameters::r, Recuit_defParametre(), Recuit_energyFunction(), Recuit_metricDistance(), Recuit_printParametre(), Recuit_printPosition(), Recuit_redirectionFlux(), and Recuit_takeStep().

Referenced by Mpi_slave().

Here is the call graph for this function:



4.6.2.2 void Recuit_defParametre (double * *x_initial*, const gsl_rng * *r*)

Definition of the initial parameters of the simulation.

Author

Amine Ghozlane

Parameters

x_initial	table of reaction parameters
r	Random number generator

Definition at line 261 of file gsl_recuit.c.

References ListParameters::nb_couples, ListParameters::parameters, and Recuit_verifParameters().

Referenced by Recuit_compute_recuit().

Here is the call graph for this function:

4.6.2.3 double Recuit_energyFunction (void * *xp*)

Simulate the petri net and compute the energy.

double Recuit_energyFunction(void *xp)

Author

Amine Ghozlane

Parameters

xp	Short table of reaction parameters

Returns

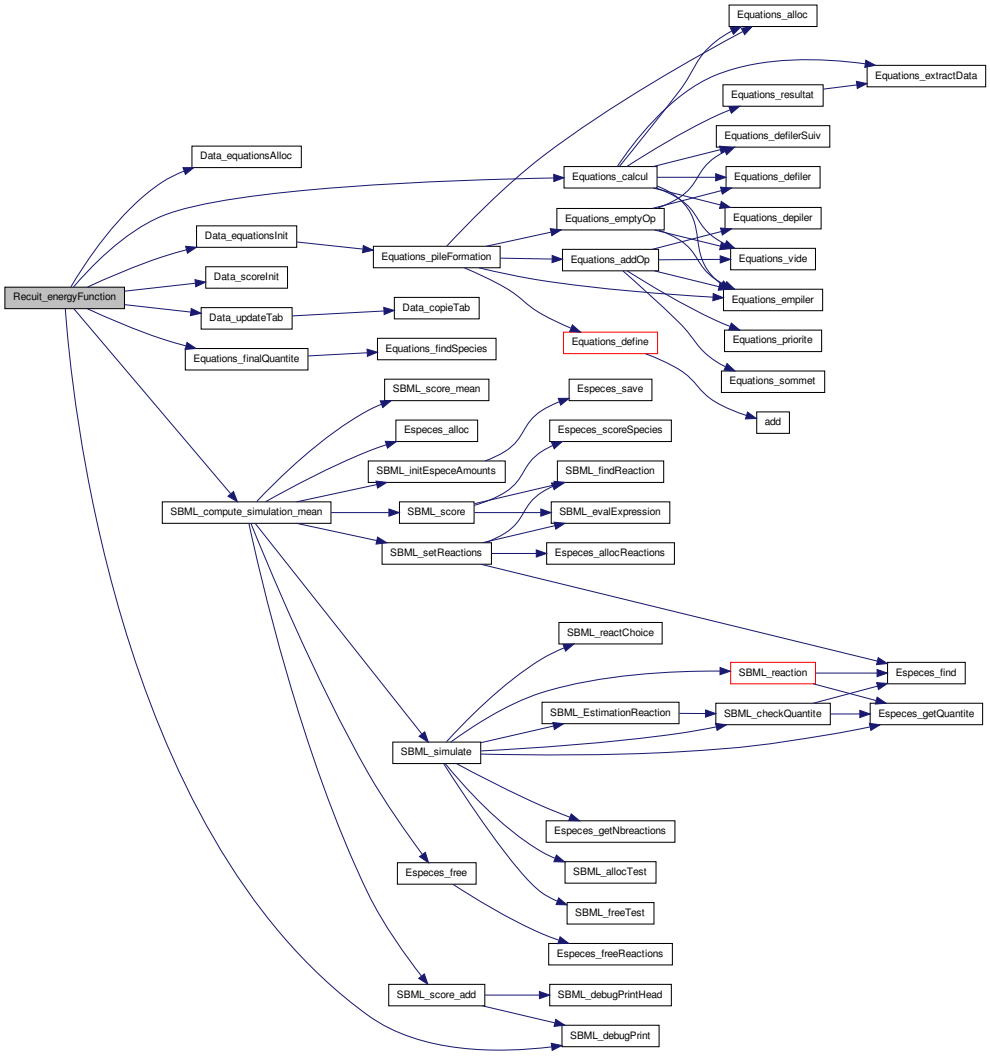
Energy value

Definition at line 65 of file gsl_recuit.c.

References ListParameters::banned, Data_equationsAlloc(), Data_equationsInit(), Data_scoreInit(), Data_updateTab(), SimParameters::debugFile, Equations_calcul(), Equations_finalQuantite(), ListParameters::model, Score::name, ListParameters::nb_banned, ListParameters::nb_equations, Score::nb_species, ListParameters::nb_triesSa, SimParameters::out, SimParameters::pile, Score::quantite, SimParameters::r, SBML_compute_simulation_mean(), SBML_debugPrint(), Score::species, Score::species_amount, Score::species_weight, Score::taille, Score::tailleSpecies, and SimParameters::y.

Referenced by Recuit_compute_recuit().

Here is the call graph for this function:



4.6.2.4 double Recuit_metricDistance (void * *xp*, void * *yp*)

Compute the distance between the two table.

Author

Amine Ghozlane

Parameters

xp	Past short table of reaction parameters
yp	New short table of reaction parameters

Returns

Distance between the two table

Definition at line 115 of file gsl_recuit.c.

References ListParameters::interest_parameters.

Referenced by Recuit_compute_recuit().

4.6.2.5 void Recuit_printParametre (double * *x_initial*)

Print initial parameters.

Author

Amine Ghozlane

Parameters

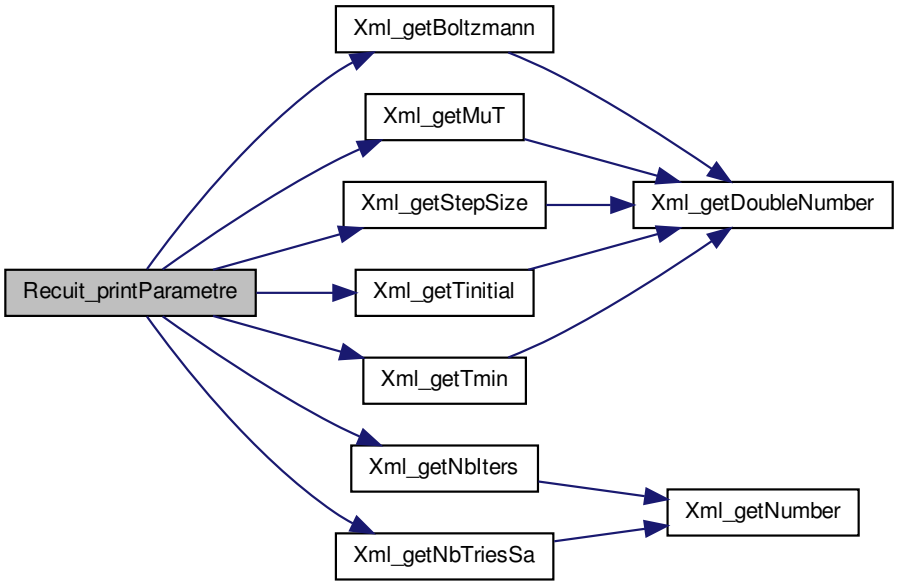
x_initial	table of reaction parameters

Definition at line 279 of file gsl_recuit.c.

References ListParameters::conf, ListParameters::interest_parameters, Xml_getBoltzmann(), Xml_getMuT(), Xml_getNbIters(), Xml_getNbTriesSa(), Xml_getStepSize(), Xml_getTinitial(), and Xml_getTmin().

Referenced by Recuit_compute_recuit().

Here is the call graph for this function:



4.6.2.6 void Recuit_printPosition (void * *xp*)

Print the table of reaction parameters.

Author

Amine Ghozlane

Parameters

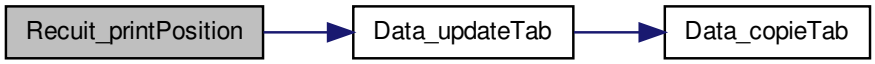
xp	Short table of reaction parameters

Definition at line 158 of file gsl_recuit.c.

References Data_updateTab(), ListParameters::nb_parameters, and SimParameters::y.

Referenced by Recuit_compute_recuit().

Here is the call graph for this function:



4.6.2.7 FILE* Recuit_redirectionFlux (xmlConfig_t * *conf*, FILE * *f*, char ** *files_path*, int *number*)

Redirect stdout flux.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t
f	File name
files_path	List of paths
number	Index of out repertory

Returns

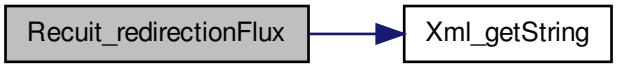
Out flux

Definition at line 184 of file gsl_recuit.c.

References Xml_getString().

Referenced by Recuit_compute_recuit().

Here is the call graph for this function:



4.6.2.8 void Recuit_takeStep (const gsl_rng * *r*, void * *xp*, double *step_size*)

Take a step through the solution space TSP.

Author

Amine Ghozlane

Parameters

r	Random number generator
xp	Short table of reaction parameters
step_size	Size of step

Definition at line 136 of file gsl_recuit.c.

References ListParameters::interest_parameters, ListParameters::nb_couples, ListParameters::parameters, and Recuit_verifParameters_2().

Referenced by Recuit_compute_recuit().

Here is the call graph for this function:

4.6.2.9 void Recuit_verifParameters (double * *x_initial*, const gsl_rng * *r*, int *debut*, int *max*)

Choice of parameters and verification of their value.

Author

Amine Ghozlane

Parameters

x_initial	table of reaction parameters
r	Random number generator
debut	Beginning
max	End

Definition at line 207 of file gsl_recuit.c.

Referenced by Recuit_defParametre().

4.6.2.10 void Recuit_verifParameters_2 (double * *new_x*, double * *old_x*, const gsl_rng * *r*, int *debut*, int *max*, double *step_size*)

Choice of parameters and verification of their value.

Author

Amine Ghozlane

Parameters

new_x	table of reaction parameters
old_x	table of reaction parameters
r	Random number generator
debut	Beginning
max	End
step_size	Size of the step

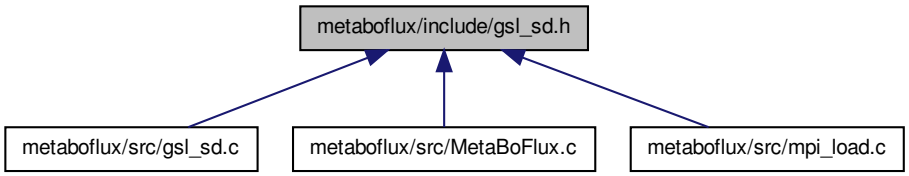
Definition at line 235 of file gsl_recuit.c.

Referenced by Recuit_takeStep().

4.7 metaboflux/include/gsl_sd.h File Reference

Compute the standard deviation analysis of the simulations.

This graph shows which files directly or indirectly include this file:



Functions

- double Sd_compute_simulation (pListParameters, pSimParameters)

  Enter in the program for standard deviation.
- void Sd_compute_standard_deviation (pListParameters, double *, double *, char **, int, int)

  Compute the standard deviation.

4.7.1 Detailed Description

Compute the standard deviation analysis of the simulations. This file is part of MetaBoFlux (http://www.cbib.u-bordeaux2.fr/metaboflux/) Copyright (C) 2010 Amine Ghozlane from LaBRI and University of Bordeaux 1

MetaBoFlux is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

MetaBoFlux is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Author

{Amine Ghozlane}

Version

2.0

Date

9 novembre 2009

Definition in file gsl_sd.h.

4.7.2 Function Documentation

4.7.2.1 double Sd_compute_simulation (pListParameters *all*, pSimParameters *sim*)

Enter in the program for standard deviation.

Author

Amine Ghozlane

Parameters

all	Global parameters : struct ListParameters
sim	Simulation parameters : struct SimParameters

Returns

Energy value

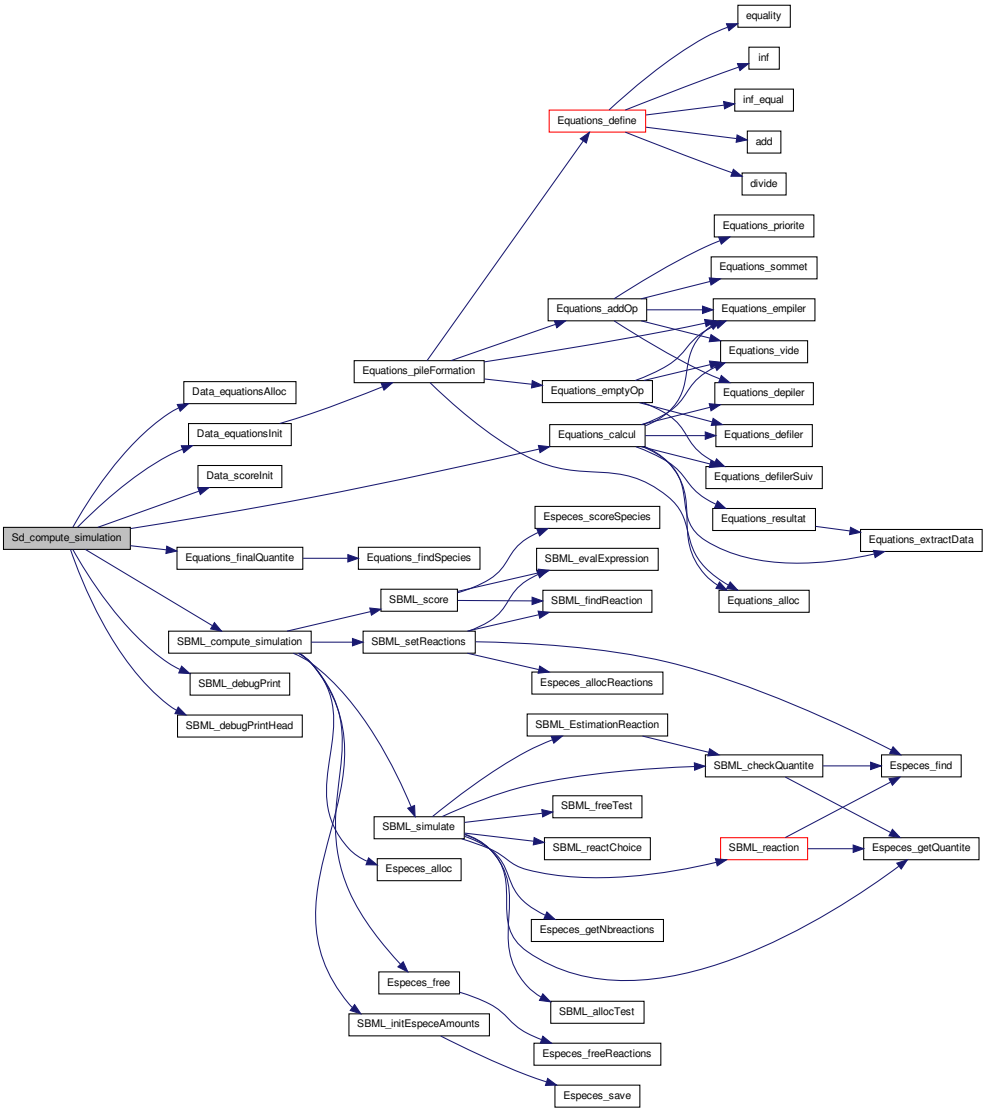
Definition at line 56 of file gsl_sd.c.

References ListParameters::banned, Data_equationsAlloc(), Data_equationsInit(), Data_scoreInit(), SimParameters::debugFile, Equations_calcul(), Equations_finalQuantite(),

ListParameters::model, Score::name, ListParameters::nb_banned, ListParameters::nb_-equations, Score::nb_species, SimParameters::out, SimParameters::pile, Score::quantite, SimParameters::r, SBML_compute_simulation(), SBML_debugPrint(), SBML_debugPrintHead(), Score::species, Score::species_amount, Score::species_weight, Score::taille, Score::tailleSpecies, and SimParameters::y.

Referenced by Sd_compute_standard_deviation().

Here is the call graph for this function:



4.7.2.2 void Sd_compute_standard_deviation (pListParameters *a*, double * *fluxRatio*, double * *result_tab*, char ** *files_path*, int *number_arg*, int *debug*)

Compute the standard deviation.

Author

Amine Ghozlane

Parameters

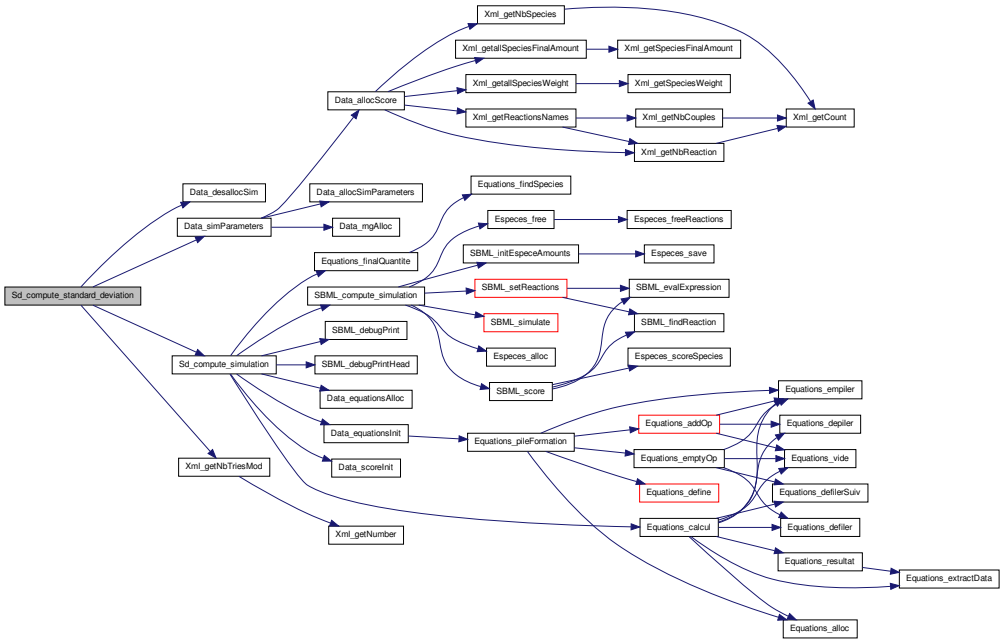
a	struct ListParameters
fluxRatio	Ratio parameters
result_tab	Result table
files_path	List of paths
number_arg	Number of simulation
debug	Debug flag

Definition at line 106 of file gsl_sd.c.

References ListParameters::conf, Data_desallocSim(), Data_simParameters(), ListParameters::nb_parameters, Sd_compute_simulation(), Xml_getNbTriesMod(), and SimParameters::y.

Referenced by Mpi_slave().

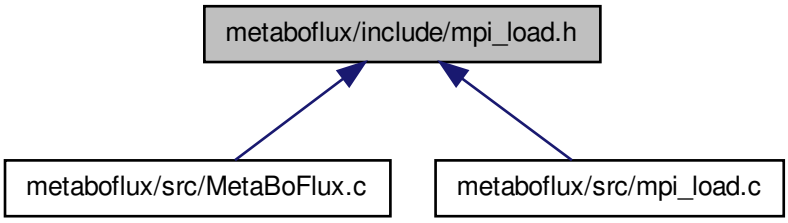
Here is the call graph for this function:



4.8 metaboflux/include/mpi_load.h File Reference

Parallelize the program.

This graph shows which files directly or indirectly include this file:



Functions

- int Mpi_connectInterface (int)

  Enter in the program for standard deviation.
- void Mpi_disconnectInterface (int)

  Disconnection to the interface.
- void Mpi_init (int, char **, int *)

  Enter in the program for standard deviation.
- void Mpi_master (pListParameters, char **, int, int, int, int, int)

  Master process.
- void Mpi_writer (pListParameters, char **, int, int, int)

  Master process.
- int Mpi_sizeResultTab (pListParameters, int, int, int)

  Determine the size of the result tab.
- double * Mpi_allocResultTab (int)

  Allocate the result tab.
- void Mpi_writeSimFile (pListParameters, FILE *, FILE *, double *, int, int, int)

  Determine the size of the result tab.
- void Mpi_writeSdFile (FILE *, double *, int)

  Determine the size of the result tab.
- void Mpi_slave (pListParameters, char **, int, int, int)
- void Mpi_finalize (void)

  End MPI execution environment.
- void compute_mpi (int, char **, char **, int, int, int, int)

  Compute the simulated annealing through mpi.

4.8.1 Detailed Description

Parallelize the program. This file is part of MetaBoFlux (http://www.cbib.u-bordeaux2.fr/metaboflux/)
Copyright (C) 2010 Amine Ghozlane from LaBRI and University of Bordeaux 1

MetaBoFlux is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

MetaBoFlux is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Author

{Amine Ghozlane}

Version

2.0

Date

9 novembre 2009

Definition in file mpi_load.h.

4.8.2 Function Documentation

4.8.2.1 void compute_mpi (int *argc,* char ** *argv,* char ** *files_path,* int *activity,* int *group,* int *debug,* int *port*)

Compute the simulated annealing through mpi.

Author

Amine Ghozlane

Parameters

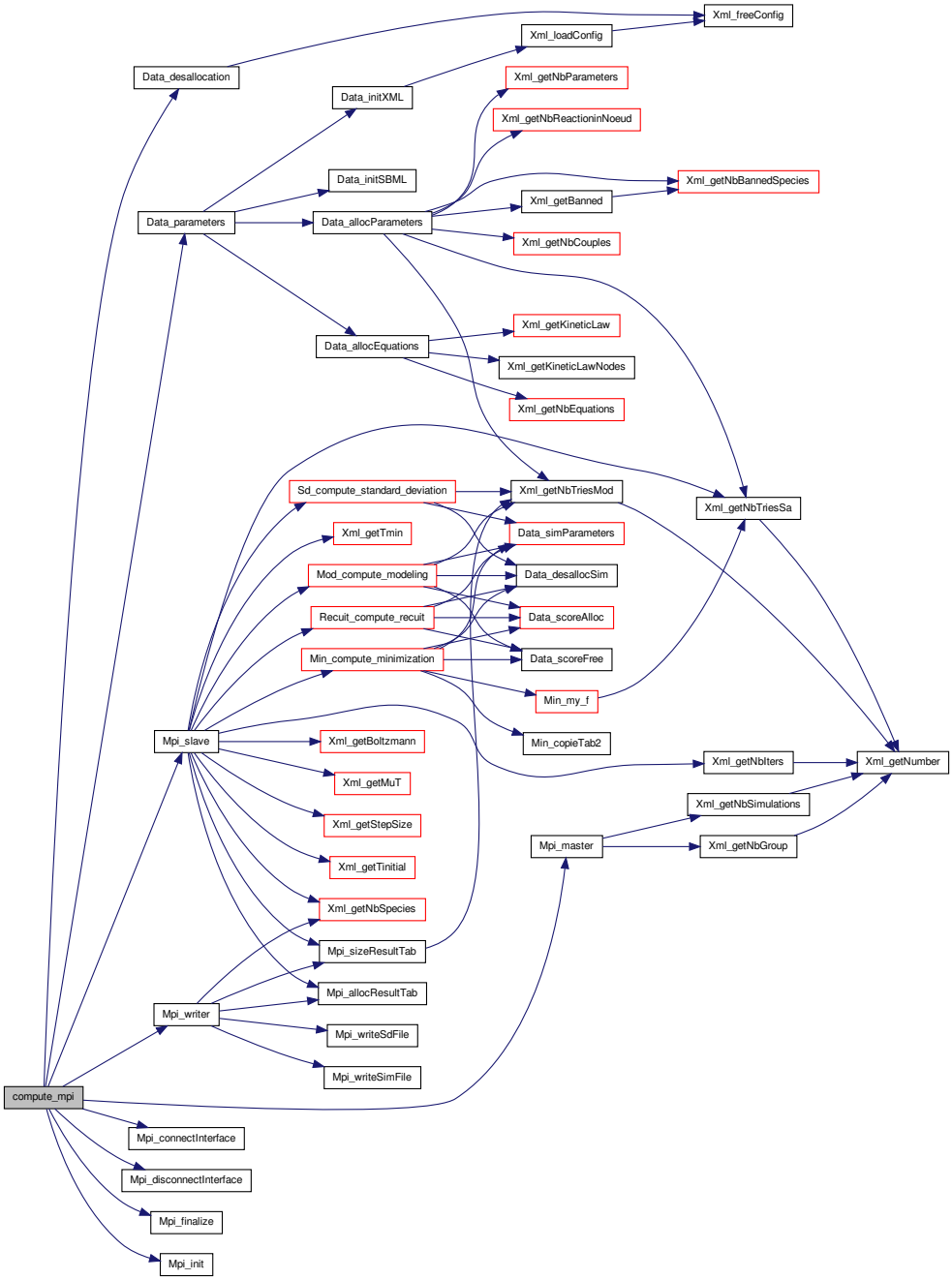
argc	Number of arguments
argv	List of arguments
files_path	List of paths
activity	Chosen activity
group	Group flag
debug	Debug flag
port	Interface port

Definition at line 546 of file mpi_load.c.

References Data_desallocation(), Data_parameters(), Mpi_connectInterface(), Mpi_disconnectInterface(), Mpi_finalize(), Mpi_init(), Mpi_master(), Mpi_slave(), and Mpi_writer().

Referenced by main().

Here is the call graph for this function:



4.8.2.2 double* Mpi_allocResultTab (int *tailleTab*)

Allocate the result tab.

Author

Amine Ghozlane

Parameters

tailleTab	Size of Result tab

Returns

Adress of the allocated space

Definition at line 355 of file mpi_load.c.

Referenced by Mpi_slave(), and Mpi_writer().

4.8.2.3 int Mpi_connectInterface (int *port*)

Enter in the program for standard deviation.

Author

Amine Ghozlane

Parameters

port	Connection port

Returns

Socket

Definition at line 69 of file mpi_load.c.

Referenced by compute_mpi().

4.8.2.4 void Mpi_disconnectInterface (int *desc*)

Disconnection to the interface.

Author

Amine Ghozlane

Parameters

desc	Socket

Definition at line 109 of file mpi_load.c.

Referenced by compute_mpi().

4.8.2.5 void Mpi_finalize (void)

End MPI execution environment.

Author

Amine Ghozlane

Definition at line 527 of file mpi_load.c.

Referenced by compute_mpi().

4.8.2.6 void Mpi_init (int *argc*, char ** *argv*, int * *tab*)

Enter in the program for standard deviation.

Author

Amine Ghozlane

Parameters

argc	Number of arguments
argv	List of arguments
tab	Table

Definition at line 123 of file mpi_load.c.

Referenced by compute_mpi().

4.8.2.7 void Mpi_master (pListParameters *allone*, char ** *files_path*, int *activity*, int *group*, int *myid*, int *numprocs*, int *desc*)

Master process.

Author

Amine Ghozlane

Parameters

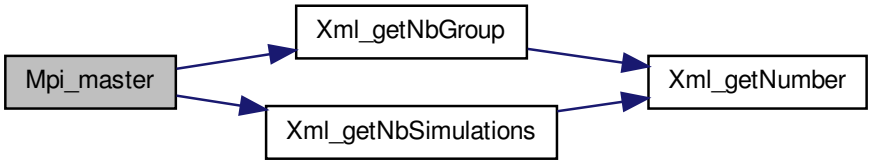
allone	Global parameters : struct ListParameters
files_path	List of paths
activity	Chosen activity
group	Group flag
myid	Id of the thread
numprocs	Number of thread
desc	Socket

Definition at line 154 of file mpi_load.c.

References ListParameters::conf, ListParameters::nb_parameters, Xml_getNbGroup(), and Xml_getNbSimulations().

Referenced by compute_mpi().

Here is the call graph for this function:



4.8.2.8 int Mpi_sizeResultTab (pListParameters *allone*, int *activity*, int *group*, int *nb_species*)

Determine the size of the result tab.

Author

Amine Ghozlane

Parameters

allone	Global parameters : struct ListParameters
activity	Chosen activity
group	Group flag
nb_species	Number of interest species

Returns

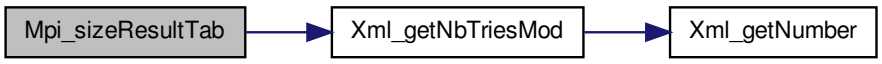
Size of Result tab

Definition at line 334 of file mpi_load.c.

References ListParameters::conf, ListParameters::model, ListParameters::nb_parameters, and Xml_getNbTriesMod().

Referenced by Mpi_slave(), and Mpi_writer().

