CaRNeSS

4.8 (20131209.62)

Generated by Doxygen 1.8.5

Tue Dec 10 2013 10:10:07

Contents

Chapter 1

Catalytic Rections Network Stochastic Simulator - CaRNeSS 4.8 (20131209.62)

Author

Alessandro Filisetti

Version

4.8 (20131209.62)

Date

2013-12-09 sourceforge repository – `https://carness.svn.sourceforge.net/svnroot/carness/`
git repository – `https://github.com/paxelito/carness`

This manual is divided in the following sections:

Chapter 2

Essential informations

The **Catalytic Reactions Network Stochastic Simulator (CaRNeSS)** is a computational model devoted to the simulation of theoretical complex catalytic networks composed of different interacting molecular species. The model takes inspiration from the original model proposed by Stuart Kauffman in 1986, and describes systems composed of molecular species interacting by means of two possible reactions only, cleavage and condensation. One polymer is divided into two short polymers in the former case while two polymers are glued together forming a longer polymer in the latter case. Each reaction must be catalyzed by another species in the system to occur, and one of the assumptions is that any chemical has an independent probability to catalyze a randomly chosen reaction. Since the version 4.8 (20131026.60) spontaneous reactions are considered too. It is important to notice that there are not indications about the chemical nature of the molecules, species "A" may be both a polipeptide, an amminoacid, a particular protein domain or an RNA strenght.

2.1 Using the simulator

To run the simulator open a terminal shell and type:

$path/executiveFile <configuration_File_Folder> <output_folder> <reaction_structures_folder>

Examples:

- Unix Based Systems:~/Documents/project/acsm2s ~/Documents/.../confFileFolder/ ~/-Documents/.../resFolder/ ~/Documents/.../StructuresFolder/
- Win Systems: C:\Documents\project\acsm2s.exe C:\Documents ..\confFileFolder\ C-:\Documents ..\resFolder\ C:\Documents ..\StructuresFolder\

2.2 System Requirement

In order to have the simulator run correctly the recommended staff is reported:

- MacOsX 10.4 or later, Linux (or in general a system UNIX based) or Windows OS (tests have been performed on Win7 and win Vista) as well
- GCC (G++) compiler, or similar, installed (if you need to compile the software on your machine)

On MacOs system compile using g++ -Wall -ansi -lm -o CaRNeSS *.cpp

2.3 Input Parameters :: acsm2s.conf

All the system parameters are stored in a file called **acsm2s.conf**. Anyone can create his own configuration file paying attention to put "=" char between the parameter name and the the parameter value (NO SPACE BETWEEN

THEM).

Notice that the simulator does not create the initial structures but it simply loads the structures created by an external software and process them. Nevertheless the configuration file is fundamental to supply all the parameters to the simulation (during the simulation new entities may be created). The simulator is provided with a structures initializator developed in MATLAB language by the group (a description of the initializator is provided in the main file "start.m") in which all the parameters we are going to describe are used to create the initial structures. All parameters are reported below divided in three categories:

```
- System
- Environment
- Dynamic

Categories are useful only to help users in the parameter recognition within the configuration file.
The following parameters are used both by the initializator and the simulator. Nvertheless it is ALWAYS neces
```

2.3.1 System

Parameters

nGen	($>$ 0) Number of generations. This parameter indicate how many times the simulation is stopped, concentration are set to the initial ones and the simulazion restart for other n-Seconds seconds.
nSIM	($>$ 0) Number of simulations per generation starting with the same initial conditions (same data structures) but different random seed
nSeconds	($>$ 0) Number of seconds
nReactions	($>$ 0) Max number of reactions (the system will be stopped after nSeconds or after n-Reactions)
randomSeed	($>$= 0) Random seed (if 0 the random seed is randomly created and the it is stored in the acsm2s.conf file saved in the results folder)
nHours	($>$=0) Runtime limit (hours)
nAttempts	($>$=0) Number of temptative in simulating the same network structure different random seed
debugLevel	($>$= -1) Debug Level Runtime: different runTime message amounts (from -1 to 4, 0 is suggested)
timeStructures-SavingInterval	($>$ 0) All system structures (species, catalysis and reactions) are saved every *timeStructures-SavingInterval* seconds (simulation time)
fileTimesSave-Interval	($>$= 0) Times data are stored in file times.csv every *fileTimesSaveInterval* seconds (If 0 reactions are stored continually)

2.3.2 Environment

Parameters

newSpecies-ProbMin-Threshold	($>$=0) Minimal new species creation probability to allow system expansion
lastFiringDisk-SpeciesID	($>$ 0) The ID of the last firing disk species.
overall-Concentration	($>$ 0) The overall initial concentration that will be divided between all the initial species according to the selected initial distribution.
EC-Concentration	($>$ 0) Incoming concentration of charged molecules per second.
alphabet	(string) Alphabet used in the simulation (e.g. *AGCT* for DNA, *ADEGFLYCWPHQIMTNKSRV* for proteins)

volume	(> 0) Volume of the container or protocell

2.3.3 Dynamic

Parameters

energy	(0 or 1) 0 no energy in the system, 1 energy constraints are applied
ratioSpecies-Energizable	(%) The probability for a species to be potentially energized by the energy carriers
nonCatalytic-MaxLength	(>= 0) Max length of non catalytic species
reaction-Probability	(from 0 to 1) Probability for a species to catalyze a reaction
cleavage-Probability	(from 0 to 1) Cleavage probability (Condensation probability is 1 - cleavage probability)
reverseReaction	(0 or 1) Set to 1 to enable reverse reactions, 0 otherwise
revRctRatio	(>0) Ratio between forward and backward reactions, it is used in the creation of new reactions only (if reverseReactions = TRUE)
spontRct	(0 or 1) If 1 spontanoues reactions are considered, otherwise no
K_ass	(>= 0) Final Condensation kinetic constant (C.A + B –> AB + C) where A.C is the molecular complex composed of C (the catalyst) and A (the first substrate)
K_diss	(>= 0) Cleavage kinetic constant (AB –> A + B)
K_cpx	(>= 0) Complex formation kinetic constant (A + C(catalyst) –> C.A)
K_cpxDiss	(>= 0) Complex Dissociation kinetic constant (C.A –> A + C)
K_nrg	(>= 0) species phosphorilation kinetic constant
K_nrg_decay	(>= 0) de-energization kinetic constant
moleculeDecay-_KineticConstant	(> 0) Molecule decay (efflux) kinetic Constant (Disregarded if the system is closed)
K_spont_ass	(>= 0) If spontaneous reactions are turned on this is the default kinetic rate for spontanoues condensations
K_spont_diss	(>= 0) If spontaneous reactions are turned on this is the default kinetic rate for spontanoues cleavages
influx_rate	(>= 0) Concentration per seconds (The species to insert in the system will be randomly chosen according to the _acsinflux.csv file). If equal to 0 the system is closed (maxLOut=0) or only the species that can cross the membrane come in and go out (maxLOut>0).
maxLOut	Maximum lenght of the species involved in the efflux process (`influx_rate` > 0), equal to 0 indicates that all the species can be involved in the efflux process (no filter). If influx_rate = 0 the parameter indicates the species that can cross the semipermeable membrane of the protocell. **THE COUPLING BETWEEN INFLUX_RATE AND MAXLOUT INDICATES IF WE ARE SIMULATING A PROTOCELL OR A FLOW REACTOR**
diffusion_-contribute	(KD) (0 or 0.5) if set to 0.5 the speed of molecules goes with the inverse of the square of the length, L^{-KD}
solubility_-threshold	(> 0) Solubility Threshold, all the species longer than solubility_threshold precipitate

2.4 Acknowledgments

- University of Bologna, Interdepartment of industrial research (C.I.R.I)
- European Centre for Living Technology http://www.ecltech.org/
- Fondazione Venezia http://www.fondazionevenezia.it
- Alex Graudenzi to take care of the initilizator.
- Chiara Damiani to contribute to the development of the software.

- Roberto Serra, Marco Villani, Timoteo Carletti, Norman Packard, Ruedi Fuchslin and Stuart Kauffman for the essential hints.
- http://www.bedaux.net/mtrand/ for the pseudo-random Marseinne-Twister library for C++.
- http://perso.wanadoo.es/antlarr/otherapps.html for the poisson distribution generator numbers (acs_longInt random_poisson(acs_double tmpLambda, MTRand& tmpRandomGenerator)).
- Dr. Luca Ansaloni (luca.ansaloni@unimore.it) for the support but especially for the file handling functions and new Python development.

Chapter 3

Initial Data Structures

In order to proceed with the simulation the system needs from **4** to **6 (open system and energy on)** initial data structures files (an example for each file is located into the source code folder):

```
- \c acsm2s.conf (described in the \ref parameters section)
- \c _acsspecies.csv - This file contains all the initial <b>species</b> with their proprieties
- \c _acsreactions.csv - This file contains all the initial <b>reactions</b> with their proprieties
- \c _acscatalysis.csv - This file contains all the <b>correspondances between species and reactions</b> with
```

and, if the system is open

```
- \c _acsinflux.csv - This file contains all the species belonging to the incoming flux
```

and. if the system is energy based

```
- \c _acsnrgbooleanfunctions.csv - This file contains all the possible boolean functions associated
```

3.1 _acsspecies.csv

Columns description (each field is delimited using "\t"):

Identification ID)	Sequence(	Concentration	Diffusion enhancement	Precipitation flag	Complex Dissociation Kinetic Constant	Complex Binding Point	Already evaluated flag	Species Age	Number of reborns	Catalyst ID	Complex ID	Phosphorilation Kinetic constant	Charged Molecules Concentration	Concentration locked

- *ID*: Species Index
- *Sequence*: Species sequence
- *Concentration*: Total concentration of the species
- *Diffusion enhancement*: Diffusion enhancement
- *Precipitation flag*: If 0 species is precipited and each new molecules of this species will be precipited
- *Dissociation Kinetic Constant*: Complex dissociation kinetic constant
- *Binding point*: If the species is a complex this field indicates the division point between catalyst and substrate
- *Evaluated*: If 1 the species is not virtual and all the reactions it catalyzes are created, if 0 the species is only potentially created
- *Species Age*: Age (in seconds) of the species since its last cretion, each time that a species amount pass from 0 to $>$ 1 the counter is resetted
- *Number of reborns*: Number of times that a species amount pass from 0 to $>$ 0
- *Catalyst ID*: If the species is a complex (e.g. catalyst C forming a complex C.A with the substrate A) this is the CATALYST ID
- *Substrate ID*: If the species is a complex (e.g. catalyst C forming a complex C.A with the substrate A) this is the SUBSTATE ID
- *Phosphorilation Kinetic constant*: NOT USED NOW!!!
- *Charged Molecules Concentration*: Concentration of the charged molecules belonging to the species.
- *Concentration locked*: 0 -$>$ Concentration of the species changes according to the reactions affecting it, 1 -$>$ The concentration of the species is locked (permeable species, the concentration of the species is assumed to be constant within the cell and in the environment)

3.2 _acsreactions.csv

Columns description (each field is delimited using "\t"):

Identificator	Reaction type	Species 1	Species 2	Species 3	Reaction counter	Energy type	k spont

- *Identificator*: Reaction ID
- *Reaction type*: 0 Condensation, 1 Cleavage
- *Species 1*: Product ID if reaction type = 1, Substrate ID reaction type = 0
- *Species 2*: Product ID if reaction type = 0, Substrate ID reaction type = 1
- *Species 3*: Product ID if reaction type = 0, Substrate ID reaction type = 1
- *Reaction counter*: Reaction occurrance counter
- *Energy Type*: The reaction energetic configuration, 1 for endoergonic 0 for esoergonic
- *Spontaneous Constant*: Spontaneous reaction constant

3.3 _acscatalysis.csv

Columns description (each field is delimited using "\t"):

Identificator	Catalyst ID	Reaction ID	Catalysis counter	K condensation	K cleavage	K Complex Association	Complex creation substrate target

- *Identificator*: Catalysis ID
- *Catalyst ID*: species (as catalyst) ID
- *Reaction ID*: Reaction ID
- *Catalysis counter*: Catalysis counter
- *K final step end condensation kinetic constant*: Final step end condensation kinetic constant
- *K Cleavage*: Cleavage Kinetic constant
- *K complex association*: Complex association kinetic constant
- *Complex creation substrate target*: Which substrate will be involved in complex creation, 1 or 2

3.4 _acsinflux.csv

Columns description (each field is delimited using "\t"):

Identificator	Probabilitity

- *Identificator*: Species ID
- *Probabilitity*: Probability to be selected when a species has to be inserted into the system

3.5 _acsnrgbooleanfunctions.csv

Columns description (each field is delimited using "\t"):

Energetic Boolean Function (decimal form)	Probabilitity

- *Energetic Boolean Function (decimal form)*: Energetic Boolean Function decimal form, e.g. 1001011010100 = 4820
- *Probabilitity*: Probability to be selected when a new reaction has to be created

Chapter 4

Outcomes

During the simulation **8** different files concerning species, reactions and catalysis tables, times, reactions parameters, living species, living species amounts and concentration are produced. In the following you find a brief description of all files and some indications about the different time storage.

4.1 Species

Every environment::timeStructuresSavingInterval seconds species structures will be saved in files called species_-<generationNumber>_<simulationNumber>_<reactionNumber>.csv (e.g. **species_1_06_0016933.csv** means species structure, generation number 1, simulation number 6, reaction number 16933).

Columns description (each field is delimited using "\t") is the same described above, section _acsspecies.csv .

4.2 Reactions

Every environment::timeStructuresSavingInterval seconds reactions structures will be saved in files called reactions-_<generationNumber>_<simulationNumber>_<reactionNumber>.csv (e.g. **reactions_1_06_0016933.csv** means reactions structure, generation number 1, simulation number 6, reaction number 16933).

Columns description (each field is delimited using "\t") is the same described above, section _acsreactions.csv .

4.3 Catalysis

Every environment::timeStructuresSavingInterval seconds catalysis structures will be saved in files called catalysis-_<generationNumber>_<simulationNumber>_<reactionNumber>.csv (e.g. **catalysis_1_06_0016933.csv** means catalysis structure, generation number 1, simulation number 6, reaction number 16933).

Columns description (each field is delimited using "\t")is the same described above, section _acscatalysis.csv :

4.4 Times

Every simulation generates a times file called times_<generationNumber>_<simulationNumber>.csv (e.g. **times-_1_03.csv** means times file, generation 1, simulation 3) containing values concerning **each events** occurred during the simulation. This file is **created at the beginning of the simulation** and it is **updated after each event**, hence this file will contain a number of rows equal to the number of reaction (of any type) occurred during the simulation.

Columns description (each field is delimited using "\t"):

Identificator (reaction)	Reaction Time	Gillespie ID selected	Reaction Type	Number of possible reactions	Computational Time (sec)	Number of Species	Number of Molecules	Number of complex species	Number of complexes	Number of bricks	Gillespie Computational Time	Reaction Process Computational Time	Various Processes Computational Time	New species creation probability

- *Identificator (reaction)*: Number of the reaction (STEP)
- *Reaction Time*: Time of the reaction
- *Gillespie ID selected*: Gillespie structure reaction ID
- *Reaction Type*: 0 condensation, 1 cleavage, 2 complex formation, 3 complex degradation, 4 species efflux, 6 endergonic cleavage, 7 endoergonic condensation, 8 endoergonic complex creation
- *Number of possible reactions*: Total number of possible reactions according to the Gillespie algorithm computation
- *Computational Time (ms)*: Computational time between two successive reactions
- *Number of Species*: Number of species with at least one molecule
- *Number of Molecules*: Number of molecules
- *Number of complex species*: Number of complex species with at least one molecule
- *Number of complexes*: Number of complex molecules
- *Number of bricks*: Number of single bricks (monomers) present in the system
- *Gillespie Computational Time*: Number of milliseconds necessary to complete the Gillespie task
- *Reaction Process Computational Time*: Number of milliseconds necessary to perform a reaction (and evaluate products) once that the Gillespie algorithm has selected the reaction
- *Various Processes Computational Time*: Number of milliseconds necessary to perform several tasks not correlated with the simulation of the phenomena
- *New species creation probability*: Given the state of the system, probability to create a new species
- *Reverse Reaction Probability*: Given the state of the system, probability for a reverse reaction to occur

4.5 Reactions_parameters

Every simulation generates a reactions parameters file called reactions_parameters_<generationNumber>_-<simulationNumber>.csv (e.g. **reactions_parameters_1_07.csv** means reactions parameters file, generation 1 simulation number 7) containing informations on **cleavage and condensation** reactions. This file is **generated at the beginning of each simulation** and it is **updated each time that a cleavage or a condensation occur**.

Columns description (each field is delimited using "\t"):

Reaction ID	Reaction Time	Reaction Type	Catalyst ID	Species 1 ID	Species 2 ID	Species 3 ID	Charged Molecules	Charged Concentration	Gillespie Score Average	Gillespie Score Standard Deviation	Entropy	New species creation probability

- *Reaction ID*: This ID referes to the reaction number (step)
- *Reaction Time*: Time of the reaction
- *Reaction Type ID*: 0 condensation, 1 cleavage, 6 endoergonic cleavage, 7 endoergonic condensation
- *Catalyst ID*: Catalyst ID into the species table
- *Species 1 ID*: product ID in case of condensation, substrate in case of cleavage
- *Species 2 ID*: substrate ID in case of condensation, product in case of cleavage
- *Species 3 ID*: substrate ID in case of condensation, product in case of cleavage
- *Charged Molecules*: Number of charged molecules
- *Charged Concentration*: Concentration of the charged molecules
- *Gillespie Score Average*: Average of the all gillespie algorithm scores
- *Gillespie Score Standard Deviation*: Standard Deviation of the all gillespie algorithm scores
- *Entropy*: Entropy Misure (test) based on the probabilities that reactions occur
- *New species creation probability*: Given the state of the system, probability to create a new species

4.6 Living_species

Each simulation generates a living species file called livingSpecies_<generationNumber>_<simulationNumber>.csv (e.g. **livingSpecies_1_07.csv** means living species file, generation 1 simulation number 7) containing all the living species IDs (living species are those species with concentration greater than 0) for each reaction in each raw. This file is **generated at the beginning of each simulation** and it is **updated each time that a cleavage or a condensation occur**. This file has a variable number of columns equal to the number of living species plus 2 (Reaction ID and time of the reaction)

Columns description (each field is delimited using "\t"):

Reaction ID	Reaction Time	living species IDs..., one for column

- *Reaction ID*: This ID referes to the reaction number (step)
- *Reaction Time*: Time of the reaction
- *Living Species IDs..., one for column*: Each column from the thirth to the last one contains living species ID

4.7 Living_species_amounts

Each simulation generates a living species amount file called livingAmount_<generationNumber>_<simulationNumber>.csv (e.g. **livingAmount_1_07.csv** means living species amount file, generation 1, simulation number 7) containing all the living species amount (living species are those species with concentration greater than 0) for each reaction in each raw. **This file is tightly correlated with the *Living_species* file**. Each row contains the total amount of molecules belonging to the species indicated by its ID in the livingSpecies file. This file is **created at the beginning of each simulation** and it is **updated each time that a cleavage or a condensation occur**. This file has a variable number of columns equals to the number of living species at time t

Columns description (each field is delimited using "\t"):

living species amounts..., one for column

- *Living Species amounts..., one for column*: Each values is related to the species indicated in the livingSpecies file. Reaction ID and time is that of the livingSpecies file

4.8 Living_species_concentration

Each simulation generates a living species amount file called livingConcentration_<generationNumber>_-<simulationNumber>.csv (e.g. **livingConcentration_1_07.csv** means living species amount file, generation 1, simulation number 7) containing all the living species amount (living species are those species with concentration greater than 0) for each reaction in each raw. **This file is tightly correlated with the *Living_species* file**. Each row contains the total amount of molecules belonging to the species indicated by its ID in the livingSpecies file. This file is **created at the beginning of each simulation** and it is **updated each time that a cleavage or a condensation occur**. This file has a variable number of columns equals to the number of living species at time t

Columns description (each field is delimited using "\t"):

living species amounts..., one for column

- *Living Species amounts..., one for column*: Each values is related to the species indicated in the livingSpecies file. Reaction ID and time is that of the livingSpecies file

Chapter 5

Gillespie Class

The Gillespie class contains the information concerning the Gillespie algorithm propensity with other useful information. In particular it has 4 properties, namely mol_I, mol_II, mol_III, mol_IV containing the species ID's related to different species type according to the different admitted reactions. In the following a table summarizing the different meanings for the different reactions is reported.

Reaction Type	Reaction Type Code	MOL 1	MOL 2	MOL 3	MOL 4
Complex Formation	2	Catalyst	First Substrate	Catalysis ID	Second Substrate
Final Condensation	0	Catalyst	Substrate	Product	Complex
Cleavage	1	Substrate	Product 1	Product 2	Catalyst
Spontaneous Cleavage	11	Substrate	Product 1	Product 2	//
Spontaneous Condensation	10	Product	Substrate 1	Substrate 2	//

Chapter 6

The initializator (a very brief description)

The initializator provided with the simulator is located in the `initializator` folder (within the source code folder) and it is developed in Matlab code. All the parameters are set in the `start.m` file (from line 22 to line 63). In addition you find the parameters related to the name of the folder that will contain the simulation (`simFolder.name`), the path where that folder will be created (`simFolder.path`) and the number of different network ensambles to create (`simFolder.nets`). It is important to notice that the initializator has not been thought to be shared, so it is not too much user friendly to be manipulated. Nevertheless it could be very useful with a little bit of practice.

6.1 Screening Parameter

To initialize structures to perform a sensitivity analysis of a specific parameters follow the following instruction:

- Change the name of the array at row 13 with the name of the parameter you want analyze. This array contains the values of the paramter. Remember to insert the old parameter you are changing in the parameters list with its single value, otherwise it would be a missing parameter.
- In the array at row 14 (`nome_folder`) you have put the same number of elements of the array containing the screening values. This array contains a numeric tag of the values (used to create the simulations folder names) contained in the values array.
- In the code of the `start.m` file (rows 122 to 150) you must change:
  - row 121: [rows,b]=size(reactionProbability); –> [rows,b]=size(name_of_the_parameter_you_have_trasformed_in_array);
  - row 146: `inizializzatore_ACS`(... reactionProbability(i), ...); -> inizializzatore_ACS(... name_of_the_parameter_you_have_trasformed_in_array(i), ...); (remember to remove the index to the reactionProbability parameter

Chapter 7

Namespace Index

7.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 8

Hierarchical Index

8.1 Class Hierarchy

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

Chapter 9

Class Index

9.1 Class List

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

Chapter 10

File Index

10.1 File List

Here is a list of all files with brief descriptions:

Chapter 11

Namespace Documentation

11.1 acsAttractorAnalysis Namespace Reference

Functions

- def zeroBeforeStrNum

Variables

- tuple StrPath = os.path.abspath(StrPath)
- tuple today = dt.date.today()
- tuple tmpDirs = sort(os.listdir(StrPath))
- list allSortedSpecies = []
- list allConcentrations = []
- list allSortedSpeciesNOINFLUX = []
- list allConcentrationsNOINFLUX = []
- tuple totDirName = os.path.join(StrPath,tmpDir)
- tuple resDirPath = os.path.abspath(os.path.join("./", "res"))
- tuple numberOfGen = len(glob.glob(os.path.join(resDirPath,'times_*')))
- tuple strZeros = zeroBeforeStrNum(ngen, numberOfGen)
- string strSpecies = 'species_'
- tuple speciesFiles = sorted(glob.glob(os.path.join(resDirPath,strSpecies)))
- list speciesFile = speciesFiles[-1]
- tuple fidSpecies = open(speciesFile, 'r')
- list seq = []
- list overallConcList = []
- list overallConcListNOINFLUX = []
- int numberOfFolders = 0
- tuple pos = seq.index(key)
- tuple pos2 = seq.index(key2)
- tuple ANG_overallResMatrix = np.zeros((numberOfFolders,numberOfFolders))
- tuple ANG_overallResMatrixNOINFLUX = np.zeros((numberOfFolders,numberOfFolders))
- tuple HAM_overallResMatrix = np.zeros((numberOfFolders,numberOfFolders))
- tuple HAM_overallResMatrixNOINFLUX = np.zeros((numberOfFolders,numberOfFolders))
- tuple EUC_overallResMatrix = np.zeros((numberOfFolders,numberOfFolders))
- tuple EUC_overallResMatrixNOINFLUX = np.zeros((numberOfFolders,numberOfFolders))
- tuple vecX = np.array(lx)
- tuple vecY = np.array(ly)
- tuple tmpCos = float(np.dot(vecX,vecY) / (np.linalg.norm(vecX) * np.linalg.norm(vecY)))

- int tmpHD = 0
- int tmpEU = 0
- string ndn = '_0_new_allStatResults'
- tuple newdirAllResults = os.path.join(os.curdir, ndn)
- string outFnameStat = 'speciesVector.csv'
- tuple saveFileStat = open(outFnameStat, 'w')
- string strTypes = ''
- int cnt = 0

11.1.1 Function Documentation

11.1.1.1 def acsAttractorAnalysis.zeroBeforeStrNum (*tmpI, tmpL*)

Definition at line 25 of file acsAttractorAnalysis.py.

11.1.2 Variable Documentation

11.1.2.1 list acsAttractorAnalysis.allConcentrations = [ ]

Definition at line 50 of file acsAttractorAnalysis.py.

11.1.2.2 list acsAttractorAnalysis.allConcentrationsNOINFLUX = [ ]

Definition at line 52 of file acsAttractorAnalysis.py.

11.1.2.3 tuple acsAttractorAnalysis.allSortedSpecies = [ ]

Definition at line 49 of file acsAttractorAnalysis.py.

11.1.2.4 tuple acsAttractorAnalysis.allSortedSpeciesNOINFLUX = [ ]

Definition at line 51 of file acsAttractorAnalysis.py.

11.1.2.5 tuple acsAttractorAnalysis.ANG_overallResMatrix = np.zeros((numberOfFolders,numberOfFolders))

Definition at line 171 of file acsAttractorAnalysis.py.

11.1.2.6 tuple acsAttractorAnalysis.ANG_overallResMatrixNOINFLUX = np.zeros((numberOfFolders,numberOfFolders))

Definition at line 172 of file acsAttractorAnalysis.py.

11.1.2.7 int acsAttractorAnalysis.cnt = 0

Definition at line 262 of file acsAttractorAnalysis.py.

11.1.2.8 tuple acsAttractorAnalysis.EUC_overallResMatrix = np.zeros((numberOfFolders,numberOfFolders))

Definition at line 175 of file acsAttractorAnalysis.py.

11.1.2.9 tuple acsAttractorAnalysis.EUC_overallResMatrixNOINFLUX = np.zeros((numberOfFolders,numberOfFolders))

Definition at line 176 of file acsAttractorAnalysis.py.

11.1.2.10 tuple acsAttractorAnalysis.fidSpecies = open(speciesFile, 'r')

Definition at line 82 of file acsAttractorAnalysis.py.

11.1.2.11 tuple acsAttractorAnalysis.HAM_overallResMatrix = np.zeros((numberOfFolders,numberOfFolders))

Definition at line 173 of file acsAttractorAnalysis.py.

11.1.2.12 tuple acsAttractorAnalysis.HAM_overallResMatrixNOINFLUX = np.zeros((numberOfFolders,numberOfFolders))

Definition at line 174 of file acsAttractorAnalysis.py.

11.1.2.13 string acsAttractorAnalysis.ndn = '_0_new_allStatResults'

Definition at line 219 of file acsAttractorAnalysis.py.

11.1.2.14 tuple acsAttractorAnalysis.newdirAllResults = os.path.join(os.curdir, ndn)

Definition at line 220 of file acsAttractorAnalysis.py.

11.1.2.15 int acsAttractorAnalysis.numberOfFolders = 0

Definition at line 109 of file acsAttractorAnalysis.py.

11.1.2.16 tuple acsAttractorAnalysis.numberOfGen = len(glob.glob(os.path.join(resDirPath,'times_*')))

Definition at line 69 of file acsAttractorAnalysis.py.

11.1.2.17 string acsAttractorAnalysis.outFnameStat = 'speciesVector.csv'

Definition at line 230 of file acsAttractorAnalysis.py.

11.1.2.18 list acsAttractorAnalysis.overallConcList = []

Definition at line 107 of file acsAttractorAnalysis.py.

11.1.2.19 list acsAttractorAnalysis.overallConcListNOINFLUX = []

Definition at line 108 of file acsAttractorAnalysis.py.

11.1.2.20 tuple acsAttractorAnalysis.pos = seq.index(key)

Definition at line 153 of file acsAttractorAnalysis.py.

11.1.2.21 tuple acsAttractorAnalysis.pos2 = seq.index(key2)

Definition at line 159 of file acsAttractorAnalysis.py.

11.1.2.22 tuple acsAttractorAnalysis.resDirPath = os.path.abspath(os.path.join("./", "res"))

Definition at line 63 of file acsAttractorAnalysis.py.

11.1.2.23 tuple acsAttractorAnalysis.saveFileStat = open(outFnameStat, 'w')

Definition at line 231 of file acsAttractorAnalysis.py.

11.1.2.24 list acsAttractorAnalysis.seq = [ ]

Definition at line 87 of file acsAttractorAnalysis.py.

11.1.2.25 list acsAttractorAnalysis.speciesFile = speciesFiles[-1]

Definition at line 78 of file acsAttractorAnalysis.py.

11.1.2.26 tuple acsAttractorAnalysis.speciesFiles = sorted(glob.glob(os.path.join(resDirPath,strSpecies)))

Definition at line 77 of file acsAttractorAnalysis.py.

11.1.2.27 tuple acsAttractorAnalysis.StrPath = os.path.abspath(StrPath)

Definition at line 42 of file acsAttractorAnalysis.py.

11.1.2.28 string acsAttractorAnalysis.strSpecies = 'species_'

Definition at line 74 of file acsAttractorAnalysis.py.

11.1.2.29 string acsAttractorAnalysis.strTypes = ''

Definition at line 233 of file acsAttractorAnalysis.py.

11.1.2.30 tuple acsAttractorAnalysis.strZeros = zeroBeforeStrNum(ngen, numberOfGen)

Definition at line 73 of file acsAttractorAnalysis.py.

11.1.2.31 tuple acsAttractorAnalysis.tmpCos = float(np.dot(vecX,vecY) / (np.linalg.norm(vecX) * np.linalg.norm(vecY)))

Definition at line 184 of file acsAttractorAnalysis.py.

11.1.2.32 tuple acsAttractorAnalysis.tmpDirs = sort(os.listdir(StrPath))

Definition at line 48 of file acsAttractorAnalysis.py.

11.1.2.33 int acsAttractorAnalysis.tmpEU = 0

Definition at line 189 of file acsAttractorAnalysis.py.

11.1.2.34 int acsAttractorAnalysis.tmpHD = 0

Definition at line 188 of file acsAttractorAnalysis.py.

11.1.2.35 tuple acsAttractorAnalysis.today = dt.date.today()

Definition at line 44 of file acsAttractorAnalysis.py.

11.1.2.36 tuple acsAttractorAnalysis.totDirName = os.path.join(StrPath,tmpDir)

Definition at line 59 of file acsAttractorAnalysis.py.

11.1.2.37 tuple acsAttractorAnalysis.vecX = np.array(lx)

Definition at line 181 of file acsAttractorAnalysis.py.

11.1.2.38 tuple acsAttractorAnalysis.vecY = np.array(ly)

Definition at line 182 of file acsAttractorAnalysis.py.

11.2 acsAttractorAnalysisInTime Namespace Reference

Functions

- def zeroBeforeStrNum

Variables

- tuple StrPath = os.path.abspath(StrPath)
- tuple today = dt.date.today()
- tuple tmpDirs = sort(os.listdir(StrPath))
- list allSortedSpecies = []
- list allConcentrations = []
- list allSortedSpeciesNOINFLUX = []
- list allConcentrationsNOINFLUX = []
- tuple totDirName = os.path.join(StrPath,tmpDir)
- tuple resDirPath = os.path.abspath(os.path.join("./", "res"))
- tuple numberOfGen = len(glob.glob(os.path.join(resDirPath,'times_*')))
- tuple strZeros = zeroBeforeStrNum(ngen, numberOfGen)
- string strSpeciesZero = 'species_'
- tuple speciesFilesZero = sorted(glob.glob(os.path.join(resDirPath,strSpeciesZero)))
- string strSpecies = 'species_'
- tuple speciesFiles = sorted(glob.glob(os.path.join(resDirPath,strSpecies)))
- list speciesFile = speciesFiles[timeFileID]
- tuple fidSpecies = open(speciesFile, 'r')
- list seq = []

- list overallConcList = []
- list overallConcListNOINFLUX = []
- int numberOfFolders = 0
- tuple pos = seq.index(key)
- tuple pos2 = seq.index(key2)
- tuple ANG_overallResMatrix = np.zeros((numberOfFolders,numberOfFolders))
- tuple ANG_overallResMatrixNOINFLUX = np.zeros((numberOfFolders,numberOfFolders))
- tuple HAM_overallResMatrix = np.zeros((numberOfFolders,numberOfFolders))
- tuple HAM_overallResMatrixNOINFLUX = np.zeros((numberOfFolders,numberOfFolders))
- tuple EUC_overallResMatrix = np.zeros((numberOfFolders,numberOfFolders))
- tuple EUC_overallResMatrixNOINFLUX = np.zeros((numberOfFolders,numberOfFolders))
- tuple vecX = np.array(lx)
- tuple vecY = np.array(ly)
- tuple tmpCos = float(np.dot(vecX,vecY) / (np.linalg.norm(vecX) ∗ np.linalg.norm(vecY)))
- int tmpHD = 0
- int tmpEU = 0
- string ndn = '_0_new_allStatResults'
- tuple newdirAllResults = os.path.join(os.curdir, ndn)
- tuple tmpZeroSaving = zeroBeforeStrNum(timeFileID, tmpNOF)
- string outFnameStat = 'speciesVector'
- tuple saveFileStat = open(outFnameStat, 'w')
- string strTypes = ''
- int cnt = 0

11.2.1 Function Documentation

11.2.1.1 def acsAttractorAnalysisInTime.zeroBeforeStrNum (*tmpI, tmpL*)

Definition at line 23 of file acsAttractorAnalysisInTime.py.

11.2.2 Variable Documentation

11.2.2.1 list acsAttractorAnalysisInTime.allConcentrations = []

Definition at line 54 of file acsAttractorAnalysisInTime.py.

11.2.2.2 list acsAttractorAnalysisInTime.allConcentrationsNOINFLUX = []

Definition at line 56 of file acsAttractorAnalysisInTime.py.

11.2.2.3 tuple acsAttractorAnalysisInTime.allSortedSpecies = []

Definition at line 53 of file acsAttractorAnalysisInTime.py.

11.2.2.4 tuple acsAttractorAnalysisInTime.allSortedSpeciesNOINFLUX = []

Definition at line 55 of file acsAttractorAnalysisInTime.py.

11.2.2.5 tuple acsAttractorAnalysisInTime.ANG_overallResMatrix = np.zeros((numberOfFolders,numberOfFolders))

Definition at line 195 of file acsAttractorAnalysisInTime.py.

11.2.2.6 tuple acsAttractorAnalysisInTime.ANG_overallResMatrixNOINFLUX = np.zeros((numberOfFolders,numberOfFolders))

Definition at line 196 of file acsAttractorAnalysisInTime.py.

11.2.2.7 int acsAttractorAnalysisInTime.cnt = 0

Definition at line 298 of file acsAttractorAnalysisInTime.py.

11.2.2.8 tuple acsAttractorAnalysisInTime.EUC_overallResMatrix = np.zeros((numberOfFolders,numberOfFolders))

Definition at line 199 of file acsAttractorAnalysisInTime.py.

11.2.2.9 tuple acsAttractorAnalysisInTime.EUC_overallResMatrixNOINFLUX = np.zeros((numberOfFolders,numberOfFolders))

Definition at line 200 of file acsAttractorAnalysisInTime.py.

11.2.2.10 tuple acsAttractorAnalysisInTime.fidSpecies = open(speciesFile, 'r')

Definition at line 96 of file acsAttractorAnalysisInTime.py.

11.2.2.11 tuple acsAttractorAnalysisInTime.HAM_overallResMatrix = np.zeros((numberOfFolders,numberOfFolders))

Definition at line 197 of file acsAttractorAnalysisInTime.py.

11.2.2.12 tuple acsAttractorAnalysisInTime.HAM_overallResMatrixNOINFLUX = np.zeros((numberOfFolders,numberOfFolders))

Definition at line 198 of file acsAttractorAnalysisInTime.py.

11.2.2.13 string acsAttractorAnalysisInTime.ndn = '_0_new_allStatResults'

Definition at line 243 of file acsAttractorAnalysisInTime.py.

11.2.2.14 tuple acsAttractorAnalysisInTime.newdirAllResults = os.path.join(os.curdir, ndn)

Definition at line 244 of file acsAttractorAnalysisInTime.py.

11.2.2.15 int acsAttractorAnalysisInTime.numberOfFolders = 0

Definition at line 123 of file acsAttractorAnalysisInTime.py.

11.2.2.16 tuple acsAttractorAnalysisInTime.numberOfGen = len(glob.glob(os.path.join(resDirPath,'times_*')))

Definition at line 73 of file acsAttractorAnalysisInTime.py.

11.2.2.17 string acsAttractorAnalysisInTime.outFnameStat = 'speciesVector'

Definition at line 266 of file acsAttractorAnalysisInTime.py.

11.2.2.18 list acsAttractorAnalysisInTime.overallConcList = [ ]

Definition at line 121 of file acsAttractorAnalysisInTime.py.

11.2.2.19 list acsAttractorAnalysisInTime.overallConcListNOINFLUX = [ ]

Definition at line 122 of file acsAttractorAnalysisInTime.py.

11.2.2.20 tuple acsAttractorAnalysisInTime.pos = seq.index(key)

Definition at line 177 of file acsAttractorAnalysisInTime.py.

11.2.2.21 tuple acsAttractorAnalysisInTime.pos2 = seq.index(key2)

Definition at line 183 of file acsAttractorAnalysisInTime.py.

11.2.2.22 tuple acsAttractorAnalysisInTime.resDirPath = os.path.abspath(os.path.join("./", "res"))

Definition at line 67 of file acsAttractorAnalysisInTime.py.

11.2.2.23 tuple acsAttractorAnalysisInTime.saveFileStat = open(outFnameStat, 'w')

Definition at line 267 of file acsAttractorAnalysisInTime.py.

11.2.2.24 list acsAttractorAnalysisInTime.seq = [ ]

Definition at line 101 of file acsAttractorAnalysisInTime.py.

11.2.2.25 list acsAttractorAnalysisInTime.speciesFile = speciesFiles[timeFileID]

Definition at line 91 of file acsAttractorAnalysisInTime.py.

11.2.2.26 tuple acsAttractorAnalysisInTime.speciesFiles = sorted(glob.glob(os.path.join(resDirPath,strSpecies)))

Definition at line 86 of file acsAttractorAnalysisInTime.py.

11.2.2.27 tuple acsAttractorAnalysisInTime.speciesFilesZero = sorted(glob.glob(os.path.join(resDirPath,strSpeciesZero)))

Definition at line 81 of file acsAttractorAnalysisInTime.py.

11.2.2.28 tuple acsAttractorAnalysisInTime.StrPath = os.path.abspath(StrPath)

Definition at line 41 of file acsAttractorAnalysisInTime.py.

11.2.2.29 string acsAttractorAnalysisInTime.strSpecies = 'species_'

Definition at line 83 of file acsAttractorAnalysisInTime.py.

11.2.2.30 string acsAttractorAnalysisInTime.strSpeciesZero = 'species_'

Definition at line 80 of file acsAttractorAnalysisInTime.py.

11.2.2.31 string acsAttractorAnalysisInTime.strTypes = ''

Definition at line 269 of file acsAttractorAnalysisInTime.py.

11.2.2.32 tuple acsAttractorAnalysisInTime.strZeros = zeroBeforeStrNum(ngen, numberOfGen)

Definition at line 77 of file acsAttractorAnalysisInTime.py.

11.2.2.33 tuple acsAttractorAnalysisInTime.tmpCos = float(np.dot(vecX,vecY) / (np.linalg.norm(vecX) * np.linalg.norm(vecY)))

Definition at line 208 of file acsAttractorAnalysisInTime.py.

11.2.2.34 tuple acsAttractorAnalysisInTime.tmpDirs = sort(os.listdir(StrPath))

Definition at line 52 of file acsAttractorAnalysisInTime.py.

11.2.2.35 int acsAttractorAnalysisInTime.tmpEU = 0

Definition at line 213 of file acsAttractorAnalysisInTime.py.

11.2.2.36 int acsAttractorAnalysisInTime.tmpHD = 0

Definition at line 212 of file acsAttractorAnalysisInTime.py.

11.2.2.37 tuple acsAttractorAnalysisInTime.tmpZeroSaving = zeroBeforeStrNum(timeFileID, tmpNOF)

Definition at line 263 of file acsAttractorAnalysisInTime.py.

11.2.2.38 tuple acsAttractorAnalysisInTime.today = dt.date.today()

Definition at line 43 of file acsAttractorAnalysisInTime.py.

11.2.2.39 tuple acsAttractorAnalysisInTime.totDirName = os.path.join(StrPath,tmpDir)

Definition at line 63 of file acsAttractorAnalysisInTime.py.

11.2.2.40 tuple acsAttractorAnalysisInTime.vecX = np.array(lx)

Definition at line 205 of file acsAttractorAnalysisInTime.py.

11.2.2.41 tuple acsAttractorAnalysisInTime.vecY = np.array(ly)

Definition at line 206 of file acsAttractorAnalysisInTime.py.

11.3 acsBufferedFluxes Namespace Reference

Functions

- def zeroBeforeStrNum

Variables

- tuple parser
- tuple args = parser.parse_args()
- tuple strPath = os.path.abspath(args.strPath)
- tuple tmpDirs = sort(os.listdir(StrPath))
- int _CONDENSATION_ = 0
- int _CLEAVAGE_ = 1
- int _ENDOCONDENSATION_ = 7
- int _ENDOCLEAVAGE_ = 6
- int _SPONTCONDENSATION_ = 10
- int _SPONTCLEAVAGE_ = 11
- int chemistry = 1
- string ndn = '_0_new_allStatResults'
- tuple newdirAllResults = os.path.join(StrPath, ndn)
- tuple totDirName = os.path.join(StrPath,tmpDir)
- tuple resDirPath = os.path.abspath(os.path.join("./", "res"))
- tuple numberOfGen = len(glob.glob(os.path.join(resDirPath,'times_*')))
- string tmpFluxFile = 'fluxDynamics_'
- tuple tmpSpeciesStatsSummaryNameFID = open(tmpFluxFile, 'w')
- tuple strZeros = zeroBeforeStrNum(ngen, numberOfGen)
- string strSpeciesZero = 'species_'
- tuple speciesFilesZero = sorted(glob.glob(os.path.join(resDirPath,strSpeciesZero)))
- string strSpecies = 'species_'
- tuple speciesFiles = sorted(glob.glob(os.path.join(resDirPath,strSpecies)))
- list lastfilespecies = speciesFiles[-1]
- tuple fidSpecies = open(lastfilespecies, 'r')
- list flux_seq = []
- tuple counters = np.zeros((nSpecies,(len(seq)*2)+7))
- string strRctPar = 'reactions_parameters_'
- tuple fidRctPar = open(strRctPar, 'r')
- int totIN = 0
- int totOUT = 0
- int deltaIO = 0
- int totBIN = 0
- int totBOUT = 0
- tuple rctTime = int(tmpRctT)
- tuple rctType = int(tmpRctType)
- tuple cat = int(tmpCat)
- tuple S1 = int(tmpS1)
- tuple S2 = int(tmpS2)
- tuple S3 = int(tmpS3)
- string tmpFileName = 'speciesStats_'
- tuple tmpFileNameFID = open(tmpFileName, 'w')
- int ID = 0
- string tmpStr = 'Total Number of Reactions\t\t\t\t'

11.3.1 Function Documentation

11.3.1.1 def acsBufferedFluxes.zeroBeforeStrNum (*tmpI, tmpL*)

Definition at line 20 of file acsBufferedFluxes.py.