Here is the call graph for this function:



4.8.2.9 void Mpi_slave (pListParameters *allone*, char ** *files_path*, int *activity*, int *group*, int *debug*)

Author

Amine Ghozlane

Parameters

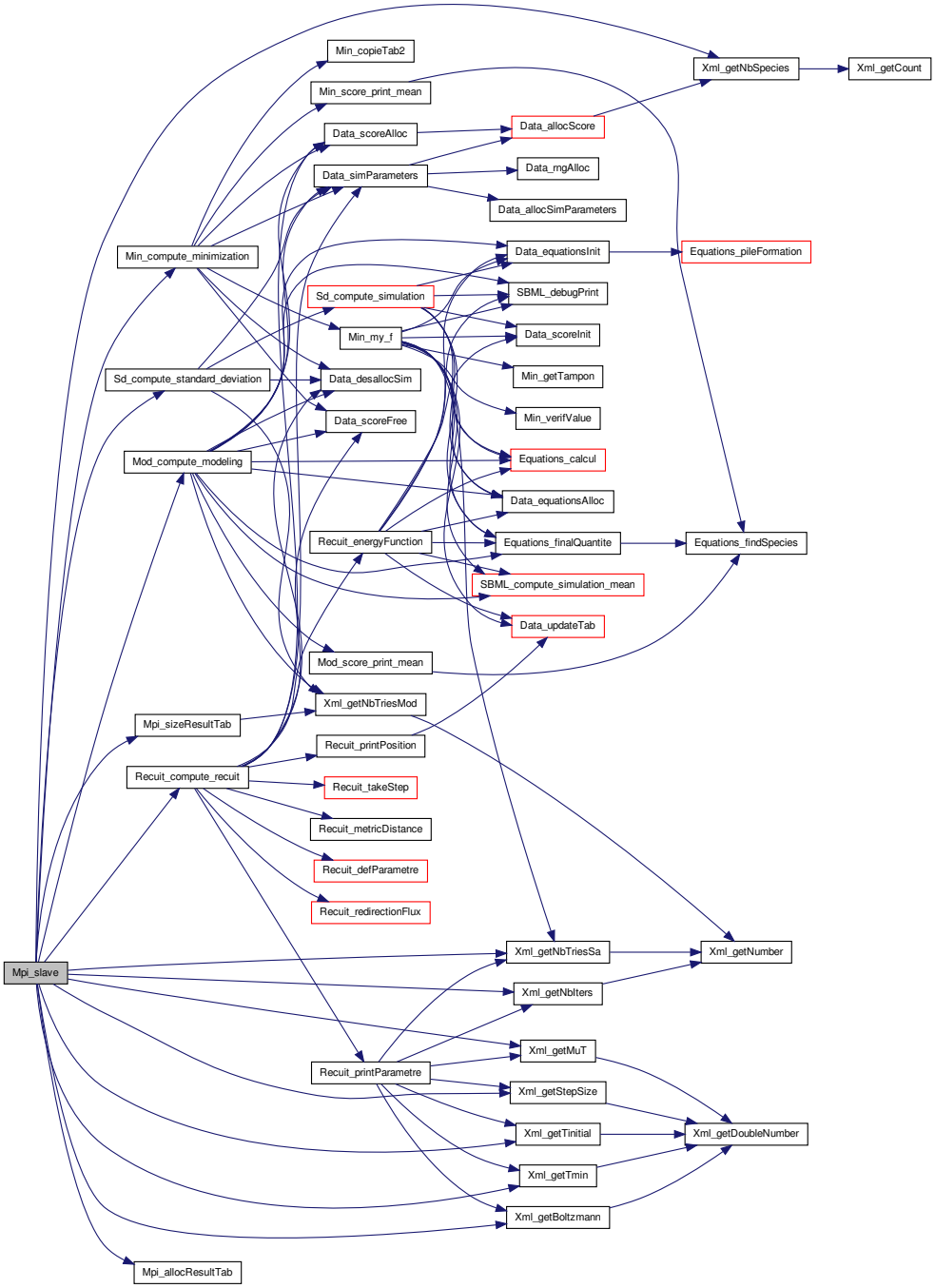
allone	Global parameters : struct ListParameters
files_path	List of paths
activity	Activity chosen
group	Group flag
debug	Debug flag

Definition at line 440 of file mpi_load.c.

References ListParameters::conf, Min_compute_minimization(), Mod_compute_modeling(), Mpi_allocResultTab(), Mpi_sizeResultTab(), ListParameters::nb_parameters, Recuit_compute_recuit(), Sd_compute_standard_deviation(), Xml_getBoltzmann(), Xml_getMuT(), Xml_getNbIters(), Xml_getNbSpecies(), Xml_getNbTriesSa(), Xml_getStepSize(), Xml_getTinitial(), and Xml_getTmin().

Referenced by compute_mpi().

Here is the call graph for this function:



4.8.2.10 void Mpi_writer (pListParameters *allone*, char ** *files_path*, int *activity*, int *group*, int *myid*)

Master process.

Author

Amine Ghozlane

Parameters

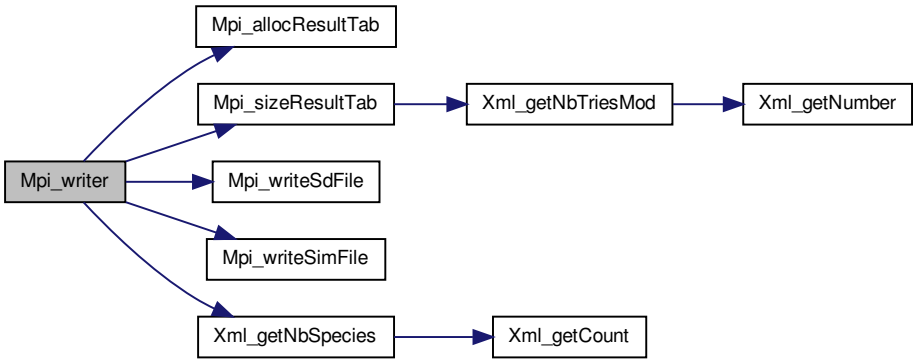
allone	Global parameters : struct ListParameters
files_path	List of paths
activity	Chosen activity
group	Group flag
myid	Id of the thread

Definition at line 273 of file mpi_load.c.

References ListParameters::conf, ListParameters::model, Mpi_allocResultTab(), Mpi_sizeResultTab(), Mpi_writeSdFile(), Mpi_writeSimFile(), and Xml_getNbSpecies().

Referenced by compute_mpi().

Here is the call graph for this function:



4.8.2.11 void Mpi_writeSdFile (FILE * *out*, double * *result_tab*, int *tailleTab*)

Determine the size of the result tab.

Author

Amine Ghozlane

Parameters

out	Result file
result_tab	Result table
tailleTab	Size of result table

Definition at line 421 of file mpi_load.c.

Referenced by Mpi_writer().

4.8.2.12 void Mpi_writeSimFile (pListParameters *allone*, FILE * *out*, FILE * *logOut*, double * *result_tab*, int *group*, int *tailleSpecies*, int *nb_species*)

Determine the size of the result tab.

Author

Amine Ghozlane

Parameters

allone	Global parameters : struct ListParameters
out	Result file
logOut	Log file
result_tab	Result table
group	Group flag
tailleSpecies	Number of species
nb_species	Number of interest species

Definition at line 375 of file mpi_load.c.

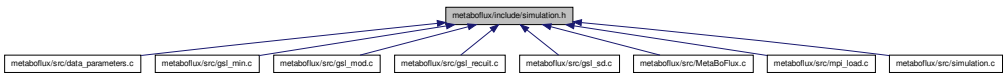
References ListParameters::nb_parameters.

Referenced by Mpi_writer().

4.9 metaboflux/include/simulation.h File Reference

Simulate a petri net.

This graph shows which files directly or indirectly include this file:



Data Structures

- struct TestReaction

Structure used to test reactions.

- struct Score

  Structure containing all information from simulations.

Functions

- void SBML_initEspeceAmounts (Model_t *, pEspeces, int)

  Alloc memory and initialize the struct Especes.
- int SBML_findReaction (char **, const char *, int)

  Determine if the reaction is study.
- double SBML_evalExpression (const char *)

  Get the reaction ratio define in the sbml.
- void SBML_setReactions (Model_t *, pEspeces, pScore, double *, int, int)

  Alloc memory and initialize the struct Especes.
- int SBML_checkQuantite (Model_t *, Reaction_t *, int, pEspeces)

  Determine the number of reaction for one molecule.
- Reaction_t * SBML_reactChoice (pEspeces, const gsl_rng *, int)

  Determine randomly the reaction to achieve for several nodes reactions.
- void SBML_reaction (Model_t *, pEspeces, Reaction_t *, int)

  Simulation of a discrete transision.
- void SBML_allocTest (pTestReaction, int)

  Alloc memory and initialize the struct pTestReaction.
- void SBML_freeTest (pTestReaction)

  Free memory of the struct TestReaction.
- int SBML_EstimationReaction (Model_t *, pTestReaction, pEspeces, int, int)

  Alloc memory and initialize the struct Especes.
- int SBML_simulate (Model_t *, pEspeces, const gsl_rng *, pTestReaction, char **, int, int, int)

  Simulate one step of petri net.
- void SBML_score (Model_t *, pEspeces, pScore, double *, int, int)

  Alloc memory and initialize the struct Especes.
- void SBML_debugPrintHead (FILE *, int, char **)

  Print the head of the debug file.
- void SBML_debugPrint (FILE *, int, int, double *, double)

  Print the debuf file.
- void SBML_compute_simulation (pScore, Model_t *, double *, gsl_rng *, char **, int)

  Simulation of metabolic network.
- void SBML_score_add (pScore, pScore, FILE *)

  Add scores.
- void SBML_score_mean (pScore, int)

  Mean quantities for score.
- void SBML_compute_simulation_mean (FILE *, pScore, pScore, Model_t *, double *, gsl_rng *, char **, int, int)

  X time simulation of metabolic network.

4.9.1 Detailed Description

Simulate a petri net. This file is part of MetaBoFlux (http://www.cbib.u-bordeaux2.fr/metaboflux
Copyright (C) 2010 Amine Ghozlane from LaBRI and University of Bordeaux 1

MetaBoFlux is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

MetaBoFlux is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Author

{Amine Ghozlane}

Version

2.0

Date

27 octobre 2009

Definition in file simulation.h.

4.9.2 Function Documentation

4.9.2.1 void SBML_allocTest (pTestReaction *T*, int *nbReactions*)

Alloc memory and initialize the struct pTestReaction.

Author

Amine Ghozlane

Parameters

T	Empty struct TestReaction
nbReactions	Number of reactions

Definition at line 279 of file simulation.c.

References TestReaction::minStepTab, and TestReaction::tabReactions.

Referenced by SBML_simulate().

4.9.2.2 int SBML_checkQuantite (Model_t * *mod*, Reaction_t * *react*, int *nbEspeces*, pEspeces *molecules*)

Determine the number of reaction for one molecule.

Author

Amine Ghozlane

Parameters

mod	Model of the SBML file
react	Reaction id
nbEspeces	Number of molecules
molecules	Struct Especes

Returns

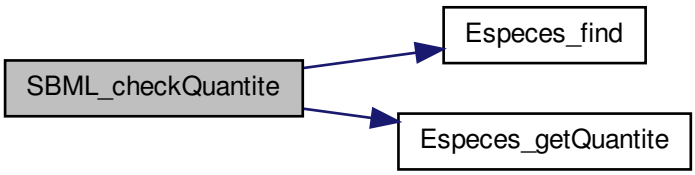
Number of reaction for one molecule

Definition at line 158 of file simulation.c.

References Especes_find(), and Especes_getQuantite().

Referenced by SBML_EstimationReaction(), and SBML_simulate().

Here is the call graph for this function:



4.9.2.3 void SBML_compute_simulation (pScore *result*, Model_t * *mod*, double * *reactions_ratio*, gsl_rng * *r*, char ** *banned*, int *nbBanned*)

Simulation of metabolic network.

Author

Amine Ghozlane

Parameters

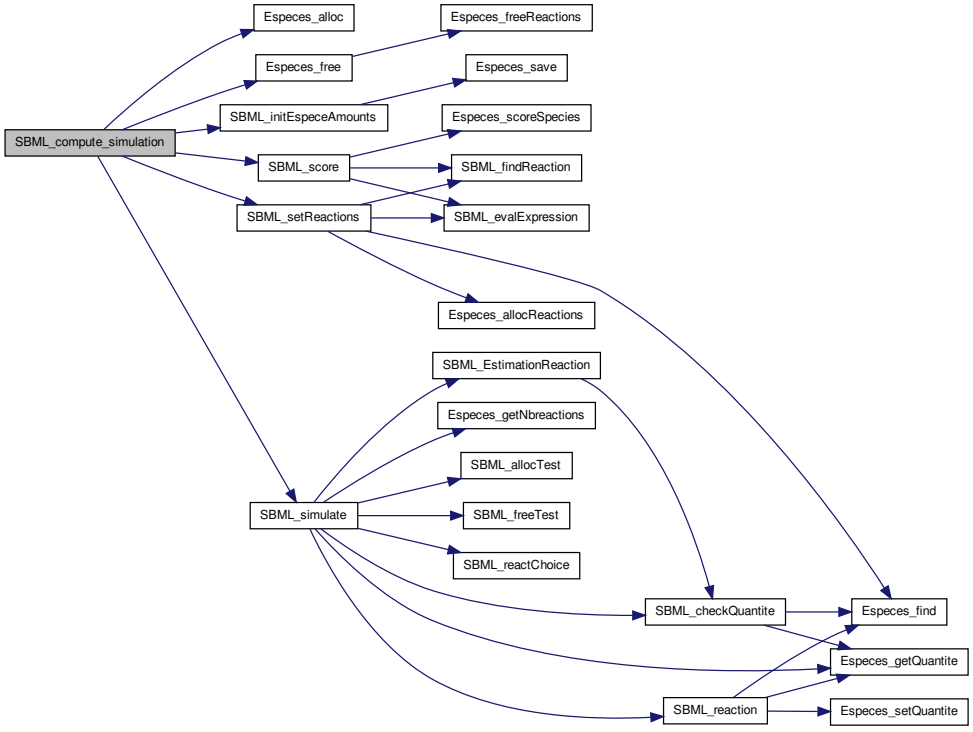
result	Struct Score
mod	Model of the SBML file
reactions_ratio	List of computed reaction ratio
r	Random number generator
banned	List of banned compound
nbBanned	Number of banned compound

Definition at line 515 of file simulation.c.

References Especes_alloc(), Especes_free(), SBML_initEspeceAmounts(), SBML_score(), SBML_setReactions(), SBML_simulate(), Score::tailleReactions, and Score::tailleSpecies.

Referenced by Sd_compute_simulation().

Here is the call graph for this function:



4.9.2.4 void SBML_compute_simulation_mean (FILE * *debugFile,* pScore *result,* pScore *result_temp,* Model_t * *mod,* double * *reactions_ratio,* gsl_rng * *r,* char ** *banned,* int *nbBanned,* int *nb_simulation*)

X time simulation of metabolic network.

Author

Amine Ghozlane

Parameters

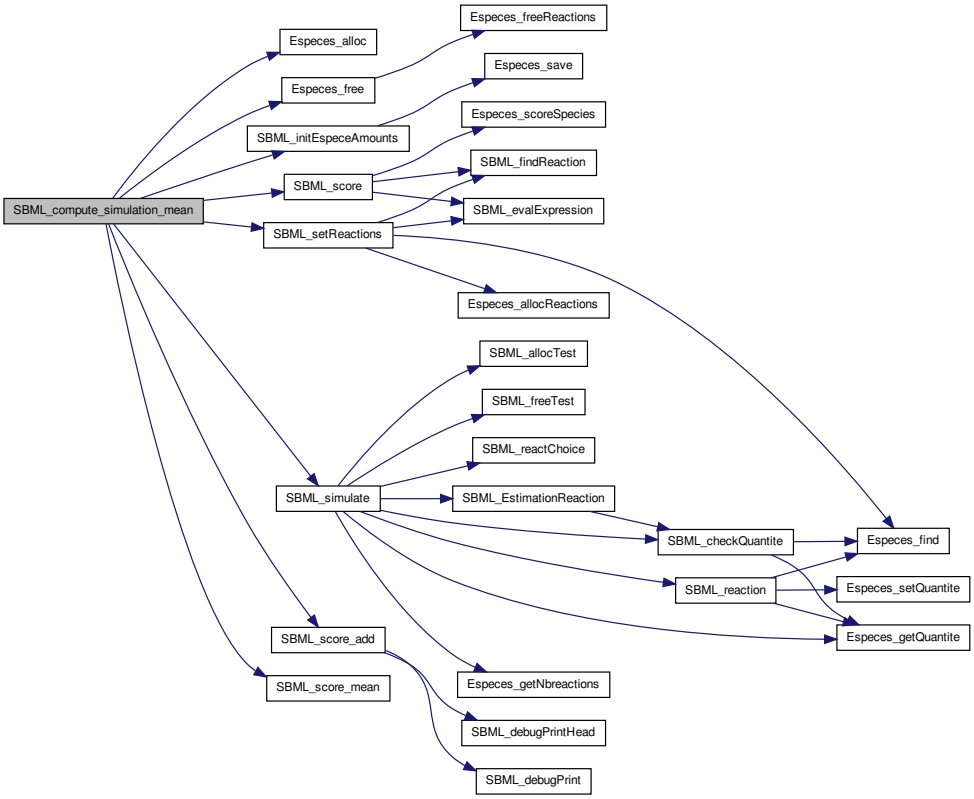
debugFile	File use for debug
result	Struct Score used for all the simulation
result_temp	Struct Score used at each simulation step
mod	Model of the SBML file
reactions_ratio	List of computed reaction ratio
r	Random number generator
banned	List of banned compound
nbBanned	Number of banned compound
nb_simulation	Number of simulation step

Definition at line 614 of file simulation.c.

References Especes_alloc(), Especes_free(), SBML_initEspeceAmounts(), SBML_score(), SBML_score_add(), SBML_score_mean(), SBML_setReactions(), and SBML_simulate().

Referenced by Min_my_f(), Mod_compute_modeling(), and Recuit_energyFunction().

Here is the call graph for this function:



4.9.2.5 void SBML_debugPrint (FILE * *debugFile*, int *tailleSpecies*, int *taille*, double * *quantite*, double *result*)

Print the debuf file.

Author

Amine Ghozlane

Parameters

debugFile	File use for debug
tailleSpecies	Number of molecules
taille	Number of molecules/reactions
quantite	Quantity of molecules/reactions
result	Energy value

Definition at line 486 of file simulation.c.

Referenced by Min_my_f(), Mod_compute_modeling(), Recuit_energyFunction(), SBML_score_add(), and Sd_compute_simulation().

4.9.2.6 void SBML_debugPrintHead (FILE * *debugFile,* int *taille,* char ** *name*)

Print the head of the debug file.

Author

Amine Ghozlane

Parameters

debugFile	File use for debug
taille	Number of molecules/reactions
name	List of molecules/reactions

Definition at line 465 of file simulation.c.

Referenced by SBML_score_add(), and Sd_compute_simulation().

4.9.2.7 int SBML_EstimationReaction (Model_t * *mod,* pTestReaction *T,* pEspeces *molecules,* int *ref,* int *nbEspeces*)

Alloc memory and initialize the struct Especes.

Author

Amine Ghozlane

Parameters

mod	Model of the SBML file
T	Struct TestReaction gives data on reaction
molecules	Struct Especes
ref	Number reference of one molecule
nbEspeces	Number of molecules

Returns

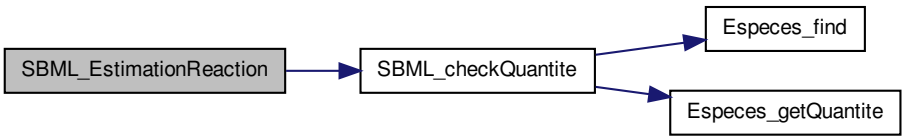
Estimated number of feasible step by reaction

Definition at line 322 of file simulation.c.

References Reaction::link, TestReaction::minStepTab, SBML_checkQuantite(), Reaction::suivant, Especes::system, and TestReaction::tabReactions.

Referenced by SBML_simulate().

Here is the call graph for this function:



4.9.2.8 double SBML_evalExpression (const char * *formule*)

Get the reaction ratio define in the sbml.

Author

Amine Ghozlane

Parameters

formule	Formule SBML

Returns

Return double value of the constraint

Definition at line 88 of file simulation.c.

Referenced by SBML_score(), and SBML_setReactions().

4.9.2.9 int SBML_findReaction (char ** *reaction*, const char * *react*, int *nb_reaction*)

Determine if the reaction is study.

Author

Amine Ghozlane

Parameters

reaction	List of reactions
react	Reaction of interest
nb_reaction	Number of reactions

Returns

Number of the molecules if it's study

Definition at line 69 of file simulation.c.

Referenced by SBML_score(), and SBML_setReactions().

4.9.2.10 void SBML_freeTest (pTestReaction *T*)

Free memory of the struct TestReaction.

Author

Amine Ghozlane

Parameters

T	Struct TestReaction gives data on reaction

Definition at line 304 of file simulation.c.

References TestReaction::minStepTab, and TestReaction::tabReactions.

Referenced by SBML_simulate().

4.9.2.11 void SBML_initEspeceAmounts (Model_t * *mod*, pEspeces *molecules*, int *nbEspeces*)

Alloc memory and initialize the struct Especes.

Author

Amine Ghozlane

Parameters

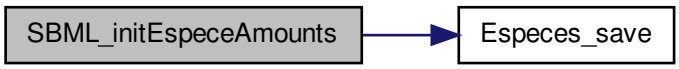
mod	Model of the SBML file
molecules	Struct Especes
nbEspeces	Number of molecules

Definition at line 46 of file simulation.c.

References Especes_save().

Referenced by SBML_compute_simulation(), and SBML_compute_simulation_mean().

Here is the call graph for this function:



4.9.2.12 Reaction_t* SBML_reactChoice (pEspeces *molecules*, const gsl_rng * *r*, int *ref*)

Determine randomly the reaction to achieve for several nodes reactions.

Author

Amine Ghozlane

Parameters

molecules	Struct Especes
r	Random number generator
ref	Number reference of one molecule

Returns

Id of the selected reaction

Definition at line 206 of file simulation.c.

References Reaction::link, Reaction::suivant, and Especes::system.

Referenced by SBML_simulate().

4.9.2.13 void SBML_reaction (Model_t * *mod*, pEspeces *molecules*, Reaction_t * *react*, int *nbEspeces*)

Simulation of a discrete transision.

Author

Amine Ghozlane

Parameters

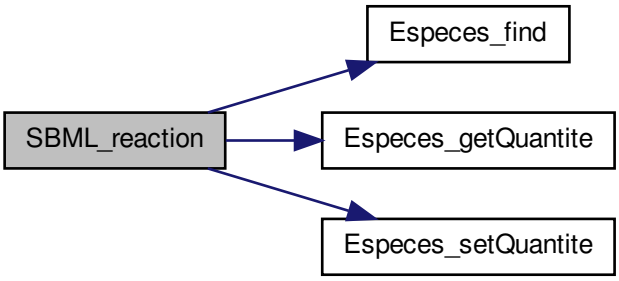
mod	Model of the SBML file
molecules	Struct Especes
react	Reaction id
nbEspeces	Number of molecules

Definition at line 244 of file simulation.c.

References Especes_find(), Especes_getQuantite(), and Especes_setQuantite().

Referenced by SBML_simulate().

Here is the call graph for this function:



4.9.2.14 void SBML_score (Model_t * *mod*, pEspeces *molecules*, pScore *result*, double * *reactions_ratio*, int *nbReactions*, int *nbEspeces*)

Alloc memory and initialize the struct Especes.

Author

Amine Ghozlane

Parameters

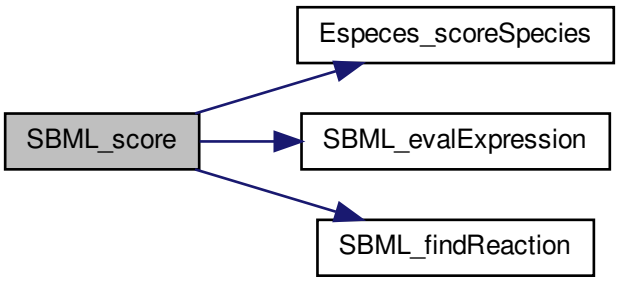
mod	Model of the SBML file
molecules	Struct Especes
result	Struct Score
reactions_ratio	List of computed reaction ratio
nbReactions	Number of reactions
nbEspeces	Number of molecules

Definition at line 420 of file simulation.c.

References Especes_scoreSpecies(), Score::name, Score::nb_reaction, Score::quantite, Score::reaction, SBML_evalExpression(), and SBML_findReaction().

Referenced by SBML_compute_simulation(), and SBML_compute_simulation_mean().

Here is the call graph for this function:



4.9.2.15 void SBML_score_add (pScore *result*, pScore *result_temp*, FILE * *debugFile*)

Add scores.

Author

Amine Ghozlane

Parameters

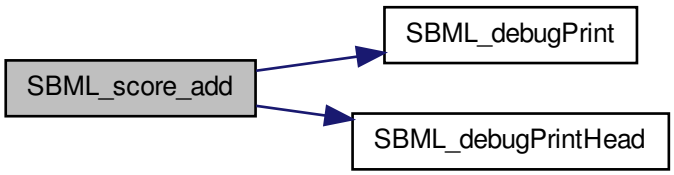
result	Struct Score used for all the simulation
result_temp	Struct Score used at each simulation step
debugFile	File use for debug

Definition at line 560 of file simulation.c.

References Score::name, Score::quantite, SBML_debugPrint(), SBML_debugPrintHead(), Score::taille, and Score::tailleSpecies.

Referenced by SBML_compute_simulation_mean().

Here is the call graph for this function:



4.9.2.16 void SBML_score_mean (pScore *result*, int *n*)

Mean quantities for score.

Author

Amine Ghozlane

Parameters

result	Struct Score
n	Number of simulation step

Definition at line 591 of file simulation.c.

References Score::quantite, and Score::taille.

Referenced by SBML_compute_simulation_mean().

4.9.2.17 void SBML_setReactions (Model_t * *mod*, pEspeces *molecules*, pScore *result*, double * *reactions_ratio*, int *nbReactions*, int *nbEspeces*)

Alloc memory and initialize the struct Especes.

Author

Amine Ghozlane

Parameters

mod	Model of the SBML file
molecules	Struct Especes
result	Struct Score
reactions_ratio	List of computed reaction ratio

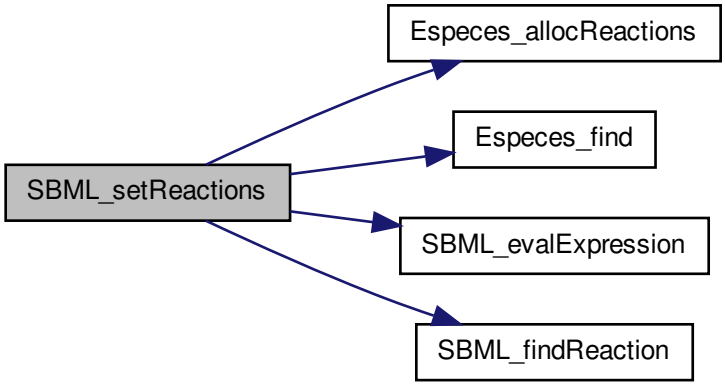
nbReactions	Number of reaction
nbEspeces	Number of molecules

Definition at line 105 of file simulation.c.

References Especes_allocReactions(), Especes_find(), Score::nb_reaction, Score::reaction, SBML_evalExpression(), and SBML_findReaction().

Referenced by SBML_compute_simulation(), and SBML_compute_simulation_mean().

Here is the call graph for this function:



4.9.2.18 int SBML_simulate (Model_t * *mod*, pEspeces *molecules*, const gsl_rng * *r*, pTestReaction *T*, char ** *banned*, int *nbBanned*, int *nbEspeces*, int *ref*)

Simulate one step of petri net.

Author

Amine Ghozlane

Parameters

mod	Model of the SBML file
molecules	Struct Especes
r	Random number generator
T	Struct TestReaction gives data on reaction
banned	List of banned compound
nbBanned	Number of banned compound

nbEspeces	Number of molecules
ref	Number reference of one molecule

Returns

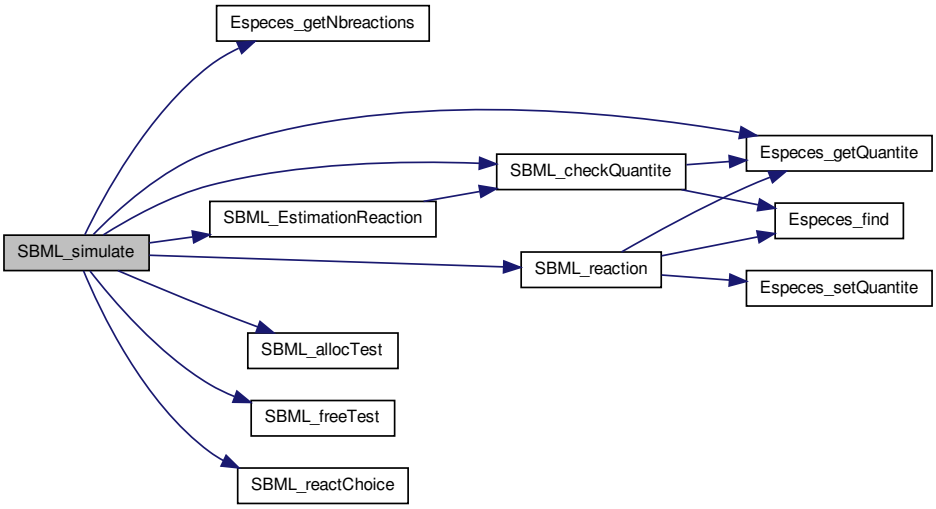
Condition of stop/pursue

Definition at line 356 of file simulation.c.

References Especes_getNbreactions(), Especes_getQuantite(), Reaction::link, SBML_allocTest(), SBML_checkQuantite(), SBML_EstimationReaction(), SBML_freeTest(), SBML_reactChoice(), SBML_reaction(), and Especes::system.

Referenced by SBML_compute_simulation(), and SBML_compute_simulation_mean().

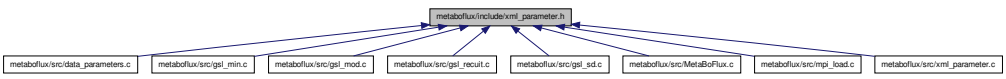
Here is the call graph for this function:



4.10 metaboflux/include/xml_parameter.h File Reference

Xml reader for parametre.xml.

This graph shows which files directly or indirectly include this file:



Data Structures

- struct xmlConfig_t

  Structure containing all information from parameter.xml.

Functions

- xmlConfig_t * Xml_loadConfig (char *)

  Initialize and load the parameter.xml.
- void Xml_freeConfig (xmlConfig_t *)

  Free the Struct xmlConfig_t.
- int Xml_getNumber (xmlConfig_t *, const char *)

  Get integer value.
- double Xml_getDoubleNumber (xmlConfig_t *, const char *)

  Get double value.
- char * Xml_getString (xmlConfig_t *)

  Get name value.
- int Xml_getNbSimulations (xmlConfig_t *)

  Get number of simulation.
- int Xml_getNbTriesMod (xmlConfig_t *)

  Get number of tries for modelling.
- int Xml_getNbTriesSa (xmlConfig_t *)

  Get number of tries for the simulated annealing and minimization.
- int Xml_getNbIters (xmlConfig_t *)

  Get number of iteration.
- int Xml_getNbGroup (xmlConfig_t *)

  Get number of group.
- double Xml_getStepSize (xmlConfig_t *)

  Get the step size.
- double Xml_getBoltzmann (xmlConfig_t *)

  Get the Boltzmann value.
- double Xml_getTinitial (xmlConfig_t *)

  Get the initial temperature.
- double Xml_getMuT (xmlConfig_t *)

  Get the variation of temperature.
- double Xml_getTmin (xmlConfig_t *)

  Get the minimum temperature.
- int Xml_getCount (xmlConfig_t *, const char *)

  Get count.
- int Xml_getNbCouples (xmlConfig_t *)

  Get the number of couples.
- int Xml_getNbParameters (xmlConfig_t *)

  Get the number of parameters.