11.3.2 Variable Documentation

11.3.2.1 int acsBufferedFluxes._CLEAVAGE_ = 1

Definition at line 42 of file acsBufferedFluxes.py.

11.3.2.2 int acsBufferedFluxes._CONDENSATION_ = 0

Definition at line 41 of file acsBufferedFluxes.py.

11.3.2.3 int acsBufferedFluxes._ENDOCLEAVAGE_ = 6

Definition at line 44 of file acsBufferedFluxes.py.

11.3.2.4 int acsBufferedFluxes._ENDOCONDENSATION_ = 7

Definition at line 43 of file acsBufferedFluxes.py.

11.3.2.5 int acsBufferedFluxes._SPONTCLEAVAGE_ = 11

Definition at line 46 of file acsBufferedFluxes.py.

11.3.2.6 int acsBufferedFluxes._SPONTCONDENSATION_ = 10

Definition at line 45 of file acsBufferedFluxes.py.

11.3.2.7 tuple acsBufferedFluxes.args = parser.parse_args()

Definition at line 34 of file acsBufferedFluxes.py.

11.3.2.8 tuple acsBufferedFluxes.cat = int(tmpCat)

Definition at line 133 of file acsBufferedFluxes.py.

11.3.2.9 int acsBufferedFluxes.chemistry = 1

Definition at line 50 of file acsBufferedFluxes.py.

11.3.2.10 tuple acsBufferedFluxes.counters = np.zeros((nSpecies,(len(seq)∗2)+7))

Definition at line 111 of file acsBufferedFluxes.py.

11.3.2.11 int acsBufferedFluxes.deltaIO = 0

Definition at line 123 of file acsBufferedFluxes.py.

11.3.2.12 tuple acsBufferedFluxes.fidRctPar = open(**strRctPar**, 'r')

Definition at line 117 of file acsBufferedFluxes.py.

11.3.2.13 tuple acsBufferedFluxes.fidSpecies = open(**lastfilespecies**, 'r')

Definition at line 96 of file acsBufferedFluxes.py.

11.3.2.14 list acsBufferedFluxes.flux_seq = [ ]

Definition at line 100 of file acsBufferedFluxes.py.

11.3.2.15 int acsBufferedFluxes.ID = 0

Definition at line 169 of file acsBufferedFluxes.py.

11.3.2.16 list acsBufferedFluxes.lastfilespecies = **speciesFiles**[-1]

Definition at line 92 of file acsBufferedFluxes.py.

11.3.2.17 string acsBufferedFluxes.ndn = '_0_new_allStatResults'

Definition at line 53 of file acsBufferedFluxes.py.

11.3.2.18 tuple acsBufferedFluxes.newdirAllResults = os.path.join(StrPath, **ndn**)

Definition at line 54 of file acsBufferedFluxes.py.

11.3.2.19 tuple acsBufferedFluxes.numberOfGen = len(glob.glob(os.path.join(**resDirPath**,'times_*')))

Definition at line 73 of file acsBufferedFluxes.py.

11.3.2.20 tuple acsBufferedFluxes.parser

Initial value:

```
1 = ArgumentParser(
2                                 description='This script characterize the buffered flux dynamics.'
3                                 , epilog='''Buffered flux dynamics. ''')
```

Definition at line 30 of file acsBufferedFluxes.py.

11.3.2.21 tuple acsBufferedFluxes.rctTime = int(tmpRctT)

Definition at line 131 of file acsBufferedFluxes.py.

11.3.2.22 tuple acsBufferedFluxes.rctType = int(tmpRctType)

Definition at line 132 of file acsBufferedFluxes.py.

11.3.2.23 tuple acsBufferedFluxes.resDirPath = os.path.abspath(os.path.join("./", "res"))

Definition at line 67 of file acsBufferedFluxes.py.

11.3.2.24 tuple acsBufferedFluxes.S1 = int(tmpS1)

Definition at line 134 of file acsBufferedFluxes.py.

11.3.2.25 tuple acsBufferedFluxes.S2 = int(tmpS2)

Definition at line 135 of file acsBufferedFluxes.py.

11.3.2.26 tuple acsBufferedFluxes.S3 = int(tmpS3)

Definition at line 136 of file acsBufferedFluxes.py.

11.3.2.27 tuple acsBufferedFluxes.speciesFiles = sorted(glob.glob(os.path.join(resDirPath,strSpecies)))

Definition at line 89 of file acsBufferedFluxes.py.

11.3.2.28 tuple acsBufferedFluxes.speciesFilesZero = sorted(glob.glob(os.path.join(resDirPath,strSpeciesZero)))

Definition at line 84 of file acsBufferedFluxes.py.

11.3.2.29 tuple acsBufferedFluxes.strPath = os.path.abspath(args.strPath)

Definition at line 37 of file acsBufferedFluxes.py.

11.3.2.30 string acsBufferedFluxes.strRctPar = 'reactions_parameters_'

Definition at line 113 of file acsBufferedFluxes.py.

11.3.2.31 string acsBufferedFluxes.strSpecies = 'species_'

Definition at line 86 of file acsBufferedFluxes.py.

11.3.2.32 string acsBufferedFluxes.strSpeciesZero = 'species_'

Definition at line 83 of file acsBufferedFluxes.py.

11.3.2.33 tuple acsBufferedFluxes.strZeros = zeroBeforeStrNum(ngen, numberOfGen)

Definition at line 81 of file acsBufferedFluxes.py.

11.3.2.34 tuple acsBufferedFluxes.tmpDirs = sort(os.listdir(StrPath))

Definition at line 38 of file acsBufferedFluxes.py.

11.3.2.35 string acsBufferedFluxes.tmpFileName = 'speciesStats_'

Definition at line 166 of file acsBufferedFluxes.py.

11.3.2.36 tuple acsBufferedFluxes.tmpFileNameFID = open(tmpFileName, 'w')

Definition at line 168 of file acsBufferedFluxes.py.

11.3.2.37 string acsBufferedFluxes.tmpFluxFile = 'fluxDynamics_'

Definition at line 78 of file acsBufferedFluxes.py.

11.3.2.38 tuple acsBufferedFluxes.tmpSpeciesStatsSummaryNameFID = open(tmpFluxFile, 'w')

Definition at line 79 of file acsBufferedFluxes.py.

11.3.2.39 tuple acsBufferedFluxes.tmpStr = 'Total Number of Reactions\t\t\t\t'

Definition at line 170 of file acsBufferedFluxes.py.

11.3.2.40 int acsBufferedFluxes.totBIN = 0

Definition at line 124 of file acsBufferedFluxes.py.

11.3.2.41 int acsBufferedFluxes.totBOUT = 0

Definition at line 125 of file acsBufferedFluxes.py.

11.3.2.42 tuple acsBufferedFluxes.totDirName = os.path.join(StrPath,tmpDir)

Definition at line 63 of file acsBufferedFluxes.py.

11.3.2.43 int acsBufferedFluxes.totIN = 0

Definition at line 121 of file acsBufferedFluxes.py.

11.3.2.44 int acsBufferedFluxes.totOUT = 0

Definition at line 122 of file acsBufferedFluxes.py.

11.4 acsDynStatInTime Namespace Reference

Variables

- tuple parser

- tuple args = parser.parse_args()
- tuple strPath = os.path.abspath(args.strPath)
- string strSubStrKey = '*'
- tuple filesToAnal = sorted(glob.glob(os.path.join(strPath,strSubStrKey)))
- tuple lenFilesToAnal = len(filesToAnal)
- int numOfTraj = 0
- tuple x = np.array(map(list, np.loadtxt(fileToAnal, str)), float)
- xsize = x.shape
- list run4Chem = xsize[0]
- tuple y = np.zeros((lenFilesToAnal,numOfTraj))
- int pos = 0
- int chem = 1
- string tmpHead = ''
- string filename = "_arranged_"

11.4.1 Variable Documentation

11.4.1.1 tuple acsDynStatInTime.args = parser.parse_args()

Definition at line 24 of file acsDynStatInTime.py.

11.4.1.2 int acsDynStatInTime.chem = 1

Definition at line 47 of file acsDynStatInTime.py.

11.4.1.3 string acsDynStatInTime.filename = "_arranged_"

Definition at line 66 of file acsDynStatInTime.py.

11.4.1.4 tuple acsDynStatInTime.filesToAnal = sorted(glob.glob(os.path.join(strPath,strSubStrKey)))

Definition at line 34 of file acsDynStatInTime.py.

11.4.1.5 tuple acsDynStatInTime.lenFilesToAnal = len(filesToAnal)

Definition at line 35 of file acsDynStatInTime.py.

11.4.1.6 tuple acsDynStatInTime.numOfTraj = 0

Definition at line 36 of file acsDynStatInTime.py.

11.4.1.7 tuple acsDynStatInTime.parser

Initial value:

```
1 = ArgumentParser(
2                               description='This script re-arrange results in a more friendly way from the
      angle analysis in time.'
3                               , epilog='''File with angle trajectories are created. ''')
```

Definition at line 17 of file acsDynStatInTime.py.

11.4.1.8 int acsDynStatInTime.pos = 0

Definition at line 46 of file acsDynStatInTime.py.

11.4.1.9 list acsDynStatInTime.run4Chem = xsize[0]

Definition at line 42 of file acsDynStatInTime.py.

11.4.1.10 tuple acsDynStatInTime.strPath = os.path.abspath(args.strPath)

Definition at line 27 of file acsDynStatInTime.py.

11.4.1.11 string acsDynStatInTime.strSubStrKey = '*'

Definition at line 33 of file acsDynStatInTime.py.

11.4.1.12 string acsDynStatInTime.tmpHead = ''

Definition at line 48 of file acsDynStatInTime.py.

11.4.1.13 tuple acsDynStatInTime.x = np.array(map(list, np.loadtxt(fileToAnal, str)), float)

Definition at line 39 of file acsDynStatInTime.py.

11.4.1.14 acsDynStatInTime.xsize = x.shape

Definition at line 40 of file acsDynStatInTime.py.

11.4.1.15 tuple acsDynStatInTime.y = np.zeros((lenFilesToAnal,numOfTraj))

Definition at line 44 of file acsDynStatInTime.py.

11.5 acsFromWim2Carness Namespace Reference

Variables

- tuple parser
- tuple args = parser.parse_args()
- string ndn = '_0_new_allStatResults'
- tuple newdirAllResults = os.path.join(args.strOut, ndn)
- tuple fname_initRafRes = os.path.join(newdirAllResults, '0_initRafAnalysis.csv')
- tuple fname_initRafResLIST = os.path.join(newdirAllResults, '0_initRafAnalysisLIST.csv')
- tuple fname_initRafResALL = os.path.join(newdirAllResults, '0_initRafAnalysisALL.csv')
- tuple fid_initRafRes = open(fname_initRafRes, 'w')
- tuple fid_initRafResLIST = open(fname_initRafResLIST, 'w')
- tuple fid_initRafResALL = open(fname_initRafResALL, 'w')
- string strToWrite = "Folder\tP\tAC\tM\tRAFsize\tClosure\tCats\tuRAF\n"
- tuple f = open(args.wimFile)
- tuple lines = f.readlines()
- int procedure = 0

- list speciesList = []
- list foodList = []
- int rct = 0
- tuple l = line.split()
- tuple catNums = len(l)
- tuple cats = np.vstack([cats,(int(rct), int(speciesList.index(l[6+catNums-1][0:len(l[6+catNums-1])-1])), int(rct), int(0), int(5), int(25), int(5), int(1))])
- tuple rafsets = raf.rafComputation(fid_initRafRes, fid_initRafResALL, fid_initRafResLIST, 'tmpDir', 0, rct/float(len(speciesList)), rcts, cats, foodList, 10)

11.5.1 Variable Documentation

11.5.1.1 tuple acsFromWim2Carness.args = parser.parse_args()

Definition at line 33 of file acsFromWim2Carness.py.

11.5.1.2 tuple acsFromWim2Carness.catNums = len(l)

Definition at line 100 of file acsFromWim2Carness.py.

11.5.1.3 tuple acsFromWim2Carness.cats = np.vstack([cats,(int(rct), int(speciesList.index(l[6+catNums-1][0:len(l[6+catNums-1])-1])), int(rct), int(0), int(5), int(25), int(5), int(1))])

Definition at line 110 of file acsFromWim2Carness.py.

11.5.1.4 tuple acsFromWim2Carness.f = open(args.wimFile)

Definition at line 54 of file acsFromWim2Carness.py.

11.5.1.5 tuple acsFromWim2Carness.fid_initRafRes = open(fname_initRafRes, 'w')

Definition at line 48 of file acsFromWim2Carness.py.

11.5.1.6 tuple acsFromWim2Carness.fid_initRafResALL = open(fname_initRafResALL, 'w')

Definition at line 50 of file acsFromWim2Carness.py.

11.5.1.7 tuple acsFromWim2Carness.fid_initRafResLIST = open(fname_initRafResLIST, 'w')

Definition at line 49 of file acsFromWim2Carness.py.

11.5.1.8 tuple acsFromWim2Carness.fname_initRafRes = os.path.join(newdirAllResults, '0_initRafAnalysis.csv')

Definition at line 45 of file acsFromWim2Carness.py.

11.5.1.9 tuple acsFromWim2Carness.fname_initRafResALL = os.path.join(newdirAllResults, '0_initRafAnalysisALL.csv')

Definition at line 47 of file acsFromWim2Carness.py.

11.5.1.10 tuple acsFromWim2Carness.fname_initRafResLIST = os.path.join(newdirAllResults, '0_initRafAnalysisLIST.csv')

Definition at line 46 of file acsFromWim2Carness.py.

11.5.1.11 list acsFromWim2Carness.foodList = [ ]

Definition at line 58 of file acsFromWim2Carness.py.

11.5.1.12 tuple acsFromWim2Carness.l = line.split()

Definition at line 74 of file acsFromWim2Carness.py.

11.5.1.13 tuple acsFromWim2Carness.lines = f.readlines()

Definition at line 55 of file acsFromWim2Carness.py.

11.5.1.14 string acsFromWim2Carness.ndn = '_0_new_allStatResults'

Definition at line 36 of file acsFromWim2Carness.py.

11.5.1.15 tuple acsFromWim2Carness.newdirAllResults = os.path.join(args.strOut, ndn)

Definition at line 37 of file acsFromWim2Carness.py.

11.5.1.16 tuple acsFromWim2Carness.parser

Initial value:

```
1 = ArgumentParser(
2                                 description='From WIM format to my format'
3                                 , epilog='''From WIM format to my format. ''')
```

Definition at line 26 of file acsFromWim2Carness.py.

11.5.1.17 int acsFromWim2Carness.procedure = 0

Definition at line 56 of file acsFromWim2Carness.py.

11.5.1.18 tuple acsFromWim2Carness.rafsets = raf.rafComputation(fid_initRafRes, fid_initRafResALL, fid_initRafResLIST, 'tmpDir', 0, rct/float(len(speciesList)), rcts, cats, foodList, 10)

Definition at line 116 of file acsFromWim2Carness.py.

11.5.1.19 int acsFromWim2Carness.rct = 0

Definition at line 59 of file acsFromWim2Carness.py.

11.5.1.20 list acsFromWim2Carness.speciesList = [ ]

Definition at line 57 of file acsFromWim2Carness.py.

11.5.1.21 string acsFromWim2Carness.strToWrite = "Folder\tP\tAC\tM\tRAFsize\tClosure\tCats\tuRAF\n"

Definition at line 51 of file acsFromWim2Carness.py.

11.6 acsRAFanalysis Namespace Reference

Variables

- tuple parser
- tuple args = parser.parse_args()
- tuple strPath = os.path.abspath(args.strPath)
- int _CLOSE_ = 0
- int _PROTO_ = 1
- int _CSTR_ = 2
- tuple conf = rf.readConfFile(strPath)
- tuple closure = dm.generateFluxList(strPath, sysType)
- tuple foodSet = deepcopy(closure)
- tuple rcts = rf.loadAllData(strPath,'_acsreactions.csv')
- tuple cats = rf.loadAllData(strPath,'_acscatalysis.csv')
- tuple RA = raf.RAcondition(strPath,closure,rcts,cats)
- tuple RAF = raf.Fcondition(strPath,closure,RA,rcts)
- tuple RAFlpre = len(RAF)
- list redRcts = rcts[RAF,:]
- int RAFlpost = 0
- tuple catalists = raf.findCatforRAF(cats, RAF, closure)

11.6.1 Variable Documentation

11.6.1.1 int acsRAFanalysis._CLOSE_ = 0

Definition at line 31 of file acsRAFanalysis.py.

11.6.1.2 int acsRAFanalysis._CSTR_ = 2

Definition at line 33 of file acsRAFanalysis.py.

11.6.1.3 int acsRAFanalysis._PROTO_ = 1

Definition at line 32 of file acsRAFanalysis.py.

11.6.1.4 tuple acsRAFanalysis.args = parser.parse_args()

Definition at line 26 of file acsRAFanalysis.py.

11.6.1.5 tuple acsRAFanalysis.catalists = raf.findCatforRAF(cats, RAF, closure)

Definition at line 76 of file acsRAFanalysis.py.

11.6.1.6 tuple acsRAFanalysis.cats = rf.loadAllData(strPath,'_acscatalysis.csv')

Definition at line 45 of file acsRAFanalysis.py.

11.6.1.7 tuple acsRAFanalysis.closure = dm.generateFluxList(strPath, sysType)

Definition at line 41 of file acsRAFanalysis.py.

11.6.1.8 tuple acsRAFanalysis.conf = rf.readConfFile(strPath)

Definition at line 34 of file acsRAFanalysis.py.

11.6.1.9 tuple acsRAFanalysis.foodSet = deepcopy(closure)

Definition at line 42 of file acsRAFanalysis.py.

11.6.1.10 tuple acsRAFanalysis.parser

Initial value:

```
1 = ArgumentParser(
2                                 description='Script to perform a RAF analysis.'
3                                 , epilog='''File with angle trajectories are created. ''')
```

Definition at line 22 of file acsRAFanalysis.py.

11.6.1.11 tuple acsRAFanalysis.RA = raf.RAcondition(strPath,closure,rcts,cats)

Definition at line 50 of file acsRAFanalysis.py.

11.6.1.12 tuple acsRAFanalysis.RAF = raf.Fcondition(strPath,closure,RA,rcts)

Definition at line 52 of file acsRAFanalysis.py.

11.6.1.13 tuple acsRAFanalysis.RAFlpost = 0

Definition at line 60 of file acsRAFanalysis.py.

11.6.1.14 tuple acsRAFanalysis.RAFlpre = len(RAF)

Definition at line 53 of file acsRAFanalysis.py.

11.6.1.15 tuple acsRAFanalysis.rcts = rf.loadAllData(strPath,'_acsreactions.csv')

Definition at line 44 of file acsRAFanalysis.py.

11.6.1.16 list acsRAFanalysis.redRcts = rcts[RAF,:]

Definition at line 55 of file acsRAFanalysis.py.

11.6.1.17 tuple acsRAFanalysis.strPath = os.path.abspath(args.strPath)

Definition at line 29 of file acsRAFanalysis.py.