- int Xml_getNbReactioninNoeud (xmlConfig_t *, int)

  Count the number of reaction in one node.
- int Xml_getNbReaction (xmlConfig_t *)

  Get the number of reactions.
- int Xml_getNbSpecies (xmlConfig_t *)

  Get the number of species.
- int Xml_getNbEquations (xmlConfig_t *)

  Get the number of equations.
- int Xml_getNbBannedSpecies (xmlConfig_t *)

  Get the number of banned species.
- char ** Xml_getReactionsNamesinNoeud (xmlConfig_t *, int)

  Get list of reactions.
- char ** Xml_getReactionsNames (xmlConfig_t *)

  Get name of reactions.
- char ** Xml_getBanned (xmlConfig_t *conf)

  Get list of banned species.
- int Xml_getSpeciesFinalAmount (xmlConfig_t *, char *)

  Get the final amount of one species.
- int * Xml_getallSpeciesFinalAmount (xmlConfig_t *, char **, int)

  Get Table of species amount.
- int Xml_getSpeciesWeight (xmlConfig_t *, char *)

  Get Species weight.
- int * Xml_getallSpeciesWeight (xmlConfig_t *, char **, int)

  Get list of species weight.
- char ** Xml_allocEquation (char **, int)

  Allocate memory for the list of equation.
- int Xml_getKineticLawNodes (xmlConfig_t *, int)

  Get the kinetic law for one node.
- char *** Xml_getKineticLaw (xmlConfig_t *, int, int)

  Get list of kinetic laws.

4.10.1 Detailed Description

Xml reader for parametre.xml. This file is part of MetaBoFlux (http://www.cbib.u-bordeaux2.fr/metaboflux/)
Copyright (C) 2010 Amine Ghozlane from LaBRI and University of Bordeaux 1

MetaBoFlux is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

MetaBoFlux is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Author

{Amine Ghozlane}

Version

2.0

Date

15 janvier 2010

Definition in file xml_parameter.h.

4.10.2 Function Documentation

4.10.2.1 char** Xml_allocEquation (char ** *equation*, int *nb_noeud*)

Allocate memory for the list of equation.

Author

Amine Ghozlane

Parameters

equation	list of equation
nb_noeud	Number of the node

Returns

List of reactions

Definition at line 789 of file xml_parameter.c.

Referenced by Xml_getKineticLaw().

4.10.2.2 void Xml_freeConfig (xmlConfig_t * *conf*)

Free the Struct xmlConfig_t.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Definition at line 97 of file xml_parameter.c.

References xmlConfig_t::ctxt, xmlConfig_t::doc, and xmlConfig_t::fichier.

Referenced by Data_desallocation(), and Xml_loadConfig().

4.10.2.3 int* Xml_getallSpeciesFinalAmount (xmlConfig_t * *conf*, char ** *species*, int *taille*)

Get Table of species amount.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t
species	Table of species
taille	Number of species

Returns

Table of species amount

Definition at line 708 of file xml_parameter.c.

References Xml_getSpeciesFinalAmount().

Referenced by Data_allocScore().

Here is the call graph for this function:

4.10.2.4 int* Xml_getallSpeciesWeight (xmlConfig_t * *conf*, char ** *species*, int *taille*)

Get list of species weight.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t
species	Table of species
taille	Number of species

Returns

List of species weight

Definition at line 770 of file xml_parameter.c.

References Xml_getSpeciesWeight().

Referenced by Data_allocScore().

Here is the call graph for this function:

4.10.2.5 char** Xml_getBanned (xmlConfig_t * *conf*)

Get list of banned species.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

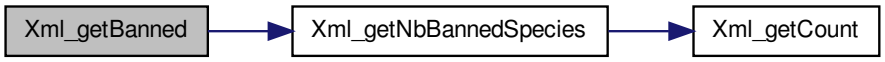
List of banned species

Definition at line 624 of file xml_parameter.c.

References xmlConfig_t::ctxt, and Xml_getNbBannedSpecies().

Referenced by Data_allocParameters().

Here is the call graph for this function:



4.10.2.6 double Xml_getBoltzmann (xmlConfig_t * *conf*)

Get the Boltzmann value.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

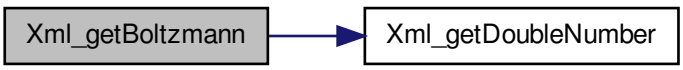
Boltzmann value

Definition at line 305 of file xml_parameter.c.

References Xml_getDoubleNumber().

Referenced by Mpi_slave(), and Recuit_printParametre().

Here is the call graph for this function:



4.10.2.7 int Xml_getCount (xmlConfig_t * *conf*, const char * *requete*)

Get count.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t
requete	Xpath query

Returns

Count

Definition at line 358 of file xml_parameter.c.

References xmlConfig_t::ctxt.

Referenced by Xml_getNbBannedSpecies(), Xml_getNbCouples(), Xml_getNbEquations(), Xml_getNbParameters(), Xml_getNbReaction(), Xml_getNbReactioninNoeud(), and Xml_-getNbSpecies().

4.10.2.8 double Xml_getDoubleNumber (xmlConfig_t * *conf*, const char * *requete*)

Get double value.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t
requete	Xpath query

Returns

Read double value in the xml

Definition at line 153 of file xml_parameter.c.

References xmlConfig_t::ctxt.

Referenced by Xml_getBoltzmann(), Xml_getMuT(), Xml_getStepSize(), Xml_getTinitial(), and Xml_getTmin().

4.10.2.9 char*** Xml_getKineticLaw (xmlConfig_t * *conf*, int *num_equation*, int *nb_noeud*)

Get list of kinetic laws.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t
num_equation	Number of equation
nb_noeud	Number nodes

Returns

List of kinetic laws

Definition at line 854 of file xml_parameter.c.

References xmlConfig_t::ctxt, and Xml_allocEquation().

Referenced by Data_allocEquations().

Here is the call graph for this function:

4.10.2.10 int Xml_getKineticLawNodes (xmlConfig_t * *conf*, int *num_equation*)

Get the kinetic law for one node.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t
num_equation	Number of the node

Returns

Ninetic law for one node

Definition at line 813 of file xml_parameter.c.

References xmlConfig_t::ctxt.

Referenced by Data_allocEquations().

4.10.2.11 double Xml_getMuT (xmlConfig_t * *conf*)

Get the variation of temperature.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

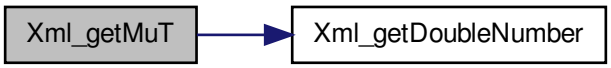
Variation of temperature

Definition at line 331 of file xml_parameter.c.

References Xml_getDoubleNumber().

Referenced by Mpi_slave(), and Recuit_printParametre().

Here is the call graph for this function:



4.10.2.12 int Xml_getNbBannedSpecies (xmlConfig_t * *conf*)

Get the number of banned species.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

Number of banned species

Definition at line 486 of file xml_parameter.c.

References Xml_getCount().

Referenced by Data_allocParameters(), and Xml_getBanned().

Here is the call graph for this function:

4.10.2.13 int Xml_getNbCouples (xmlConfig_t * *conf*)

Get the number of couples.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

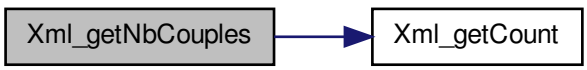
Number of couples

Definition at line 396 of file xml_parameter.c.

References Xml_getCount().

Referenced by Data_allocParameters(), and Xml_getReactionsNames().

Here is the call graph for this function:



4.10.2.14 int Xml_getNbEquations (xmlConfig_t * *conf*)

Get the number of equations.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

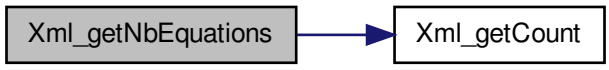
Number of equations

Definition at line 473 of file xml_parameter.c.

References Xml_getCount().

Referenced by Data_allocEquations().

Here is the call graph for this function:



4.10.2.15 int Xml_getNbGroup (xmlConfig_t * *conf*)

Get number of group.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

Number of iteration

Definition at line 279 of file xml_parameter.c.

References Xml_getNumber().

Referenced by Mpi_master().

Here is the call graph for this function:

4.10.2.16 int Xml_getNbIters (xmlConfig_t * *conf*)

Get number of iteration.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

Number of iteration

Definition at line 266 of file xml_parameter.c.

References Xml_getNumber().

Referenced by Mpi_slave(), and Recuit_printParametre().

Here is the call graph for this function:

4.10.2.17 int Xml_getNbParameters (xmlConfig_t * *conf*)

Get the number of parameters.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

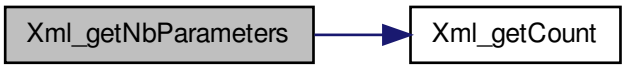
Number of parameters

Definition at line 409 of file xml_parameter.c.

References Xml_getCount().

Referenced by Data_allocParameters().

Here is the call graph for this function:



4.10.2.18 int Xml_getNbReaction (xmlConfig_t * *conf*)

Get the number of reactions.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

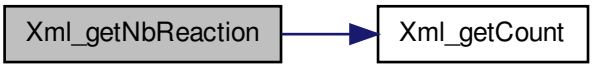
Number of reactions

Definition at line 447 of file xml_parameter.c.

References Xml_getCount().

Referenced by Data_allocScore(), and Xml_getReactionsNames().

Here is the call graph for this function:



4.10.2.19 int Xml_getNbReactioninNoeud (xmlConfig_t * *conf*, int *noeud*)

Count the number of reaction in one node.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t
noeud	Number of the node

Returns

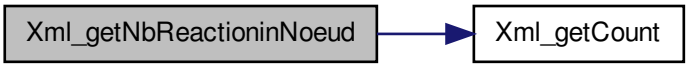
Number of reaction in one node

Definition at line 423 of file xml_parameter.c.

References Xml_getCount().

Referenced by Data_allocParameters(), and Xml_getReactionsNamesinNoeud().

Here is the call graph for this function:



4.10.2.20 int Xml_getNbSimulations (xmlConfig_t * *conf*)

Get number of simulation.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

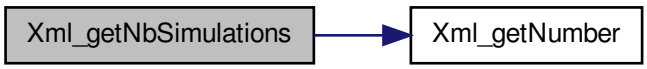
Number of simulation

Definition at line 227 of file xml_parameter.c.

References Xml_getNumber().

Referenced by Mpi_master().

Here is the call graph for this function:



4.10.2.21 int Xml_getNbSpecies (xmlConfig_t * *conf*)

Get the number of species.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

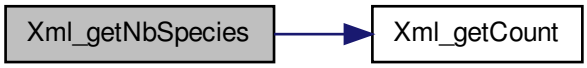
Number of species

Definition at line 460 of file xml_parameter.c.

References Xml_getCount().

Referenced by Data_allocScore(), Mpi_slave(), and Mpi_writer().

Here is the call graph for this function:



4.10.2.22 int Xml_getNbTriesMod (xmlConfig_t * *conf*)

Get number of tries for modelling.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

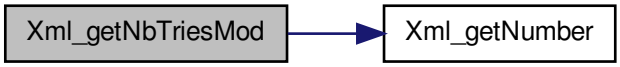
Number of tries

Definition at line 240 of file xml_parameter.c.

References Xml_getNumber().

Referenced by Data_allocParameters(), Mod_compute_modeling(), Mpi_sizeResultTab(), and Sd_compute_standard_deviation().

Here is the call graph for this function:



4.10.2.23 int Xml_getNbTriesSa (xmlConfig_t * *conf*)

Get number of tries for the simulated annealing and minimization.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

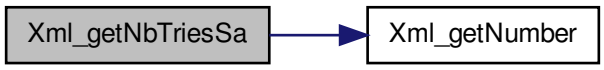
Number of tries

Definition at line 253 of file xml_parameter.c.

References Xml_getNumber().

Referenced by Data_allocParameters(), Min_my_f(), Mpi_slave(), and Recuit_printParametre().

Here is the call graph for this function:



4.10.2.24 int Xml_getNumber (xmlConfig_t * *conf*, const char * *requete*)

Get integer value.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t
requete	Xpath query

Returns

Read integer value in the xml

Definition at line 115 of file xml_parameter.c.

References xmlConfig_t::ctxt.

Referenced by Xml_getNbGroup(), Xml_getNbIters(), Xml_getNbSimulations(), Xml_-getNbTriesMod(), and Xml_getNbTriesSa().

4.10.2.25 char** Xml_getReactionsNames (xmlConfig_t * *conf*)

Get name of reactions.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

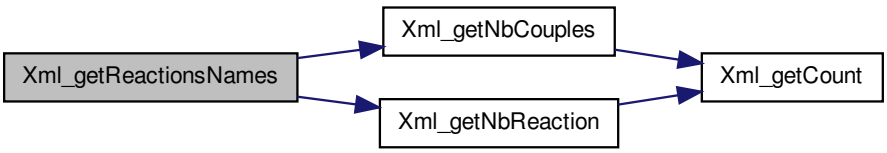
Name of reactions

Definition at line 541 of file xml_parameter.c.

References xmlConfig_t::ctxt, Xml_getNbCouples(), and Xml_getNbReaction().

Referenced by Data_allocScore().

Here is the call graph for this function:



4.10.2.26 char** Xml_getReactionsNamesinNoeud (xmlConfig_t * *conf*, int *numero*)

Get list of reactions.

Get list of species.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t
numero	Number of the node

Returns

List of reactions

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

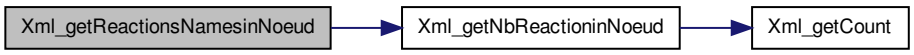
Returns

List of species

Definition at line 500 of file xml_parameter.c.

References xmlConfig_t::ctxt, and Xml_getNbReactioninNoeud().

Here is the call graph for this function:



4.10.2.27 int Xml_getSpeciesFinalAmount (xmlConfig_t * *conf*, char * *species*)

Get the final amount of one species.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t
species	Name of one species

Returns

Final amount of one species

Definition at line 665 of file xml_parameter.c.

References xmlConfig_t::ctxt.

Referenced by Xml_getallSpeciesFinalAmount().

4.10.2.28 int Xml_getSpeciesWeight (xmlConfig_t * *conf*, char * *species*)

Get Species weight.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t
species	Name of one species

Returns

Species weight

Definition at line 727 of file xml_parameter.c.

References xmlConfig_t::ctxt.

Referenced by Xml_getallSpeciesWeight().

4.10.2.29 double Xml_getStepSize (xmlConfig_t * *conf*)

Get the step size.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

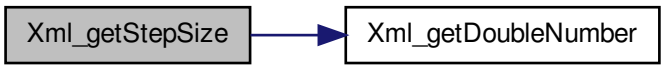
Step size

Definition at line 292 of file xml_parameter.c.

References Xml_getDoubleNumber().

Referenced by Mpi_slave(), and Recuit_printParametre().

Here is the call graph for this function:



4.10.2.30 char* Xml_getString (xmlConfig_t * *conf*)

Get name value.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

Read name value in the xml

Definition at line 190 of file xml_parameter.c.

References xmlConfig_t::ctxt.

Referenced by Recuit_redirectionFlux().

4.10.2.31 double Xml_getTinitial (xmlConfig_t * *conf*)

Get the initial temperature.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

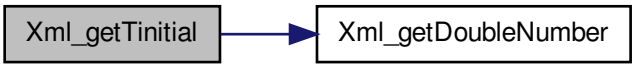
Initial temperature

Definition at line 318 of file xml_parameter.c.

References Xml_getDoubleNumber().

Referenced by Mpi_slave(), and Recuit_printParametre().

Here is the call graph for this function:



4.10.2.32 double Xml_getTmin (xmlConfig_t * *conf*)

Get the minimum temperature.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

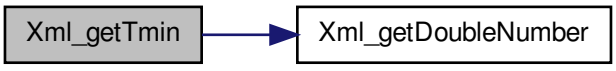
Minimum temperature

Definition at line 344 of file xml_parameter.c.

References Xml_getDoubleNumber().

Referenced by Mpi_slave(), and Recuit_printParametre().

Here is the call graph for this function:



4.10.2.33 xmlConfig_t* Xml_loadConfig (char * *fichier*)

Initialize and load the parameter.xml.

Author

Amine Ghozlane

Parameters

fichier	Xml file name

Returns

Struct xmlConfig_t

Definition at line 42 of file xml_parameter.c.

References xmlConfig_t::ctxt, xmlConfig_t::doc, xmlConfig_t::fichier, xmlConfig_t::racine, and Xml_freeConfig().

Referenced by Data_initXML().

Here is the call graph for this function:

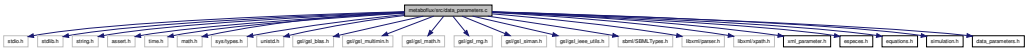
4.11 metaboflux/src/data_parameters.c File Reference

Load data parameters.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <assert.h>
#include <time.h>
#include <math.h>
#include <sys/types.h>
#include <unistd.h>
```

```
#include <gsl/gsl_blas.h>
#include <gsl/gsl_multimin.h>
#include <gsl/gsl_math.h>
#include <gsl/gsl_rng.h>
#include <gsl/gsl_siman.h>
#include <gsl/gsl_ieee_utils.h>
#include <sbml/SBMLTypes.h>
#include <libxml/parser.h>
#include <libxml/xpath.h>
#include "xml_parameter.h"
#include "especes.h"
#include "equations.h"
#include "simulation.h"
#include "data_parameters.h"
```

Include dependency graph for data_parameters.c:



Functions

- xmlConfig_t * Data_initXML (char *xml_file)

  Load parameter file.
- Model_t * Data_initSBML (char *sbml_file)

  Load SBML file.
- gsl_rng * Data_rngAlloc (gsl_rng *r)

  Allocation of the random number generator.
- Equations ** Data_equationsAlloc (pListParameters current, Equations **pile)

  Allocate the struct Equations.
- void Data_equationsInit (pListParameters current, Equations **pile)

  Initialize equations.
- void Data_allocEquations (pListParameters a)

  Allocate the different elements related to he struct Equations.
- void Data_allocScore (pScore out, xmlConfig_t *conf, Model_t *model)

  Copy the partial table x in the full table y.
- pScore Data_scoreAlloc (pListParameters a)

  Allocation of the struct Score.
- void Data_scoreInit (pScore out)

Initialize the struct Score.

- void Data_scoreFree (pScore out)

  Free the struct Score.

- int Data_copieTab (pSimParameters simulated, double *x, int a, int debut, int fin)

  Copy the partial table x in the full table y.

- void Data_updateTab (pListParameters current, pSimParameters simulated, double *x)

  Copy table.

- void Data_allocSimParameters (double *y, pListParameters a)

  Allocation of tables needed for simulations.

- void Data_allocParameters (pListParameters a)

  Copy the partial table x in the full table y.

- void Data_parameters (pListParameters a, char **files_path)

  Record parameters of the simulation.

- void Data_desallocation (pListParameters a)

  Free memory allocated to the various objects of the program.

- void Data_simParameters (pListParameters a, pSimParameters sim, char *out, char *texte, int number, int debug)

  Recording parameters specific to each simulation.

- void Data_desallocSim (pSimParameters sim)

  Free struct parameters specific to each simulation.

4.11.1 Detailed Description

Load data parameters. This file is part of MetaBoFlux (http://www.cbib.u-bordeaux2.fr/metabofl
Copyright (C) 2010 Amine Ghozlane from LaBRI and University of Bordeaux 1

MetaBoFlux is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

MetaBoFlux is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Author

{Amine Ghozlane}

Version

2.0

Date

10 novembre 2010

Definition in file data_parameters.c.

4.11.2 Function Documentation

4.11.2.1 void Data_allocEquations (pListParameters *a*)

Allocate the different elements related to he struct Equations.

Author

Amine Ghozlane

Parameters

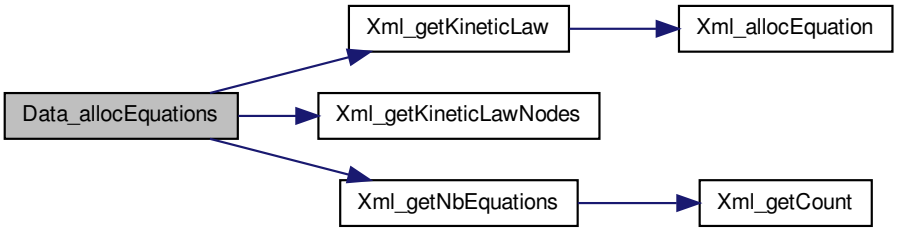
a	Global parameters : struct ListParameters

Definition at line 202 of file data_parameters.c.

References ListParameters::conf, ListParameters::equation, ListParameters::nb_equations, ListParameters::noeud, Xml_getKineticLaw(), Xml_getKineticLawNodes(), and Xml_-getNbEquations().

Referenced by Data_parameters().

Here is the call graph for this function:



4.11.2.2 void Data_allocParameters (pListParameters *a*)

Copy the partial table x in the full table y.

Author

Amine Ghozlane

Parameters

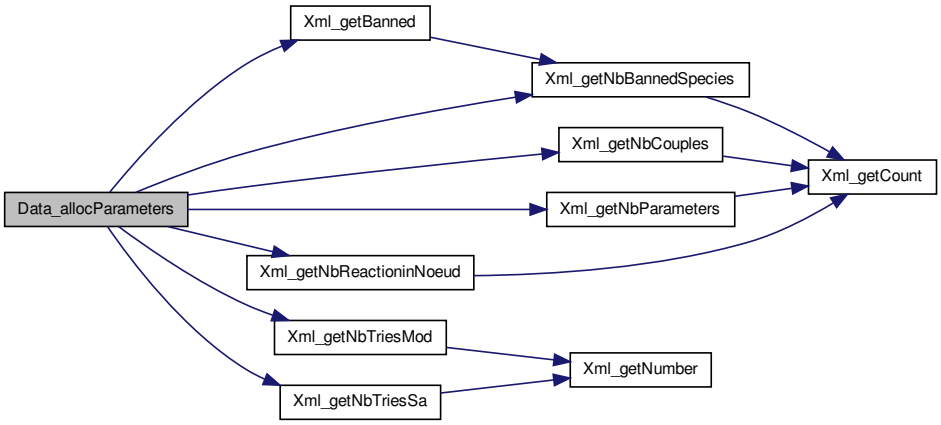
a	Global parameters : struct ListParameters

Definition at line 408 of file data_parameters.c.

References ListParameters::banned, ListParameters::conf, ListParameters::interest_parameters, ListParameters::nb_banned, ListParameters::nb_couples, ListParameters::nb_parameters, ListParameters::nb_triesMod, ListParameters::nb_triesSa, ListParameters::parameters, Xml_getBanned(), Xml_getNbBannedSpecies(), Xml_getNbCouples(), Xml_getNbParameters(), Xml_getNbReactioninNoeud(), Xml_getNbTriesMod(), and Xml_getNbTriesSa().

Referenced by Data_parameters().

Here is the call graph for this function:



4.11.2.3 void Data_allocScore (pScore *out*, xmlConfig_t * *conf*, Model_t * *model*)

Copy the partial table x in the full table y.

Author

Amine Ghozlane

Parameters

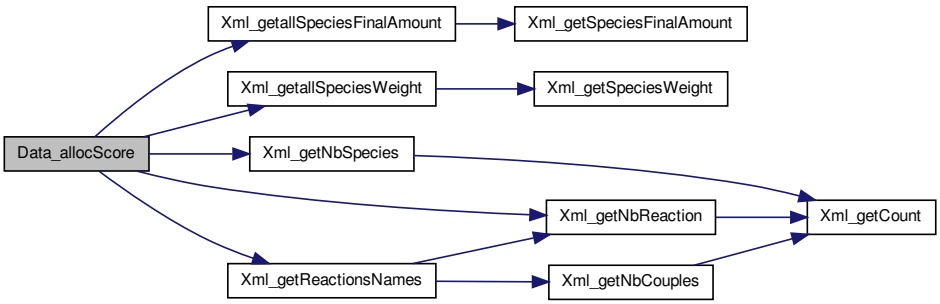
out	Struct Score
conf	Struct xmlConfig_t
model	Model of the SBML file

Definition at line 238 of file data_parameters.c.

References Score::name, Score::nb_reaction, Score::nb_species, Score::quantite, Score::reaction, Score::species, Score::species_amount, Score::species_weight, Score::taille, Score::tailleReactions, Score::tailleSpecies, Xml_getallSpeciesFinalAmount(), Xml_getallSpeciesWeight(), Xml_-getNbReaction(), Xml_getNbSpecies(), and Xml_getReactionsNames().

Referenced by Data_scoreAlloc(), and Data_simParameters().

Here is the call graph for this function:



4.11.2.4 void Data_allocSimParameters (double * *y*, pListParameters *a*)

Allocation of tables needed for simulations.

void Data_allocSimParameters(double *y, pListParameters a)

Author

Amine Ghozlane

Parameters

y	table of reaction parameters
a	Global parameters : struct ListParameters

Definition at line 391 of file data_parameters.c.

References ListParameters::nb_parameters.

Referenced by Data_simParameters().

4.11.2.5 int Data_copieTab (pSimParameters *simulated*, double * *x*, int *a*, int *debut*, int *fin*)

Copy the partial table x in the full table y.

Author

Amine Ghozlane

Parameters

simulated	Simulation parameters : struct SimParameters
x	Short table of reaction parameters
a	Line
debut	Beginning
fin	End

Returns

Number of copied element

Definition at line 347 of file data_parameters.c.

References SimParameters::y.

Referenced by Data_updateTab().

4.11.2.6 void Data_desallocation (pListParameters *a*)

Free memory allocated to the various objects of the program.

Author

Amine Ghozlane

Parameters

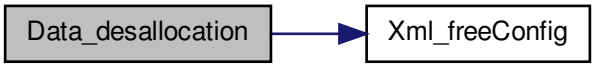
a	Global parameters : struct ListParameters

Definition at line 463 of file data_parameters.c.

References ListParameters::banned, ListParameters::conf, ListParameters::equation, ListParameters::model, ListParameters::nb_equations, ListParameters::noeud, ListParameters::parameters, and Xml_freeConfig().

Referenced by compute_mpi().

Here is the call graph for this function:



4.11.2.7 void Data_desallocSim (pSimParameters *sim*)

Free struct parameters specific to each simulation.

Author

Amine Ghozlane

Parameters

sim	Simulation parameters : struct SimParameters

Definition at line 549 of file data_parameters.c.

References SimParameters::debugFile, Score::name, Score::nb_reaction, SimParameters::out, Score::quantite, SimParameters::r, Score::reaction, Score::species, Score::species_amount, Score::species_weight, Score::taille, and SimParameters::y.

Referenced by Min_compute_minimization(), Mod_compute_modeling(), Recuit_compute_recuit(), and Sd_compute_standard_deviation().

4.11.2.8 Equations ** Data_equationsAlloc (pListParameters *current*, Equations ** *pile*)

Allocate the struct Equations.

Author

Amine Ghozlane

Parameters

current	Current parameters : struct ListParameters
pile	Pile des equations : struct Equations

Returns

Allocated struct Equations

Definition at line 169 of file data_parameters.c.

References ListParameters::nb_equations.

Referenced by Min_my_f(), Mod_compute_modeling(), Recuit_energyFunction(), and Sd_compute_simulation().

4.11.2.9 void Data_equationsInit (pListParameters *current*, Equations ** *pile*)

Initialize equations.

Author

Amine Ghozlane

Parameters

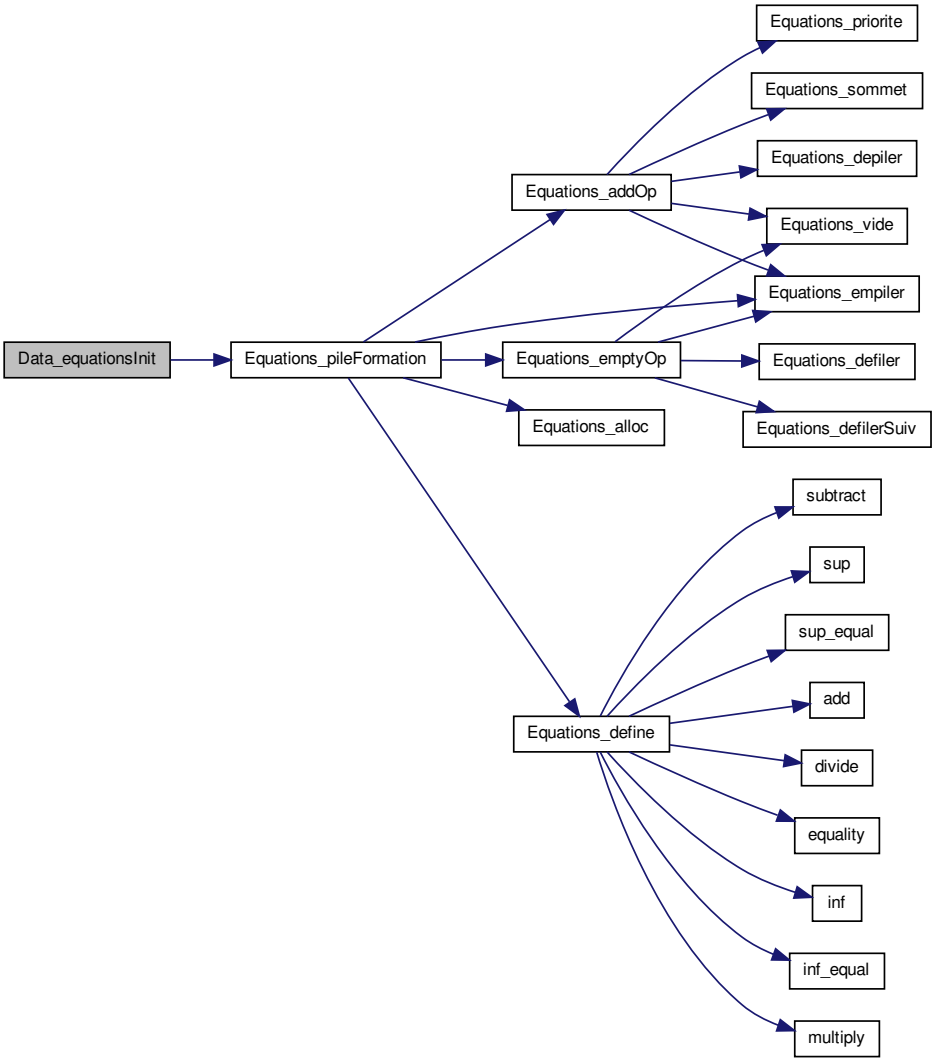
current	Current parameters : struct ListParameters
pile	Pile des equations : struct Equations

Definition at line 184 of file data_parameters.c.

References ListParameters::equation, Equations_pileFormation(), ListParameters::nb_-equations, and ListParameters::noeud.

Referenced by Min_my_f(), Mod_compute_modeling(), Recuit_energyFunction(), and Sd_compute_simulation().

Here is the call graph for this function:



4.11.2.10 Model_t * Data_initSBML (char * *sbml_file*)

Load SBML file.

Author

Amine Ghozlane

Parameters

sbml_file	Name of the SBML file

Returns

Model of the SBML file

Definition at line 81 of file data_parameters.c.

Referenced by Data_parameters().

4.11.2.11 xmlConfig_t * Data_initXML (char * *xml_file*)

Load parameter file.

Author

Amine Ghozlane

Parameters

xml_file	Name of the parameter file

Returns

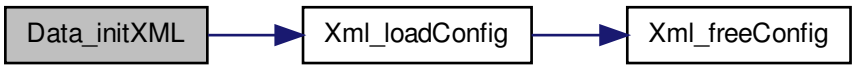
Model of the parameter file =:Struct xmlConfig_t

Definition at line 58 of file data_parameters.c.

References Xml_loadConfig().

Referenced by Data_parameters().

Here is the call graph for this function:



4.11.2.12 void Data_parameters (pListParameters *a*, char ** *files_path*)

Record parameters of the simulation.

Author

Amine Ghozlane

Parameters

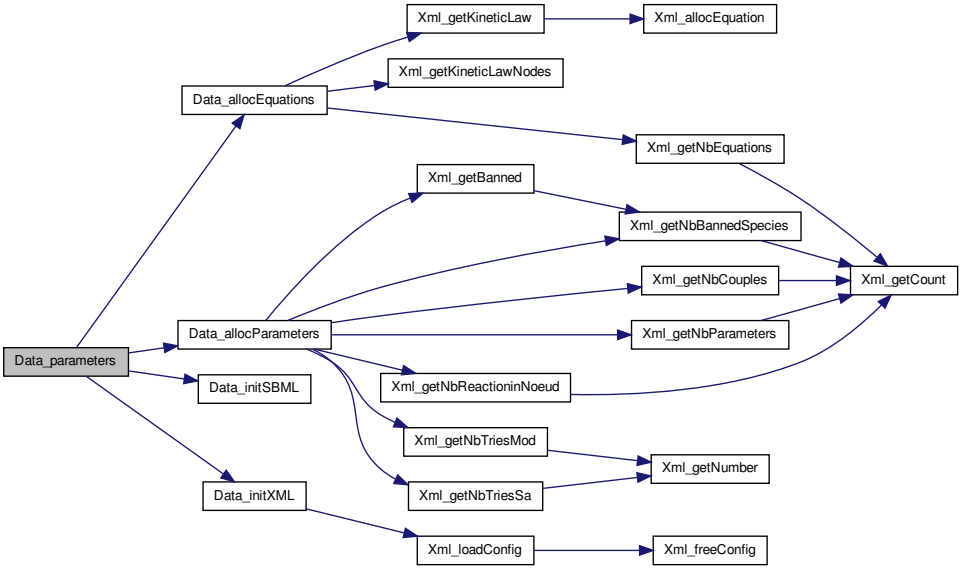
a	Global parameters : struct ListParameters
files_path	List of paths

Definition at line 442 of file data_parameters.c.

References ListParameters::conf, Data_allocEquations(), Data_allocParameters(), Data_initSBML(), Data_initXML(), and ListParameters::model.

Referenced by compute_mpi().

Here is the call graph for this function:



4.11.2.13 gsl_rng * Data_rngAlloc (gsl_rng * *r*)

Allocation of the random number generator.

Author

Amine Ghozlane

Parameters

r	Random number generator

Returns

Random number generator

Definition at line 135 of file data_parameters.c.

Referenced by Data_simParameters().

4.11.2.14 pScore Data_scoreAlloc (pListParameters *a*)

Allocation of the struct Score.

Author

Amine Ghozlane

Parameters

a	Global parameters : struct ListParameters

Returns

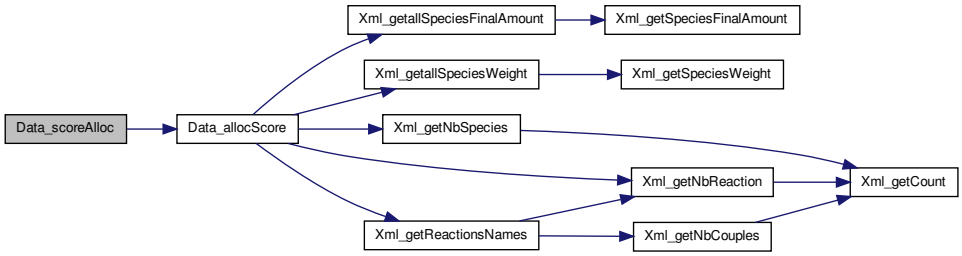
Allocated struct Score

Definition at line 274 of file data_parameters.c.

References ListParameters::conf, Data_allocScore(), and ListParameters::model.

Referenced by Min_compute_minimization(), Mod_compute_modeling(), and Recuit_-compute_recuit().

Here is the call graph for this function:



4.11.2.15 void Data_scoreFree (pScore *out*)

Free the struct Score.

Author

Amine Ghozlane

Parameters

out	Struct score

Definition at line 307 of file data_parameters.c.

References Score::name, Score::nb_reaction, Score::quantite, Score::reaction, Score::species_amount, Score::species_weight, and Score::taille.

Referenced by Min_compute_minimization(), Mod_compute_modeling(), and Recuit_compute_recuit().

4.11.2.16 void Data_scoreInit (pScore *out*)

Initialize the struct Score.

Author

Amine Ghozlane

Parameters

out	Empty struct Score

Definition at line 290 of file data_parameters.c.

References Score::quantite, and Score::taille.

Referenced by Min_my_f(), Recuit_energyFunction(), and Sd_compute_simulation().

4.11.2.17 void Data_simParameters (pListParameters *a*, pSimParameters *sim*, char * *out*, char * *texte*, int *number*, int *debug*)

Recording parameters specific to each simulation.

Author

Amine Ghozlane

Parameters

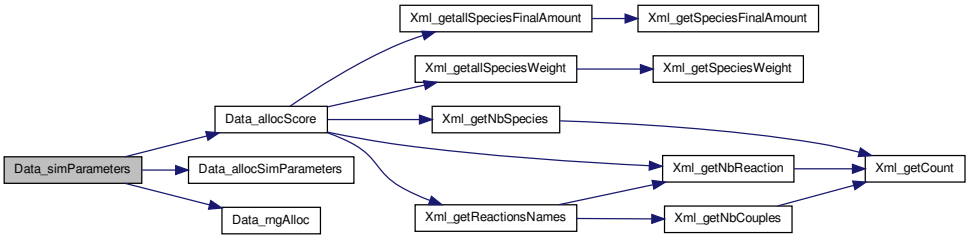
a	Global parameters : struct ListParameters
sim	Simulation parameters : struct SimParameters
out	Output repertory
texte	File name
number	Number of the debug file
debug	Determine debug

Definition at line 506 of file data_parameters.c.

References ListParameters::conf, Data_allocScore(), Data_allocSimParameters(), Data_rngAlloc(), SimParameters::debugFile, ListParameters::model, ListParameters::nb_parameters, SimParameters::out, SimParameters::pile, SimParameters::r, and SimParameters::y.

Referenced by Min_compute_minimization(), Mod_compute_modeling(), Recuit_compute_-recuit(), and Sd_compute_standard_deviation().

Here is the call graph for this function:



4.11.2.18 void Data_updateTab (pListParameters *current*, pSimParameters *simulated*, double * *x*)

Copy table.

Author

Amine Ghozlane

Parameters

current	Current parameters : struct ListParameters
simulated	Simulation parameters : struct SimParameters
x	Short table of reaction parameters

Definition at line 371 of file data_parameters.c.

References Data_copieTab(), ListParameters::nb_couples, and ListParameters::parameters.

Referenced by Min_my_f(), Recuit_energyFunction(), and Recuit_printPosition().

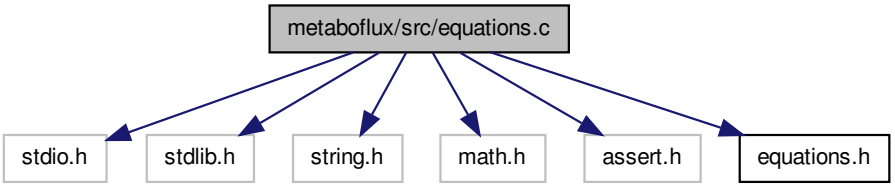
Here is the call graph for this function:

4.12 metaboflux/src/equations.c File Reference

Processes an equation in MathML format.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <assert.h>
#include "equations.h"
```