11.7 acsSCCanalysis Namespace Reference

Functions

- def zeroBeforeStrNum
- def loadReactionGraph
- def loadSpecificReactionGraph
- def loadSpecificReactionSubGraph
- def saveGraphToFile
- def saveGraphSUBToFile
- def saveNrgToFile
- def saveGillToFile

Variables

- tuple StrPath = os.path.abspath(StrPath)
- tuple today = dt.date.today()
- tuple mswindows = (sys.platform == "win32")
- string cmdFileName = StrPath+'/'
- tuple cmdFileFid = open(cmdFileName, 'a')
- string strToWrite = "\tReaction Probability"
- int initilizeGraphStructure = 0
- tuple graph = loadSpecificReactionGraph()
- tuple graphSUB = loadSpecificReactionSubGraph()
- tuple tmpRctFileToLoadSplitString = tmpRctFileToLoad.split("_")
- tuple initRctId = int(tmpRctFileToLoadSplitString[len(tmpRctFileToLoadSplitString)-2])
- list initTempTime = tmpRctFileToLoadSplitString[len(tmpRctFileToLoadSplitString)-1]
- tuple initTime = float(initTempTime[0:len(initTempTime)-4])
- tuple tmpDirs = sort(os.listdir(StrPath))
- tuple totDirName = os.path.join(StrPath,tmpDir)
- tuple resDirPath = os.path.abspath(os.path.join("./", "res"))
- tuple numberOfGen = len(glob.glob(os.path.join(resDirPath,'times_*')))
- tuple strZeros = zeroBeforeStrNum(nGen, numberOfGen)
- string paramFile = "acsm2s.conf"
- int simFolder = 0
- tuple fid = open(paramFile, 'r')
- tuple strLine = line.split('=')
- tuple rp = float(strLine[1])
- tuple decayTime = int(strLine[1])
- tuple totTimes = int(strLine[1])
- tuple nrgType = int(strLine[1])
- tuple totalRcts = int(strLine[1])
- tuple nrgConc = float(strLine[1])
- tuple influx_rate = float(strLine[1])
- tuple maxLOut = float(strLine[1])
- int _ANALTIMEINTERAVAL_ = totTimes/10
- int _ANALTIMEINTERAVALNOSAVE_ = totTimes/100
- int nrg = 1
- tuple speciesInFlux = range(0,int(pow(2,(maxLOut+1)) - 2))
- tuple fidflux = open('_acsinflux.csv', 'r')
- tuple overThreshold = float(0)
- tuple overThresholdTOT = float(0)
- string tmpFilesToSearch = 'species_'
- tuple speciesFiles = sorted(glob.glob(os.path.join(resDirPath,tmpFilesToSearch)))

- list speciesFile = speciesFiles[-1]
- tuple fidSpecies = open(speciesFile, 'r')
- int ok = 0
- list IDsOverThreshold = []
- list concVec = []
- tuple index = int(tmpID)
- tuple conc = float(tmpConc)
- tuple cpxCut = int(tmpCpxCut)
- tuple age = float(tmpAge)
- realT = threshold
- string folderCat = '__0_iGraph_CAT_'
- string folderSub = '__0_iGraph_SUB_'
- tuple newdir = os.path.join(os.curdir, folderCat)
- tuple newdirSUB = os.path.join(os.curdir, folderSub)
- string filextPre = '_'
- string rctParamFileQ = 'reactions_parameters_'
- tuple rctParamFile = sorted(glob.glob(os.path.join(resDirPath,rctParamFileQ)))
- float rctIDshow = 1.0
- float rctIDshowNoSave = 1.0
- rctID = initRctId
- int previousTime = 0
- int endo_condensation_counter = 0
- int condensation_counter = 0
- int endo_cleavage_counter = 0
- int cleavage_counter = 0
- tuple nrgTimeSeries = np.array([[0,0,0]])
- tuple gillTimeSeries = np.array([[0,0,0,0]])
- int tmpFlagLastRctSaved = 0
- tuple reaction = int(tmpReaction)
- tuple rtime = float(tmpTime)
- tuple cc = int(tmpcc)
- tuple cat = int(tmpCat)
- tuple mol_I = int(tmpMol_I)
- tuple mol_II = int(tmpMol_II)
- tuple mol_III = int(tmpMol_III)
- tuple loadedMolsConc = float(tmpLoadedMolsConc)
- tuple loadedMols = int(tmpLoadedMols)
- tuple gillMean = float(tmpGillMean)
- tuple gillSD = float(tmpGillSD)
- tuple gillEntropy = float(tmpGillEntropy)
- tuple newSpeciesCreationProb = float(tmpNSCprob)
- tuple reverseProbability = float(tmpRevProb)
- int printTemporalMessage = 1
- timeInterval = rtime-previousTime
- tuple position = ((graph[:,0] == cat) & (graph[:,1] == mol_I))
- int realSccs = 0
- tuple Gcatpro = nx.DiGraph()
- tuple scc = nx.strongly_connected_components(Gcatpro)
- tuple sccN = nx.number_strongly_connected_components(Gcatpro)
- tuple selfLoops = Gcatpro.number_of_selfloops()
- tuple selfLoopsEgdes = Gcatpro.selfloop_edges()
- int prod_inSCC = 0
- int prod_chain = 0
- int prod_bySCC = 0
- int prod_overlap = 0

- int sccID = 0
- int autocatalysis = 0
- int prod_inSCC_weight = 0
- int prod_chain_weight = 0
- int prod_bySCC_weight = 0
- int prod_overlap_weight = 0
- int self_loop_weight = 0
- int conc_inSCC = 0
- int conc_chain = 0
- int conc_bySCC = 0
- int conc_overLap = 0
- int conc_selfCat = 0
- int wasteSpecies = 0
- int alreadyAdded_ACS = 0
- int alreadyAdded_leaves = 0
- int alreadyAdded_chain = 0
- int tmpProd_chain = 0
- tuple incomingNodes = Gcatpro.predecessors(IdsOT)
- int tempProd_chain_weight = 0
- int noInAcs = 1
- int inSCCFlag = 0
- list weightToDistribute = graph[((graph[:,0] == sngInNode) & (graph[:,1] == IdsOT)),5]
- tuple inDegreeMean = mean(Gcatpro.in_degree().values())
- tuple meanOverThreshold = float(overThreshold)

11.7.1 Function Documentation

11.7.1.1 def acsSCCanalysis.loadReactionGraph ( )

Definition at line 31 of file acsSCCanalysis.py.

11.7.1.2 def acsSCCanalysis.loadSpecificReactionGraph ( )

Definition at line 64 of file acsSCCanalysis.py.

11.7.1.3 def acsSCCanalysis.loadSpecificReactionSubGraph ( )

Definition at line 93 of file acsSCCanalysis.py.

11.7.1.4 def acsSCCanalysis.saveGillToFile ( )

Definition at line 154 of file acsSCCanalysis.py.

11.7.1.5 def acsSCCanalysis.saveGraphSUBToFile ( )

Definition at line 133 of file acsSCCanalysis.py.

Here is the call graph for this function:

11.7.1.6 def acsSCCanalysis.saveGraphToFile ( )

Definition at line 122 of file acsSCCanalysis.py.

Here is the call graph for this function:

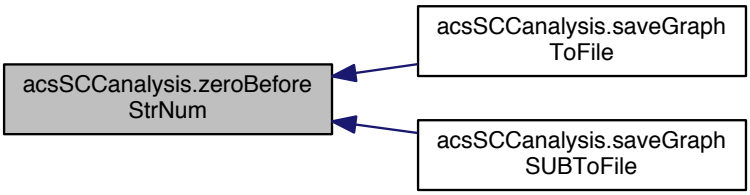
11.7.1.7 def acsSCCanalysis.saveNrgToFile ( )

Definition at line 144 of file acsSCCanalysis.py.

11.7.1.8 def acsSCCanalysis.zeroBeforeStrNum (*tmpI, tmpL*)

Definition at line 22 of file acsSCCanalysis.py.

Here is the caller graph for this function:



11.7.2 Variable Documentation

11.7.2.1 int acsSCCanalysis._ANALTIMEINTERAVAL_ = totTimes/10

Definition at line 269 of file acsSCCanalysis.py.

11.7.2.2 int acsSCCanalysis._ANALTIMEINTERAVALNOSAVE_ = totTimes/100

Definition at line 270 of file acsSCCanalysis.py.

11.7.2.3 tuple acsSCCanalysis.age = float(tmpAge)

Definition at line 329 of file acsSCCanalysis.py.

11.7.2.4 int acsSCCanalysis.alreadyAdded_ACS = 0

Definition at line 600 of file acsSCCanalysis.py.

11.7.2.5 int acsSCCanalysis.alreadyAdded_chain = 0

Definition at line 602 of file acsSCCanalysis.py.

11.7.2.6 int acsSCCanalysis.alreadyAdded_leaves = 0

Definition at line 601 of file acsSCCanalysis.py.

11.7.2.7 int acsSCCanalysis.autocatalysis = 0

Definition at line 573 of file acsSCCanalysis.py.

11.7.2.8 tuple acsSCCanalysis.cat = int(tmpCat)

Definition at line 411 of file acsSCCanalysis.py.

11.7.2.9 tuple acsSCCanalysis.cc = int(tmpcc)

Definition at line 410 of file acsSCCanalysis.py.

11.7.2.10 int acsSCCanalysis.cleavage_counter = 0

Definition at line 392 of file acsSCCanalysis.py.

11.7.2.11 tuple acsSCCanalysis.cmdFileFid = open(cmdFileName, 'a')

Definition at line 185 of file acsSCCanalysis.py.

11.7.2.12 string acsSCCanalysis.cmdFileName = StrPath+'/'

Definition at line 182 of file acsSCCanalysis.py.

11.7.2.13 tuple acsSCCanalysis.conc = float(tmpConc)

Definition at line 327 of file acsSCCanalysis.py.

11.7.2.14 int acsSCCanalysis.conc_bySCC = 0

Definition at line 581 of file acsSCCanalysis.py.

11.7.2.15 int acsSCCanalysis.conc_chain = 0

Definition at line 580 of file acsSCCanalysis.py.

11.7.2.16 int acsSCCanalysis.conc_inSCC = 0

Definition at line 579 of file acsSCCanalysis.py.

11.7.2.17 int acsSCCanalysis.conc_overLap = 0

Definition at line 582 of file acsSCCanalysis.py.

11.7.2.18 int acsSCCanalysis.conc_selfCat = 0

Definition at line 583 of file acsSCCanalysis.py.

11.7.2.19 list acsSCCanalysis.concVec = [ ]

Definition at line 318 of file acsSCCanalysis.py.

11.7.2.20 int acsSCCanalysis.condensation_counter = 0

Definition at line 390 of file acsSCCanalysis.py.

11.7.2.21 tuple acsSCCanalysis.cpxCut = int(tmpCpxCut)

Definition at line 328 of file acsSCCanalysis.py.

11.7.2.22 tuple acsSCCanalysis.decayTime = int(strLine[1])

Definition at line 255 of file acsSCCanalysis.py.

11.7.2.23 int acsSCCanalysis.endo_cleavage_counter = 0

Definition at line 391 of file acsSCCanalysis.py.

11.7.2.24 int acsSCCanalysis.endo_condensation_counter = 0

Definition at line 389 of file acsSCCanalysis.py.

11.7.2.25 tuple acsSCCanalysis.fid = open(paramFile, 'r')

Definition at line 244 of file acsSCCanalysis.py.

11.7.2.26 tuple acsSCCanalysis.fidflux = open('_acsinflux.csv', 'r')

Definition at line 285 of file acsSCCanalysis.py.

11.7.2.27 tuple acsSCCanalysis.fidSpecies = open(speciesFile, 'r')

Definition at line 311 of file acsSCCanalysis.py.

11.7.2.28 string acsSCCanalysis.filextPre = '_'

Definition at line 376 of file acsSCCanalysis.py.

11.7.2.29 string acsSCCanalysis.folderCat = '__0_iGraph_CAT_'

Definition at line 348 of file acsSCCanalysis.py.

11.7.2.30 string acsSCCanalysis.folderSub = '__0_iGraph_SUB_'

Definition at line 349 of file acsSCCanalysis.py.

11.7.2.31 tuple acsSCCanalysis.Gcatpro = nx.DiGraph()

Definition at line 555 of file acsSCCanalysis.py.

11.7.2.32 tuple acsSCCanalysis.gillEntropy = float(tmpGillEntropy)

Definition at line 419 of file acsSCCanalysis.py.

11.7.2.33 tuple acsSCCanalysis.gillMean = float(tmpGillMean)

Definition at line 417 of file acsSCCanalysis.py.

11.7.2.34 tuple acsSCCanalysis.gillSD = float(tmpGillSD)

Definition at line 418 of file acsSCCanalysis.py.

11.7.2.35 tuple acsSCCanalysis.gillTimeSeries = np.array([[0,0,0,0]])

Definition at line 399 of file acsSCCanalysis.py.

11.7.2.36 tuple acsSCCanalysis.graph = loadSpecificReactionGraph()

Definition at line 193 of file acsSCCanalysis.py.

11.7.2.37 tuple acsSCCanalysis.graphSUB = loadSpecificReactionSubGraph()

Definition at line 194 of file acsSCCanalysis.py.

11.7.2.38 list acsSCCanalysis.IDsOverThreshold = [ ]

Definition at line 317 of file acsSCCanalysis.py.

11.7.2.39 tuple acsSCCanalysis.incomingNodes = Gcatpro.predecessors(IdsOT)

Definition at line 605 of file acsSCCanalysis.py.

11.7.2.40 tuple acsSCCanalysis.inDegreeMean = mean(Gcatpro.in_degree().values())

Definition at line 682 of file acsSCCanalysis.py.

11.7.2.41 tuple acsSCCanalysis.index = int(tmpID)

Definition at line 326 of file acsSCCanalysis.py.

11.7.2.42 tuple acsSCCanalysis.influx_rate = float(strLine[1])

Definition at line 264 of file acsSCCanalysis.py.

11.7.2.43 int acsSCCanalysis.initilizeGraphStructure = 0

Definition at line 189 of file acsSCCanalysis.py.

11.7.2.44 int acsSCCanalysis.initRctId = int(tmpRctFileToLoadSplitString[len(tmpRctFileToLoadSplitString)-2])

Definition at line 198 of file acsSCCanalysis.py.

11.7.2.45 list acsSCCanalysis.initTempTime = tmpRctFileToLoadSplitString[len(tmpRctFileToLoadSplitString)-1]

Definition at line 199 of file acsSCCanalysis.py.

11.7.2.46 int acsSCCanalysis.initTime = float(initTempTime[0:len(initTempTime)-4])

Definition at line 200 of file acsSCCanalysis.py.

11.7.2.47 int acsSCCanalysis.inSCCFlag = 0

Definition at line 616 of file acsSCCanalysis.py.

11.7.2.48 tuple acsSCCanalysis.loadedMols = int(tmpLoadedMols)

Definition at line 416 of file acsSCCanalysis.py.

11.7.2.49 tuple acsSCCanalysis.loadedMolsConc = float(tmpLoadedMolsConc)

Definition at line 415 of file acsSCCanalysis.py.

11.7.2.50 tuple acsSCCanalysis.maxLOut = float(strLine[1])

Definition at line 266 of file acsSCCanalysis.py.

11.7.2.51 tuple acsSCCanalysis.meanOverThreshold = float(overThreshold)

Definition at line 703 of file acsSCCanalysis.py.

11.7.2.52 tuple acsSCCanalysis.mol_I = int(tmpMol_I)

Definition at line 412 of file acsSCCanalysis.py.

11.7.2.53 tuple acsSCCanalysis.mol_II = int(tmpMol_II)

Definition at line 413 of file acsSCCanalysis.py.

11.7.2.54 tuple acsSCCanalysis.mol_III = int(tmpMol_III)

Definition at line 414 of file acsSCCanalysis.py.

11.7.2.55 tuple acsSCCanalysis.mswindows = (sys.platform == "win32")

Definition at line 179 of file acsSCCanalysis.py.

11.7.2.56 tuple acsSCCanalysis.newdir = os.path.join(os.curdir, folderCat)

Definition at line 356 of file acsSCCanalysis.py.

11.7.2.57 tuple acsSCCanalysis.newdirSUB = os.path.join(os.curdir, folderSub)

Definition at line 365 of file acsSCCanalysis.py.

11.7.2.58 tuple acsSCCanalysis.newSpeciesCreationProb = float(tmpNSCprob)

Definition at line 420 of file acsSCCanalysis.py.

11.7.2.59 int acsSCCanalysis.noInAcs = 1

Definition at line 612 of file acsSCCanalysis.py.

11.7.2.60 int acsSCCanalysis.nrg = 1

Definition at line 273 of file acsSCCanalysis.py.

11.7.2.61 tuple acsSCCanalysis.nrgConc = float(strLine[1])

Definition at line 262 of file acsSCCanalysis.py.

11.7.2.62 tuple acsSCCanalysis.nrgTimeSeries = np.array([[0,0,0]])

Definition at line 397 of file acsSCCanalysis.py.

11.7.2.63 tuple acsSCCanalysis.nrgType = int(strLine[1])

Definition at line 258 of file acsSCCanalysis.py.

11.7.2.64 tuple acsSCCanalysis.numberOfGen = len(glob.glob(os.path.join(resDirPath,'times_*')))

Definition at line 230 of file acsSCCanalysis.py.

11.7.2.65 int acsSCCanalysis.ok = 0

Definition at line 316 of file acsSCCanalysis.py.

11.7.2.66 tuple acsSCCanalysis.overThreshold = float(0)

Definition at line 299 of file acsSCCanalysis.py.

11.7.2.67 tuple acsSCCanalysis.overThresholdTOT = float(0)

Definition at line 300 of file acsSCCanalysis.py.

11.7.2.68 string acsSCCanalysis.paramFile = "acsm2s.conf"

Definition at line 241 of file acsSCCanalysis.py.

11.7.2.69 tuple acsSCCanalysis.position = ((graph[:,0] == cat) & (graph[:,1] == mol_I))

Definition at line 482 of file acsSCCanalysis.py.

11.7.2.70 acsSCCanalysis.previousTime = 0

Definition at line 386 of file acsSCCanalysis.py.

11.7.2.71 int acsSCCanalysis.printTemporalMessage = 1

Definition at line 428 of file acsSCCanalysis.py.

11.7.2.72 int acsSCCanalysis.prod_bySCC = 0

Definition at line 570 of file acsSCCanalysis.py.

11.7.2.73 int acsSCCanalysis.prod_bySCC_weight = 0

Definition at line 576 of file acsSCCanalysis.py.

11.7.2.74 int acsSCCanalysis.prod_chain = 0

Definition at line 569 of file acsSCCanalysis.py.

11.7.2.75 int acsSCCanalysis.prod_chain_weight = 0

Definition at line 575 of file acsSCCanalysis.py.

11.7.2.76 int acsSCCanalysis.prod_inSCC = 0

Definition at line 568 of file acsSCCanalysis.py.

11.7.2.77 int acsSCCanalysis.prod_inSCC_weight = 0

Definition at line 574 of file acsSCCanalysis.py.

11.7.2.78 int acsSCCanalysis.prod_overlap = 0

Definition at line 571 of file acsSCCanalysis.py.

11.7.2.79 int acsSCCanalysis.prod_overlap_weight = 0

Definition at line 577 of file acsSCCanalysis.py.

11.7.2.80 int acsSCCanalysis.rctID = initRctId

Definition at line 385 of file acsSCCanalysis.py.

11.7.2.81 int acsSCCanalysis.rctIDshow = 1.0

Definition at line 383 of file acsSCCanalysis.py.

11.7.2.82 int acsSCCanalysis.rctIDshowNoSave = 1.0

Definition at line 384 of file acsSCCanalysis.py.

11.7.2.83 tuple acsSCCanalysis.rctParamFile = sorted(glob.glob(os.path.join(resDirPath,rctParamFileQ)))

Definition at line 380 of file acsSCCanalysis.py.

11.7.2.84 string acsSCCanalysis.rctParamFileQ = 'reactions_parameters_'

Definition at line 379 of file acsSCCanalysis.py.

11.7.2.85 tuple acsSCCanalysis.reaction = int(tmpReaction)

Definition at line 408 of file acsSCCanalysis.py.

11.7.2.86 int acsSCCanalysis.realSccs = 0

Definition at line 554 of file acsSCCanalysis.py.

11.7.2.87 acsSCCanalysis.realT = threshold

Definition at line 333 of file acsSCCanalysis.py.

11.7.2.88 tuple acsSCCanalysis.resDirPath = os.path.abspath(os.path.join("./", "res"))

Definition at line 223 of file acsSCCanalysis.py.

11.7.2.89 tuple acsSCCanalysis.reverseProbability = float(tmpRevProb)

Definition at line 421 of file acsSCCanalysis.py.

11.7.2.90 tuple acsSCCanalysis.rp = float(strLine[1])

Definition at line 252 of file acsSCCanalysis.py.

11.7.2.91 tuple acsSCCanalysis.rtime = float(tmpTime)

Definition at line 409 of file acsSCCanalysis.py.

11.7.2.92 tuple acsSCCanalysis.scc = nx.strongly_connected_components(Gcatpro)

Definition at line 561 of file acsSCCanalysis.py.

11.7.2.93 int acsSCCanalysis.sccID = 0

Definition at line 572 of file acsSCCanalysis.py.

11.7.2.94 tuple acsSCCanalysis.sccN = nx.number_strongly_connected_components(Gcatpro)

Definition at line 562 of file acsSCCanalysis.py.

11.7.2.95 int acsSCCanalysis.self_loop_weight = 0

Definition at line 578 of file acsSCCanalysis.py.

11.7.2.96 tuple acsSCCanalysis.selfLoops = Gcatpro.number_of_selfloops()

Definition at line 563 of file acsSCCanalysis.py.

11.7.2.97 tuple acsSCCanalysis.selfLoopsEgdes = Gcatpro.selfloop_edges()

Definition at line 564 of file acsSCCanalysis.py.

11.7.2.98 int acsSCCanalysis.simFolder = 0

Definition at line 242 of file acsSCCanalysis.py.

11.7.2.99 list acsSCCanalysis.speciesFile = speciesFiles[-1]

Definition at line 305 of file acsSCCanalysis.py.

11.7.2.100 tuple acsSCCanalysis.speciesFiles = sorted(glob.glob(os.path.join(resDirPath,tmpFilesToSearch)))

Definition at line 304 of file acsSCCanalysis.py.

11.7.2.101 list acsSCCanalysis.speciesInFlux = range(0,int(pow(2,(maxLOut+1)) - 2))

Definition at line 280 of file acsSCCanalysis.py.

11.7.2.102 tuple acsSCCanalysis.strLine = line.split('=')

Definition at line 250 of file acsSCCanalysis.py.

11.7.2.103 tuple acsSCCanalysis.StrPath = os.path.abspath(StrPath)

Definition at line 177 of file acsSCCanalysis.py.

11.7.2.104 string acsSCCanalysis.strToWrite = "\tReaction Probability"

Definition at line 186 of file acsSCCanalysis.py.

11.7.2.105 tuple acsSCCanalysis.strZeros = zeroBeforeStrNum(nGen, numberOfGen)

Definition at line 238 of file acsSCCanalysis.py.

11.7.2.106 int acsSCCanalysis.tempProd_chain_weight = 0

Definition at line 607 of file acsSCCanalysis.py.

11.7.2.107 acsSCCanalysis.timeInterval = rtime-previousTime

Definition at line 451 of file acsSCCanalysis.py.

11.7.2.108 tuple acsSCCanalysis.tmpDirs = sort(os.listdir(StrPath))

Definition at line 209 of file acsSCCanalysis.py.

11.7.2.109 string acsSCCanalysis.tmpFilesToSearch = 'species_'

Definition at line 303 of file acsSCCanalysis.py.

11.7.2.110 int acsSCCanalysis.tmpFlagLastRctSaved = 0

Definition at line 401 of file acsSCCanalysis.py.

11.7.2.111 int acsSCCanalysis.tmpProd_chain = 0

Definition at line 603 of file acsSCCanalysis.py.

11.7.2.112 tuple acsSCCanalysis.tmpRctFileToLoadSplitString = tmpRctFileToLoad.split("_")

Definition at line 196 of file acsSCCanalysis.py.