Include dependency graph for equations.c:



Functions

- double add (double arg1, double arg2)

  Add two numbers.
- double subtract (double arg1, double arg2)

  Subtraction two numbers.
- double multiply (double arg1, double arg2)

  Multiply two numbers.
- double divide (double arg1, double arg2)

  Divide two numbers.
- double equality (double arg1, double arg2)

  Test the equality of two numbers.
- double sup (double arg1, double arg2)

  Test the superiority between two numbers.
- double sup_equal (double arg1, double arg2)

  Test the superiority or equality between two numbers.
- double inf (double arg1, double arg2)

  Test the inferiority between two numbers.

- double inf_equal (double arg1, double arg2)

  Test the inferiority or equality between two numbers.
- pEquations Equations_alloc (void)

  Alloc memory and initialize the struct Equations.
- void Equations_define (pEquations new, char *operateur)

  Identify the mathematical operator used.
- int Equations_vide (pEquations liste)

  Test if the struct Equations is empty.
- pEquations Equations_empiler (pEquations liste, pEquations new)

  Stack an element to the struct Equations.
- pEquations Equations_depiler (pEquations liste)

  Unstack the last element of the struct Equations.
- pEquations Equations_sommet (pEquations liste)

  Look for the last element of the struct Equations.
- pEquations Equations_defiler (pEquations liste)

  Look for the first element of the struct Equations.
- pEquations Equations_defilerSuiv (pEquations liste)

  Get the next element of the struct Equations.
- int Equations_priorite (pEquations new)

  Give the priority of the operator. Multiplication and Division have higher priority than addition or subtration...
- void Equations_print (pEquations liste)

  Print all information on the element of the struct Equations.
- pEquations Equations_addOp (pEquations result, pEquations op, pEquations new)

  Build the list of the Struct Equations result used to compute the equation.
- void Equations_emptyOp (pEquations result, pEquations op)

  Empty the Struct Equations op for the struct Equation result.
- pEquations Equations_pileFormation (char ***equation, int nb_noeud)

  Build pile.
- double Equations_extractData (pEquations new, char **name, double *quantite, int taille)

  Extract information on element of type Struct Equation.
- double Equations_findSpecies (char *species, char **name, double *quantite, int taille)

  Look for the quantity of a molecule in the list.
- double Equations_resultat (pEquations result, pEquations result1, pEquations eq, char **name, double *quantite, int taille)

  Compute the result of two pile.
- double Equations_calcul (pEquations liste, char **name, double *quantite, int taille)

  Compute the score of the equation.
- double Equations_finalQuantite (int file_nb_especes, char **file_species, int *file_amount, int *file_weight, char **sim_name, double *sim_quantite, int sim_taille)

Compute the score of the quantity which means the difference between what is expected to what is simulated.

4.12.1 Detailed Description

Processes an equation in MathML format. This file is part of Metaboflux (http://www.cbib.u-bordeaux2.fr/metab
Copyright (C) 2010 Amine Ghozlane from LaBRI and University of Bordeaux 1

Metaboflux is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

Metaboflux is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Author

{Amine Ghozlane}

Version

2.0

Date

15 janvier 2010

Definition in file equations.c.

4.12.2 Function Documentation

4.12.2.1 double add (double *arg1*, double *arg2*)

Add two numbers.

Author

Amine Ghozlane

Parameters

arg1	Value 1
arg2	Value 2

Returns

Result

Definition at line 43 of file equations.c.

Referenced by Equations_define().

4.12.2.2 double divide (double *arg1*, double *arg2*)

Divide two numbers.

Author

Amine Ghozlane

Parameters

arg1	Value 1
arg2	Value 2

Returns

Result

Definition at line 82 of file equations.c.

Referenced by Equations_define().

4.12.2.3 double equality (double *arg1*, double *arg2*)

Test the equality of two numbers.

Author

Amine Ghozlane

Parameters

arg1	Value 1
arg2	Value 2

Returns

Result

Definition at line 95 of file equations.c.

Referenced by Equations_define().

4.12.2.4 pEquations Equations_addOp (pEquations *result*, pEquations *op*, pEquations *new*)

Build the list of the Struct Equations result used to compute the equation.

Author

Amine Ghozlane

Parameters

result	Struct Equations used for computation
op	Struct Equations used to store the operator
new	One element of the Struct Equations

Returns

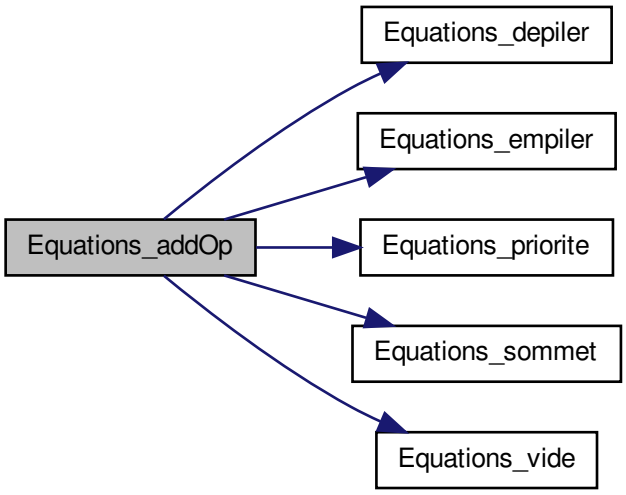
List of operator

Definition at line 383 of file equations.c.

References equal, Equations_depiler(), Equations_empiler(), Equations_priorite(), Equations_-sommet(), Equations_vide(), inferior, inferior_equal, superior, and superior_equal.

Referenced by Equations_pileFormation().

Here is the call graph for this function:



4.12.2.5 pEquations Equations_alloc (void)

Alloc memory and initialize the struct Equations.

Author

Amine Ghozlane

Returns

Allocated struct Equations

Definition at line 158 of file equations.c.

Referenced by Equations_calcul(), and Equations_pileFormation().

4.12.2.6 double Equations_calcul (pEquations *liste*, char ** *name*, double * *quantite*, int *taille*)

Compute the score of the equation.

Author

Amine Ghozlane

Parameters

liste	pile of type Struct Equations
name	List of molecules
quantite	List of the quantity of the molecules
taille	Number of molecules in the list

Returns

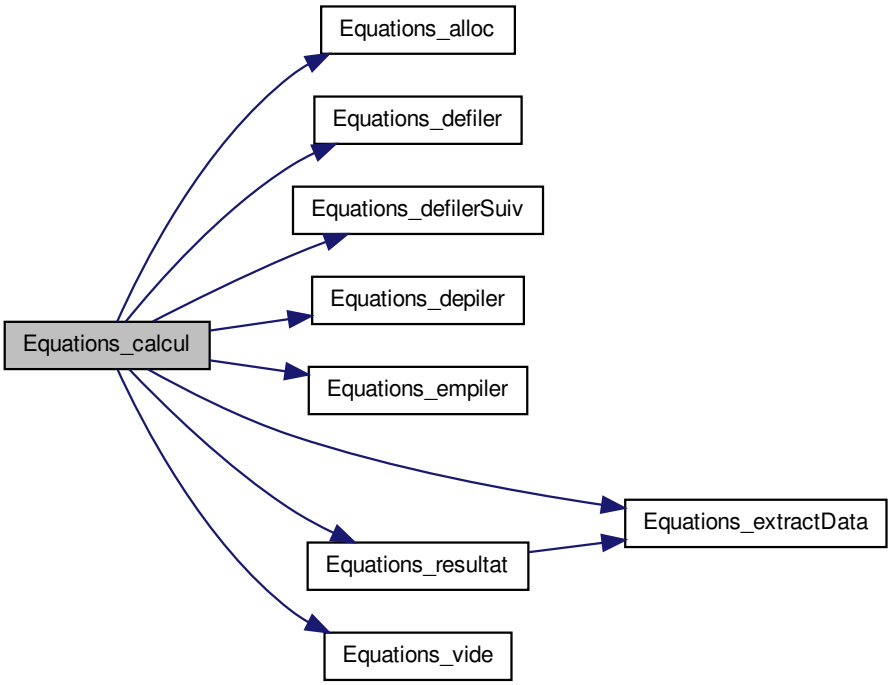
Result of the equation (First part of the energy)

Definition at line 614 of file equations.c.

References addition, constant, division, equal, Equations_alloc(), Equations_defiler(), Equations_defilerSuiv(), Equations_depiler(), Equations_empiler(), Equations_extractData(), Equations_resultat(), Equations_vide(), inferior, inferior_equal, multiplication, soustraction, superior, and superior_equal.

Referenced by Min_my_f(), Mod_compute_modeling(), Recuit_energyFunction(), and Sd_compute_simulation().

Here is the call graph for this function:



4.12.2.7 pEquations Equations_defiler (pEquations *liste*)

Look for the first element of the struct Equations.

Author

Amine Ghozlane

Parameters

liste	Struct Equations

Returns

Pointer on the first element of struct Equations

Definition at line 320 of file equations.c.

Referenced by Equations_calcul(), and Equations_emptyOp().

4.12.2.8 pEquations Equations_defilerSuiv (pEquations *liste*)

Get the next element of the struct Equations.

Author

Amine Ghozlane

Parameters

liste	Struct Equations

Returns

Pointer on the next element of struct Equations

Definition at line 334 of file equations.c.

Referenced by Equations_calcul(), and Equations_emptyOp().

4.12.2.9 void Equations_define (pEquations *new*, char * *operateur*)

Identify the mathematical operator used.

Author

Amine Ghozlane

Parameters

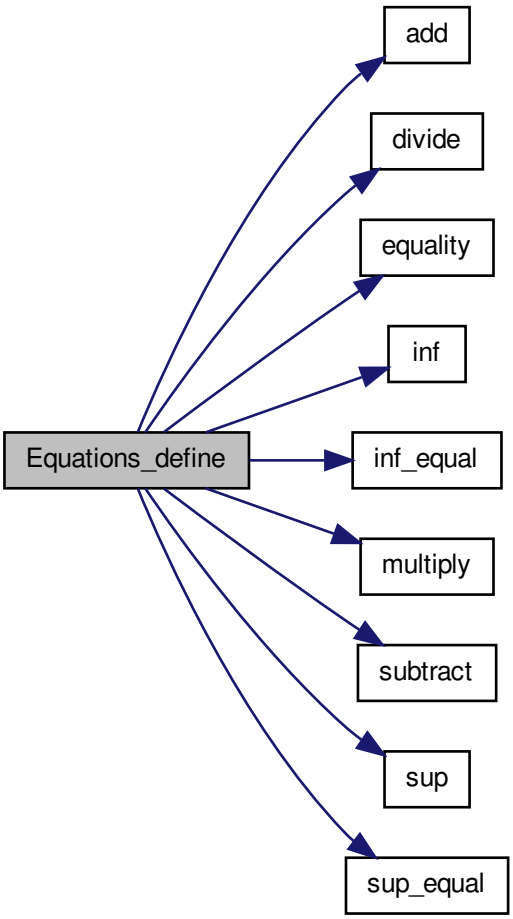
new	Struct Equations
operateur	Read operator

Definition at line 174 of file equations.c.

References add(), addition, divide(), division, equal, equality(), inf(), inf_equal(), inferior, inferior_equal, multiplication, multiply(), soustraction, subtract(), sup(), sup_equal(), superior, and superior_equal.

Referenced by Equations_pileFormation().

Here is the call graph for this function:



4.12.2.10 pEquations Equations_depiler (pEquations *liste*)

Unstack the last element of the struct Equations.

Author

Amine Ghozlane

Parameters

liste	Struct Equations.

Returns

Pointer on the "unstack" element of struct Equations.

Definition at line 282 of file equations.c.

Referenced by Equations_addOp(), and Equations_calcul().

4.12.2.11 pEquations Equations_empiler (pEquations *liste,* pEquations *new*)

Stack an element to the struct Equations.

Author

Amine Ghozlane

Parameters

liste	Struct Equations
new	New struct Equations element

Returns

Pointer on the last element of struct Equations

Definition at line 258 of file equations.c.

Referenced by Equations_addOp(), Equations_calcul(), Equations_emptyOp(), and Equations_pileFormation().

4.12.2.12 void Equations_emptyOp (pEquations *result,* pEquations *op*)

Empty the Struct Equations op for the struct Equation result.

Author

Amine Ghozlane

Parameters

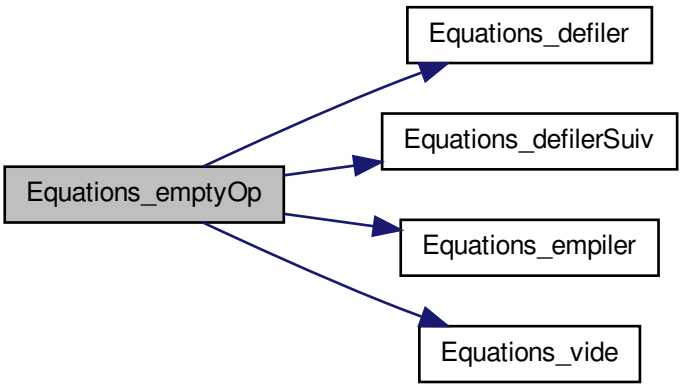
result	Struct Equations used for computation
op	Struct Equation used to store the operator

Definition at line 421 of file equations.c.

References Equations_defiler(), Equations_defilerSuiv(), Equations_empiler(), and Equations_vide().

Referenced by Equations_pileFormation().

Here is the call graph for this function:



4.12.2.13 double Equations_extractData (pEquations *new*, char ** *name*, double * *quantite*, int *taille*)

Extract information on element of type Struct Equation.

Author

Amine Ghozlane

Parameters

new	Element of type Struct Equation
name	Name of the interest element
quantite	List of the quantity of the molecules
taille	Number of molecules in the list

Returns

Quantity of the molecule of interest

Definition at line 497 of file equations.c.

References constant.

Referenced by Equations_calcul(), and Equations_resultat().

4.12.2.14 double Equations_finalQuantite (int *file_nb_especes*, char ** *file_species*, int * *file_amount*, int * *file_weight*, char ** *sim_name*, double * *sim_quantite*, int *sim_taille*)

Compute the score of the quantity which means the difference between what is expected to what is simulated.

Author

Amine Ghozlane

Parameters

file_nb_especes	Number of species in the parameter file
file_species	List of species in the parameter file
file_amount	List of species expected quantity
file_weight	Weight defined for the species
sim_name	List of species simulated (sbml file)
sim_quantite	List of the quantity of the species simulated (sbml file)
sim_taille	Number of species simulated (sbml file)

Returns

Result of the difference (Second part of the Energy)

Definition at line 683 of file equations.c.

References Equations_findSpecies().

Referenced by Min_my_f(), Mod_compute_modeling(), Recuit_energyFunction(), and Sd_compute_simulation().

Here is the call graph for this function:

4.12.2.15 double Equations_findSpecies (char * *species*, char ** *name*, double * *quantite*, int *taille*)

Look for the quantity of a molecule in the list.

Author

Amine Ghozlane

Parameters

species	Name of a molecule
name	List of molecules
quantite	List of the quantity of the molecules
taille	Number of molecules in the list

Returns

Quantity of the molecule of interest

Definition at line 533 of file equations.c.

Referenced by Equations_finalQuantite(), Min_score_print_mean(), and Mod_score_print_mean().

4.12.2.16 pEquations Equations_pileFormation (char *** *equation*, int *nb_noeud*)

Build pile.

Author

Amine Ghozlane

Parameters

equation	Table with the equation in Mathml format
nb_noeud	Number of element inside the equation

Returns

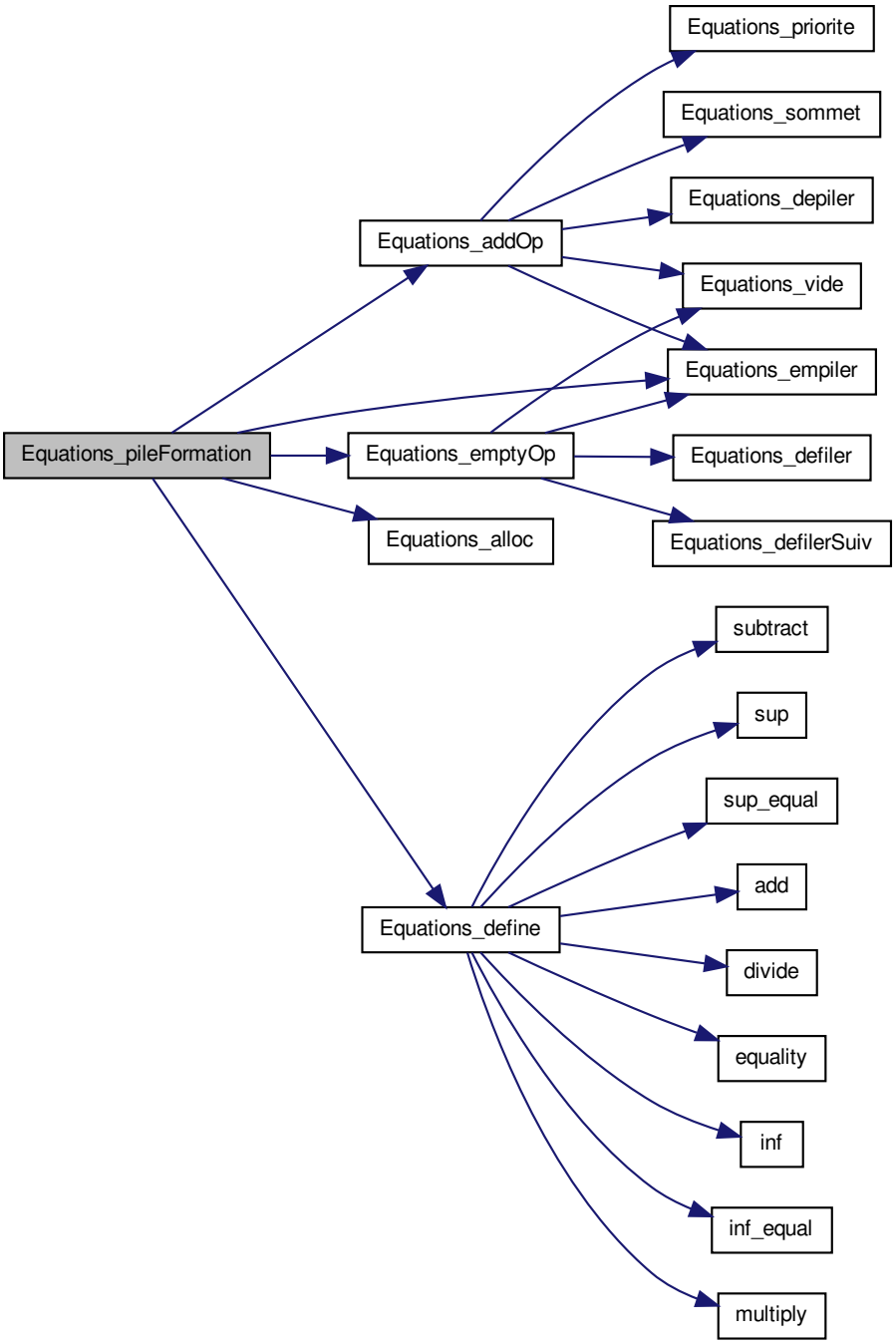
Struct Equation needed for compute the equation

Definition at line 441 of file equations.c.

References constant, Equations_addOp(), Equations_alloc(), Equations_define(), Equations_empiler(), Equations_emptyOp(), and variable.

Referenced by Data_equationsInit().

Here is the call graph for this function:



4.12.2.17 void Equations_print (pEquations *liste*)

Print all information on the element of the struct Equations.

Author

Amine Ghozlane

Parameters

liste	Struct Equations

Definition at line 359 of file equations.c.

References constant, Equdata::data, Equations::info, Equations::type, Equdata::var, and variable.

4.12.2.18 int Equations_priorite (pEquations *new*)

Give the priority of the operator. Multiplication and Division have higher priority than addition or subtration...

Author

Amine Ghozlane

Parameters

new	One element of the struct Equations

Returns

Priority of the operator

Definition at line 346 of file equations.c.

References addition, equal, inferior, inferior_equal, soustraction, superior, and superior_equal.

Referenced by Equations_addOp().

4.12.2.19 double Equations_resultat (pEquations *result*, pEquations *result1*, pEquations *eq*, char ** *name*, double * *quantite*, int *taille*)

Compute the result of two pile.

Author

Amine Ghozlane

Parameters

result	Struct Equation result

result1	Struct Equation result after the operator
eq	Operator
name	List of molecules
quantite	List of the quantity of the molecules
taille	Number of molecules in the list

Returns

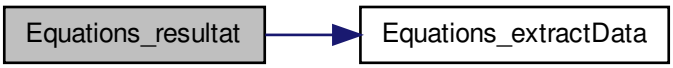
Result of the two pile

Definition at line 561 of file equations.c.

References equal, Equations_extractData(), inferior, inferior_equal, superior, superior_-equal, and Equations::type.

Referenced by Equations_calcul().

Here is the call graph for this function:



4.12.2.20 pEquations Equations_sommet (pEquations *liste*)

Look for the last element of the struct Equations.

Author

Amine Ghozlane

Parameters

liste	Struct Equations

Returns

Pointer on the last element of struct Equations

Definition at line 305 of file equations.c.

Referenced by Equations_addOp().

4.12.2.21 int Equations_vide (pEquations *liste*)

Test if the struct Equations is empty.

Author

Amine Ghozlane

Parameters

liste	Struct Equations

Returns

Number of elements

Definition at line 237 of file equations.c.

Referenced by Equations_addOp(), Equations_calcul(), and Equations_emptyOp().

4.12.2.22 double inf (double *arg1*, double *arg2*)

Test the inferiority between two numbers.

Author

Amine Ghozlane

Parameters

arg1	Value 1
arg2	Value 2

Returns

Result

Definition at line 134 of file equations.c.

Referenced by Equations_define().

4.12.2.23 double inf_equal (double *arg1*, double *arg2*)

Test the inferiority or equality between two numbers.

Author

Amine Ghozlane

Parameters

arg1	Value 1
arg2	Value 2

Returns

Result

Definition at line 147 of file equations.c.

Referenced by Equations_define().

4.12.2.24 double multiply (double *arg1*, double *arg2*)

Multiply two numbers.

Author

Amine Ghozlane

Parameters

arg1	Value 1
arg2	Value 2

Returns

Result

Definition at line 69 of file equations.c.