11.7.2.113 tuple acsSCCanalysis.today = dt.date.today()

Definition at line 178 of file acsSCCanalysis.py.

11.7.2.114 tuple acsSCCanalysis.totalRcts = int(strLine[1])

Definition at line 260 of file acsSCCanalysis.py.

11.7.2.115 tuple acsSCCanalysis.totDirName = os.path.join(StrPath,tmpDir)

Definition at line 218 of file acsSCCanalysis.py.

11.7.2.116 tuple acsSCCanalysis.totTimes = int(strLine[1])

Definition at line 256 of file acsSCCanalysis.py.

11.7.2.117 int acsSCCanalysis.wasteSpecies = 0

Definition at line 584 of file acsSCCanalysis.py.

11.7.2.118 list acsSCCanalysis.weightToDistribute = graph[((graph[:,0] == sngInNode) & (graph[:,1] == IdsOT)),5]

Definition at line 622 of file acsSCCanalysis.py.

11.8 acsSpeciesActivities Namespace Reference

Functions

- def zeroBeforeStrNum

Variables

- tuple StrPath = os.path.abspath(StrPath)
- tuple tmpDirs = sort(os.listdir(StrPath))
- int chemistry = 1
- string ndn = '_0_new_allStatResults'
- tuple newdirAllResults = os.path.join(StrPath, ndn)
- tuple newdirAllResultsInner = os.path.join(StrPath,'_0_new_allStatResults',ndn)
- tuple totDirName = os.path.join(StrPath,tmpDir)
- tuple resDirPath = os.path.abspath(os.path.join("./", "res"))
- tuple numberOfGen = len(glob.glob(os.path.join(resDirPath,'times_*')))
- string tmpSpeciesStatsSummaryName = 'speciesStatsSummary_'
- tuple tmpSpeciesStatsSummaryNameFID = open(tmpSpeciesStatsSummaryName, 'w')
- string tmpStr = '\n------- CHEMISTRY '
- tuple strZeros = zeroBeforeStrNum(ngen, numberOfGen)
- string strSpeciesZero = 'species_'
- tuple speciesFilesZero = sorted(glob.glob(os.path.join(resDirPath,strSpeciesZero)))
- string strSpecies = 'species_'
- tuple speciesFiles = sorted(glob.glob(os.path.join(resDirPath,strSpecies)))
- list lastfilespecies = speciesFiles[-1]
- tuple fidSpecies = open(lastfilespecies, 'r')
- list seq = []
- tuple counters = np.zeros((nSpecies,3))
- string strRctPar = 'reactions_parameters_'
- tuple fidRctPar = open(strRctPar, 'r')
- tuple rctType = int(tmpRctType)
- tuple cat = int(tmpCat)
- tuple S1 = int(tmpS1)
- tuple S2 = int(tmpS2)
- tuple S3 = int(tmpS3)
- string tmpFileName = 'speciesStats_'
- tuple tmpFileNameFID = open(tmpFileName, 'w')
- int ID = 0

11.8.1 Function Documentation

11.8.1.1 def acsSpeciesActivities.zeroBeforeStrNum (*tmpI, tmpL*)

Definition at line 20 of file acsSpeciesActivities.py.

11.8.2 Variable Documentation

11.8.2.1 tuple acsSpeciesActivities.cat = int(tmpCat)

Definition at line 130 of file acsSpeciesActivities.py.

11.8.2.2 int acsSpeciesActivities.chemistry = 1

Definition at line 43 of file acsSpeciesActivities.py.

11.8.2.3 tuple acsSpeciesActivities.counters = np.zeros((nSpecies,3))

Definition at line 116 of file acsSpeciesActivities.py.

11.8.2.4 tuple acsSpeciesActivities.fidRctPar = open(strRctPar, 'r')

Definition at line 122 of file acsSpeciesActivities.py.

11.8.2.5 tuple acsSpeciesActivities.fidSpecies = open(lastfilespecies, 'r')

Definition at line 105 of file acsSpeciesActivities.py.

11.8.2.6 int acsSpeciesActivities.ID = 0

Definition at line 162 of file acsSpeciesActivities.py.

11.8.2.7 list acsSpeciesActivities.lastfilespecies = speciesFiles[-1]

Definition at line 101 of file acsSpeciesActivities.py.

11.8.2.8 string acsSpeciesActivities.ndn = '_0_new_allStatResults'

Definition at line 46 of file acsSpeciesActivities.py.

11.8.2.9 tuple acsSpeciesActivities.newdirAllResults = os.path.join(StrPath, ndn)

Definition at line 47 of file acsSpeciesActivities.py.

11.8.2.10 tuple acsSpeciesActivities.newdirAllResultsInner = os.path.join(StrPath,'_0_new_allStatResults',ndn)

Definition at line 55 of file acsSpeciesActivities.py.

11.8.2.11 tuple acsSpeciesActivities.numberOfGen = len(glob.glob(os.path.join(resDirPath,'times_*')))

Definition at line 74 of file acsSpeciesActivities.py.

11.8.2.12 tuple acsSpeciesActivities.rctType = int(tmpRctType)

Definition at line 129 of file acsSpeciesActivities.py.

11.8.2.13 tuple acsSpeciesActivities.resDirPath = os.path.abspath(os.path.join("./", "res"))

Definition at line 68 of file acsSpeciesActivities.py.

11.8.2.14 tuple acsSpeciesActivities.S1 = int(tmpS1)

Definition at line 131 of file acsSpeciesActivities.py.

11.8.2.15 tuple acsSpeciesActivities.S2 = int(tmpS2)

Definition at line 132 of file acsSpeciesActivities.py.

11.8.2.16 tuple acsSpeciesActivities.S3 = int(tmpS3)

Definition at line 133 of file acsSpeciesActivities.py.

11.8.2.17 list acsSpeciesActivities.seq = [ ]

Definition at line 109 of file acsSpeciesActivities.py.

11.8.2.18 tuple acsSpeciesActivities.speciesFiles = sorted(glob.glob(os.path.join(resDirPath,strSpecies)))

Definition at line 98 of file acsSpeciesActivities.py.

11.8.2.19 tuple acsSpeciesActivities.speciesFilesZero = sorted(glob.glob(os.path.join(resDirPath,strSpeciesZero)))

Definition at line 93 of file acsSpeciesActivities.py.

11.8.2.20 tuple acsSpeciesActivities.StrPath = os.path.abspath(StrPath)

Definition at line 36 of file acsSpeciesActivities.py.

11.8.2.21 string acsSpeciesActivities.strRctPar = 'reactions_parameters_'

Definition at line 118 of file acsSpeciesActivities.py.

11.8.2.22 string acsSpeciesActivities.strSpecies = 'species_'

Definition at line 95 of file acsSpeciesActivities.py.

11.8.2.23 string acsSpeciesActivities.strSpeciesZero = 'species_'

Definition at line 92 of file acsSpeciesActivities.py.

11.8.2.24 tuple acsSpeciesActivities.strZeros = zeroBeforeStrNum(ngen, numberOfGen)

Definition at line 90 of file acsSpeciesActivities.py.

11.8.2.25 tuple acsSpeciesActivities.tmpDirs = sort(os.listdir(StrPath))

Definition at line 38 of file acsSpeciesActivities.py.

11.8.2.26 string acsSpeciesActivities.tmpFileName = 'speciesStats_'

Definition at line 159 of file acsSpeciesActivities.py.

11.8.2.27 tuple acsSpeciesActivities.tmpFileNameFID = open(tmpFileName, 'w')

Definition at line 161 of file acsSpeciesActivities.py.

11.8.2.28 string acsSpeciesActivities.tmpSpeciesStatsSummaryName = 'speciesStatsSummary_'

Definition at line 79 of file acsSpeciesActivities.py.

11.8.2.29 tuple acsSpeciesActivities.tmpSpeciesStatsSummaryNameFID = open(tmpSpeciesStatsSummaryName, 'w')

Definition at line 80 of file acsSpeciesActivities.py.

11.8.2.30 tuple acsSpeciesActivities.tmpStr = '\n------- CHEMISTRY '

Definition at line 82 of file acsSpeciesActivities.py.

11.8.2.31 tuple acsSpeciesActivities.totDirName = os.path.join(StrPath,tmpDir)

Definition at line 64 of file acsSpeciesActivities.py.

11.9 acsStatesAnalysis Namespace Reference

Functions

- def zeroBeforeStrNum
- def returnZeroSpeciesList
- def distanceMisures

Variables

- tuple today = dt.date.today()
- tuple StrPath = os.path.abspath(StrPath)
- tuple tmpDirs = sort(os.listdir(StrPath))
- string currentDir = ''
- string ndn = currentDir+'_0_new_allStatResults'
- tuple newdirAllResults = os.path.join(os.curdir, ndn)
- tuple previousFILE_FID = open('STAT_t_tminus_1.csv', 'w')
- tuple previousNOINFLUX_FILE_FID = open('STAT_t_tminus_1_NOINFLUX.csv', 'w')
- tuple startFILE_FID = open('STAT_t_start.csv', 'w')
- tuple startNOINFLUX_FILE_FID = open('STAT_t_start_NOINFLUX.csv', 'w')
- tuple HAM_previousFILE_FID = open('STAT_HAM_t_tminus_1.csv', 'w')
- tuple HAM_previousNOINFLUX_FILE_FID = open('STAT_HAM_t_tminus_1_NOINFLUX.csv', 'w')
- tuple HAM_startFILE_FID = open('STAT_HAM_t_start.csv', 'w')
- tuple HAM_startNOINFLUX_FILE_FID = open('STAT_HAM_t_start_NOINFLUX.csv', 'w')
- tuple EUC_previousFILE_FID = open('STAT_EUC_t_tminus_1.csv', 'w')
- tuple EUC_previousNOINFLUX_FILE_FID = open('STAT_EUC_t_tminus_1_NOINFLUX.csv', 'w')
- tuple EUC_startFILE_FID = open('STAT_EUC_t_start.csv', 'w')
- tuple EUC_startNOINFLUX_FILE_FID = open('STAT_EUC_t_start_NOINFLUX.csv', 'w')
- tuple ANG_middlePreviousFILE_FID = open('STAT_ANG_t_middle_NOINFLUX.csv', 'w')
- tuple HAM_middlePreviousFILE_FID = open('STAT_HAM_t_middle_NOINFLUX.csv', 'w')
- tuple EUC_middlePreviousFILE_FID = open('STAT_EUC_t_middle_NOINFLUX.csv', 'w')
- tuple previousFILE_FID_group = open('STAT_t_tminus_1_group.csv', 'w')
- tuple previousNOINFLUX_FILE_FID_group = open('STAT_t_tminus_1_NOINFLUX_group.csv', 'w')
- tuple startFILE_FID_group = open('STAT_t_start_group.csv', 'w')
- tuple startNOINFLUX_FILE_FID_group = open('STAT_t_start_NOINFLUX_group.csv', 'w')

- tuple HAM_previousFILE_FID_group = open('STAT_HAM_t_tminus_1_group.csv', 'w')
- tuple HAM_previousNOINFLUX_FILE_FID_group = open('STAT_HAM_t_tminus_1_NOINFLUX_group.csv', 'w')
- tuple HAM_startFILE_FID_group = open('STAT_HAM_t_start_group.csv', 'w')
- tuple HAM_startNOINFLUX_FILE_FID_group = open('STAT_HAM_t_start_NOINFLUX_group.csv', 'w')
- tuple EUC_previousFILE_FID_group = open('STAT_EUC_t_tminus_1_group.csv', 'w')
- tuple EUC_previousNOINFLUX_FILE_FID_group = open('STAT_EUC_t_tminus_1_NOINFLUX_group.csv', 'w')
- tuple EUC_startFILE_FID_group = open('STAT_EUC_t_start_group.csv', 'w')
- tuple EUC_startNOINFLUX_FILE_FID_group = open('STAT_EUC_t_start_NOINFLUX_group.csv', 'w')
- tuple newSpecies_FID = open('STAT_GENERAL_newSpecies.csv', 'w')
- tuple livingSpecies_FID = open('STAT_GENERAL_livingSpecies.csv', 'w')
- tuple mols_FID = open('STAT_GENERAL_mols.csv', 'w')
- tuple totMass_FID = open('STAT_GENERAL_overallMass.csv', 'w')
- tuple totOverallMass_FID = open('STAT_GENERAL_overallTotMass.csv', 'w')
- tuple complex_FID = open('STAT_GENERAL_complex.csv', 'w')
- tuple complexMols_FID = open('STAT_GENERAL_complexMols.csv', 'w')
- tuple evaluatedFID = open('STAT_GENERAL_evaluated.csv', 'w')
- tuple zeroOneSpeciesFID = open('STAT_GENERAL_zeroOneSpecies.csv', 'w')
- tuple biodeversityFID = open('STAT_GENERAL_bioDiversity.csv', 'w')
- int validDir = 1
- tuple totDirName = os.path.join(StrPath,tmpDir)
- tuple resDirPath = os.path.abspath(os.path.join("./", "res"))
- tuple numberOfGen = len(glob.glob(os.path.join(resDirPath,'times_*')))
- list group_A_prev = []
- list group_A_start = []
- list group_A_prev_NI = []
- list group_A_start_NI = []
- tuple strZeros = zeroBeforeStrNum(ngen, numberOfGen)
- string strSpeciesZero = 'species_'
- tuple speciesFilesZero = sorted(glob.glob(os.path.join(resDirPath,strSpeciesZero)))
- string strSpecies = 'species_'
- tuple speciesFiles = sorted(glob.glob(os.path.join(resDirPath,strSpecies)))
- tuple zeroList = returnZeroSpeciesList(speciesFiles[-1])
- tuple speciesConcs = np.zeros((len(speciesFiles)+1,len(zeroList)))
- list seqOLD = []
- list seqSTART = []
- list totMass = []
- list seqMIDDLE_NOINFLUX = []
- int oldNumberOfSpecies = 0
- tuple fidSpecies = open(sngSpeciesFile, 'r')
- list seq = []
- int tmpMols = 0
- int bioDivInd = 0
- deltaNspecies = numberOfSpecies-oldNumberOfSpecies
- tuple strtoW = str(deltaNspecies)
- tuple tmpMisure = distanceMisures(seq, conc, seqOLD, concOLD, idS)
- list seqOLDNOINFLUX = seqNOINFLUX[:]
- list concOLD = conc[:]
- string filename = "STAT_species_Concentrations_"

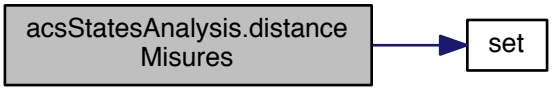
11.9.1 Function Documentation

11.9.1.1 def acsStatesAnalysis.distanceMisures (*tmpSeqX, tmpConcX, tmpSeqY, tmpConcY, tmpIDs*)

```
Function to compute the angle between two multidimensional vectors
```

Definition at line 45 of file acsStatesAnalysis.py.

Here is the call graph for this function:



11.9.1.2 def acsStatesAnalysis.returnZeroSpeciesList (*tmpLastSpeciesFile*)

```
Function to create a zero vector for each species (NO COMPLEXES)
```

Definition at line 31 of file acsStatesAnalysis.py.

11.9.1.3 def acsStatesAnalysis.zeroBeforeStrNum (*tmpI, tmpL*)

```
Function to create string zero string vector before graph filename.
According to the total number of reactions N zeros will be add before the instant reaction number
(e.g. reaction 130 of 10000 the string became '00130')
```

Definition at line 21 of file acsStatesAnalysis.py.

11.9.2 Variable Documentation

11.9.2.1 tuple acsStatesAnalysis.ANG_middlePreviousFILE_FID = open('STAT_ANG_t_middle_NOINFLUX.csv', 'w')

Definition at line 140 of file acsStatesAnalysis.py.

11.9.2.2 tuple acsStatesAnalysis.biodeversityFID = open('STAT_GENERAL_bioDiversity.csv', 'w')

Definition at line 168 of file acsStatesAnalysis.py.

11.9.2.3 tuple acsStatesAnalysis.bioDivInd = 0

Definition at line 273 of file acsStatesAnalysis.py.

11.9.2.4 tuple acsStatesAnalysis.complex_FID = open('STAT_GENERAL_complex.csv', 'w')

Definition at line 164 of file acsStatesAnalysis.py.

11.9.2.5 tuple acsStatesAnalysis.complexMols_FID = open('STAT_GENERAL_complexMols.csv', 'w')

Definition at line 165 of file acsStatesAnalysis.py.

11.9.2.6 list acsStatesAnalysis.concOLD = conc[:]

Definition at line 351 of file acsStatesAnalysis.py.

11.9.2.7 string acsStatesAnalysis.currentDir = ''

Definition at line 113 of file acsStatesAnalysis.py.

11.9.2.8 acsStatesAnalysis.deltaNspecies = numberOfSpecies-oldNumberOfSpecies

Definition at line 279 of file acsStatesAnalysis.py.

11.9.2.9 tuple acsStatesAnalysis.EUC_middlePreviousFILE_FID = open('STAT_EUC_t_middle_NOINFLUX.csv', 'w')

Definition at line 142 of file acsStatesAnalysis.py.

11.9.2.10 tuple acsStatesAnalysis.EUC_previousFILE_FID = open('STAT_EUC_t_tminus_1.csv', 'w')

Definition at line 135 of file acsStatesAnalysis.py.

11.9.2.11 tuple acsStatesAnalysis.EUC_previousFILE_FID_group = open('STAT_EUC_t_tminus_1_group.csv', 'w')

Definition at line 154 of file acsStatesAnalysis.py.

11.9.2.12 tuple acsStatesAnalysis.EUC_previousNOINFLUX_FILE_FID = open('STAT_EUC_t_tminus_1_NOINFLUX.csv', 'w')

Definition at line 136 of file acsStatesAnalysis.py.

11.9.2.13 tuple acsStatesAnalysis.EUC_previousNOINFLUX_FILE_FID_group = open('STAT_EUC_t_tminus_1_NOINFLUX_-group.csv', 'w')

Definition at line 155 of file acsStatesAnalysis.py.

11.9.2.14 tuple acsStatesAnalysis.EUC_startFILE_FID = open('STAT_EUC_t_start.csv', 'w')

Definition at line 137 of file acsStatesAnalysis.py.

11.9.2.15 tuple acsStatesAnalysis.EUC_startFILE_FID_group = open('STAT_EUC_t_start_group.csv', 'w')

Definition at line 156 of file acsStatesAnalysis.py.

11.9.2.16 tuple acsStatesAnalysis.EUC_startNOINFLUX_FILE_FID = open('STAT_EUC_t_start_NOINFLUX.csv', 'w')

Definition at line 138 of file acsStatesAnalysis.py.

11.9.2.17 tuple acsStatesAnalysis.EUC_startNOINFLUX_FILE_FID_group = open('STAT_EUC_t_start_NOINFLUX_group.csv', 'w')

Definition at line 157 of file acsStatesAnalysis.py.

11.9.2.18 tuple acsStatesAnalysis.evaluatedFID = open('STAT_GENERAL_evaluated.csv', 'w')

Definition at line 166 of file acsStatesAnalysis.py.

11.9.2.19 tuple acsStatesAnalysis.fidSpecies = open(sngSpeciesFile, 'r')

Definition at line 230 of file acsStatesAnalysis.py.

11.9.2.20 string acsStatesAnalysis.filename = "STAT_species_Concentrations_"

Definition at line 410 of file acsStatesAnalysis.py.

11.9.2.21 list acsStatesAnalysis.group_A_prev = [ ]

Definition at line 186 of file acsStatesAnalysis.py.

11.9.2.22 list acsStatesAnalysis.group_A_prev_NI = [ ]

Definition at line 188 of file acsStatesAnalysis.py.

11.9.2.23 list acsStatesAnalysis.group_A_start = [ ]

Definition at line 187 of file acsStatesAnalysis.py.

11.9.2.24 list acsStatesAnalysis.group_A_start_NI = [ ]

Definition at line 189 of file acsStatesAnalysis.py.

11.9.2.25 tuple acsStatesAnalysis.HAM_middlePreviousFILE_FID = open('STAT_HAM_t_middle_NOINFLUX.csv', 'w')

Definition at line 141 of file acsStatesAnalysis.py.

11.9.2.26 tuple acsStatesAnalysis.HAM_previousFILE_FID = open('STAT_HAM_t_tminus_1.csv', 'w')

Definition at line 130 of file acsStatesAnalysis.py.

11.9.2.27 tuple acsStatesAnalysis.HAM_previousFILE_FID_group = open('STAT_HAM_t_tminus_1_group.csv', 'w')

Definition at line 149 of file acsStatesAnalysis.py.

11.9.2.28 tuple acsStatesAnalysis.HAM_previousNOINFLUX_FILE_FID = open('STAT_HAM_t_tminus_1_NOINFLUX.csv', 'w')

Definition at line 131 of file acsStatesAnalysis.py.

11.9.2.29 tuple acsStatesAnalysis.HAM_previousNOINFLUX_FILE_FID_group = open('STAT_HAM_t_tminus_1_NOINFLUX_group.csv', 'w')

Definition at line 150 of file acsStatesAnalysis.py.

11.9.2.30 tuple acsStatesAnalysis.HAM_startFILE_FID = open('STAT_HAM_t_start.csv', 'w')

Definition at line 132 of file acsStatesAnalysis.py.

11.9.2.31 tuple acsStatesAnalysis.HAM_startFILE_FID_group = open('STAT_HAM_t_start_group.csv', 'w')

Definition at line 151 of file acsStatesAnalysis.py.

11.9.2.32 tuple acsStatesAnalysis.HAM_startNOINFLUX_FILE_FID = open('STAT_HAM_t_start_NOINFLUX.csv', 'w')

Definition at line 133 of file acsStatesAnalysis.py.

11.9.2.33 tuple acsStatesAnalysis.HAM_startNOINFLUX_FILE_FID_group = open('STAT_HAM_t_start_NOINFLUX_group.csv', 'w')

Definition at line 152 of file acsStatesAnalysis.py.