Referenced by Equations_define().

4.12.2.25 double subtract (double *arg1*, double *arg2*)

Subtraction two numbers.

Author

Amine Ghozlane

Parameters

arg1	Value 1
arg2	Value 2

Returns

Result

Definition at line 56 of file equations.c.

Referenced by Equations_define().

4.12.2.26 double sup (double *arg1*, double *arg2*)

Test the superiority between two numbers.

Author

Amine Ghozlane

Parameters

arg1	Value 1
arg2	Value 2

Returns

Result

Definition at line 108 of file equations.c.

Referenced by Equations_define().

4.12.2.27 double sup_equal (double *arg1*, double *arg2*)

Test the superiority or equality between two numbers.

Author

Amine Ghozlane

Parameters

arg1	Value 1
arg2	Value 2

Returns

Result

Definition at line 121 of file equations.c.

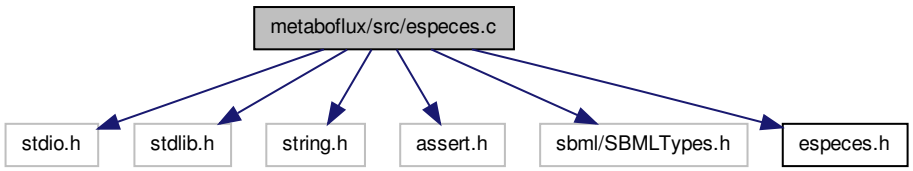
Referenced by Equations_define().

4.13 metaboflux/src/especes.c File Reference

Modelize a molecule.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <assert.h>
#include <sbml/SBMLTypes.h>
#include "especes.h"
```

Include dependency graph for especes.c:



Functions

- pEspeces Especes_alloc (int nbEspeces)

  Alloc memory and initialize the struct Especes.
- void Especes_allocReactions (pEspeces molecules, int Espece, Reaction_t *local, double reactRatio)

  Alloc memory and initialize the struct Reaction for each molecule.
- void Especes_free (pEspeces molecules, int nbEspeces)

  Free the struct pEspeces.
- void Especes_freeReactions (Especes molecules)

  Free the struct Reaction.
- void Especes_save (pEspeces molecules, int i, double quant, const char *dye)

  Save data on one molecule.
- void Especes_scoreSpecies (pEspeces molecules, int nbEspeces, char **name, double *quantite)

  Save the quantite of all molecules for scoring.
- void Especes_print (pEspeces molecules, int nbEspeces)

  Print data on each molecule.
- void Especes_print_2 (pEspeces molecules, int nbEspeces)

  Print Id and quantity on each molecule.
- int Especes_find (pEspeces molecules, const char *especeId, int nbEspeces)

  Find a molecule thank to it ID.
- void Especes_setQuantite (pEspeces molecules, int Especes, double quant)

  Change the quantity of a molecule.
- double Especes_getQuantite (pEspeces molecules, int Especes)

  Get the quanty of one molecule by its number.
- int Especes_getNbreactions (pEspeces molecules, int ref)

  Get number of reaction where one molecule is used (like a reactif)

4.13.1 Detailed Description

Modelize a molecule. This file is part of MetaBoFlux (http://www.cbib.u-bordeaux2.fr/metaboflux/)

Author

{Amine Ghozlane}

Version

2.0

Date

27 octobre 2009

Definition in file especes.c.

4.13.2 Function Documentation

4.13.2.1 pEspeces Especes_alloc (int *nbEspeces*)

Alloc memory and initialize the struct Especes.

Author

Amine Ghozlane

Parameters

nbEspeces	Number of molecules

Returns

Allocated struct Espece

Definition at line 40 of file especes.c.

References Especes::quantite, and Especes::system.

Referenced by SBML_compute_simulation(), and SBML_compute_simulation_mean().

4.13.2.2 void Especes_allocReactions (pEspeces *molecules*, int *Especе*, Reaction_t * *local*, double *reactRatio*)

Alloc memory and initialize the struct Reaction for each molecule.

Author

Amine Ghozlane

Parameters

molecules	List of molecules
Especе	Number of the selected molecule
local	Reaction sbml reference
reactRatio	Ratio of the reaction

Definition at line 64 of file especes.c.

References Reaction::link, Reaction::suivant, and Especes::system.

Referenced by SBML_setReactions().

4.13.2.3 int Especes_find (pEspeces *molecules*, const char * *especeId*, int *nbEspeces*)

Find a molecule thank to it ID.

Author

Amine Ghozlane

Parameters

molecules	List of molecules
especeId	SBML ID of the molecule
nbEspeces	Number of molecules

Returns

Number of a selected molecule

Definition at line 214 of file especes.c.

Referenced by SBML_checkQuantite(), SBML_reaction(), and SBML_setReactions().

4.13.2.4 void Especes_free (pEspeces *molecules*, int *nbEspeces*)

Free the struct pEspeces.

Author

Amine Ghozlane

Parameters

molecules	List of molecules
nbEspeces	Number of molecules

Definition at line 94 of file especes.c.

References Especes_freeReactions().

Referenced by SBML_compute_simulation(), and SBML_compute_simulation_mean().

Here is the call graph for this function:

4.13.2.5 void Especes_freeReactions (Especes *molecules*)

Free the struct Reaction.

Author

Amine Ghozlane

Parameters

molecules	List of molecules

Definition at line 109 of file especes.c.

References Reaction::suivant, and Especes::system.

Referenced by Especes_free().

4.13.2.6 int Especes_getNbreactions (pEspeces *molecules*, int *ref*)

Get number of reaction where one molecule is used (like a reactif)

Author

Amine Ghozlane

Parameters

molecules	List of molecules
ref	Number of a selected molecule

Returns

Number of reactions

Definition at line 265 of file especes.c.

References Reaction::suivant, and Especes::system.

Referenced by SBML_simulate().

4.13.2.7 double Especes_getQuantite (pEspeces *molecules*, int *Especes*)

Get the quanty of one molecule by its number.

Author

Amine Ghozlane

Parameters

molecules	List of molecules
Especes	Number of a selected molecule

Returns

Quantity of a selected molecule

Definition at line 252 of file especes.c.

References Especes::quantite.

Referenced by SBML_checkQuantite(), SBML_reaction(), and SBML_simulate().

4.13.2.8 void Especes_print (pEspeces *molecules*, int *nbEspeces*)

Print data on each molecule.

Author

Amine Ghozlane

Parameters

molecules	List of molecules
nbEspeces	Number of molecules

Definition at line 170 of file especes.c.

References Reaction::link, Reaction::suivant, and Especes::system.

4.13.2.9 void Especes_print_2 (pEspeces *molecules*, int *nbEspeces*)

Print Id and quantity on each molecule.

Author

Amine Ghozlane

Parameters

molecules	List of molecules
nbEspeces	Number of molecules

Definition at line 196 of file especes.c.

4.13.2.10 void Especes_save (pEspeces *molecules*, int *i*, double *quant*, const char * *dye*)

Save data on one molecule.

Author

Amine Ghozlane

Parameters

molecules	List of molecules
i	Number of the selected molecule
quant	Quantity of the molecule
dye	ID of the molecule

Definition at line 134 of file especes.c.

References Especes::id, and Especes::quantite.

Referenced by SBML_initEspeceAmounts().

4.13.2.11 void Especes_scoreSpecies (pEspeces *molecules*, int *nbEspeces*, char ** *name*, double * *quantite*)

Save the quantite of all molecules for scoring.

Author

Amine Ghozlane

Parameters

molecules	List of molecules
nbEspeces	Number of molecules
name	ID of the selected molecule
quantite	Table of molecule Quantity

Definition at line 149 of file especes.c.

References Especes::quantite.

Referenced by SBML_score().

4.13.2.12 void Especes_setQuantite (pEspeces *molecules*, int *Especes*, double *quant*)

Change the quantity of a molecule.

Author

Amine Ghozlane

Parameters

molecules	List of molecules
Especes	Number of a selected molecule
quant	Quantity of the molecule

Definition at line 239 of file especes.c.

References Especes::quantite.

Referenced by SBML_reaction().

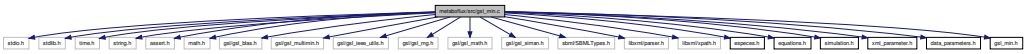
4.14 metaboflux/src/gsl_min.c File Reference

Compute minimization of scenarii.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <string.h>
#include <assert.h>
#include <math.h>
#include <gsl/gsl_blas.h>
#include <gsl/gsl_multimin.h>
#include <gsl/gsl_ieee_utils.h>
#include <gsl/gsl_rng.h>
#include <gsl/gsl_math.h>
#include <gsl/gsl_siman.h>
#include <sbml/SBMLTypes.h>
#include <libxml/parser.h>
#include <libxml/xpath.h>
#include "especes.h"
#include "equations.h"
#include "simulation.h"
```

```
#include "xml_parameter.h"
#include "data_parameters.h"
#include "gsl_min.h"
```

Include dependency graph for gsl_min.c:



Functions

- int Min_copieTab2 (double ∗x, int current, int debut, int fin)

  Copy table of reaction parameters.
- int Min_copieTab3 (gsl_multimin_fminimizer ∗s, int current, int debut, int fin)

  Copy short table of parameter into the larger table.
- void Min_verifValue (gsl_vector ∗v, double ∗x, int debut, int max)

  Checking reaction ratio parameters.
- double Min_my_f (const gsl_vector ∗v, void ∗params)

  Enter in the program for standard deviation.
- void Min_getTampon (double ∗energie_temp, double energie)

  Enter in the program for standard deviation.
- void Min_score_print_mean (double ∗result_tab)

  Save the minimization result.
- void Min_compute_minimization (pListParameters allone, double ∗fluxRatio, double ∗result_tab, char ∗∗files_path, int number_arg, int debug)

  Compute the minimization.

4.14.1 Detailed Description

Compute minimization of scenarii. This file is part of MetaBoFlux (http://www.cbib.u-bordeaux2.fr/metaboflu
Copyright (C) 2010 Amine Ghozlane from LaBRI and University of Bordeaux 1

MetaBoFlux is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

MetaBoFlux is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Author

{Amine Ghozlane}

Version

2.0

Date

9 novembre 2009

Definition in file gsl_min.c.

4.14.2 Function Documentation

4.14.2.1 void Min_compute_minimization (pListParameters *allone*, double * *fluxRatio*, double * *result_tab*, char ** *files_path*, int *number_arg*, int *debug*)

Compute the minimization.

Author

Amine Ghozlane

Parameters

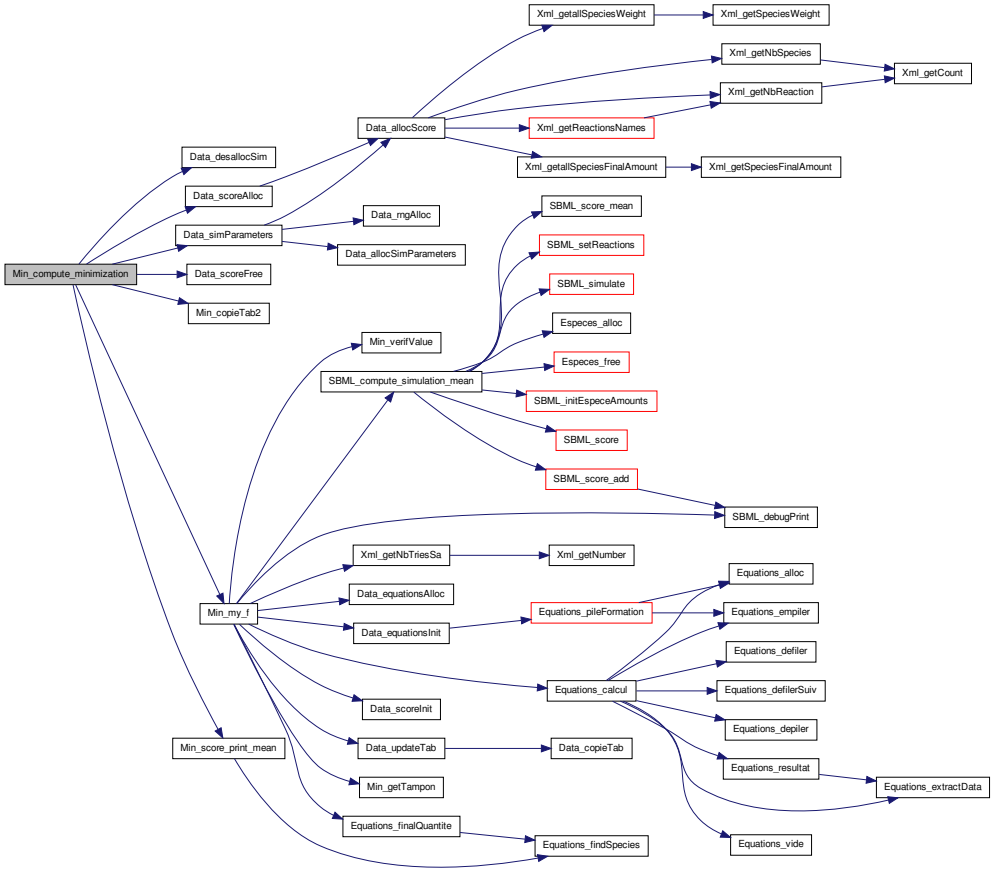
allone	struct ListParameters
fluxRatio	Ratio parameters
result_tab	Result table
files_path	List of paths
number_arg	Number of simulation
debug	Debug flag

Definition at line 274 of file gsl_min.c.

References Data_desallocSim(), Data_scoreAlloc(), Data_scoreFree(), Data_simParameters(), ListParameters::interest_parameters, Min_copieTab2(), Min_my_f(), Min_score_print_-mean(), ListParameters::nb_couples, ListParameters::nb_parameters, ListParameters::parameters, and SimParameters::y.

Referenced by Mpi_slave().

Here is the call graph for this function:



4.14.2.2 int Min_copieTab2 (double * *x*, int *current*, int *debut*, int *fin*)

Copy table of reaction parameters.

Author

Amine Ghozlane

Parameters

x	Short table of reaction parameters
current	Line
debut	Beginning
fin	End

Returns

Number of copied element

Definition at line 68 of file gsl_min.c.

References SimParameters::y.

Referenced by Min_compute_minimization().

4.14.2.3 int Min_copieTab3 (gsl_multimin_fminimizer * *s*, int *current*, int *debut*, int *fin*)

Copy short table of parameter into the larger table.

Author

Amine Ghozlane

Parameters

s	Minimizer parameter
current	Line
debut	Beginning
fin	End

Returns

Number of copied element

Definition at line 90 of file gsl_min.c.

References SimParameters::y.

4.14.2.4 void Min_getTampon (double * *energie_temp*, double *energie*)

Enter in the program for standard deviation.

Author

Amine Ghozlane

Parameters

energie_temp	Current energy
energie	Best energy

Definition at line 208 of file gsl_min.c.

References ListParameters::nb_parameters, SimParameters::out, Score::quantite, Score::taille, and SimParameters::y.

Referenced by Min_my_f().

4.14.2.5 double Min_my_f (const gsl_vector * *v*, void * *params*)

Enter in the program for standard deviation.

Author

Amine Ghozlane

Parameters

v	Vector of reaction parameters
params	Unused parameter define by GSL

Returns

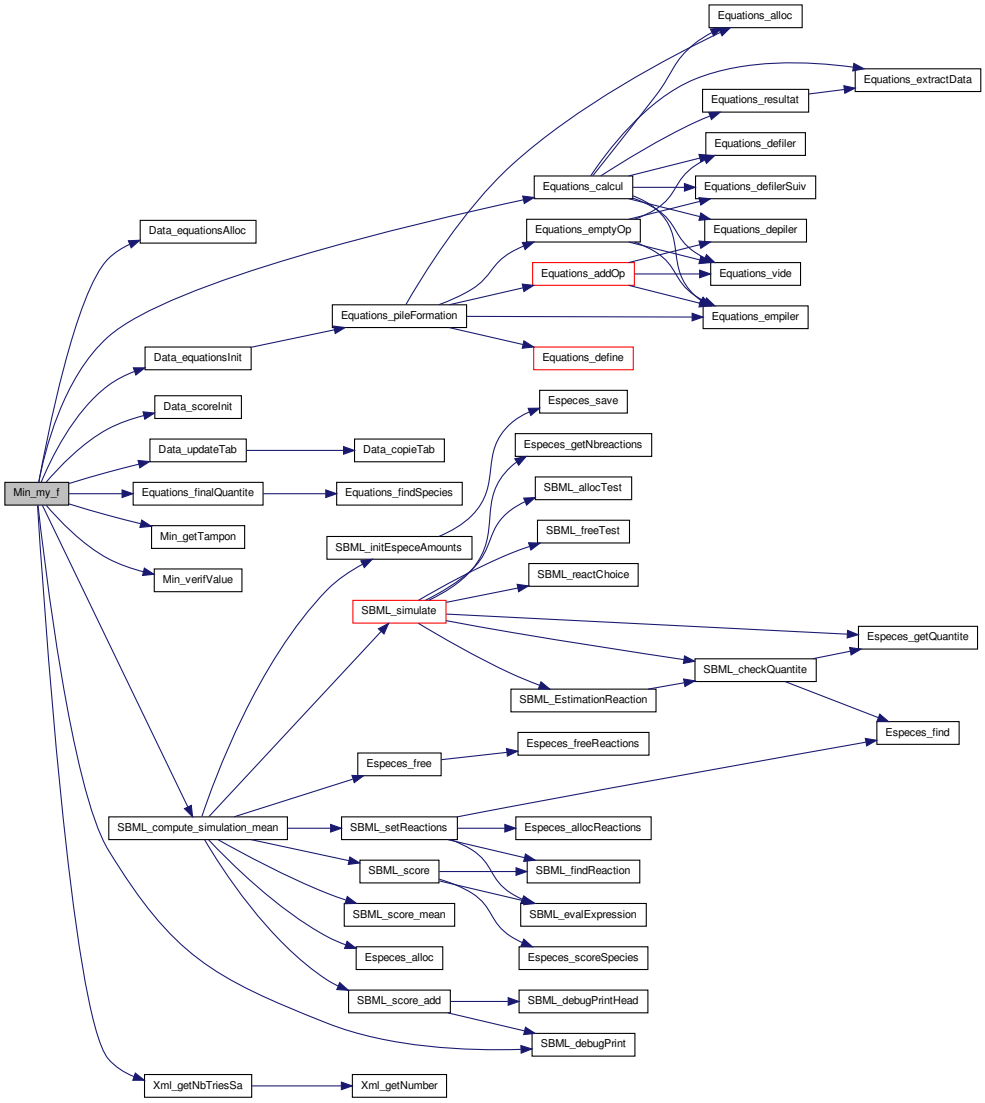
Energy value

Definition at line 149 of file gsl_min.c.

References ListParameters::banned, ListParameters::conf, Data_equationsAlloc(), Data_equationsInit(), Data_scoreInit(), Data_updateTab(), SimParameters::debugFile, Equations_calcul(), Equations_finalQuantite(), Min_getTampon(), Min_verifValue(), ListParameters::model, Score::name, ListParameters::nb_banned, ListParameters::nb_couples, ListParameters::nb_equations, ListParameters::nb_parameters, Score::nb_species, SimParameters::out, ListParameters::parameters, SimParameters::pile, Score::quantite, SimParameters::r, SBML_compute_simulation_mean(), SBML_debugPrint(), Score::species, Score::species_amount, Score::species_weight, Score::taille, Score::tailleSpecies, Xml_getNbTriesSa(), and SimParameters::y.

Referenced by Min_compute_minimization().

Here is the call graph for this function:



4.14.2.6 void Min_score_print_mean (double * *result_tab*)

Save the minimization result.

Author

Amine Ghozlane

Parameters

result_tab	Result table

Definition at line 230 of file gsl_min.c.

References Equations_findSpecies(), Score::name, ListParameters::nb_parameters, Score::nb_-species, SimParameters::out, Score::quantite, Score::species, Score::species_amount, Score::taille, Score::tailleSpecies, and SimParameters::y.

Referenced by Min_compute_minimization().

Here is the call graph for this function:

4.14.2.7 void Min_verifValue (gsl_vector * *v*, double * *x*, int *debut*, int *max*)

Checking reaction ratio parameters.

Author

Amine Ghozlane

Parameters

v	Vector of reaction parameters
x	Short table of reaction parameters
debut	Beginning
max	End

Definition at line 114 of file gsl_min.c.

Referenced by Min_my_f().

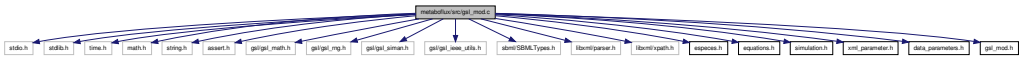
4.15 metaboflux/src/gsl_mod.c File Reference

Compute the modeling of scenarii.

```
#include <stdio.h>
#include <stdlib.h>
```

```
#include <time.h>
#include <math.h>
#include <string.h>
#include <assert.h>
#include <gsl/gsl_math.h>
#include <gsl/gsl_rng.h>
#include <gsl/gsl_siman.h>
#include <gsl/gsl_ieee_utils.h>
#include <sbml/SBMLTypes.h>
#include <libxml/parser.h>
#include <libxml/xpath.h>
#include "especes.h"
#include "equations.h"
#include "simulation.h"
#include "xml_parameter.h"
#include "data_parameters.h"
#include "gsl_mod.h"
```

Include dependency graph for gsl_mod.c:



Functions

- void Mod_score_print_mean (pListParameters a, pSimParameters simu, double *result_tab, double energie, int number_group, int group)

  Print the mean score.
- void Mod_compute_modeling (pListParameters a, double *fluxRatio, double *result_tab, char **files_path, int number_group, int group, int debug)

  Compute the modeling.

4.15.1 Detailed Description

Compute the modeling of scenarii. This file is part of MetaBoFlux (http://www.cbib.u-bordeaux2.fr/m
Copyright (C) 2010 Amine Ghozlane from LaBRI and University of Bordeaux 1

MetaBoFlux is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation,

either version 3 of the License, or (at your option) any later version.

MetaBoFlux is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Author

{Amine Ghozlane}

Version

2.0

Date

9 novembre 2009

Definition in file gsl_mod.c.

4.15.2 Function Documentation

4.15.2.1 void Mod_compute_modeling (pListParameters *a*, double * *fluxRatio*, double * *result_tab*, char ** *files_path*, int *number_group*, int *group*, int *debug*)

Compute the modeling.

Author

Amine Ghozlane

Parameters

a	struct ListParameters
fluxRatio	Ratio parameters
result_tab	Result table
files_path	List of paths
number_group	Number of the group
group	Group flag
debug	Debug flag

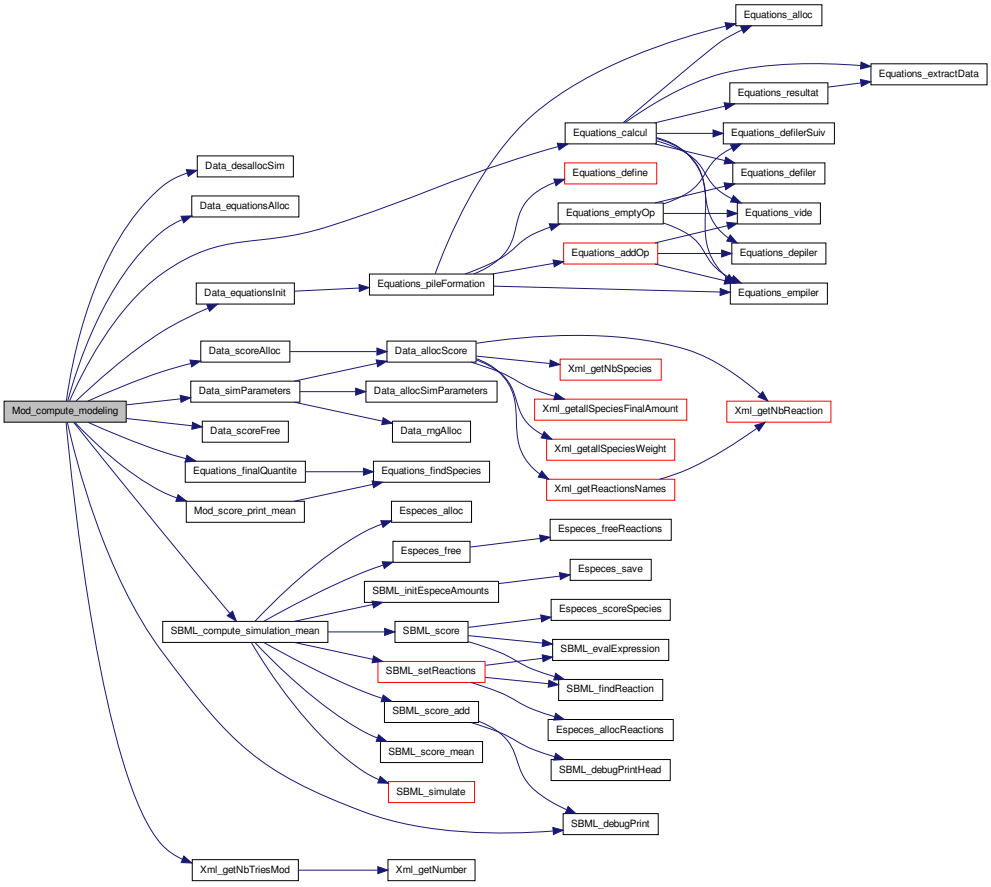
Definition at line 109 of file gsl_mod.c.

References ListParameters::banned, ListParameters::conf, Data_desallocSim(), Data_equationsAlloc(), Data_equationsInit(), Data_scoreAlloc(), Data_scoreFree(), Data_simParameters(), SimParameters::debugFile, Equations_calcul(), Equations_finalQuantite(), Mod_score_print_mean(), ListParameters::model, Score::name, ListParameters::nb_banned, ListParameters::nb_equations, ListParameters::nb_parameters, Score::nb_species, SimParameters::out, Sim-

Parameters::pile, Score::quantite, SimParameters::r, SBML_compute_simulation_mean(), SBML_debugPrint(), Score::species, Score::species_amount, Score::species_weight, Score::taille, Score::tailleSpecies, Xml_getNbTriesMod(), and SimParameters::y.

Referenced by Mpi_slave().

Here is the call graph for this function:



4.15.2.2 void Mod_score_print_mean (pListParameters *a,* pSimParameters *simu,* double * *result_tab,* double *energie,* int *number_group,* int *group*)

Print the mean score.

Author

Amine Ghozlane

Parameters

a	Global parameters
simu	Simulation parameters
result_tab	Result table
energie	Energy of the simulation
number_group	Number of simulation
group	Number of the group (if a group is simulated)

Definition at line 58 of file gsl_mod.c.

References Equations_findSpecies(), Score::name, ListParameters::nb_parameters, Score::nb_species, SimParameters::out, Score::quantite, Score::species, Score::species_amount, Score::taille, Score::tailleSpecies, and SimParameters::y.

Referenced by Mod_compute_modeling().

Here is the call graph for this function:

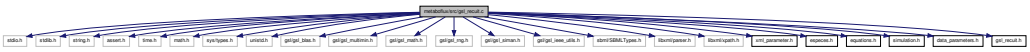
4.16 metaboflux/src/gsl_recuit.c File Reference

Compute the simulated annealing.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <assert.h>
#include <time.h>
#include <math.h>
#include <sys/types.h>
#include <unistd.h>
#include <gsl/gsl_blas.h>
#include <gsl/gsl_multimin.h>
#include <gsl/gsl_math.h>
```

```
#include <gsl/gsl_rng.h>
#include <gsl/gsl_siman.h>
#include <gsl/gsl_ieee_utils.h>
#include <sbml/SBMLTypes.h>
#include <libxml/parser.h>
#include <libxml/xpath.h>
#include "xml_parameter.h"
#include "especes.h"
#include "equations.h"
#include "simulation.h"
#include "data_parameters.h"
#include "gsl_recuit.h"
```

Include dependency graph for gsl_recuit.c:



Functions

- double Recuit_energyFunction (void *xp)

  Simulate the petri net and compute the energy.
- double Recuit_metricDistance (void *xp, void *yp)

  Compute the distance between the two table.
- void Recuit_takeStep (const gsl_rng *r, void *xp, double step_size)

  Take a step through the solution space TSP.
- void Recuit_printPosition (void *xp)

  Print the table of reaction parameters.
- FILE * Recuit_redirectionFlux (xmlConfig_t *conf, FILE *f, char **files_path, int number)

  Redirect stdout flux.
- void Recuit_verifParameters (double *x_initial, const gsl_rng *r, int debut, int max)

  Choice of parameters and verification of their value.
- void Recuit_verifParameters_2 (double *new_x, double *old_x, const gsl_rng *r, int debut, int max, double step_size)

  Choice of parameters and verification of their value.
- void Recuit_defParametre (double *x_initial, const gsl_rng *r)

  Definition of the initial parameters of the simulation.
- void Recuit_printParametre (double *x_initial)

Print initial parameters.

- void Recuit_compute_recuit (char **files_path, int debug, int number, pListParameters allone, gsl_siman_params_t params)

  Compute the simulated annealing.

4.16.1 Detailed Description

Compute the simulated annealing. This file is part of MetaBoFlux (http://www.cbib.u-bordeaux2.fr/metaboflu
Copyright (C) 2010 Amine Ghozlane from LaBRI and University of Bordeaux 1

MetaBoFlux is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

MetaBoFlux is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Author

{Amine Ghozlane}

Version

2.0

Date

9 novembre 2009

Definition in file gsl_recuit.c.

4.16.2 Function Documentation

4.16.2.1 void Recuit_compute_recuit (char ** *files_path*, int *debug*, int *number*, pListParameters *allone*, gsl_siman_params_t *params*)

Compute the simulated annealing.

Author

Amine Ghozlane

Parameters

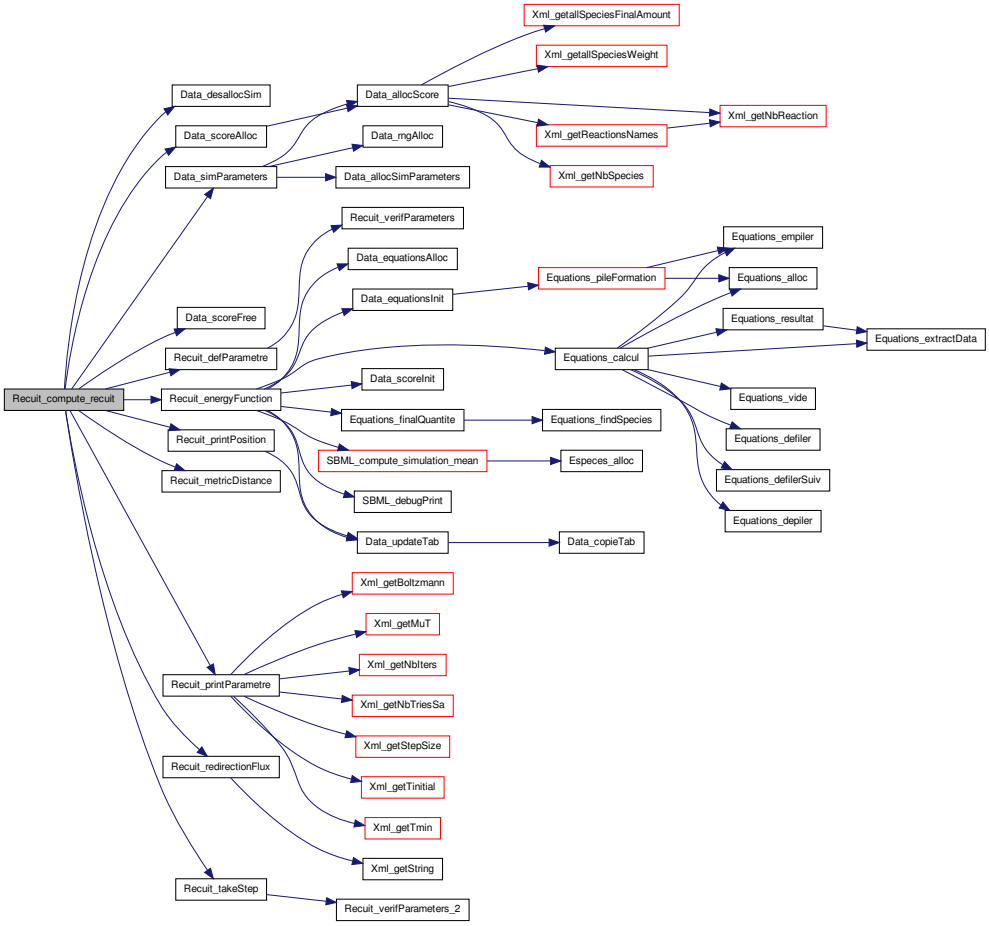
files_path	List of paths
debug	Debug flag
number	Number of the simulation
allone	Global parameters : struct ListParameters
params	Gsl parameters

Definition at line 304 of file gsl_recuit.c.

References ListParameters::conf, Data_desallocSim(), Data_scoreAlloc(), Data_scoreFree(), Data_simParameters(), ListParameters::interest_parameters, SimParameters::r, Recuit_-defParametre(), Recuit_energyFunction(), Recuit_metricDistance(), Recuit_printParametre(), Recuit_printPosition(), Recuit_redirectionFlux(), and Recuit_takeStep().

Referenced by Mpi_slave().

Here is the call graph for this function:



4.16.2.2 void Recuit_defParametre (double * *x_initial*, const gsl_rng * *r*)

Definition of the initial parameters of the simulation.

Author

Amine Ghozlane

Parameters

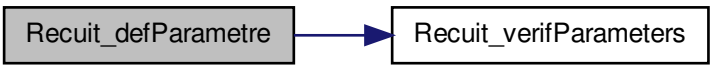
x_initial	table of reaction parameters
r	Random number generator

Definition at line 261 of file gsl_recuit.c.

References ListParameters::nb_couples, ListParameters::parameters, and Recuit_verifParameters().

Referenced by Recuit_compute_recuit().

Here is the call graph for this function:



4.16.2.3 double Recuit_energyFunction (void * *xp*)

Simulate the petri net and compute the energy.

double Recuit_energyFunction(void *xp)

Author

Amine Ghozlane

Parameters

xp	Short table of reaction parameters

Returns

Energy value

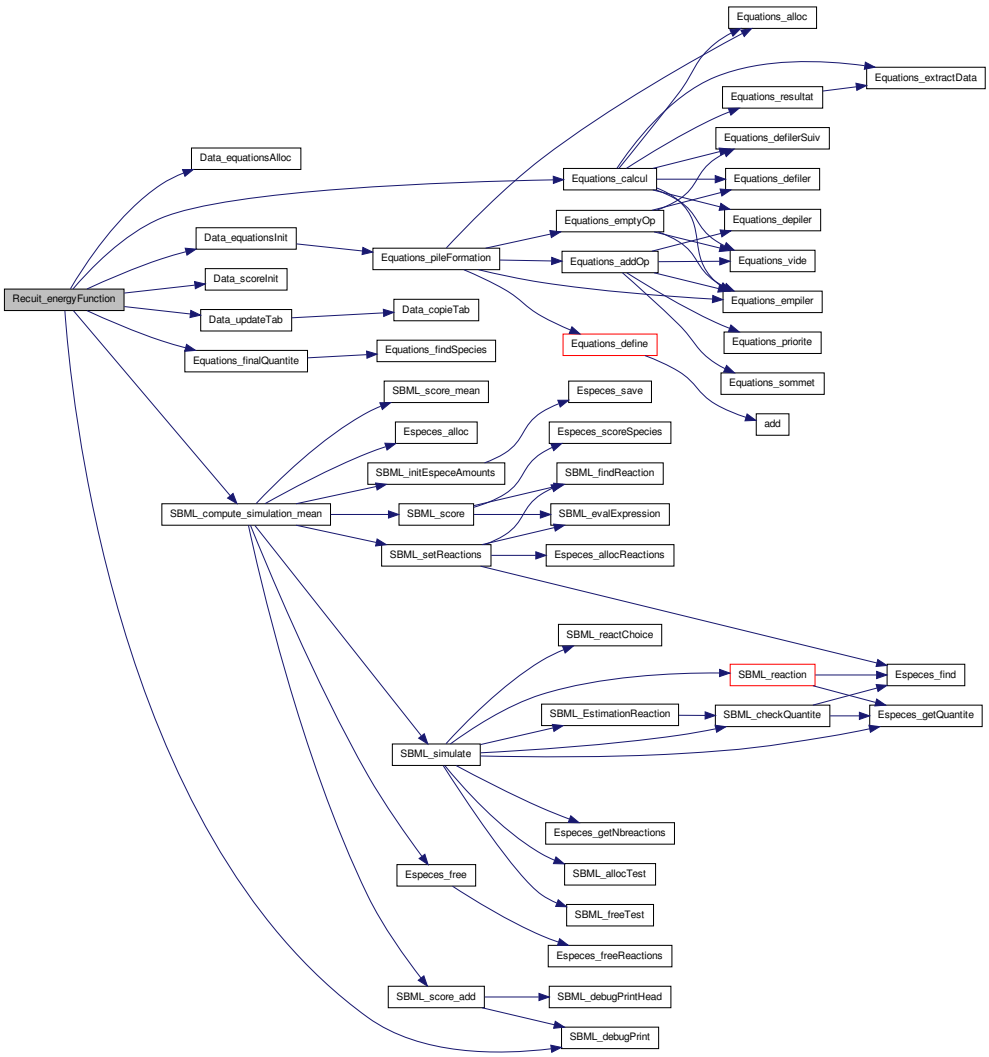
Definition at line 65 of file gsl_recuit.c.

References ListParameters::banned, Data_equationsAlloc(), Data_equationsInit(), Data_scoreInit(), Data_updateTab(), SimParameters::debugFile, Equations_calcul(), Equations_finalQuantite(), ListParameters::model, Score::name, ListParameters::nb_banned, ListParameters::nb_equations, Score::nb_species, ListParameters::nb_triesSa, SimParameters::out, SimParameters::pile, Score::quantite, SimParameters::r, SBML_compute_simulation_mean(),

SBML_debugPrint(), Score::species, Score::species_amount, Score::species_weight, Score::taille, Score::tailleSpecies, and SimParameters::y.

Referenced by Recuit_compute_recuit().

Here is the call graph for this function:



4.16.2.4 double Recuit_metricDistance (void * *xp*, void * *yp*)

Compute the distance between the two table.

Author

Amine Ghozlane

Parameters

xp	Past short table of reaction parameters
yp	New short table of reaction parameters

Returns

Distance between the two table

Definition at line 115 of file gsl_recuit.c.

References ListParameters::interest_parameters.

Referenced by Recuit_compute_recuit().

4.16.2.5 void Recuit_printParametre (double * *x_initial*)

Print initial parameters.

Author

Amine Ghozlane

Parameters

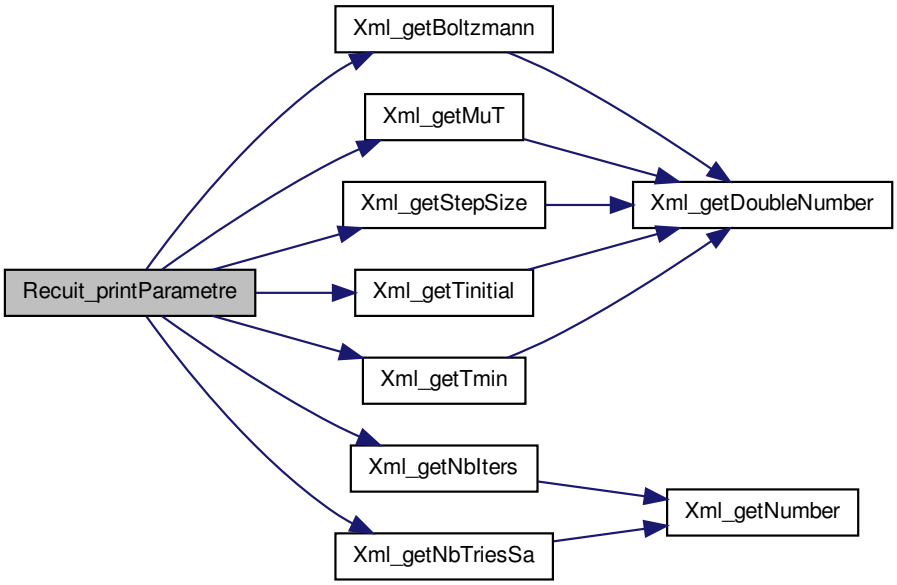
x_initial	table of reaction parameters

Definition at line 279 of file gsl_recuit.c.

References ListParameters::conf, ListParameters::interest_parameters, Xml_getBoltzmann(), Xml_getMuT(), Xml_getNbIters(), Xml_getNbTriesSa(), Xml_getStepSize(), Xml_getTinitial(), and Xml_getTmin().

Referenced by Recuit_compute_recuit().

Here is the call graph for this function:



4.16.2.6 void Recuit_printPosition (void * *xp*)

Print the table of reaction parameters.

Author

Amine Ghozlane

Parameters

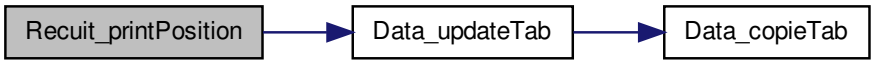
xp	Short table of reaction parameters

Definition at line 158 of file gsl_recuit.c.

References Data_updateTab(), ListParameters::nb_parameters, and SimParameters::y.

Referenced by Recuit_compute_recuit().

Here is the call graph for this function:



4.16.2.7 FILE * Recuit_redirectionFlux (xmlConfig_t * *conf*, FILE * *f*, char ** *files_path*, int *number*)

Redirect stdout flux.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t
f	File name
files_path	List of paths
number	Index of out repertory

Returns

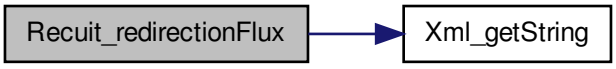
Out flux

Definition at line 184 of file gsl_recuit.c.

References Xml_getString().

Referenced by Recuit_compute_recuit().

Here is the call graph for this function:



4.16.2.8 void Recuit_takeStep (const gsl_rng * r, void * xp, double step_size)

Take a step through the solution space TSP.

Author

Amine Ghozlane

Parameters

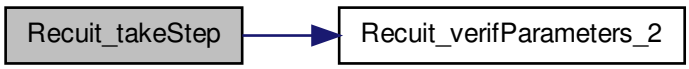
r	Random number generator
xp	Short table of reaction parameters
step_size	Size of step

Definition at line 136 of file gsl_recuit.c.

References ListParameters::interest_parameters, ListParameters::nb_couples, ListParameters::parameters, and Recuit_verifParameters_2().

Referenced by Recuit_compute_recuit().

Here is the call graph for this function:



4.16.2.9 void Recuit_verifParameters (double * x_initial, const gsl_rng * r, int debut, int max)

Choice of parameters and verification of their value.

Author

Amine Ghozlane

Parameters

x_initial	table of reaction parameters
r	Random number generator
debut	Beginning
max	End

Definition at line 207 of file gsl_recuit.c.

Referenced by Recuit_defParametre().

4.16.2.10 void Recuit_verifParameters_2 (double * *new_x*, double * *old_x*, const gsl_rng * *r*, int *debut*, int *max*, double *step_size*)

Choice of parameters and verification of their value.

Author

Amine Ghozlane

Parameters

new_x	table of reaction parameters
old_x	table of reaction parameters
r	Random number generator
debut	Beginning
max	End
step_size	Size of the step

Definition at line 235 of file gsl_recuit.c.

Referenced by Recuit_takeStep().

4.17 metaboflux/src/gsl_sd.c File Reference

Compute the standard deviation analysis of the simulations.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
#include <string.h>
#include <assert.h>
#include <gsl/gsl_math.h>
#include <gsl/gsl_rng.h>
#include <gsl/gsl_siman.h>
#include <gsl/gsl_ieee_utils.h>
#include <gsl/gsl_statistics_double.h>
#include <sbml/SBMLTypes.h>
#include <libxml/parser.h>
#include <libxml/xpath.h>
#include "especes.h"
#include "equations.h"
```

```
#include "simulation.h"
#include "xml_parameter.h"
#include "data_parameters.h"
#include "gsl_sd.h"
```

Include dependency graph for gsl_sd.c:

Functions

- double Sd_compute_simulation (pListParameters all, pSimParameters sim)

  Enter in the program for standard deviation.
- void Sd_compute_standard_deviation (pListParameters a, double ∗fluxRatio, double ∗result_tab, char ∗∗files_path, int number_arg, int debug)

  Compute the standard deviation.

4.17.1 Detailed Description

Compute the standard deviation analysis of the simulations. This file is part of MetaBoFlux (http://www.cbib.u-bordeaux2.fr/metaboflux/) Copyright (C) 2010 Amine Ghozlane from LaBRI and University of Bordeaux 1

MetaBoFlux is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

MetaBoFlux is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Author

{Amine Ghozlane}

Version

2.0

Date

9 novembre 2009

Definition in file gsl_sd.c.

4.17.2 Function Documentation

4.17.2.1 double Sd_compute_simulation (pListParameters *all*, pSimParameters *sim*)

Enter in the program for standard deviation.

Author

Amine Ghozlane

Parameters

all	Global parameters : struct ListParameters
sim	Simulation parameters : struct SimParameters

Returns

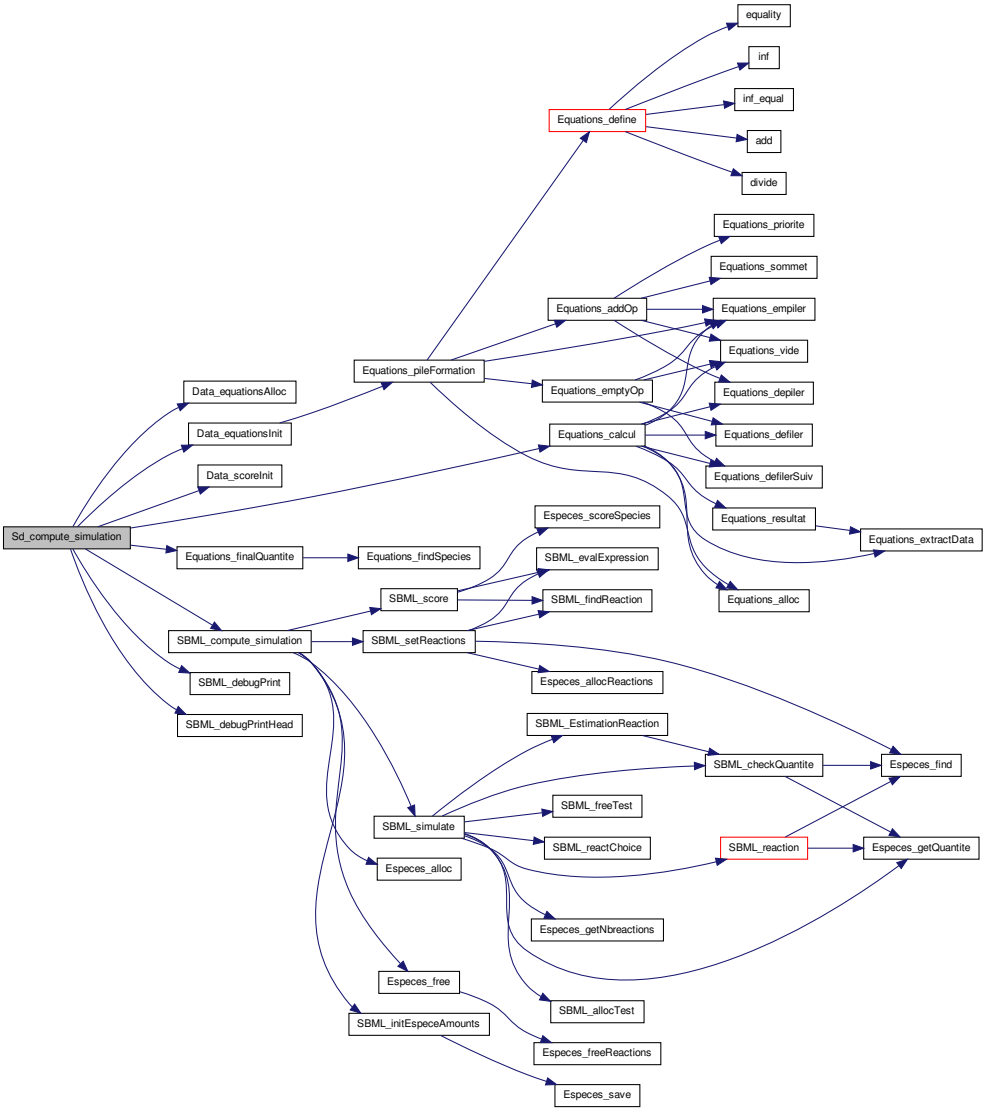
Energy value

Definition at line 56 of file gsl_sd.c.

References ListParameters::banned, Data_equationsAlloc(), Data_equationsInit(), Data_scoreInit(), SimParameters::debugFile, Equations_calcul(), Equations_finalQuantite(), ListParameters::model, Score::name, ListParameters::nb_banned, ListParameters::nb_equations, Score::nb_species, SimParameters::out, SimParameters::pile, Score::quantite, SimParameters::r, SBML_compute_simulation(), SBML_debugPrint(), SBML_debugPrintHead(), Score::species, Score::species_amount, Score::species_weight, Score::taille, Score::tailleSpecies, and SimParameters::y.

Referenced by Sd_compute_standard_deviation().

Here is the call graph for this function:



4.17.2.2 void Sd_compute_standard_deviation (pListParameters *a*, double * *fluxRatio*, double * *result_tab*, char ** *files_path*, int *number_arg*, int *debug*)

Compute the standard deviation.

Author

Amine Ghozlane

Parameters

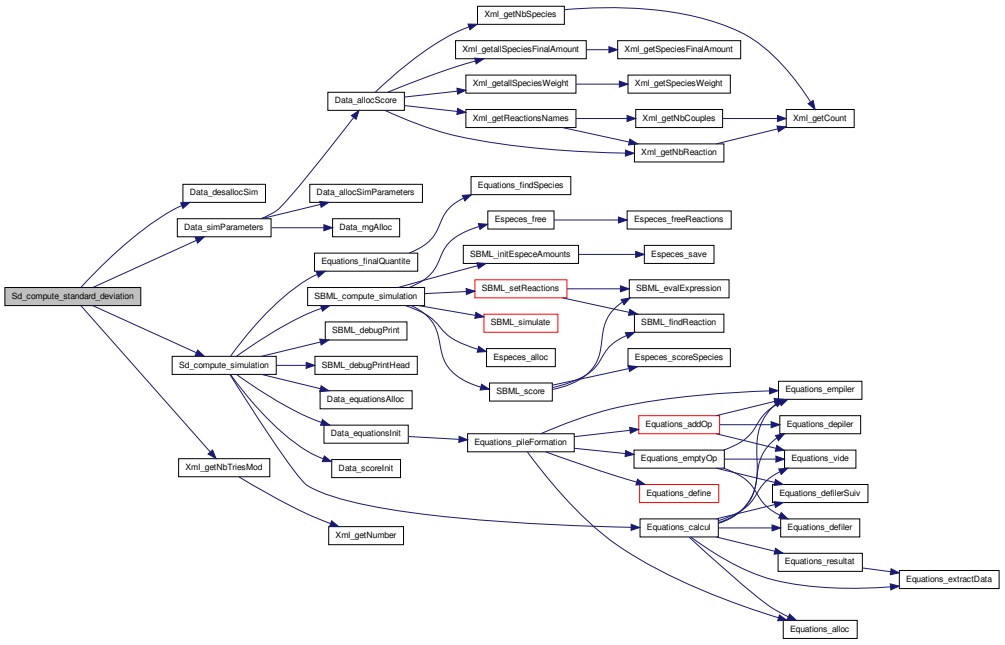
a	struct ListParameters
fluxRatio	Ratio parameters
result_tab	Result table
files_path	List of paths
number_arg	Number of simulation
debug	Debug flag

Definition at line 106 of file gsl_sd.c.

References ListParameters::conf, Data_desallocSim(), Data_simParameters(), ListParameters::nb_-parameters, Sd_compute_simulation(), Xml_getNbTriesMod(), and SimParameters::y.

Referenced by Mpi_slave().

Here is the call graph for this function:



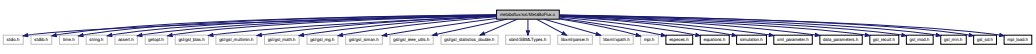
4.18 metaboflux/src/MetaBoFlux.c File Reference

Program for simulated annealing analysis.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
```

```
#include <string.h>
#include <assert.h>
#include <getopt.h>
#include <gsl/gsl_blas.h>
#include <gsl/gsl_multimin.h>
#include <gsl/gsl_math.h>
#include <gsl/gsl_rng.h>
#include <gsl/gsl_siman.h>
#include <gsl/gsl_ieee_utils.h>
#include <gsl/gsl_statistics_double.h>
#include <sbml/SBMLTypes.h>
#include <libxml/parser.h>
#include <libxml/xpath.h>
#include "mpi.h"
#include "especes.h"
#include "equations.h"
#include "simulation.h"
#include "xml_parameter.h"
#include "data_parameters.h"
#include "gsl_recuit.h"
#include "gsl_mod.h"
#include "gsl_min.h"
#include "gsl_sd.h"
#include "mpi_load.h"
```

Include dependency graph for MetaBoFlux.c:



Data Structures

- struct ListArguments

  List of arguments.

Functions

- pListArguments alloc_arguments (void)

  Alloc memory for the struct pListArguments.
- void free_arguments (pListArguments args)

  Free the struct pListArguments.
- void help_print (void)

  Print the usage help.
- void error_activity (int activity)

  Print the error notification.
- void argument_analysis (int argc, char **argv, pListArguments args)

  Print the error notification.
- void check_common (pListArguments args)

  Check obligatory arguments.
- void check_arguments (pListArguments args)

  Check program arguments.
- int main (int argc, char **argv)

  Enter in the program for simulated annealing.

4.18.1 Detailed Description

Program for simulated annealing analysis. This file is part of MetaBoFlux (http://www.cbib.u-bordeaux2.fr/meta
Copyright (C) 2010 Amine Ghozlane from LaBRI and University of Bordeaux 1

MetaBoFlux is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

MetaBoFlux is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Author

{Amine Ghozlane}

Version

3.0

Date

9 novembre 2009

Definition in file MetaBoFlux.c.

4.18.2 Function Documentation

4.18.2.1 pListArguments alloc_arguments (void)

Alloc memory for the struct pListArguments.

Author

Amine Ghozlane

Definition at line 103 of file MetaBoFlux.c.

References ListArguments::activity, ListArguments::debug, ListArguments::files_path, ListArguments::group, and ListArguments::port.

Referenced by main().

4.18.2.2 void argument_analysis (int *argc*, char ** *argv*, pListArguments *args*)

Print the error notification.

Author

Amine Ghozlane

Parameters

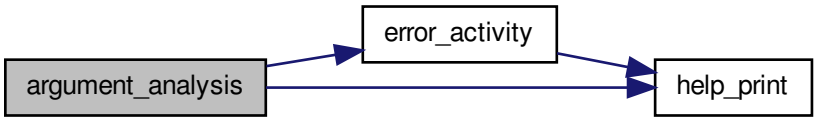
argc	Number of arguments
argv	List of arguments
args	struct ListArguments

Definition at line 185 of file MetaBoFlux.c.

References ListArguments::activity, ListArguments::debug, error_activity(), ListArguments::files_path, ListArguments::group, help_print(), and ListArguments::port.

Referenced by main().

Here is the call graph for this function:



4.18.2.3 void check_arguments (pListArguments *args*)

Check program arguments.

Author

Amine Ghozlane

Parameters

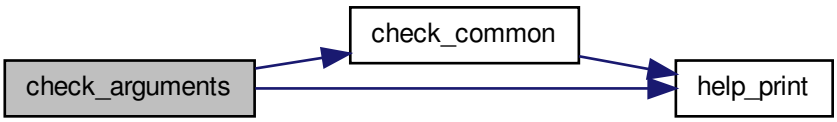
args	struct ListArguments

Definition at line 361 of file MetaBoFlux.c.

References ListArguments::activity, check_common(), ListArguments::files_path, and help_print().

Referenced by main().

Here is the call graph for this function:



4.18.2.4 void check_common (pListArguments *args*)

Check obligatory arguments.

Author

Amine Ghozlane

Parameters

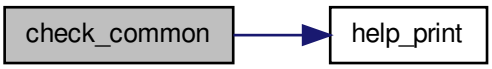
args	struct ListArguments

Definition at line 332 of file MetaBoFlux.c.

References ListArguments::files_path, and help_print().

Referenced by check_arguments().

Here is the call graph for this function:



4.18.2.5 void error_activity (int *activity*)

Print the error notification.

Author

Amine Ghozlane

Parameters

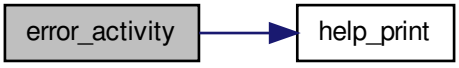
activity	Activity chosen

Definition at line 170 of file MetaBoFlux.c.

References help_print().

Referenced by argument_analysis().

Here is the call graph for this function:



4.18.2.6 void free_arguments (pListArguments *arguments*)

Free the struct pListArguments.

Author

Amine Ghozlane

Parameters

arguments	struct pListArguments

Definition at line 131 of file MetaBoFlux.c.

References ListArguments::files_path.

Referenced by main().

4.18.2.7 void help_print (void)

Print the usage help.

Author

Amine Ghozlane

Definition at line 142 of file MetaBoFlux.c.

Referenced by argument_analysis(), check_arguments(), check_common(), and error_activity().

4.18.2.8 int main (int *argc*, char ** *argv*)

Enter in the program for simulated annealing.

Author

Amine Ghozlane

Parameters

argc	Number of arguments
argv	List of arguments

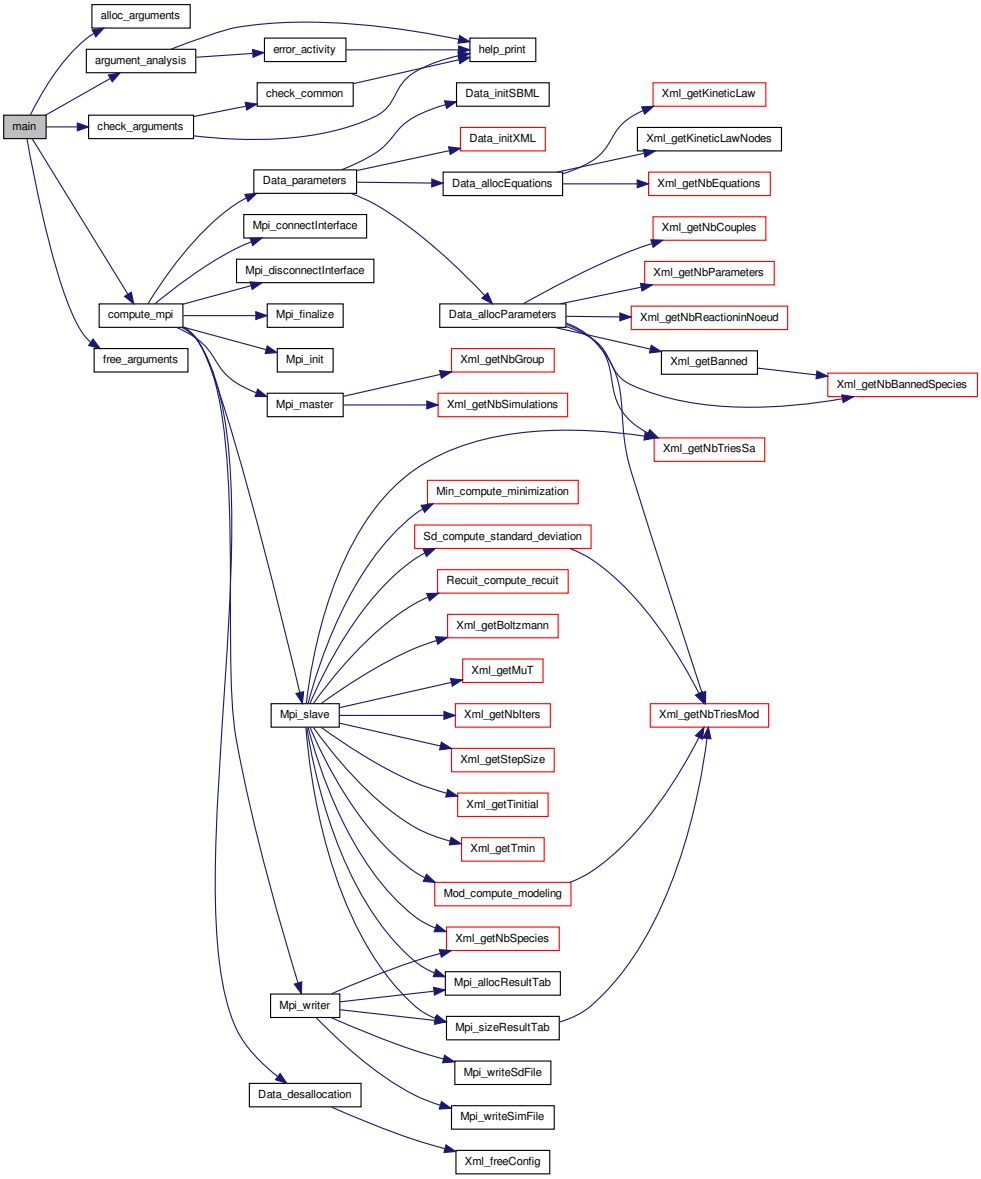
Returns

EXIT_SUCCESS Normal stop of the program

Definition at line 399 of file MetaBoFlux.c.

References ListArguments::activity, alloc_arguments(), argument_analysis(), check_arguments(), compute_mpi(), ListArguments::debug, ListArguments::files_path, free_arguments(), ListArguments::group, and ListArguments::port.

Here is the call graph for this function:



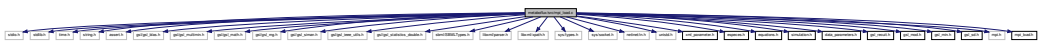
4.19 metaboflux/src/mpi_load.c File Reference

Parallelize the program.