11.9.2.34 tuple acsStatesAnalysis.livingSpecies_FID = open('STAT_GENERAL_livingSpecies.csv', 'w')

Definition at line 160 of file acsStatesAnalysis.py.

11.9.2.35 tuple acsStatesAnalysis.mols_FID = open('STAT_GENERAL_mols.csv', 'w')

Definition at line 161 of file acsStatesAnalysis.py.

11.9.2.36 string acsStatesAnalysis.ndn = currentDir+'_0_new_allStatResults'

Definition at line 114 of file acsStatesAnalysis.py.

11.9.2.37 tuple acsStatesAnalysis.newdirAllResults = os.path.join(os.curdir, ndn)

Definition at line 115 of file acsStatesAnalysis.py.

11.9.2.38 tuple acsStatesAnalysis.newSpecies_FID = open('STAT_GENERAL_newSpecies.csv', 'w')

Definition at line 159 of file acsStatesAnalysis.py.

11.9.2.39 tuple acsStatesAnalysis.numberOfGen = len(glob.glob(os.path.join(resDirPath,'times_*')))

Definition at line 184 of file acsStatesAnalysis.py.

11.9.2.40 acsStatesAnalysis.oldNumberOfSpecies = 0

Definition at line 222 of file acsStatesAnalysis.py.

11.9.2.41 tuple acsStatesAnalysis.previousFILE_FID = open('STAT_t_tminus_1.csv', 'w')

Definition at line 125 of file acsStatesAnalysis.py.

11.9.2.42 tuple acsStatesAnalysis.previousFILE_FID_group = open('STAT_t_tminus_1_group.csv', 'w')

Definition at line 144 of file acsStatesAnalysis.py.

11.9.2.43 tuple acsStatesAnalysis.previousNOINFLUX_FILE_FID = open('STAT_t_tminus_1_NOINFLUX.csv', 'w')

Definition at line 126 of file acsStatesAnalysis.py.

11.9.2.44 tuple acsStatesAnalysis.previousNOINFLUX_FILE_FID_group = open('STAT_t_tminus_1_NOINFLUX_group.csv', 'w')

Definition at line 145 of file acsStatesAnalysis.py.

11.9.2.45 tuple acsStatesAnalysis.resDirPath = os.path.abspath(os.path.join("./", "res"))

Definition at line 178 of file acsStatesAnalysis.py.

11.9.2.46 list acsStatesAnalysis.seq = [ ]

Definition at line 234 of file acsStatesAnalysis.py.

11.9.2.47 list acsStatesAnalysis.seqMIDDLE_NOINFLUX = [ ]

Definition at line 220 of file acsStatesAnalysis.py.

11.9.2.48 list acsStatesAnalysis.seqOLD = [ ]

Definition at line 217 of file acsStatesAnalysis.py.

11.9.2.49 list acsStatesAnalysis.seqOLDNOINFLUX = seqNOINFLUX[:]

Definition at line 350 of file acsStatesAnalysis.py.

11.9.2.50 list acsStatesAnalysis.seqSTART = [ ]

Definition at line 218 of file acsStatesAnalysis.py.

11.9.2.51 list acsStatesAnalysis.speciesConcs = np.zeros((len(speciesFiles)+1,len(zeroList)))

Definition at line 214 of file acsStatesAnalysis.py.

11.9.2.52 acsStatesAnalysis.speciesFiles = sorted(glob.glob(os.path.join(resDirPath,strSpecies)))

Definition at line 204 of file acsStatesAnalysis.py.

11.9.2.53 tuple acsStatesAnalysis.speciesFilesZero = sorted(glob.glob(os.path.join(resDirPath,strSpeciesZero)))

Definition at line 199 of file acsStatesAnalysis.py.

11.9.2.54 tuple acsStatesAnalysis.startFILE_FID = open('STAT_t_start.csv', 'w')

Definition at line 127 of file acsStatesAnalysis.py.

11.9.2.55 tuple acsStatesAnalysis.startFILE_FID_group = open('STAT_t_start_group.csv', 'w')

Definition at line 146 of file acsStatesAnalysis.py.

11.9.2.56 tuple acsStatesAnalysis.startNOINFLUX_FILE_FID = open('STAT_t_start_NOINFLUX.csv', 'w')

Definition at line 128 of file acsStatesAnalysis.py.

11.9.2.57 tuple acsStatesAnalysis.startNOINFLUX_FILE_FID_group = open('STAT_t_start_NOINFLUX_group.csv', 'w')

Definition at line 147 of file acsStatesAnalysis.py.

11.9.2.58 tuple acsStatesAnalysis.StrPath = os.path.abspath(StrPath)

Definition at line 101 of file acsStatesAnalysis.py.

11.9.2.59 string acsStatesAnalysis.strSpecies = 'species_'

Definition at line 201 of file acsStatesAnalysis.py.

11.9.2.60 string acsStatesAnalysis.strSpeciesZero = 'species_'

Definition at line 198 of file acsStatesAnalysis.py.

11.9.2.61 tuple acsStatesAnalysis.strtoW = str(deltaNspecies)

Definition at line 281 of file acsStatesAnalysis.py.

11.9.2.62 tuple acsStatesAnalysis.strZeros = zeroBeforeStrNum(ngen, numberOfGen)

Definition at line 195 of file acsStatesAnalysis.py.

11.9.2.63 tuple acsStatesAnalysis.tmpDirs = sort(os.listdir(StrPath))

Definition at line 103 of file acsStatesAnalysis.py.

11.9.2.64 tuple acsStatesAnalysis.tmpMisure = distanceMisures(seq, conc, seqOLD, concOLD, idS)

Definition at line 302 of file acsStatesAnalysis.py.

11.9.2.65 int acsStatesAnalysis.tmpMols = 0

Definition at line 235 of file acsStatesAnalysis.py.

11.9.2.66 tuple acsStatesAnalysis.today = dt.date.today()

Definition at line 99 of file acsStatesAnalysis.py.

11.9.2.67 tuple acsStatesAnalysis.totDirName = os.path.join(StrPath,tmpDir)

Definition at line 174 of file acsStatesAnalysis.py.

11.9.2.68 list acsStatesAnalysis.totMass = [ ]

Definition at line 219 of file acsStatesAnalysis.py.

11.9.2.69 tuple acsStatesAnalysis.totMass_FID = open('STAT_GENERAL_overallMass.csv', 'w')

Definition at line 162 of file acsStatesAnalysis.py.

11.9.2.70 tuple acsStatesAnalysis.totOverallMass_FID = open('STAT_GENERAL_overallTotMass.csv', 'w')

Definition at line 163 of file acsStatesAnalysis.py.

11.9.2.71 int acsStatesAnalysis.validDir = 1

Definition at line 171 of file acsStatesAnalysis.py.

11.9.2.72 tuple acsStatesAnalysis.zeroList = returnZeroSpeciesList(speciesFiles[-1])

Definition at line 207 of file acsStatesAnalysis.py.

11.9.2.73 tuple acsStatesAnalysis.zeroOneSpeciesFID = open('STAT_GENERAL_zeroOneSpecies.csv', 'w')

Definition at line 167 of file acsStatesAnalysis.py.

11.10 bufferedFluxAnalysis Namespace Reference

Variables

- list StrPath = sys.argv[1]
- list tmpResFold = sys.argv[2]
- tuple today = dt.date.today()
- string simF = StrPath+'/'
- tuple tmpDirs = sort(glob.glob(simF))
- tuple newdir = os.path.join(os.curdir, '0_statistics')
- tuple matrixTimeLITE = np.zeros((101,len(tmpDirs)))
- tuple matrixFluxLITE = np.zeros((101,len(tmpDirs)))
- tuple matrixAbsLITE = np.zeros((101,len(tmpDirs)))

- tuple matrixExpLITE = np.zeros((101,len(tmpDirs)))
- int tmpDirsCnt = 0
- tuple speciesFiles = sort(glob.glob('species_*'))
- tuple fidSpecies = open(speciesFiles[0], 'r')
- int ok = 0
- list fluxIndexes = []
- list fluxLengths = []
- tuple index = int(tmpID)
- tuple speciesSeq = len(tmpSeq)
- tuple concFixed = int(tmpConcFixed)
- tuple fileslist = sort(glob.glob('reactions_parameters_*'))
- int rctParFileNum = 1
- int rctID = 1
- list totFluxDyn = []
- list absorbedBricks = []
- int tempAbsorbedBricks = 0
- list expelledBricks = []
- int tempExpelledBricks = 0
- list totTimes = []
- list totFluxDyn_LITE = []
- list absorbedBricks_LITE = []
- list expelledBricks_LITE = []
- list totTimes_LITE = []
- tuple fid = open(file, 'r')
- int okmonitor = 1
- int oksaveLite = 0
- int deltaRct = 0
- tuple reaction = int(tmpReaction)
- tuple rtime = float(tmpTime)
- tuple cc = int(tmpcc)
- tuple cat = int(tmpCat)
- tuple mol_I = int(tmpMol_I)
- tuple mol_II = int(tmpMol_II)
- tuple mol_III = int(tmpMol_III)
- tuple loadedMolsConc = float(tmpLoadedMolsConc)
- tuple loadedMols = int(tmpLoadedMols)
- tuple gillMean = float(tmpGillMean)
- tuple gillSD = float(tmpGillSD)
- tuple gillEntropy = float(tmpGillEntropy)
- tuple savingMatrix = np.zeros((len(totTimes),4))
- tuple tmpDirSplit = tmpDir.split("/")
- string cmdFileName = StrPath+'0_statistics/0_fluxDynamics_'
- tuple cmdFileFid = open(cmdFileName, 'a')

11.10.1 Variable Documentation

11.10.1.1 list bufferedFluxAnalysis.absorbedBricks = []

Definition at line 99 of file bufferedFluxAnalysis.py.

11.10.1.2 list bufferedFluxAnalysis.absorbedBricks_LITE = []

Definition at line 107 of file bufferedFluxAnalysis.py.

11.10.1.3 tuple bufferedFluxAnalysis.cat = int(tmpCat)

Definition at line 131 of file bufferedFluxAnalysis.py.

11.10.1.4 tuple bufferedFluxAnalysis.cc = int(tmpcc)

Definition at line 130 of file bufferedFluxAnalysis.py.

11.10.1.5 tuple bufferedFluxAnalysis.cmdFileFid = open(cmdFileName, 'a')

Definition at line 206 of file bufferedFluxAnalysis.py.

11.10.1.6 string bufferedFluxAnalysis.cmdFileName = StrPath+'0_statistics/0_fluxDynamics_'

Definition at line 203 of file bufferedFluxAnalysis.py.

11.10.1.7 tuple bufferedFluxAnalysis.concFixed = int(tmpConcFixed)

Definition at line 85 of file bufferedFluxAnalysis.py.

11.10.1.8 list bufferedFluxAnalysis.deltaRct = 0

Definition at line 123 of file bufferedFluxAnalysis.py.

11.10.1.9 list bufferedFluxAnalysis.expelledBricks = [ ]

Definition at line 101 of file bufferedFluxAnalysis.py.

11.10.1.10 list bufferedFluxAnalysis.expelledBricks_LITE = [ ]

Definition at line 108 of file bufferedFluxAnalysis.py.

11.10.1.11 tuple bufferedFluxAnalysis.fid = open(file, 'r')

Definition at line 116 of file bufferedFluxAnalysis.py.

11.10.1.12 tuple bufferedFluxAnalysis.fidSpecies = open(speciesFiles[0], 'r')

Definition at line 70 of file bufferedFluxAnalysis.py.

11.10.1.13 tuple bufferedFluxAnalysis.fileslist = sort(glob.glob('reactions_parameters_*'))

Definition at line 94 of file bufferedFluxAnalysis.py.

11.10.1.14 list bufferedFluxAnalysis.fluxIndexes = [ ]

Definition at line 76 of file bufferedFluxAnalysis.py.

11.10.1.15 list bufferedFluxAnalysis.fluxLengths = [ ]

Definition at line 77 of file bufferedFluxAnalysis.py.

11.10.1.16 tuple bufferedFluxAnalysis.gillEntropy = float(tmpGillEntropy)

Definition at line 139 of file bufferedFluxAnalysis.py.

11.10.1.17 tuple bufferedFluxAnalysis.gillMean = float(tmpGillMean)

Definition at line 137 of file bufferedFluxAnalysis.py.

11.10.1.18 tuple bufferedFluxAnalysis.gillSD = float(tmpGillSD)

Definition at line 138 of file bufferedFluxAnalysis.py.

11.10.1.19 tuple bufferedFluxAnalysis.index = int(tmpID)

Definition at line 83 of file bufferedFluxAnalysis.py.

11.10.1.20 tuple bufferedFluxAnalysis.loadedMols = int(tmpLoadedMols)

Definition at line 136 of file bufferedFluxAnalysis.py.

11.10.1.21 tuple bufferedFluxAnalysis.loadedMolsConc = float(tmpLoadedMolsConc)

Definition at line 135 of file bufferedFluxAnalysis.py.

11.10.1.22 tuple bufferedFluxAnalysis.matrixAbsLITE = np.zeros((101,len(tmpDirs)))

Definition at line 48 of file bufferedFluxAnalysis.py.

11.10.1.23 tuple bufferedFluxAnalysis.matrixExpLITE = np.zeros((101,len(tmpDirs)))

Definition at line 49 of file bufferedFluxAnalysis.py.

11.10.1.24 tuple bufferedFluxAnalysis.matrixFluxLITE = np.zeros((101,len(tmpDirs)))

Definition at line 47 of file bufferedFluxAnalysis.py.

11.10.1.25 tuple bufferedFluxAnalysis.matrixTimeLITE = np.zeros((101,len(tmpDirs)))

Definition at line 46 of file bufferedFluxAnalysis.py.

11.10.1.26 tuple bufferedFluxAnalysis.mol_I = int(tmpMol_I)

Definition at line 132 of file bufferedFluxAnalysis.py.

11.10.1.27 tuple bufferedFluxAnalysis.mol_II = int(tmpMol_II)

Definition at line 133 of file bufferedFluxAnalysis.py.

11.10.1.28 tuple bufferedFluxAnalysis.mol_III = int(tmpMol_III)

Definition at line 134 of file bufferedFluxAnalysis.py.

11.10.1.29 tuple bufferedFluxAnalysis.newdir = os.path.join(os.curdir, '0_statistics')

Definition at line 39 of file bufferedFluxAnalysis.py.

11.10.1.30 int bufferedFluxAnalysis.ok = 0

Definition at line 75 of file bufferedFluxAnalysis.py.

11.10.1.31 int bufferedFluxAnalysis.okmonitor = 1

Definition at line 118 of file bufferedFluxAnalysis.py.

11.10.1.32 int bufferedFluxAnalysis.oksaveLite = 0

Definition at line 119 of file bufferedFluxAnalysis.py.

11.10.1.33 int bufferedFluxAnalysis.rctID = 1

Definition at line 96 of file bufferedFluxAnalysis.py.

11.10.1.34 int bufferedFluxAnalysis.rctParFileNum = 1

Definition at line 95 of file bufferedFluxAnalysis.py.

11.10.1.35 tuple bufferedFluxAnalysis.reaction = int(tmpReaction)

Definition at line 128 of file bufferedFluxAnalysis.py.

11.10.1.36 tuple bufferedFluxAnalysis.rtime = float(tmpTime)

Definition at line 129 of file bufferedFluxAnalysis.py.

11.10.1.37 tuple bufferedFluxAnalysis.savingMatrix = np.zeros((len(totTimes),4))

Definition at line 195 of file bufferedFluxAnalysis.py.

11.10.1.38 string bufferedFluxAnalysis.simF = StrPath+'/'

Definition at line 31 of file bufferedFluxAnalysis.py.

11.10.1.39 tuple bufferedFluxAnalysis.speciesFiles = sort(glob.glob('species_*'))

Definition at line 64 of file bufferedFluxAnalysis.py.

11.10.1.40 tuple bufferedFluxAnalysis.speciesSeq = len(tmpSeq)

Definition at line 84 of file bufferedFluxAnalysis.py.

11.10.1.41 list bufferedFluxAnalysis.StrPath = sys.argv[1]

Definition at line 18 of file bufferedFluxAnalysis.py.

11.10.1.42 list bufferedFluxAnalysis.tempAbsorbedBricks = 0

Definition at line 100 of file bufferedFluxAnalysis.py.

11.10.1.43 list bufferedFluxAnalysis.tempExpelledBricks = 0

Definition at line 102 of file bufferedFluxAnalysis.py.

11.10.1.44 tuple bufferedFluxAnalysis.tmpDirs = sort(glob.glob(simF))

Definition at line 32 of file bufferedFluxAnalysis.py.

11.10.1.45 int bufferedFluxAnalysis.tmpDirsCnt = 0

Definition at line 51 of file bufferedFluxAnalysis.py.

11.10.1.46 tuple bufferedFluxAnalysis.tmpDirSplit = tmpDir.split("/")

Definition at line 202 of file bufferedFluxAnalysis.py.

11.10.1.47 list bufferedFluxAnalysis.tmpResFold = sys.argv[2]

Definition at line 19 of file bufferedFluxAnalysis.py.

11.10.1.48 tuple bufferedFluxAnalysis.today = dt.date.today()

Definition at line 25 of file bufferedFluxAnalysis.py.

11.10.1.49 list bufferedFluxAnalysis.totFluxDyn = [ ]

Definition at line 98 of file bufferedFluxAnalysis.py.

11.10.1.50 list bufferedFluxAnalysis.totFluxDyn_LITE = [ ]

Definition at line 106 of file bufferedFluxAnalysis.py.

11.10.1.51 list bufferedFluxAnalysis.totTimes = [ ]

Definition at line 103 of file bufferedFluxAnalysis.py.

11.10.1.52 list bufferedFluxAnalysis.totTimes_LITE = [ ]

Definition at line 109 of file bufferedFluxAnalysis.py.

11.11 fromWithin2Between Namespace Reference

Functions

- def zeroBeforeStrNum

Variables

- tuple zeroSIM = zeroBeforeStrNum(i,numSim)
- tuple zeroGEN = zeroBeforeStrNum(j,numGen)
- string folderNew = "s_"
- tuple resdir = os.path.join(os.curdir, "res")

11.11.1 Function Documentation

11.11.1.1 def fromWithin2Between.zeroBeforeStrNum (*tmpI, tmpL*)

Definition at line 10 of file fromWithin2Between.py.

11.11.2 Variable Documentation

11.11.2.1 string fromWithin2Between.folderNew = "s_"

Definition at line 41 of file fromWithin2Between.py.

11.11.2.2 tuple fromWithin2Between.resdir = os.path.join(os.curdir, "res")

Definition at line 52 of file fromWithin2Between.py.

11.11.2.3 tuple fromWithin2Between.zeroGEN = zeroBeforeStrNum(j,numGen)

Definition at line 39 of file fromWithin2Between.py.

11.11.2.4 tuple fromWithin2Between.zeroSIM = zeroBeforeStrNum(i,numSim)

Definition at line 35 of file fromWithin2Between.py.

11.12 init Namespace Reference

Variables

- tuple parser
- tuple args = parser.parse_args()
- string ndn = '_0_new_allStatResults'
- tuple newdirAllResults = os.path.join(args.strOut, ndn)
- tuple fname_initRafRes = os.path.join(newdirAllResults, '0_initRafAnalysis.csv')
- tuple fname_initRafResSUM = os.path.join(newdirAllResults, '0_initRafAnalysisSUM.csv')
- tuple fname_initRafResLIST = os.path.join(newdirAllResults, '0_initRafAnalysisLIST.csv')
- tuple fname_initRafResALL = os.path.join(newdirAllResults, '0_initRafAnalysisALL.csv')
- tuple fid_initRafRes = open(fname_initRafRes, 'w')
- tuple fid_initRafResSUM = open(fname_initRafResSUM, 'w')
- tuple fid_initRafResLIST = open(fname_initRafResLIST, 'w')
- tuple fid_initRafResALL = open(fname_initRafResALL, 'w')
- string strToWrite = "Folder\tP\tAC\tM\tRAFsize\tClosure\tCats\tuRAF\n"
- tuple foodList = range(args.lastFood+1)
- tuple avgCon = dn.rangeFloat(float(args.avgCon[0]), float(args.avgCon[1]), float(args.avgCon[2]))
- int raffound = 0
- list alphabet = ['A', 'B']
- list species = []
- int totSpecies = 2
- tuple totCleavage = sum(map(lambda x: len(x)-1,species))
- int totCond = totSpecies**2
- totRcts = totCleavage+totCond
- tuple rctToCat = int(round(totRcts * totSpecies * prob))
- int nCleavage = 0
- int nCondensa = 0
- tuple initSpeciesListLength = len(species)
- tuple conf = (1,1,2000,0,200000,0,0,2,args.lastFood,prob)
- int rctType = 1
- tuple molToCleav = ran.choice(species[len(alphabet):initSpeciesListLength-1])
- tuple cutPt = ran.randint(1,len(molToCleav)-1)
- list tmp1 = molToCleav[0:cutPt]
- tuple tmp1id = species.index(tmp1)
- find1 = True
- list tmp2 = molToCleav[cutPt:len(molToCleav)]
- tuple tmp2id = species.index(tmp2)
- tuple sub1 = ran.choice(species[:initSpeciesListLength-1])
- tuple idsub1 = species.index(sub1)
- tuple sub2 = ran.choice(species[:initSpeciesListLength-1])
- tuple idsub2 = species.index(sub2)
- prod = sub1+sub2
- tuple tmpprodid = species.index(prod)
- int catalyst = -1
- catFound = False
- tuple rafsets = raf.rafComputation(fid_initRafRes, fid_initRafResALL, fid_initRafResLIST, 'tmpDir', prob, averageConn, rcts, cats, foodList, maxlength)

11.12.1 Variable Documentation

11.12.1.1 list init.alphabet = ['A', 'B']

Definition at line 79 of file init.py.