```
#include <stdio.h>
```

```
#include <stdlib.h>
#include <time.h>
#include <string.h>
#include <assert.h>
#include <gsl/gsl_blas.h>
#include <gsl/gsl_multimin.h>
#include <gsl/gsl_math.h>
#include <gsl/gsl_rng.h>
#include <gsl/gsl_siman.h>
#include <gsl/gsl_ieee_utils.h>
#include <gsl/gsl_statistics_double.h>
#include <sbml/SBMLTypes.h>
#include <libxml/parser.h>
#include <libxml/xpath.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <unistd.h>
#include "xml_parameter.h"
#include "especes.h"
#include "equations.h"
#include "simulation.h"
#include "data_parameters.h"
#include "gsl_recuit.h"
#include "gsl_mod.h"
#include "gsl_min.h"
#include "gsl_sd.h"
#include "mpi.h"
#include "mpi_load.h"
```

Include dependency graph for mpi_load.c:



Functions

- int Mpi_connectInterface (int port)

  Enter in the program for standard deviation.
- void Mpi_disconnectInterface (int desc)

  Disconnection to the interface.
- void Mpi_init (int argc, char **argv, int *tab)

  Enter in the program for standard deviation.
- void Mpi_master (pListParameters allone, char **files_path, int activity, int group, int myid, int numprocs, int desc)

  Master process.
- void Mpi_writer (pListParameters allone, char **files_path, int activity, int group, int myid)

  Master process.
- int Mpi_sizeResultTab (pListParameters allone, int activity, int group, int nb_species)

  Determine the size of the result tab.
- double * Mpi_allocResultTab (int tailleTab)

  Allocate the result tab.
- void Mpi_writeSimFile (pListParameters allone, FILE *out, FILE *logOut, double *result_tab, int group, int tailleSpecies, int nb_species)

  Determine the size of the result tab.
- void Mpi_writeSdFile (FILE *out, double *result_tab, int tailleTab)

  Determine the size of the result tab.
- void Mpi_slave (pListParameters allone, char **files_path, int activity, int group, int debug)
- void Mpi_finalize (void)

  End MPI execution environment.
- void compute_mpi (int argc, char **argv, char **files_path, int activity, int group, int debug, int port)

  Compute the simulated annealing through mpi.

4.19.1 Detailed Description

Parallelize the program. This file is part of MetaBoFlux (http://www.cbib.u-bordeaux2.fr/metabof
Copyright (C) 2010 Amine Ghozlane from LaBRI and University of Bordeaux 1

MetaBoFlux is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

MetaBoFlux is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Author

{Amine Ghozlane}

Version

2.0

Date

9 novembre 2009

Definition in file mpi_load.c.

4.19.2 Function Documentation

4.19.2.1 void compute_mpi (int *argc*, char ** *argv*, char ** *files_path*, int *activity*, int *group*, int *debug*, int *port*)

Compute the simulated annealing through mpi.

Author

Amine Ghozlane

Parameters

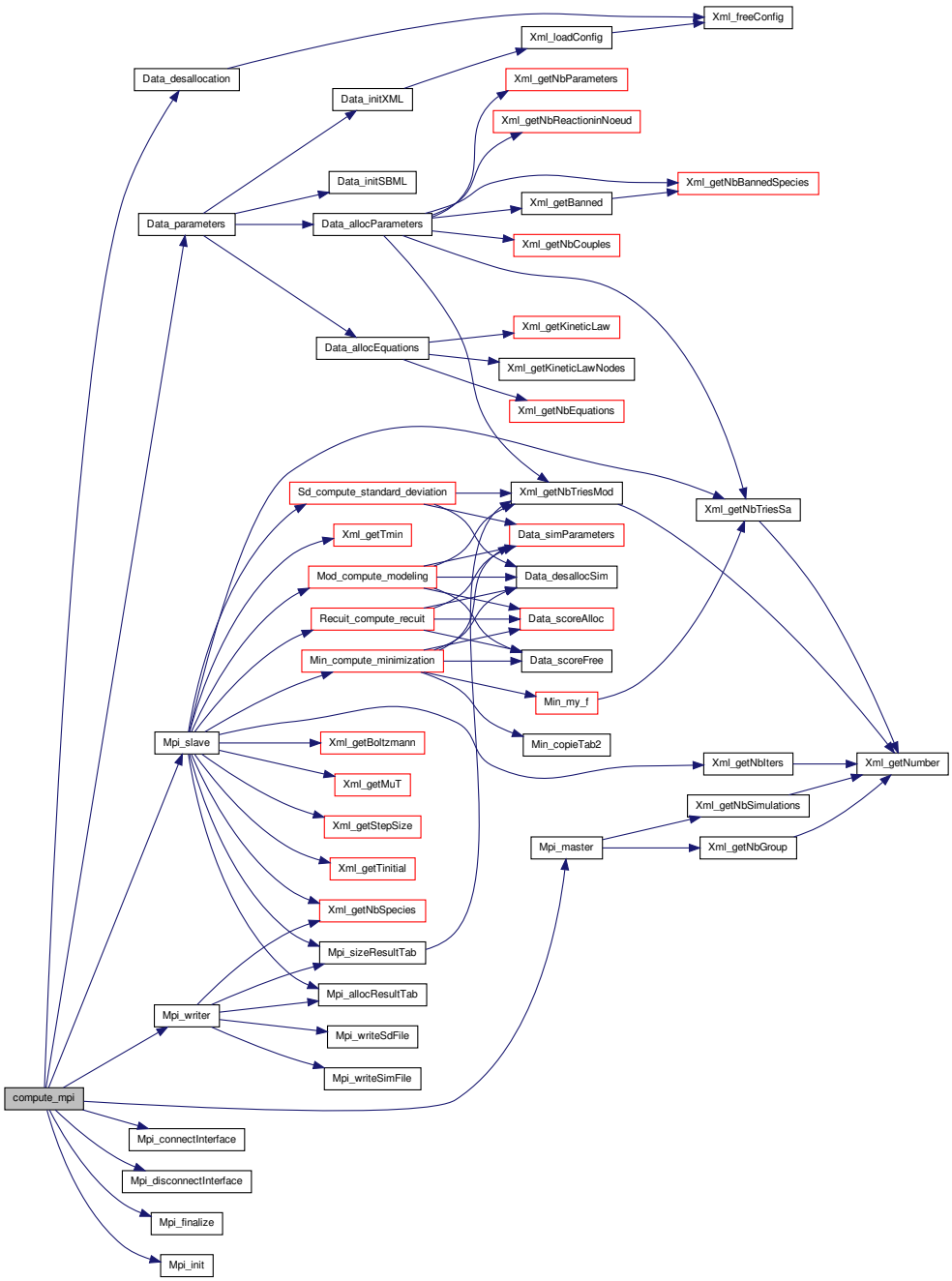
argc	Number of arguments
argv	List of arguments
files_path	List of paths
activity	Chosen activity
group	Group flag
debug	Debug flag
port	Interface port

Definition at line 546 of file mpi_load.c.

References Data_desallocation(), Data_parameters(), Mpi_connectInterface(), Mpi_disconnectInterface(), Mpi_finalize(), Mpi_init(), Mpi_master(), Mpi_slave(), and Mpi_writer().

Referenced by main().

Here is the call graph for this function:



4.19.2.2 double * Mpi_allocResultTab (int *tailleTab*)

Allocate the result tab.

Author

Amine Ghozlane

Parameters

tailleTab	Size of Result tab

Returns

Adress of the allocated space

Definition at line 355 of file mpi_load.c.

Referenced by Mpi_slave(), and Mpi_writer().

4.19.2.3 int Mpi_connectInterface (int *port*)

Enter in the program for standard deviation.

Author

Amine Ghozlane

Parameters

port	Connection port

Returns

Socket

Definition at line 69 of file mpi_load.c.

Referenced by compute_mpi().

4.19.2.4 void Mpi_disconnectInterface (int *desc*)

Disconnection to the interface.

Author

Amine Ghozlane

Parameters

desc	Socket

Definition at line 109 of file mpi_load.c.

Referenced by compute_mpi().

4.19.2.5 void Mpi_finalize (void)

End MPI execution environment.

Author

Amine Ghozlane

Definition at line 527 of file mpi_load.c.

Referenced by compute_mpi().

4.19.2.6 void Mpi_init (int *argc*, char ** *argv*, int * *tab*)

Enter in the program for standard deviation.

Author

Amine Ghozlane

Parameters

argc	Number of arguments
argv	List of arguments
tab	Table

Definition at line 123 of file mpi_load.c.

Referenced by compute_mpi().

4.19.2.7 void Mpi_master (pListParameters *allone*, char ** *files_path*, int *activity*, int *group*, int *myid*, int *numprocs*, int *desc*)

Master process.

Author

Amine Ghozlane

Parameters

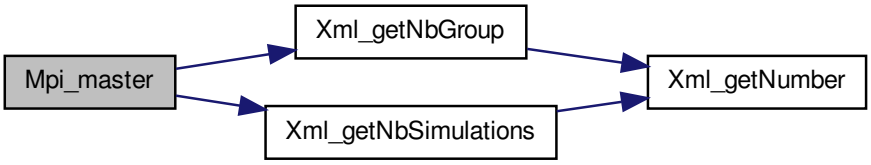
allone	Global parameters : struct ListParameters
files_path	List of paths
activity	Chosen activity
group	Group flag
myid	Id of the thread
numprocs	Number of thread
desc	Socket

Definition at line 154 of file mpi_load.c.

References ListParameters::conf, ListParameters::nb_parameters, Xml_getNbGroup(), and Xml_getNbSimulations().

Referenced by compute_mpi().

Here is the call graph for this function:



4.19.2.8 int Mpi_sizeResultTab (pListParameters *allone*, int *activity*, int *group*, int *nb_species*)

Determine the size of the result tab.

Author

Amine Ghozlane

Parameters

allone	Global parameters : struct ListParameters
activity	Chosen activity
group	Group flag
nb_species	Number of interest species

Returns

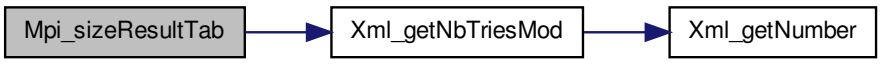
Size of Result tab

Definition at line 334 of file mpi_load.c.

References ListParameters::conf, ListParameters::model, ListParameters::nb_parameters, and Xml_getNbTriesMod().

Referenced by Mpi_slave(), and Mpi_writer().

Here is the call graph for this function:



4.19.2.9 void Mpi_slave (pListParameters *allone*, char ** *files_path*, int *activity*, int *group*, int *debug*)

Author

Amine Ghozlane

Parameters

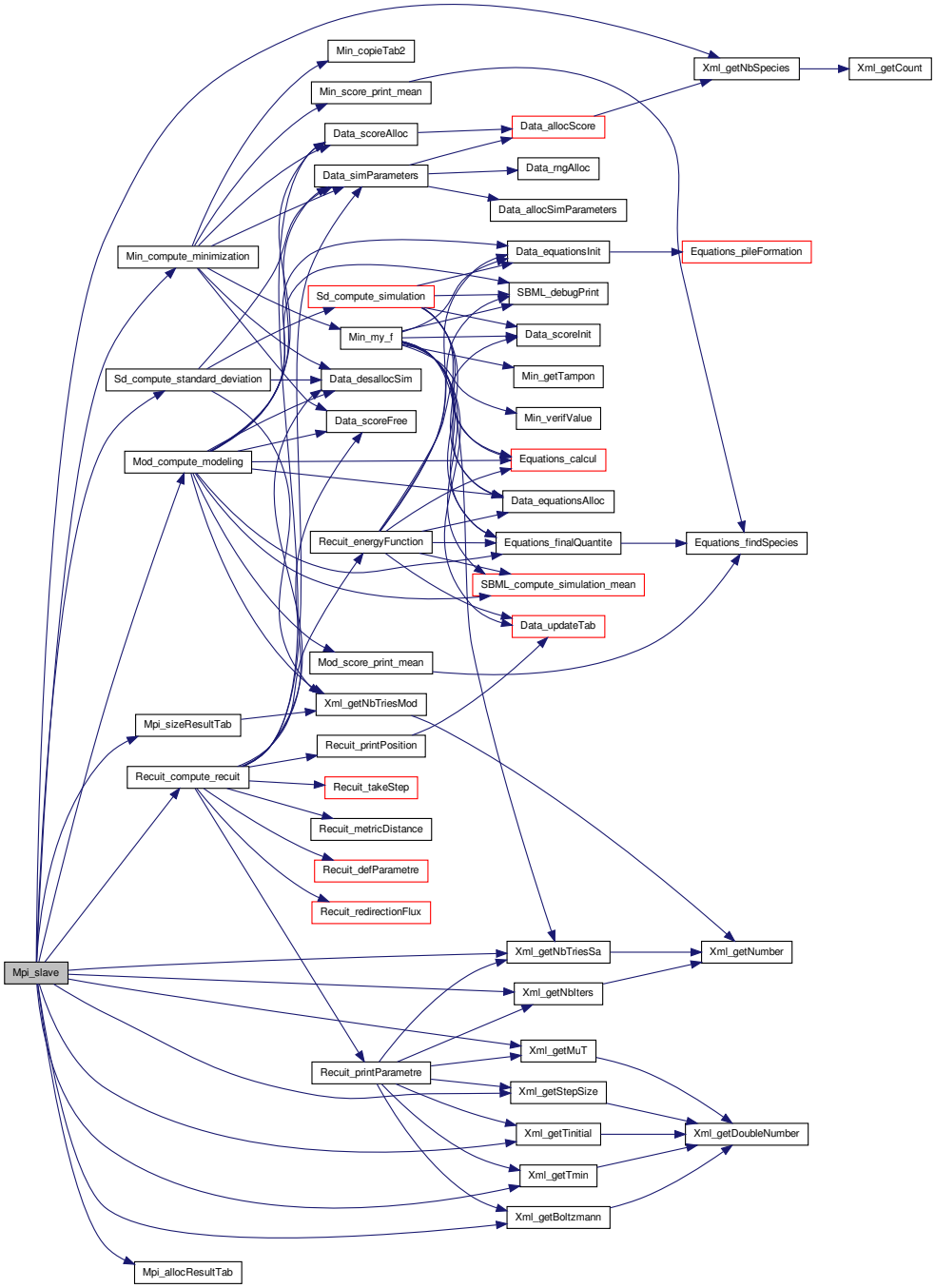
allone	Global parameters : struct ListParameters
files_path	List of paths
activity	Activity chosen
group	Group flag
debug	Debug flag

Definition at line 440 of file mpi_load.c.

References ListParameters::conf, Min_compute_minimization(), Mod_compute_modeling(), Mpi_allocResultTab(), Mpi_sizeResultTab(), ListParameters::nb_parameters, Recuit_compute_recuit(), Sd_compute_standard_deviation(), Xml_getBoltzmann(), Xml_getMuT(), Xml_getNbIters(), Xml_getNbSpecies(), Xml_getNbTriesSa(), Xml_getStepSize(), Xml_getTinitial(), and Xml_getTmin().

Referenced by compute_mpi().

Here is the call graph for this function:



4.19.2.10 void Mpi_writer (pListParameters *allone*, char ** *files_path*, int *activity*, int *group*, int *myid*)

Master process.

Author

Amine Ghozlane

Parameters

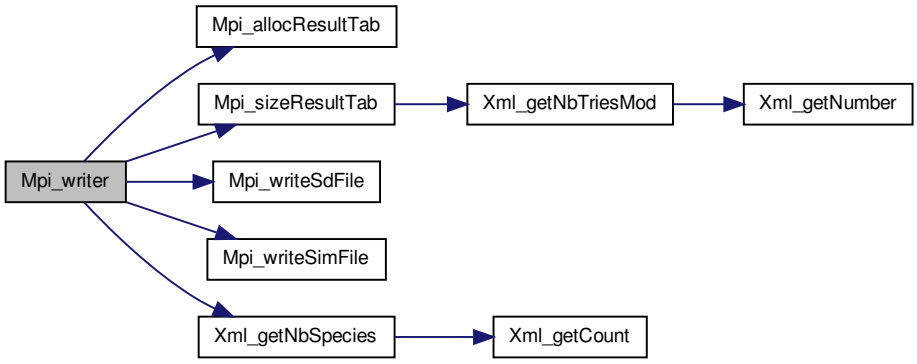
allone	Global parameters : struct ListParameters
files_path	List of paths
activity	Chosen activity
group	Group flag
myid	Id of the thread

Definition at line 273 of file mpi_load.c.

References ListParameters::conf, ListParameters::model, Mpi_allocResultTab(), Mpi_sizeResultTab(), Mpi_writeSdFile(), Mpi_writeSimFile(), and Xml_getNbSpecies().

Referenced by compute_mpi().

Here is the call graph for this function:



4.19.2.11 void Mpi_writeSdFile (FILE * *out*, double * *result_tab*, int *tailleTab*)

Determine the size of the result tab.

Author

Amine Ghozlane

Parameters

out	Result file
result_tab	Result table
tailleTab	Size of result table

Definition at line 421 of file mpi_load.c.

Referenced by Mpi_writer().

4.19.2.12 void Mpi_writeSimFile (pListParameters *allone*, FILE * *out*, FILE * *logOut*, double * *result_tab*, int *group*, int *tailleSpecies*, int *nb_species*)

Determine the size of the result tab.

Author

Amine Ghozlane

Parameters

allone	Global parameters : struct ListParameters
out	Result file
logOut	Log file
result_tab	Result table
group	Group flag
tailleSpecies	Number of species
nb_species	Number of interest species

Definition at line 375 of file mpi_load.c.

References ListParameters::nb_parameters.

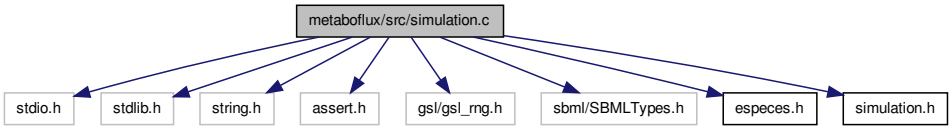
Referenced by Mpi_writer().

4.20 metaboflux/src/simulation.c File Reference

Simulate a petri net.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <assert.h>
#include <gsl/gsl_rng.h>
#include <sbml/SBMLTypes.h>
#include "especes.h"
#include "simulation.h"
```

Include dependency graph for simulation.c:



Functions

- void SBML_initEspeceAmounts (Model_t ∗mod, pEspeces molecules, int nbEspeces)

  Alloc memory and initialize the struct Especes.
- int SBML_findReaction (char ∗∗reaction, const char ∗react, int nb_reaction)

  Determine if the reaction is study.
- double SBML_evalExpression (const char ∗formule)

  Get the reaction ratio define in the sbml.
- void SBML_setReactions (Model_t ∗mod, pEspeces molecules, pScore result, double ∗reactions_ratio, int nbReactions, int nbEspeces)

  Alloc memory and initialize the struct Especes.
- int SBML_checkQuantite (Model_t ∗mod, Reaction_t ∗react, int nbEspeces, pEspeces molecules)

  Determine the number of reaction for one molecule.
- Reaction_t ∗ SBML_reactChoice (pEspeces molecules, const gsl_rng ∗r, int ref)

  Determine randomly the reaction to achieve for several nodes reactions.
- void SBML_reaction (Model_t ∗mod, pEspeces molecules, Reaction_t ∗react, int nbEspeces)

  Simulation of a discrete transision.
- void SBML_allocTest (pTestReaction T, int nbReactions)

  Alloc memory and initialize the struct pTestReaction.
- void SBML_freeTest (pTestReaction T)

  Free memory of the struct TestReaction.
- int SBML_EstimationReaction (Model_t ∗mod, pTestReaction T, pEspeces molecules, int ref, int nbEspeces)

  Alloc memory and initialize the struct Especes.
- int SBML_simulate (Model_t ∗mod, pEspeces molecules, const gsl_rng ∗r, pTestReaction T, char ∗∗banned, int nbBanned, int nbEspeces, int ref)

  Simulate one step of petri net.
- void SBML_score (Model_t ∗mod, pEspeces molecules, pScore result, double ∗reactions_ratio, int nbReactions, int nbEspeces)

Alloc memory and initialize the struct Especes.

- void SBML_debugPrintHead (FILE *debugFile, int taille, char **name)

  Print the head of the debug file.

- void SBML_debugPrint (FILE *debugFile, int tailleSpecies, int taille, double *quantite, double result)

  Print the debuf file.

- void SBML_compute_simulation (pScore result, Model_t *mod, double *reactions_ratio, gsl_rng *r, char **banned, int nbBanned)

  Simulation of metabolic network.

- void SBML_score_add (pScore result, pScore result_temp, FILE *debugFile)

  Add scores.

- void SBML_score_mean (pScore result, int n)

  Mean quantities for score.

- void SBML_compute_simulation_mean (FILE *debugFile, pScore result, pScore result_temp, Model_t *mod, double *reactions_ratio, gsl_rng *r, char **banned, int nbBanned, int nb_simulation)

  X time simulation of metabolic network.

4.20.1 Detailed Description

Simulate a petri net. This file is part of MetaBoFlux (http://www.cbib.u-bordeaux2.fr/metaboflux/)
Copyright (C) 2010 Amine Ghozlane from LaBRI and University of Bordeaux 1

MetaBoFlux is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

MetaBoFlux is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Author

{Amine Ghozlane}

Version

2.0

Date

27 octobre 2009

Definition in file simulation.c.

4.20.2 Function Documentation

4.20.2.1 void SBML_allocTest (pTestReaction *T,* int *nbReactions*)

Alloc memory and initialize the struct pTestReaction.

Author

Amine Ghozlane

Parameters

T	Empty struct TestReaction
nbReactions	Number of reactions

Definition at line 279 of file simulation.c.

References TestReaction::minStepTab, and TestReaction::tabReactions.

Referenced by SBML_simulate().

4.20.2.2 int SBML_checkQuantite (Model_t * *mod,* Reaction_t * *react,* int *nbEspeces,* pEspeces *molecules*)

Determine the number of reaction for one molecule.

Author

Amine Ghozlane

Parameters

mod	Model of the SBML file
react	Reaction id
nbEspeces	Number of molecules
molecules	Struct Especes

Returns

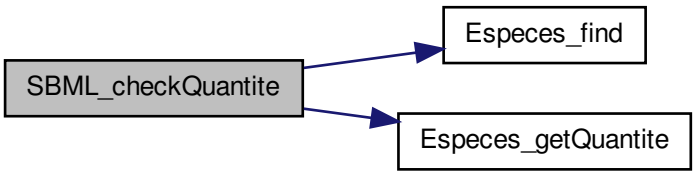
Number of reaction for one molecule

Definition at line 158 of file simulation.c.

References Especes_find(), and Especes_getQuantite().

Referenced by SBML_EstimationReaction(), and SBML_simulate().

Here is the call graph for this function:



4.20.2.3 void SBML_compute_simulation (pScore *result*, Model_t * *mod*, double * *reactions_ratio*, gsl_rng * *r*, char ** *banned*, int *nbBanned*)

Simulation of metabolic network.

Author

Amine Ghozlane

Parameters

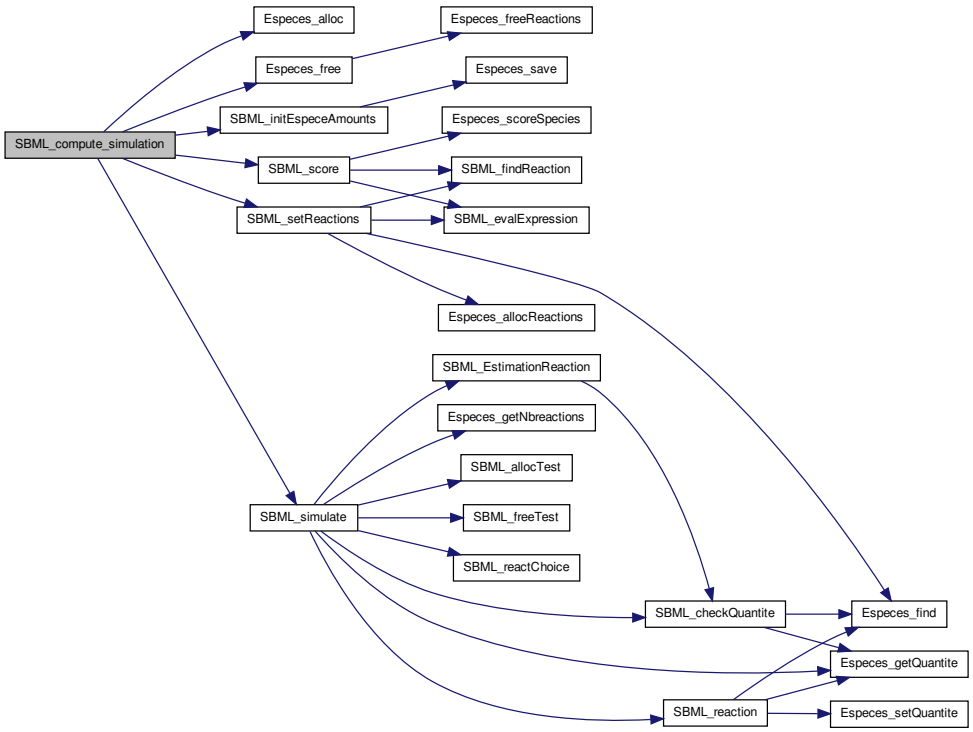
result	Struct Score
mod	Model of the SBML file
reactions_ratio	List of computed reaction ratio
r	Random number generator
banned	List of banned compound
nbBanned	Number of banned compound

Definition at line 515 of file simulation.c.

References Especes_alloc(), Especes_free(), SBML_initEspeceAmounts(), SBML_score(), SBML_setReactions(), SBML_simulate(), Score::tailleReactions, and Score::tailleSpecies.

Referenced by Sd_compute_simulation().

Here is the call graph for this function:



4.20.2.4 void SBML_compute_simulation_mean (FILE * *debugFile*, pScore *result*, pScore *result_temp*, Model_t * *mod*, double * *reactions_ratio*, gsl_rng * *r*, char ** *banned*, int *nbBanned*, int *nb_simulation*)

X time simulation of metabolic network.

Author

Amine Ghozlane

Parameters

debugFile	File use for debug
result	Struct Score used for all the simulation
result_temp	Struct Score used at each simulation step
mod	Model of the SBML file
reactions_ratio	List of computed reaction ratio
r	Random number generator

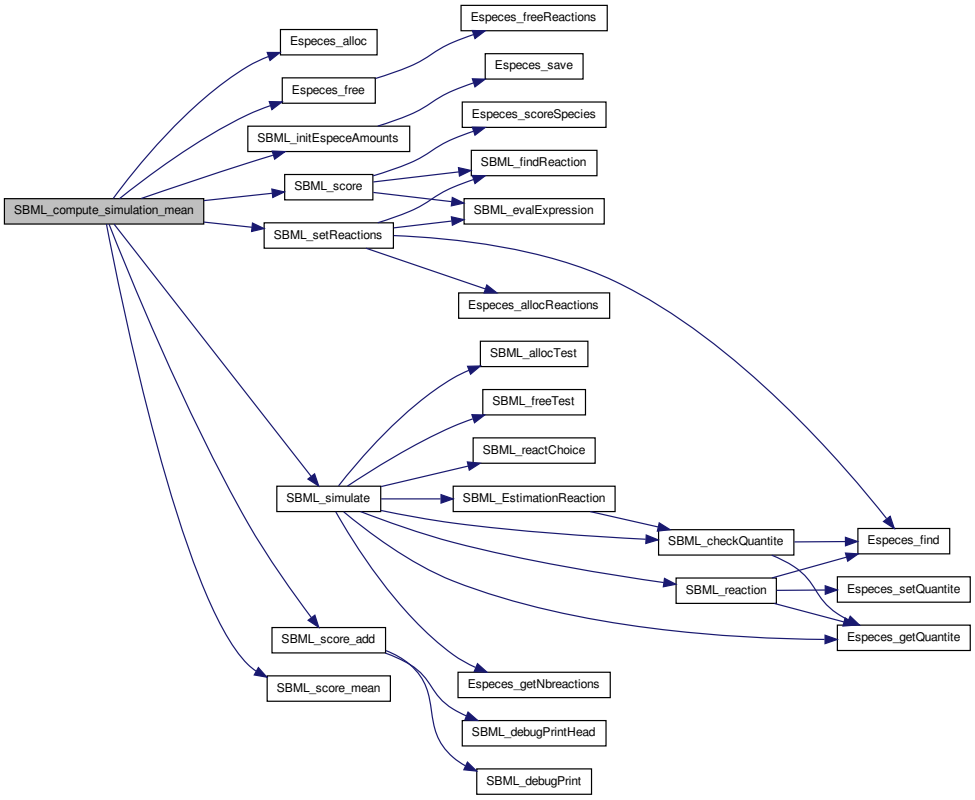
banned	List of banned compound
nbBanned	Number of banned compound
nb_simulation	Number of simulation step

Definition at line 614 of file simulation.c.

References Especes_alloc(), Especes_free(), SBML_initEspeceAmounts(), SBML_score(), SBML_score_add(), SBML_score_mean(), SBML_setReactions(), and SBML_simulate().

Referenced by Min_my_f(), Mod_compute_modeling(), and Recuit_energyFunction().

Here is the call graph for this function:



4.20.2.5 void SBML_debugPrint (FILE * *debugFile,* int *tailleSpecies,* int *taille,* double * *quantite,* double *result*)

Print the debuf file.

Author

Amine Ghozlane

Parameters

debugFile	File use for debug
tailleSpecies	Number of molecules
taille	Number of molecules/reactions
quantite	Quantity of molecules/reactions
result	Energy value

Definition at line 486 of file simulation.c.

Referenced by Min_my_f(), Mod_compute_modeling(), Recuit_energyFunction(), SBML_score_add(), and Sd_compute_simulation().

4.20.2.6 void SBML_debugPrintHead (FILE * *debugFile*, int *taille*, char ** *name*)

Print the head of the debug file.

Author

Amine Ghozlane

Parameters

debugFile	File use for debug
taille	Number of molecules/reactions
name	List of molecules/reactions

Definition at line 465 of file simulation.c.

Referenced by SBML_score_add(), and Sd_compute_simulation().

4.20.2.7 int SBML_EstimationReaction (Model_t * *mod*, pTestReaction *T*, pEspeces *molecules*, int *ref*, int *nbEspeces*)

Alloc memory and initialize the struct Especes.

Author

Amine Ghozlane

Parameters

mod	Model of the SBML file
T	Struct TestReaction gives data on reaction
molecules	Struct Especes
ref	Number reference of one molecule
nbEspeces	Number of molecules

Returns

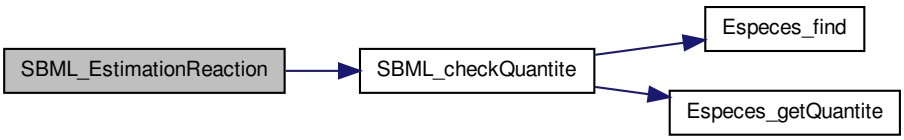
Estimated number of feasible step by reaction

Definition at line 322 of file simulation.c.

References Reaction::link, TestReaction::minStepTab, SBML_checkQuantite(), Reaction::suivant, Especes::system, and TestReaction::tabReactions.

Referenced by SBML_simulate().

Here is the call graph for this function:



4.20.2.8 double SBML_evalExpression (const char * *formule*)

Get the reaction ratio define in the sbml.

Author

Amine Ghozlane

Parameters

formule	Formule SBML

Returns

Return double value of the constraint

Definition at line 88 of file simulation.c.

Referenced by SBML_score(), and SBML_setReactions().

4.20.2.9 int SBML_findReaction (char ** *reaction,* const char * *react,* int *nb_reaction*)

Determine if the reaction is study.

Author

Amine Ghozlane

Parameters

reaction	List of reactions
react	Reaction of interest
nb_reaction	Number of reactions

Returns

Number of the molecules if it's study

Definition at line 69 of file simulation.c.

Referenced by SBML_score(), and SBML_setReactions().

4.20.2.10 void SBML_freeTest (pTestReaction *T*)

Free memory of the struct TestReaction.

Author

Amine Ghozlane

Parameters

T	Struct TestReaction gives data on reaction

Definition at line 304 of file simulation.c.

References TestReaction::minStepTab, and TestReaction::tabReactions.

Referenced by SBML_simulate().

4.20.2.11 void SBML_initEspeceAmounts (Model_t * *mod*, pEspeces *molecules*, int *nbEspeces*)

Alloc memory and initialize the struct Especes.

Author

Amine Ghozlane

Parameters

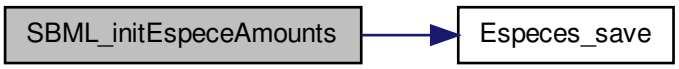
mod	Model of the SBML file
molecules	Struct Especes
nbEspeces	Number of molecules

Definition at line 46 of file simulation.c.

References Especes_save().

Referenced by SBML_compute_simulation(), and SBML_compute_simulation_mean().

Here is the call graph for this function:



4.20.2.12 Reaction_t * SBML_reactChoice (pEspeces *molecules*, const gsl_rng * *r*, int *ref*)

Determine randomly the reaction to achieve for several nodes reactions.

Author

Amine Ghozlane

Parameters

molecules	Struct Especes
r	Random number generator
ref	Number reference of one molecule

Returns

Id of the selected reaction

Definition at line 206 of file simulation.c.

References Reaction::link, Reaction::suivant, and Especes::system.

Referenced by SBML_simulate().

4.20.2.13 void SBML_reaction (Model_t * *mod*, pEspeces *molecules*, Reaction_t * *react*, int *nbEspeces*)

Simulation of a discrete transision.

Author

Amine Ghozlane

Parameters

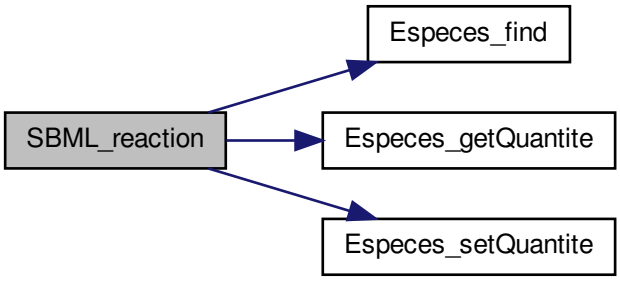
mod	Model of the SBML file
molecules	Struct Especes
react	Reaction id
nbEspeces	Number of molecules

Definition at line 244 of file simulation.c.

References Especes_find(), Especes_getQuantite(), and Especes_setQuantite().

Referenced by SBML_simulate().

Here is the call graph for this function:



4.20.2.14 void SBML_score (Model_t * *mod*, pEspeces *molecules*, pScore *result*, double * *reactions_ratio*, int *nbReactions*, int *nbEspeces*)

Alloc memory and initialize the struct Especes.

Author

Amine Ghozlane

Parameters

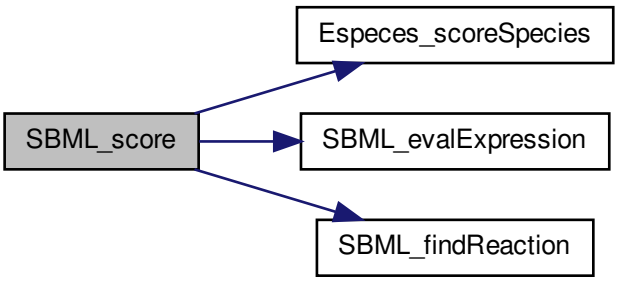
mod	Model of the SBML file
molecules	Struct Especes
result	Struct Score
reactions_ratio	List of computed reaction ratio
nbReactions	Number of reactions
nbEspeces	Number of molecules

Definition at line 420 of file simulation.c.

References Especes_scoreSpecies(), Score::name, Score::nb_reaction, Score::quantite, Score::reaction, SBML_evalExpression(), and SBML_findReaction().

Referenced by SBML_compute_simulation(), and SBML_compute_simulation_mean().

Here is the call graph for this function:



4.20.2.15 void SBML_score_add (pScore *result*, pScore *result_temp*, FILE * *debugFile*)

Add scores.

Author

Amine Ghozlane

Parameters

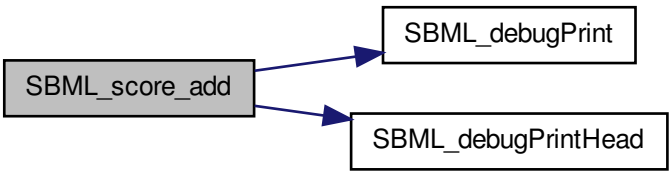
result	Struct Score used for all the simulation
result_temp	Struct Score used at each simulation step
debugFile	File use for debug

Definition at line 560 of file simulation.c.

References Score::name, Score::quantite, SBML_debugPrint(), SBML_debugPrintHead(), Score::taille, and Score::tailleSpecies.

Referenced by SBML_compute_simulation_mean().

Here is the call graph for this function:



4.20.2.16 void SBML_score_mean (pScore *result*, int *n*)

Mean quantities for score.

Author

Amine Ghozlane

Parameters

result	Struct Score
n	Number of simulation step

Definition at line 591 of file simulation.c.

References Score::quantite, and Score::taille.

Referenced by SBML_compute_simulation_mean().

4.20.2.17 void SBML_setReactions (Model_t * *mod*, pEspeces *molecules*, pScore *result*, double * *reactions_ratio*, int *nbReactions*, int *nbEspeces*)

Alloc memory and initialize the struct Especes.

Author

Amine Ghozlane

Parameters

mod	Model of the SBML file
molecules	Struct Especes
result	Struct Score
reactions_ratio	List of computed reaction ratio

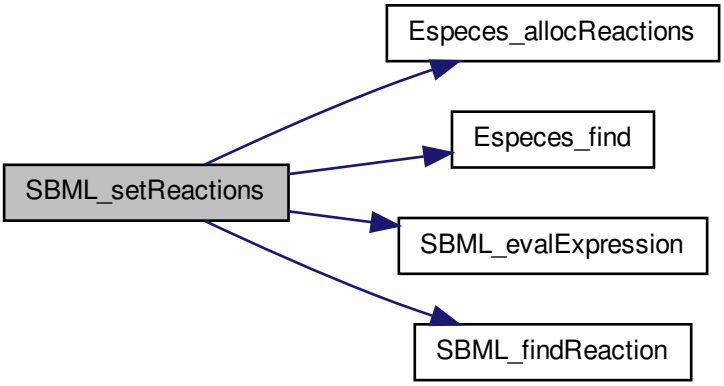
nbReactions	Number of reaction
nbEspeces	Number of molecules

Definition at line 105 of file simulation.c.

References Especes_allocReactions(), Especes_find(), Score::nb_reaction, Score::reaction, SBML_evalExpression(), and SBML_findReaction().

Referenced by SBML_compute_simulation(), and SBML_compute_simulation_mean().

Here is the call graph for this function:



4.20.2.18 int SBML_simulate (Model_t * *mod*, pEspeces *molecules*, const gsl_rng * *r*, pTestReaction *T*, char ** *banned*, int *nbBanned*, int *nbEspeces*, int *ref*)

Simulate one step of petri net.

Author

Amine Ghozlane

Parameters

mod	Model of the SBML file
molecules	Struct Especes
r	Random number generator
T	Struct TestReaction gives data on reaction
banned	List of banned compound
nbBanned	Number of banned compound

nbEspeces	Number of molecules
ref	Number reference of one molecule

Returns

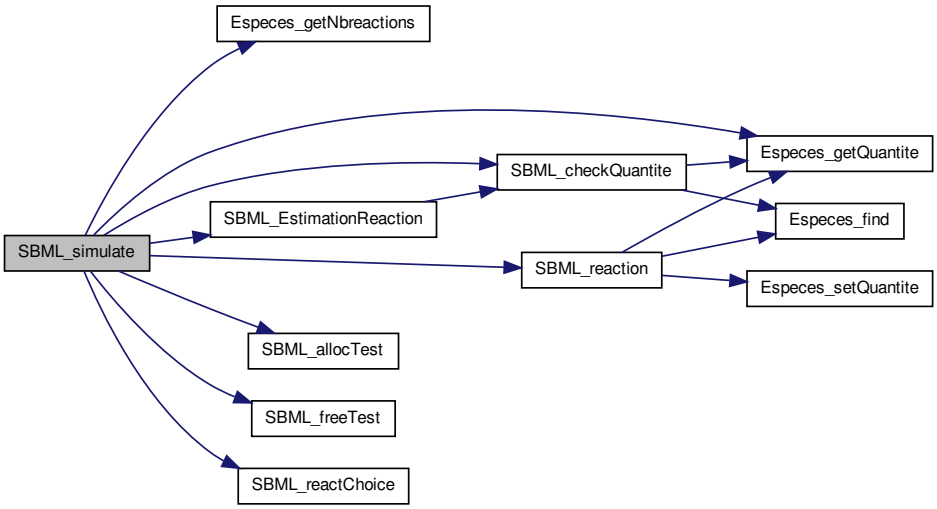
Condition of stop/pursue

Definition at line 356 of file simulation.c.

References Especes_getNbreactions(), Especes_getQuantite(), Reaction::link, SBML_allocTest(), SBML_checkQuantite(), SBML_EstimationReaction(), SBML_freeTest(), SBML_reactChoice(), SBML_reaction(), and Especes::system.

Referenced by SBML_compute_simulation(), and SBML_compute_simulation_mean().

Here is the call graph for this function:



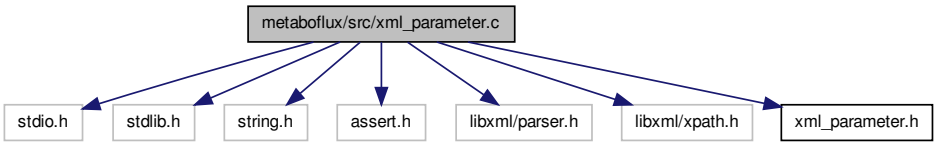
4.21 metaboflux/src/xml_parameter.c File Reference

Xml reader for parametre.xml.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <assert.h>
#include <libxml/parser.h>
```

```
#include <libxml/xpath.h>
#include "xml_parameter.h"
```

Include dependency graph for xml_parameter.c:



Functions

- xmlConfig_t * Xml_loadConfig (char *fichier)
  Initialize and load the parameter.xml.
- void Xml_freeConfig (xmlConfig_t *conf)
  Free the Struct xmlConfig_t.
- int Xml_getNumber (xmlConfig_t *conf, const char *requete)
  Get integer value.
- double Xml_getDoubleNumber (xmlConfig_t *conf, const char *requete)
  Get double value.
- char * Xml_getString (xmlConfig_t *conf)
  Get name value.
- int Xml_getNbSimulations (xmlConfig_t *conf)
  Get number of simulation.
- int Xml_getNbTriesMod (xmlConfig_t *conf)
  Get number of tries for modelling.
- int Xml_getNbTriesSa (xmlConfig_t *conf)
  Get number of tries for the simulated annealing and minimization.
- int Xml_getNbIters (xmlConfig_t *conf)
  Get number of iteration.
- int Xml_getNbGroup (xmlConfig_t *conf)
  Get number of group.
- double Xml_getStepSize (xmlConfig_t *conf)
  Get the step size.
- double Xml_getBoltzmann (xmlConfig_t *conf)
  Get the Boltzmann value.
- double Xml_getTinitial (xmlConfig_t *conf)
  Get the initial temperature.
- double Xml_getMuT (xmlConfig_t *conf)

Get the variation of temperature.

- double Xml_getTmin (xmlConfig_t *conf)

  Get the minimum temperature.
- int Xml_getCount (xmlConfig_t *conf, const char *requete)

  Get count.
- int Xml_getNbCouples (xmlConfig_t *conf)

  Get the number of couples.
- int Xml_getNbParameters (xmlConfig_t *conf)

  Get the number of parameters.
- int Xml_getNbReactioninNoeud (xmlConfig_t *conf, int noeud)

  Count the number of reaction in one node.
- int Xml_getNbReaction (xmlConfig_t *conf)

  Get the number of reactions.
- int Xml_getNbSpecies (xmlConfig_t *conf)

  Get the number of species.
- int Xml_getNbEquations (xmlConfig_t *conf)

  Get the number of equations.
- int Xml_getNbBannedSpecies (xmlConfig_t *conf)

  Get the number of banned species.
- char ** Xml_getReactionsNamesinNoeud (xmlConfig_t *conf, int numero)

  Get list of reactions.
- char ** Xml_getReactionsNames (xmlConfig_t *conf)

  Get name of reactions.
- char ** Xml_getBanned (xmlConfig_t *conf)

  Get list of banned species.
- int Xml_getSpeciesFinalAmount (xmlConfig_t *conf, char *species)

  Get the final amount of one species.
- int * Xml_getallSpeciesFinalAmount (xmlConfig_t *conf, char **species, int taille)

  Get Table of species amount.
- int Xml_getSpeciesWeight (xmlConfig_t *conf, char *species)

  Get Species weight.
- int * Xml_getallSpeciesWeight (xmlConfig_t *conf, char **species, int taille)

  Get list of species weight.
- char ** Xml_allocEquation (char **equation, int nb_noeud)

  Allocate memory for the list of equation.
- int Xml_getKineticLawNodes (xmlConfig_t *conf, int num_equation)

  Get the kinetic law for one node.
- char *** Xml_getKineticLaw (xmlConfig_t *conf, int num_equation, int nb_noeud)

  Get list of kinetic laws.

4.21.1 Detailed Description

Xml reader for parametre.xml. This file is part of MetaBoFlux (http://www.cbib.u-bordeaux2.fr/metaboflux/)

Author

{Amine Ghozlane}

Version

2.0

Date

15 janvier 2010

Definition in file xml_parameter.c.

4.21.2 Function Documentation

4.21.2.1 char ** Xml_allocEquation (char ** *equation*, int *nb_noeud*)

Allocate memory for the list of equation.

Author

Amine Ghozlane

Parameters

equation	list of equation
nb_noeud	Number of the node

Returns

List of reactions

Definition at line 789 of file xml_parameter.c.

Referenced by Xml_getKineticLaw().

4.21.2.2 void Xml_freeConfig (xmlConfig_t * *conf*)

Free the Struct xmlConfig_t.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Definition at line 97 of file xml_parameter.c.

References xmlConfig_t::ctxt, xmlConfig_t::doc, and xmlConfig_t::fichier.

Referenced by Data_desallocation(), and Xml_loadConfig().

4.21.2.3 int * Xml_getallSpeciesFinalAmount (xmlConfig_t * *conf*, char ** *species*, int *taille*)

Get Table of species amount.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t
species	Table of species
taille	Number of species

Returns

Table of species amount

Definition at line 708 of file xml_parameter.c.

References Xml_getSpeciesFinalAmount().

Referenced by Data_allocScore().

Here is the call graph for this function:

4.21.2.4 int * Xml_getallSpeciesWeight (xmlConfig_t * *conf*, char ** *species*, int *taille*)

Get list of species weight.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t
species	Table of species
taille	Number of species

Returns

List of species weight

Definition at line 770 of file xml_parameter.c.

References Xml_getSpeciesWeight().

Referenced by Data_allocScore().

Here is the call graph for this function:

4.21.2.5 char ** Xml_getBanned (xmlConfig_t * *conf*)

Get list of banned species.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

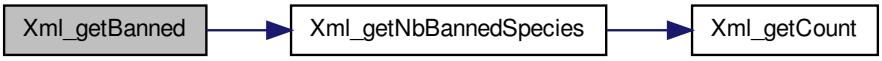
List of banned species

Definition at line 624 of file xml_parameter.c.

References xmlConfig_t::ctxt, and Xml_getNbBannedSpecies().

Referenced by Data_allocParameters().

Here is the call graph for this function:



4.21.2.6 double Xml_getBoltzmann (xmlConfig_t * *conf*)

Get the Boltzmann value.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

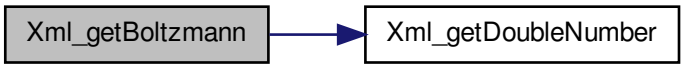
Boltzmann value

Definition at line 305 of file xml_parameter.c.

References Xml_getDoubleNumber().

Referenced by Mpi_slave(), and Recuit_printParametre().

Here is the call graph for this function:



4.21.2.7 int Xml_getCount (xmlConfig_t * *conf*, const char * *requete*)

Get count.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t
requete	Xpath query

Returns

Count

Definition at line 358 of file xml_parameter.c.

References xmlConfig_t::ctxt.

Referenced by Xml_getNbBannedSpecies(), Xml_getNbCouples(), Xml_getNbEquations(), Xml_getNbParameters(), Xml_getNbReaction(), Xml_getNbReactioninNoeud(), and Xml_-getNbSpecies().

4.21.2.8 double Xml_getDoubleNumber (xmlConfig_t * *conf*, const char * *requete*)

Get double value.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t
requete	Xpath query

Returns

Read double value in the xml

Definition at line 153 of file xml_parameter.c.

References xmlConfig_t::ctxt.

Referenced by Xml_getBoltzmann(), Xml_getMuT(), Xml_getStepSize(), Xml_getTinitial(), and Xml_getTmin().

4.21.2.9 char *** Xml_getKineticLaw (xmlConfig_t * *conf*, int *num_equation*, int *nb_noeud*)

Get list of kinetic laws.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t
num_equation	Number of equation
nb_noeud	Number nodes

Returns

List of kinetic laws

Definition at line 854 of file xml_parameter.c.

References xmlConfig_t::ctxt, and Xml_allocEquation().

Referenced by Data_allocEquations().

Here is the call graph for this function:

4.21.2.10 int Xml_getKineticLawNodes (xmlConfig_t * *conf*, int *num_equation*)

Get the kinetic law for one node.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t
num_equation	Number of the node

Returns

Ninetic law for one node

Definition at line 813 of file xml_parameter.c.

References xmlConfig_t::ctxt.

Referenced by Data_allocEquations().

4.21.2.11 double Xml_getMuT (xmlConfig_t * *conf*)

Get the variation of temperature.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

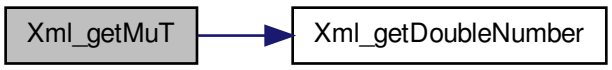
Variation of temperature

Definition at line 331 of file xml_parameter.c.

References Xml_getDoubleNumber().

Referenced by Mpi_slave(), and Recuit_printParametre().

Here is the call graph for this function:



4.21.2.12 int Xml_getNbBannedSpecies (xmlConfig_t * *conf*)

Get the number of banned species.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

Number of banned species

Definition at line 486 of file xml_parameter.c.

References Xml_getCount().

Referenced by Data_allocParameters(), and Xml_getBanned().

Here is the call graph for this function:

4.21.2.13 int Xml_getNbCouples (xmlConfig_t * *conf*)

Get the number of couples.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

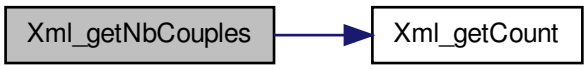
Number of couples

Definition at line 396 of file xml_parameter.c.

References Xml_getCount().

Referenced by Data_allocParameters(), and Xml_getReactionsNames().

Here is the call graph for this function:



4.21.2.14 int Xml_getNbEquations (xmlConfig_t * conf)

Get the number of equations.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

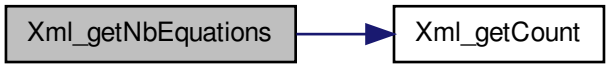
Number of equations

Definition at line 473 of file xml_parameter.c.

References Xml_getCount().

Referenced by Data_allocEquations().

Here is the call graph for this function:



4.21.2.15 int Xml_getNbGroup (xmlConfig_t * conf)

Get number of group.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

Number of iteration

Definition at line 279 of file xml_parameter.c.

References Xml_getNumber().

Referenced by Mpi_master().

Here is the call graph for this function:

4.21.2.16 int Xml_getNbIters (xmlConfig_t * *conf*)

Get number of iteration.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

Number of iteration

Definition at line 266 of file xml_parameter.c.

References Xml_getNumber().

Referenced by Mpi_slave(), and Recuit_printParametre().

Here is the call graph for this function:

4.21.2.17 int Xml_getNbParameters (xmlConfig_t * *conf*)

Get the number of parameters.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

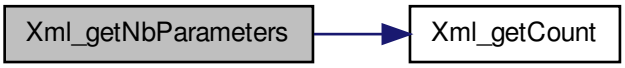
Number of parameters

Definition at line 409 of file xml_parameter.c.

References Xml_getCount().

Referenced by Data_allocParameters().

Here is the call graph for this function:



4.21.2.18 int Xml_getNbReaction (xmlConfig_t * *conf*)

Get the number of reactions.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

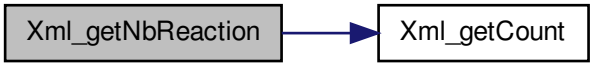
Number of reactions

Definition at line 447 of file xml_parameter.c.

References Xml_getCount().

Referenced by Data_allocScore(), and Xml_getReactionsNames().

Here is the call graph for this function:



4.21.2.19 int Xml_getNbReactioninNoeud (xmlConfig_t * *conf*, int *noeud*)

Count the number of reaction in one node.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t
noeud	Number of the node

Returns

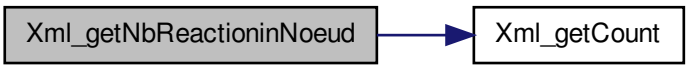
Number of reaction in one node

Definition at line 423 of file xml_parameter.c.

References Xml_getCount().

Referenced by Data_allocParameters(), and Xml_getReactionsNamesinNoeud().

Here is the call graph for this function:



4.21.2.20 int Xml_getNbSimulations (xmlConfig_t * conf)

Get number of simulation.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

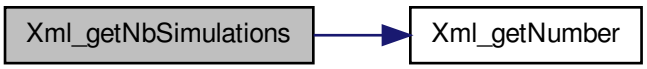
Number of simulation

Definition at line 227 of file xml_parameter.c.

References Xml_getNumber().

Referenced by Mpi_master().

Here is the call graph for this function:



4.21.2.21 int Xml_getNbSpecies (xmlConfig_t * conf)

Get the number of species.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

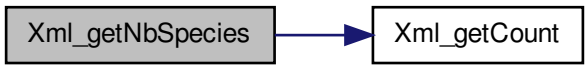
Number of species

Definition at line 460 of file xml_parameter.c.

References Xml_getCount().

Referenced by Data_allocScore(), Mpi_slave(), and Mpi_writer().

Here is the call graph for this function:



4.21.2.22 int Xml_getNbTriesMod (xmlConfig_t * *conf*)

Get number of tries for modelling.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

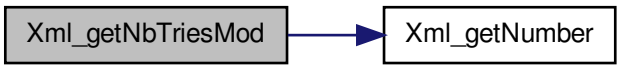
Number of tries

Definition at line 240 of file xml_parameter.c.

References Xml_getNumber().

Referenced by Data_allocParameters(), Mod_compute_modeling(), Mpi_sizeResultTab(), and Sd_compute_standard_deviation().

Here is the call graph for this function:



4.21.2.23 int Xml_getNbTriesSa (xmlConfig_t * *conf*)

Get number of tries for the simulated annealing and minimization.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

Number of tries

Definition at line 253 of file xml_parameter.c.

References Xml_getNumber().

Referenced by Data_allocParameters(), Min_my_f(), Mpi_slave(), and Recuit_printParametre().

Here is the call graph for this function:

4.21.2.24 int Xml_getNumber (xmlConfig_t * *conf*, const char * *requete*)

Get integer value.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t
requete	Xpath query

Returns

Read integer value in the xml

Definition at line 115 of file xml_parameter.c.

References xmlConfig_t::ctxt.

Referenced by Xml_getNbGroup(), Xml_getNbIters(), Xml_getNbSimulations(), Xml_-getNbTriesMod(), and Xml_getNbTriesSa().

4.21.2.25 char ** Xml_getReactionsNames (xmlConfig_t * *conf*)

Get name of reactions.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

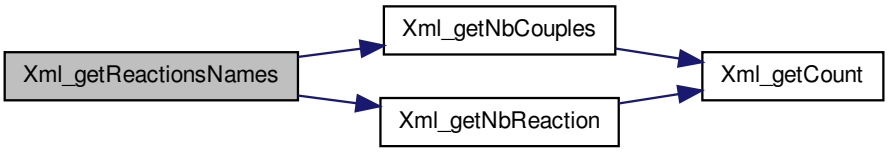
Name of reactions

Definition at line 541 of file xml_parameter.c.

References xmlConfig_t::ctxt, Xml_getNbCouples(), and Xml_getNbReaction().

Referenced by Data_allocScore().

Here is the call graph for this function:



4.21.2.26 char ** Xml_getReactionsNamesinNoeud (xmlConfig_t * *conf*, int *numero*)

Get list of reactions.

Get list of species.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t
numero	Number of the node

Returns

List of reactions

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

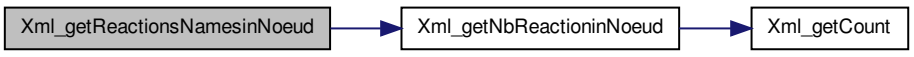
Returns

List of species

Definition at line 500 of file xml_parameter.c.

References xmlConfig_t::ctxt, and Xml_getNbReactioninNoeud().

Here is the call graph for this function:



4.21.2.27 int Xml_getSpeciesFinalAmount (xmlConfig_t * *conf*, char * *species*)

Get the final amount of one species.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t
species	Name of one species

Returns

Final amount of one species

Definition at line 665 of file xml_parameter.c.

References xmlConfig_t::ctxt.

Referenced by Xml_getallSpeciesFinalAmount().

4.21.2.28 int Xml_getSpeciesWeight (xmlConfig_t * *conf*, char * *species*)

Get Species weight.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t
species	Name of one species

Returns

Species weight

Definition at line 727 of file xml_parameter.c.

References xmlConfig_t::ctxt.

Referenced by Xml_getallSpeciesWeight().

4.21.2.29 double Xml_getStepSize (xmlConfig_t * *conf*)

Get the step size.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

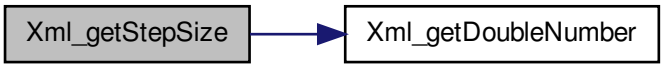
Step size

Definition at line 292 of file xml_parameter.c.

References Xml_getDoubleNumber().

Referenced by Mpi_slave(), and Recuit_printParametre().

Here is the call graph for this function:



4.21.2.30 char * Xml_getString (xmlConfig_t * *conf*)

Get name value.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

Read name value in the xml

Definition at line 190 of file xml_parameter.c.

References xmlConfig_t::ctxt.

Referenced by Recuit_redirectionFlux().

4.21.2.31 double Xml_getTinitial (xmlConfig_t * *conf*)

Get the initial temperature.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

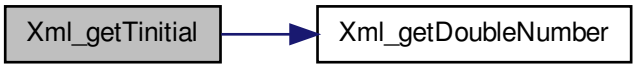
Initial temperature

Definition at line 318 of file xml_parameter.c.

References Xml_getDoubleNumber().

Referenced by Mpi_slave(), and Recuit_printParametre().

Here is the call graph for this function:



4.21.2.32 double Xml_getTmin (xmlConfig_t * *conf*)

Get the minimum temperature.

Author

Amine Ghozlane

Parameters

conf	Struct xmlConfig_t

Returns

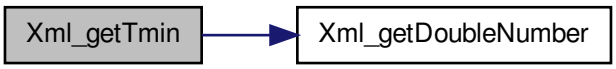
Minimum temperature

Definition at line 344 of file xml_parameter.c.

References Xml_getDoubleNumber().

Referenced by Mpi_slave(), and Recuit_printParametre().

Here is the call graph for this function:



4.21.2.33 xmlConfig_t * Xml_loadConfig (char * *fichier*)

Initialize and load the parameter.xml.

Author

Amine Ghozlane

Parameters

fichier	Xml file name

Returns

Struct xmlConfig_t

Definition at line 42 of file xml_parameter.c.

References xmlConfig_t::ctxt, xmlConfig_t::doc, xmlConfig_t::fichier, xmlConfig_t::racine, and Xml_freeConfig().

Referenced by Data_initXML().

Here is the call graph for this function:

Index