11.12.1.2 tuple init.args = parser.parse_args()

Definition at line 39 of file init.py.

11.12.1.3 tuple init.avgCon = dn.rangeFloat(float(args.avgCon[0]), float(args.avgCon[1]), float(args.avgCon[2]))

Definition at line 72 of file init.py.

11.12.1.4 tuple init.catalyst = -1

Definition at line 158 of file init.py.

11.12.1.5 init.catFound = False

Definition at line 159 of file init.py.

11.12.1.6 tuple init.conf = (1,1,2000,0,200000,0,0,2,args.lastFood,prob)

Definition at line 104 of file init.py.

11.12.1.7 tuple init.cutPt = ran.randint(1,len(molToCleav)-1)

Definition at line 121 of file init.py.

11.12.1.8 tuple init.fid_initRafRes = open(fname_initRafRes, 'w')

Definition at line 59 of file init.py.

11.12.1.9 tuple init.fid_initRafResALL = open(fname_initRafResALL, 'w')

Definition at line 62 of file init.py.

11.12.1.10 tuple init.fid_initRafResLIST = open(fname_initRafResLIST, 'w')

Definition at line 61 of file init.py.

11.12.1.11 tuple init.fid_initRafResSUM = open(fname_initRafResSUM, 'w')

Definition at line 60 of file init.py.

11.12.1.12 init.find1 = True

Definition at line 125 of file init.py.

11.12.1.13 tuple init.fname_initRafRes = os.path.join(newdirAllResults, '0_initRafAnalysis.csv')

Definition at line 55 of file init.py.

11.12.1.14 tuple init.fname_initRafResALL = os.path.join(newdirAllResults, '0_initRafAnalysisALL.csv')

Definition at line 58 of file init.py.

11.12.1.15 tuple init.fname_initRafResLIST = os.path.join(newdirAllResults, '0_initRafAnalysisLIST.csv')

Definition at line 57 of file init.py.

11.12.1.16 tuple init.fname_initRafResSUM = os.path.join(newdirAllResults, '0_initRafAnalysisSUM.csv')

Definition at line 56 of file init.py.

11.12.1.17 tuple init.foodList = range(args.lastFood+1)

Definition at line 69 of file init.py.

11.12.1.18 tuple init.idsub1 = species.index(sub1)

Definition at line 143 of file init.py.

11.12.1.19 tuple init.idsub2 = species.index(sub2)

Definition at line 146 of file init.py.

11.12.1.20 tuple init.initSpeciesListLength = len(species)

Definition at line 101 of file init.py.

11.12.1.21 tuple init.molToCleav = ran.choice(species[len(alphabet):initSpeciesListLength-1])

Definition at line 120 of file init.py.

11.12.1.22 int init.nCleavage = 0

Definition at line 99 of file init.py.

11.12.1.23 int init.nCondensa = 0

Definition at line 100 of file init.py.

11.12.1.24 string init.ndn = '_0_new_allStatResults'

Definition at line 45 of file init.py.

11.12.1.25 tuple init.newdirAllResults = os.path.join(args.strOut, ndn)

Definition at line 46 of file init.py.

11.12.1.26 tuple init.parser

Initial value:

```
1 = ArgumentParser(
2                                 description='This script re-arrange results in a more friendly way from the
      angle analysis in time.'
3                                 , epilog='''File with angle trajectories are created. ''')
```

Definition at line 23 of file init.py.

11.12.1.27 init.prod = sub1+sub2

Definition at line 147 of file init.py.

11.12.1.28 int init.raffound = 0

Definition at line 76 of file init.py.

11.12.1.29 tuple init.rafsets = raf.rafComputation(fid_initRafRes, fid_initRafResALL, fid_initRafResLIST, 'tmpDir', prob, averageConn, rcts, cats, foodList, maxlength)

Definition at line 167 of file init.py.

11.12.1.30 tuple init.rctToCat = int(round(totRcts * totSpecies * prob))

Definition at line 96 of file init.py.

11.12.1.31 int init.rctType = 1

Definition at line 110 of file init.py.

11.12.1.32 list init.species = [ ]

Definition at line 80 of file init.py.

11.12.1.33 tuple init.strToWrite = "Folder\tP\tAC\tM\tRAFsize\tClosure\tCats\tuRAF\n"

Definition at line 63 of file init.py.

11.12.1.34 tuple init.sub1 = ran.choice(species[:initSpeciesListLength-1])

Definition at line 142 of file init.py.

11.12.1.35 tuple init.sub2 = ran.choice(species[:initSpeciesListLength-1])

Definition at line 145 of file init.py.

11.12.1.36 list init.tmp1 = molToCleav[0:cutPt]

Definition at line 122 of file init.py.

11.12.1.37 tuple init.tmp1id = species.index(tmp1)

Definition at line 124 of file init.py.

11.12.1.38 list init.tmp2 = molToCleav[cutPt:len(molToCleav)]

Definition at line 130 of file init.py.

11.12.1.39 tuple init.tmp2id = species.index(tmp2)

Definition at line 132 of file init.py.

11.12.1.40 tuple init.tmpprodid = species.index(prod)

Definition at line 149 of file init.py.

11.12.1.41 tuple init.totCleavage = sum(map(lambda x: len(x)-1,species))

Definition at line 86 of file init.py.

11.12.1.42 int init.totCond = totSpecies∗∗2

Definition at line 87 of file init.py.

11.12.1.43 init.totRcts = totCleavage+totCond

Definition at line 88 of file init.py.

11.12.1.44 int init.totSpecies = 2

Definition at line 84 of file init.py.

11.13 lib Namespace Reference

Namespaces

- dyn
- graph
- IO

11.14 lib.dyn Namespace Reference

Namespaces

- dynamics

11.15 lib.dyn.dynamics Namespace Reference

Functions

- def generateFluxList
- def rangeFloat
- def fluxAnalysis

11.15.1 Function Documentation

11.15.1.1 def lib.dyn.dynamics.fluxAnalysis (*tmpDir, resDirPath, strZeros, ngen*)

Definition at line 37 of file dynamics.py.

11.15.1.2 def lib.dyn.dynamics.generateFluxList (*tmpPath, tmpSysType, tmpLastID* = `None`)

Definition at line 17 of file dynamics.py.

11.15.1.3 def lib.dyn.dynamics.rangeFloat (*start, step, stop*)

Definition at line 31 of file dynamics.py.

11.16 lib.graph Namespace Reference

Namespaces

- network
- raf

11.17 lib.graph.network Namespace Reference

Functions

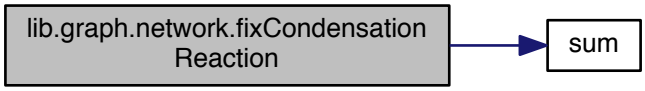
- def removeRareRcts
- def fixCondensationReaction

11.17.1 Function Documentation

11.17.1.1 def lib.graph.network.fixCondensationReaction (*m1, m2, m3, rcts*)

Definition at line 25 of file network.py.

Here is the call graph for this function:



11.17.1.2 def lib.graph.network.removeRareRcts (*graph, dt, life, nrg, deltat*)

Definition at line 16 of file network.py.

11.18 lib.graph.raf Namespace Reference

Functions

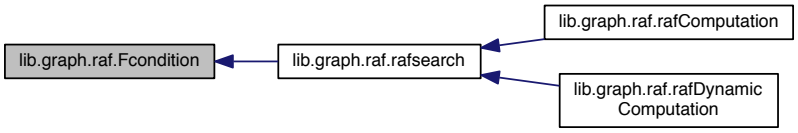
- def generateClosure
- def RAcondition
- def Fcondition
- def findCatforRAF
- def rafsearch
- def rafComputation
- def rafDynamicComputation
- def findRAFrcts

11.18.1 Function Documentation

11.18.1.1 def lib.graph.raf.Fcondition (*tmpCL, tmpRA, rcts, debug* = `False`)

Definition at line 60 of file raf.py.

Here is the caller graph for this function:



11.18.1.2 def lib.graph.raf.findCatforRAF (*tmpCat, tmpRAF, tmpClosure, debug* = `False`)

Definition at line 82 of file raf.py.

Here is the caller graph for this function:

11.18.1.3 def lib.graph.raf.findRAFrcts (*RAF, rcts, actrcts*)

Definition at line 160 of file raf.py.

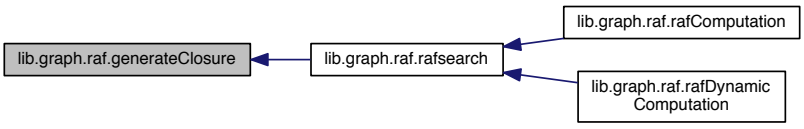
Here is the call graph for this function:

Here is the caller graph for this function:

11.18.1.4 def lib.graph.raf.generateClosure (*tmpF, rcts, debug* = `False`)

Definition at line 20 of file raf.py.

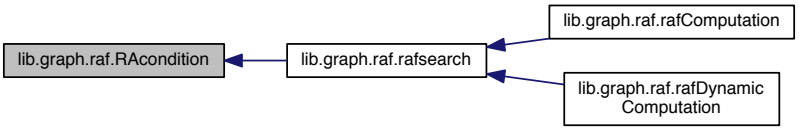
Here is the caller graph for this function:



11.18.1.5 def lib.graph.raf.RAcondition (*tmpCL, rcts, cats, debug* = `False`)

Definition at line 46 of file raf.py.

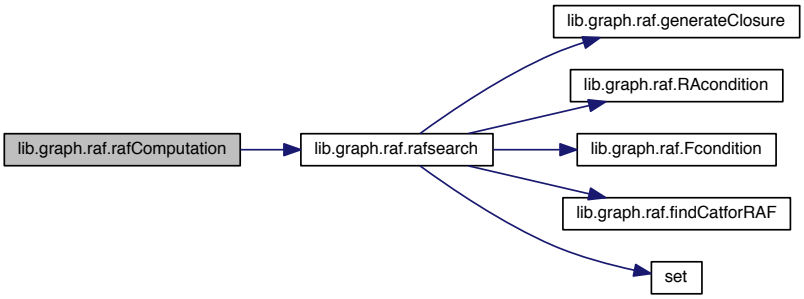
Here is the caller graph for this function:



11.18.1.6 def lib.graph.raf.rafComputation (*fid_initRafRes, fid_initRafResALL, fid_initRafResLIST, tmpDir, rctProb, avgCon, rcts, cats, foodList, maxDim, debug* = `False`)

Definition at line 134 of file raf.py.

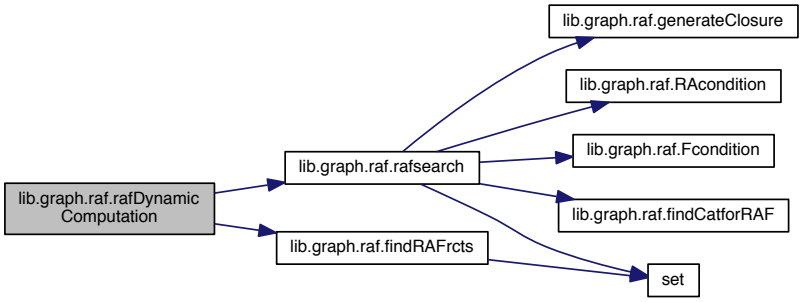
Here is the call graph for this function:



11.18.1.7 def lib.graph.raf.rafDynamicComputation (*fid_dynRafRes, tmpTime, rcts, cats, foodList, growth* = `False`, *rctsALL* = `None`, *catsALL* = `None`, *completeRCTS* = `None`, *debug* = `False`)

Definition at line 144 of file raf.py.

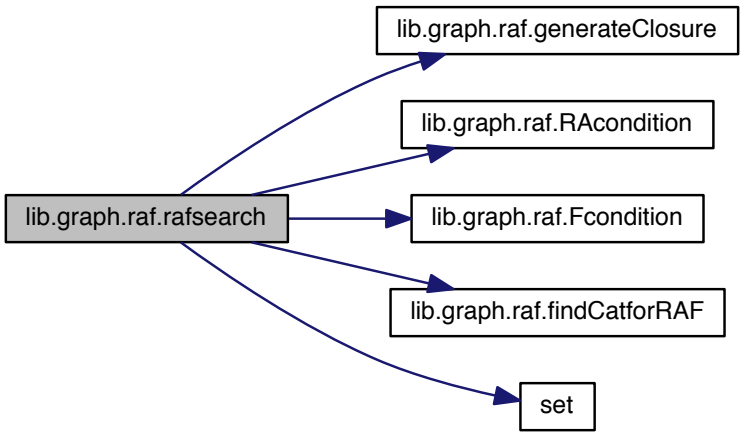
Here is the call graph for this function:



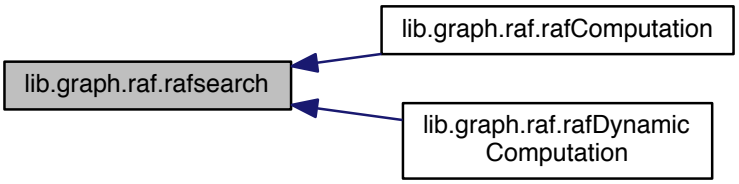
11.18.1.8 def lib.graph.raf.rafsearch (*rcts, cats, closure, debug* = `False`)

Definition at line 91 of file raf.py.

Here is the call graph for this function:



Here is the caller graph for this function:



11.19 lib.IO Namespace Reference

Namespaces

- readfiles
- writefiles

11.20 lib.IO.readfiles Namespace Reference

Functions

- def readConfFile
- def readInitConfFile
- def readBufferedID
- def readCSTRflux
- def loadAllData
- def zeroBeforeStrNum
- def splitRctParsLine

11.20.1 Function Documentation

11.20.1.1 def lib.IO.readfiles.loadAllData (*tmpPath, tmpFname*)

Definition at line 120 of file readfiles.py.

11.20.1.2 def lib.IO.readfiles.readBufferedID (*tmpPath*)

Definition at line 89 of file readfiles.py.

11.20.1.3 def lib.IO.readfiles.readConfFile (*tmpPath*)

Definition at line 14 of file readfiles.py.

11.20.1.4 def lib.IO.readfiles.readCSTRflux (*tmpPath*)

Definition at line 105 of file readfiles.py.

11.20.1.5 def lib.IO.readfiles.readInitConfFile (*tmpPath*)

Definition at line 52 of file readfiles.py.

11.20.1.6 def lib.IO.readfiles.splitRctParsLine (*tmpLine*)

Definition at line 135 of file readfiles.py.

11.20.1.7 def lib.IO.readfiles.zeroBeforeStrNum (*tmpI, tmpL*)

Definition at line 127 of file readfiles.py.

11.21 lib.IO.writefiles Namespace Reference

Functions

- def write_init_raf_list
- def write_init_raf_all

11.21.1 Function Documentation

11.21.1.1 def lib.IO.writefiles.write_init_raf_all (*fid, rafinfo, folder, rcts, cats*)

Definition at line 23 of file writefiles.py.

11.21.1.2 def lib.IO.writefiles.write_init_raf_list (*fid, rafinfo, folder*)

Definition at line 14 of file writefiles.py.

11.22 main Namespace Reference

Variables

- tuple parser
- tuple args = parser.parse_args()
- tuple strPath = os.path.abspath(args.strPath)
- tuple tmpDirs = sort(os.listdir(strPath))
- string ndn = '_0_new_allStatResults'
- tuple newdirAllResults = os.path.join(strPath, ndn)
- int _CLOSE_ = 0
- int _PROTO_ = 1
- int _CSTR_ = 2
- tuple fname_initRafRes = os.path.join(newdirAllResults, '0_initRafAnalysis.csv')
- tuple fname_initRafResLIST = os.path.join(newdirAllResults, '0_initRafAnalysisLIST.csv')
- tuple fname_initRafResALL = os.path.join(newdirAllResults, '0_initRafAnalysisALL.csv')
- tuple fid_initRafRes = open(fname_initRafRes, 'w')
- tuple fid_initRafResLIST = open(fname_initRafResLIST, 'w')
- tuple fid_initRafResALL = open(fname_initRafResALL, 'w')
- string strToWrite = "Folder\tP\tAC\tM\tRAFsize\tClosureSize\tCatsSize\tuRAF\n"
- tuple totDirName = os.path.join(strPath,tmpDir)

- tuple resDirPath = os.path.abspath(os.path.join("./", args.resFolder))
- tuple conf = readfiles.readConfFile(totDirName)
- tuple foodList = dm.generateFluxList(totDirName, sysType)
- tuple rcts = readfiles.loadAllData(totDirName,'_acsreactions.csv')
- tuple cats = readfiles.loadAllData(totDirName,'_acscatalysis.csv')
- tuple numberOfGen = len(glob.glob(os.path.join(resDirPath,'times_*')))
- tuple strZeros = readfiles.zeroBeforeStrNum(ngen, numberOfGen)
- string fName = 'RAF_structuresInTime_analysis_gen_'
- tuple fname_inTimeRafRes = os.path.join(newdirAllResults, fName)
- tuple fid_inTimeRafRes = open(fname_inTimeRafRes, 'w')
- potential = False
- string strRctZero = 'reactions_'
- string strCatZero = 'catalysis_'
- tuple rctFilesZero = sorted(glob.glob(os.path.join(resDirPath,strRctZero)))
- tuple catFilesZero = sorted(glob.glob(os.path.join(resDirPath,strCatZero)))
- string strRct = 'reactions_'
- string strCat = 'catalysis_'
- tuple rctFiles = sorted(glob.glob(os.path.join(resDirPath,strRct)))
- tuple catFiles = sorted(glob.glob(os.path.join(resDirPath,strCat)))
- list sngTime = conf[2]
- int actTime = 0
- list procrcts = rcts[rcts[:,5] > 0,:]
- list proccats = cats[cats[:,3] > 0,:]
- tuple R = raf.rafDynamicComputation(fid_inTimeRafRes, actTime, procrcts[:,0:5], proccats[:,0:5], foodList, potential, rcts, cats, debug=args.debug)
- tuple lastRct = readfiles.loadAllData(totDirName,rctFiles[-1])
- tuple lastCat = readfiles.loadAllData(totDirName,catFiles[-1])
- tuple fname_dynRafRes = os.path.join(newdirAllResults, fName)
- tuple fid_dynRafRes = open(fname_dynRafRes, 'w')
- string strRctPar = 'reactions_parameters_'
- tuple rctParamFile = sorted(glob.glob(os.path.join(resDirPath,strRctPar)))
- tuple fid = open(rctParamFile[0], 'r')
- int previousTime = 0
- decayTime = args.decay
- int condensation_counter = 0
- int endo_condensation_counter = 0
- int cleavage_counter = 0
- int endo_cleavage_counter = 0
- int nAnal = 1
- int rctCurrID = 0
- int catCurrID = 0
- timeInterval = rtime-previousTime
- tuple graph = network.removeRareRcts(graph,2,3,4,timeInterval)
- tuple graphSUB = network.removeRareRcts(graphSUB,2,3,4,timeInterval)
- tuple onrcts = network.removeRareRcts(onrcts,5,6,7,timeInterval)
- tuple oncats = network.removeRareRcts(oncats,3,4,5,timeInterval)
- tuple positionR = ((onrcts[:,1] == cc) & (onrcts[:,2] == mol_I) & (onrcts[:,3] == mol_II))
- tuple position = ((oncats[:,1] == cat) & (oncats[:,2] == onrcts[positionR,0]))

11.22.1 Variable Documentation

11.22.1.1 int main._CLOSE_ = 0

Definition at line 57 of file main.py.

11.22.1.2 int main._CSTR_ = 2

Definition at line 59 of file main.py.

11.22.1.3 int main._PROTO_ = 1

Definition at line 58 of file main.py.

11.22.1.4 int main.actTime = 0

Definition at line 144 of file main.py.

11.22.1.5 tuple main.args = parser.parse_args()

Definition at line 40 of file main.py.

11.22.1.6 int main.catCurrID = 0

Definition at line 191 of file main.py.

11.22.1.7 tuple main.catFiles = sorted(glob.glob(os.path.join(resDirPath,strCat)))

Definition at line 137 of file main.py.

11.22.1.8 tuple main.catFilesZero = sorted(glob.glob(os.path.join(resDirPath,strCatZero)))

Definition at line 132 of file main.py.

11.22.1.9 tuple main.cats = readfiles.loadAllData(totDirName,'_acscatalysis.csv')

Definition at line 95 of file main.py.

11.22.1.10 int main.cleavage_counter = 0

Definition at line 187 of file main.py.

11.22.1.11 int main.condensation_counter = 0

Definition at line 185 of file main.py.

11.22.1.12 tuple main.conf = readfiles.readConfFile(totDirName)

Definition at line 84 of file main.py.

11.22.1.13 main.decayTime = args.decay

Definition at line 184 of file main.py.

11.22.1.14 int main.endo_cleavage_counter = 0

Definition at line 188 of file main.py.

11.22.1.15 int main.endo_condensation_counter = 0

Definition at line 186 of file main.py.

11.22.1.16 tuple main.fid = open(**rctParamFile**[0], 'r')

Definition at line 182 of file main.py.

11.22.1.17 tuple main.fid_dynRafRes = open(**fname_dynRafRes**, 'w')

Definition at line 174 of file main.py.

11.22.1.18 tuple main.fid_initRafRes = open(**fname_initRafRes**, 'w')

Definition at line 65 of file main.py.

11.22.1.19 tuple main.fid_initRafResALL = open(**fname_initRafResALL**, 'w')

Definition at line 67 of file main.py.

11.22.1.20 tuple main.fid_initRafResLIST = open(**fname_initRafResLIST**, 'w')

Definition at line 66 of file main.py.

11.22.1.21 tuple main.fid_inTimeRafRes = open(**fname_inTimeRafRes**, 'w')

Definition at line 119 of file main.py.

11.22.1.22 string main.fName = 'RAF_structuresInTime_analysis_gen_'

Definition at line 117 of file main.py.

11.22.1.23 tuple main.fname_dynRafRes = os.path.join(**newdirAllResults**, **fName**)

Definition at line 173 of file main.py.

11.22.1.24 tuple main.fname_initRafRes = os.path.join(**newdirAllResults**, '0_initRafAnalysis.csv')

Definition at line 62 of file main.py.

11.22.1.25 tuple main.fname_initRafResALL = os.path.join(**newdirAllResults**, '0_initRafAnalysisALL.csv')

Definition at line 64 of file main.py.

11.22.1.26 tuple main.fname_initRafResLIST = os.path.join(newdirAllResults, '0_initRafAnalysisLIST.csv')

Definition at line 63 of file main.py.

11.22.1.27 tuple main.fname_inTimeRafRes = os.path.join(newdirAllResults, fName)

Definition at line 118 of file main.py.

11.22.1.28 tuple main.foodList = dm.generateFluxList(totDirName, sysType)

Definition at line 89 of file main.py.

11.22.1.29 tuple main.graph = network.removeRareRcts(graph,2,3,4,timeInterval)

Definition at line 205 of file main.py.

11.22.1.30 tuple main.graphSUB = network.removeRareRcts(graphSUB,2,3,4,timeInterval)

Definition at line 206 of file main.py.

11.22.1.31 tuple main.lastCat = readfiles.loadAllData(totDirName,catFiles[-1])

Definition at line 166 of file main.py.

11.22.1.32 tuple main.lastRct = readfiles.loadAllData(totDirName,rctFiles[-1])

Definition at line 165 of file main.py.

11.22.1.33 int main.nAnal = 1

Definition at line 189 of file main.py.

11.22.1.34 string main.ndn = '_0_new_allStatResults'

Definition at line 50 of file main.py.

11.22.1.35 tuple main.newdirAllResults = os.path.join(strPath, ndn)

Definition at line 51 of file main.py.

11.22.1.36 tuple main.numberOfGen = len(glob.glob(os.path.join(resDirPath,'times_*')))

Definition at line 102 of file main.py.

11.22.1.37 tuple main.oncats = network.removeRareRcts(oncats,3,4,5,timeInterval)

Definition at line 208 of file main.py.

11.22.1.38 tuple main.onrcts = network.removeRareRcts(onrcts,5,6,7,timeInterval)

Definition at line 207 of file main.py.

11.22.1.39 tuple main.parser

Initial value:

```
1 = ArgumentParser(
2                                 description='Main script of ACS analysis.'
3                                 , epilog='''ACS ANALYSIS Main File. ''')
```

Definition at line 28 of file main.py.

11.22.1.40 tuple main.position = ((oncats[:,1] == cat) & (oncats[:,2] == onrcts[positionR,0]))

Definition at line 233 of file main.py.

11.22.1.41 list main.positionR = ((onrcts[:,1] == cc) & (onrcts[:,2] == mol_I) & (onrcts[:,3] == mol_II))

Definition at line 221 of file main.py.

11.22.1.42 main.potential = False

Definition at line 122 of file main.py.

11.22.1.43 main.previousTime = 0

Definition at line 183 of file main.py.

11.22.1.44 list main.proccats = cats[cats[:,3] $>$ 0,:]

Definition at line 153 of file main.py.

11.22.1.45 list main.procrcts = rcts[rcts[:,5] $>$ 0,:]

Definition at line 152 of file main.py.

11.22.1.46 tuple main.R = raf.rafDynamicComputation(fid_inTimeRafRes, actTime, procrcts[:,0:5], proccats[:,0:5], foodList, potential, rcts, cats, debug=args.debug)

Definition at line 154 of file main.py.

11.22.1.47 int main.rctCurrID = 0

Definition at line 190 of file main.py.

11.22.1.48 tuple main.rctFiles = sorted(glob.glob(os.path.join(resDirPath,strRct)))

Definition at line 136 of file main.py.

11.22.1.49 tuple main.rctFilesZero = sorted(glob.glob(os.path.join(resDirPath,strRctZero)))

Definition at line 131 of file main.py.

11.22.1.50 tuple main.rctParamFile = sorted(glob.glob(os.path.join(resDirPath,strRctPar)))

Definition at line 181 of file main.py.

11.22.1.51 tuple main.rcts = readfiles.loadAllData(totDirName,'_acsreactions.csv')

Definition at line 94 of file main.py.

11.22.1.52 tuple main.resDirPath = os.path.abspath(os.path.join("./", args.resFolder))

Definition at line 79 of file main.py.

11.22.1.53 list main.sngTime = conf[2]

Definition at line 143 of file main.py.

11.22.1.54 string main.strCat = 'catalysis_'

Definition at line 134 of file main.py.

11.22.1.55 string main.strCatZero = 'catalysis_'

Definition at line 130 of file main.py.

11.22.1.56 tuple main.strPath = os.path.abspath(args.strPath)

Definition at line 43 of file main.py.

11.22.1.57 string main.strRct = 'reactions_'

Definition at line 133 of file main.py.

11.22.1.58 string main.strRctPar = 'reactions_parameters_'

Definition at line 180 of file main.py.

11.22.1.59 string main.strRctZero = 'reactions_'

Definition at line 129 of file main.py.

11.22.1.60 string main.strToWrite = "Folder\tP\tAC\tM\tRAFsize\tClosureSize\tCatsSize\tuRAF\n"

Definition at line 68 of file main.py.

11.22.1.61 tuple main.strZeros = readfiles.zeroBeforeStrNum(ngen, numberOfGen)

Definition at line 111 of file main.py.

11.22.1.62 main.timeInterval = rtime-previousTime

Definition at line 200 of file main.py.

11.22.1.63 tuple main.tmpDirs = sort(os.listdir(strPath))

Definition at line 44 of file main.py.

11.22.1.64 tuple main.totDirName = os.path.join(strPath,tmpDir)

Definition at line 73 of file main.py.

11.23 prepareNewSim Namespace Reference

Functions

- def zeroBeforeStrNum

Variables

- tuple parser
- tuple args = parser.parse_args()
- tuple StrFrom = os.path.abspath(args.StrFrom)
- tuple StrTo = os.path.abspath(args.StrTo)
- tuple StrFileSpeciesToGetConc = os.path.abspath(args.FileSpeciesToGetConc)
- tuple origin = os.getcwd()
- int _LASTSPECIES_ = 29
- tuple _REVRCTS_ = int(args.revRct)
- tuple _RATIOREV_ = int(args.k_revRct)
- float _CLEAVAGE_ = 25.0
- float _CONDENSATION_ = 50.0
- float _COMPLEXFORM_ = 50.0
- tuple _COMPLEXDISS_ = float(args.kDiss)
- tuple _INITSPECIESCONC_ = float(args.singleInitConc)
- tuple fileDest = os.path.join(StrTo,"_acsinflux.csv")
- tuple sourceResFolder = os.path.join(StrFrom,"res")
- tuple lastSpeciesFile = sorted(glob.glob('species_*'))
- tuple lastReactionsFile = sorted(glob.glob('reactions_1*'))
- tuple lastCatalysisFile = sorted(glob.glob('catalysis_*'))
- tuple mod = open("acsm2s.conf")
- int id = 0
- tuple linesplitted = line.split("=")
- tuple file = open("acsm2s.conf", "w")
- list concs = []
- tuple specFileLines = open(StrFileSpeciesToGetConc)
- tuple mod_rct = open("_acsreactions.csv")
- int flag = 0
- tuple catRct = linecache.getline('_acsreactions.csv', int(linesplitted[2])+1)
- tuple carRctSplit = catRct.split("\t")

11.23.1 Function Documentation

11.23.1.1 def prepareNewSim.zeroBeforeStrNum (*tmpI*, *tmpL*)

Definition at line 12 of file prepareNewSim.py.

11.23.2 Variable Documentation

11.23.2.1 float prepareNewSim._CLEAVAGE_ = 25.0

Definition at line 51 of file prepareNewSim.py.

11.23.2.2 tuple prepareNewSim._COMPLEXDISS_ = float(args.kDiss)

Definition at line 54 of file prepareNewSim.py.

11.23.2.3 float prepareNewSim._COMPLEXFORM_ = 50.0

Definition at line 53 of file prepareNewSim.py.

11.23.2.4 float prepareNewSim._CONDENSATION_ = 50.0

Definition at line 52 of file prepareNewSim.py.

11.23.2.5 tuple prepareNewSim._INITSPECIESCONC_ = float(args.singleInitConc)

Definition at line 55 of file prepareNewSim.py.

11.23.2.6 int prepareNewSim._LASTSPECIES_ = 29

Definition at line 48 of file prepareNewSim.py.

11.23.2.7 tuple prepareNewSim._RATIOREV_ = int(args.k_revRct)

Definition at line 50 of file prepareNewSim.py.

11.23.2.8 tuple prepareNewSim._REVRCTS_ = int(args.revRct)

Definition at line 49 of file prepareNewSim.py.

11.23.2.9 tuple prepareNewSim.args = parser.parse_args()

Definition at line 36 of file prepareNewSim.py.

11.23.2.10 tuple prepareNewSim.carRctSplit = catRct.split("\t")

Definition at line 217 of file prepareNewSim.py.

11.23.2.11 tuple prepareNewSim.catRct = linecache.getline('_acsreactions.csv', int(linesplitted[2])+1)

Definition at line 216 of file prepareNewSim.py.

11.23.2.12 list prepareNewSim.concs = [ ]

Definition at line 142 of file prepareNewSim.py.

11.23.2.13 tuple prepareNewSim.file = open("acsm2s.conf", "w")

Definition at line 131 of file prepareNewSim.py.

11.23.2.14 tuple prepareNewSim.fileDest = os.path.join(StrTo,"_acsinflux.csv")

Definition at line 61 of file prepareNewSim.py.

11.23.2.15 int prepareNewSim.flag = 0

Definition at line 211 of file prepareNewSim.py.

11.23.2.16 int prepareNewSim.id = 0

Definition at line 100 of file prepareNewSim.py.

11.23.2.17 tuple prepareNewSim.lastCatalysisFile = sorted(glob.glob('catalysis_*'))

Definition at line 74 of file prepareNewSim.py.

11.23.2.18 tuple prepareNewSim.lastReactionsFile = sorted(glob.glob('reactions_1*'))

Definition at line 73 of file prepareNewSim.py.

11.23.2.19 tuple prepareNewSim.lastSpeciesFile = sorted(glob.glob('species_*'))

Definition at line 71 of file prepareNewSim.py.

11.23.2.20 tuple prepareNewSim.linesplitted = line.split("=")

Definition at line 103 of file prepareNewSim.py.

11.23.2.21 tuple prepareNewSim.mod = open("acsm2s.conf")

Definition at line 99 of file prepareNewSim.py.

11.23.2.22 tuple prepareNewSim.mod_rct = open("_acsreactions.csv")

Definition at line 208 of file prepareNewSim.py.

11.23.2.23 tuple prepareNewSim.origin = os.getcwd()

Definition at line 47 of file prepareNewSim.py.

11.23.2.24 tuple prepareNewSim.parser

Initial value:

```
1 = ArgumentParser(
2                                 description='Script to create new init simulation files starting from the
      end of a previous simulation.'
3                                 , epilog='''Species concentration are initilized according to the sequence
      uploaded, pay attention to the file selected. ''')
```

Definition at line 22 of file prepareNewSim.py.

11.23.2.25 tuple prepareNewSim.sourceResFolder = os.path.join(StrFrom,"res")

Definition at line 67 of file prepareNewSim.py.

11.23.2.26 tuple prepareNewSim.specFileLines = open(StrFileSpeciesToGetConc)

Definition at line 143 of file prepareNewSim.py.

11.23.2.27 tuple prepareNewSim.StrFileSpeciesToGetConc = os.path.abspath(args.FileSpeciesToGetConc)

Definition at line 41 of file prepareNewSim.py.

11.23.2.28 tuple prepareNewSim.StrFrom = os.path.abspath(args.StrFrom)

Definition at line 39 of file prepareNewSim.py.

11.23.2.29 tuple prepareNewSim.StrTo = os.path.abspath(args.StrTo)

Definition at line 40 of file prepareNewSim.py.

11.24 resetForNewSimulations Namespace Reference

Functions

- def zeroBeforeStrNum

Variables

- int foldersSIMS = 10
- int foldersREP = 10
- tuple zerosSIMS = zeroBeforeStrNum(i,foldersSIMS)
- tuple zerosREPS = zeroBeforeStrNum(j,foldersREP)
- string folderName = "s_inv_1e-2_"
- string folderNew = "s_inv_1e-1_"
- tuple resdir = os.path.join(os.curdir, "res")

- tuple crtSimFolder = os.path.join(StrTo,folderName)
- tuple fileDest = os.path.join(StrTo,folderNew,"_acsinflux.csv")
- tuple speciesFiles = sorted(glob.glob("species_1_∗"))
- tuple mod = open("_acsspecies.csv")
- int id = 0
- tuple linesplitted = line.split("\t")
- tuple file = open("_acsspecies.csv", "w")

11.24.1 Function Documentation

11.24.1.1 def resetForNewSimulations.zeroBeforeStrNum (*tmpl, tmpL*)

Definition at line 5 of file resetForNewSimulations.py.

11.24.2 Variable Documentation

11.24.2.1 tuple resetForNewSimulations.crtSimFolder = os.path.join(StrTo,folderName)

Definition at line 51 of file resetForNewSimulations.py.

11.24.2.2 tuple resetForNewSimulations.file = open("_acsspecies.csv", "w")

Definition at line 99 of file resetForNewSimulations.py.

11.24.2.3 tuple resetForNewSimulations.fileDest = os.path.join(StrTo,folderNew,"_acsinflux.csv")

Definition at line 57 of file resetForNewSimulations.py.

11.24.2.4 string resetForNewSimulations.folderName = "s_inv_1e-2_"

Definition at line 31 of file resetForNewSimulations.py.

11.24.2.5 string resetForNewSimulations.folderNew = "s_inv_1e-1_"

Definition at line 32 of file resetForNewSimulations.py.

11.24.2.6 int resetForNewSimulations.foldersREP = 10

Definition at line 22 of file resetForNewSimulations.py.

11.24.2.7 int resetForNewSimulations.foldersSIMS = 10

Definition at line 21 of file resetForNewSimulations.py.

11.24.2.8 int resetForNewSimulations.id = 0

Definition at line 83 of file resetForNewSimulations.py.

11.24.2.9 tuple resetForNewSimulations.linesplitted = line.split("\t")

Definition at line 85 of file resetForNewSimulations.py.

11.24.2.10 tuple resetForNewSimulations.mod = open("_acsspecies.csv")

Definition at line 82 of file resetForNewSimulations.py.

11.24.2.11 tuple resetForNewSimulations.resdir = os.path.join(os.curdir, "res")

Definition at line 42 of file resetForNewSimulations.py.

11.24.2.12 tuple resetForNewSimulations.speciesFiles = sorted(glob.glob("species_1_*"))

Definition at line 66 of file resetForNewSimulations.py.

11.24.2.13 tuple resetForNewSimulations.zerosREPS = zeroBeforeStrNum(j,foldersREP)

Definition at line 29 of file resetForNewSimulations.py.

11.24.2.14 tuple resetForNewSimulations.zerosSIMS = zeroBeforeStrNum(i,foldersSIMS)

Definition at line 28 of file resetForNewSimulations.py.

Chapter 12

Class Documentation

12.1 catalysis Class Reference

CATALYSIS class.

`#include <catalysis.h>`

Public Member Functions

- catalysis ()
- catalysis (acs_longInt tmpCatId, acs_longInt tmpCat, acs_longInt tmpRctId, acs_longInt tmpAmount, acs_-double tmpKass, acs_double tmpKdiss, acs_double tmpK_cpx, acs_int tmpCpxTarget)
- ~catalysis ()
- acs_longInt getCatId () const
- acs_longInt getCat () const
- acs_longInt getReactionID () const
- acs_longInt getTotAmount () const
- acs_double getKass () const
- acs_double getKdiss () const
- acs_double getK_cpx () const
- acs_int getCpxTarget () const
- void updateTotAmount ()
- void resetEventsCounter ()

12.1.1 Detailed Description

CATALYSIS class.

This class contains catalysis proprieties and methods

Author

Alessandro Filisetti

Version

0.1

Date

2009-04-16

Definition at line 16 of file catalysis.h.

12.1.2 Constructor & Destructor Documentation

12.1.2.1 catalysis::catalysis ( )

12.1.2.2 catalysis::catalysis (acs_longInt *tmpCatId,* acs_longInt *tmpCat,* acs_longInt *tmpRctId,* acs_longInt *tmpAmount,* acs_double *tmpKass,* acs_double *tmpKdiss,* acs_double *tmpK_cpx,* acs_int *tmpCpxTarget*)

```
catalysis class constructor (FROM FILE)
```

Version

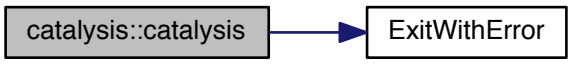
0.1

Date

2010-03-16

Definition at line 19 of file catalysis.cpp.

Here is the call graph for this function:



12.1.2.3 catalysis::~catalysis ( ) [inline]

Definition at line 35 of file catalysis.h.

12.1.3 Member Function Documentation

12.1.3.1 acs_longInt catalysis::getCat ( ) const [inline]

Definition at line 39 of file catalysis.h.

12.1.3.2 acs_longInt catalysis::getCatId ( ) const [inline]

Definition at line 38 of file catalysis.h.

12.1.3.3 acs_int catalysis::getCpxTarget ( ) const [inline]

Definition at line 46 of file catalysis.h.

12.1.3.4 acs_double catalysis::getK_cpx ( ) const [inline]

Definition at line 44 of file catalysis.h.

12.1.3.5 acs_double catalysis::getKass ( ) const [inline]

Definition at line 42 of file catalysis.h.

12.1.3.6 acs_double catalysis::getKdiss ( ) const [inline]

Definition at line 43 of file catalysis.h.

12.1.3.7 acs_longInt catalysis::getReactionID ( ) const [inline]

Definition at line 40 of file catalysis.h.

12.1.3.8 acs_longInt catalysis::getTotAmount ( ) const [inline]

Definition at line 41 of file catalysis.h.

12.1.3.9 void catalysis::resetEventsCounter ( ) [inline]

Definition at line 50 of file catalysis.h.

12.1.3.10 void catalysis::updateTotAmount ( ) [inline]

Definition at line 49 of file catalysis.h.

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

- /Users/alessandrofilisetti/Documents/GIT/carness/catalysis.h
- /Users/alessandrofilisetti/Documents/GIT/carness/catalysis.cpp

12.2 commonFunctions Class Reference

This class contains all the common function of the system.

`#include <commonFunctions.h>`

12.2.1 Detailed Description

This class contains all the common function of the system.

This class contains all the functions useful in general

Authors

alessandro filisetti

Date

2011/12/10

Version

1.0

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

- /Users/alessandrofilisetti/Documents/GIT/carness/commonFunctions.h

12.3 environment Class Reference

environment class

`#include <environment.h>`

Public Member Functions

- environment (string tmpInitialPath)
- ~environment ()
- acs_int getNgen () const
- acs_int getNsim () const
- acs_double getActualTime () const
- acs_double getNseconds () const
- acs_int getNreactions () const
- acs_double getMAXhours () const
- acs_int getMAXattempts () const
- acs_int getCurrentAttempts () const
- acs_double getTimeStructuresSavingInterval () const
- acs_double getFileTimesSavingInterval () const
- acs_int getLastFiringDiskSpeciesID () const
- acs_double getOverallConcentration () const
- acs_int getMaxNonCatalyticLength () const
- acs_double getRctProb () const
- acs_double getCleavProb () const
- bool getReverseReactions () const
- acs_int getEnergy () const
- acs_double getRatioSpeciesEnergizable () const
- acs_int getADP () const
- acs_int getATP () const
- acs_longInt getMols () const
- acs_longInt getNewMols () const
- acs_longInt getNspecies () const
- acs_longInt getNnewSpecies () const
- acs_longInt getNcpx () const
- acs_longInt getNcpxMols () const
- acs_double getGillespieMean () const
- acs_double getgillespieSD () const
- acs_double getgillespieEntropy () const
- acs_double getRatioBetweenNewGillTotGill () const
- acs_double getRatioBetweenBackandForw () const
- bool getSystemExpFlag () const
- acs_double getKdiss () const
- acs_double getKass () const
- acs_double getKcpx () const
- acs_double getKcpxDiss () const
- acs_double getKnrg () const
- acs_double getKirrad () const
- acs_double getK_spont_diss () const
- acs_double getK_spont_ass () const
- acs_double getCleavageKC () const
- acs_double getComplexKC () const
- acs_double getCondensationKC () const
- acs_double getComplexDegKC () const

- acs_double getMoleculeDecayKC () const
- acs_int getMaxLOut () const
- acs_int getSolubilityThreshold () const
- acs_double getDiffusionContribute () const
- acs_double getInflux () const
- acs_double getRefillInterval () const
- string getAlphabet () const
- acs_double getVolume () const
- acs_double getRandomSeed () const
- vector< species > getMoleculesPopulation () const
- acs_longInt getTotalNumberOfSpecies ()
- acs_longInt getTotalNumberOfPossibleCatalysts ()
- acs_longInt getTotalNumberOfMolecules ()
- acs_longInt getTotalNumberOfComplexSpecies ()
- acs_longInt getTotalNumberOfComplexes ()
- acs_longInt getTotalNumberOfMonomers ()
- vector< reactions > getReactionsLayer () const
- int getDebugLevel () const
- acs_longInt getNumberOfTheoreticalSpecies () const
- acs_longInt getNumberOfReactions () const
- acs_longInt getNumberOfCatalysis () const
- acs_longInt getNumberOfGillespieCOPYpossibleRcts () const
- acs_longInt getNumberOfGillespiePossibleRcts () const
- void setLivingSpeciesIDsAndAmounts ()
- void setNotChargedAndChargedSpeciesIDsAndAmounts ()
- acs_longInt getCleavageCounter () const
- acs_longInt getEndoCleavageCounter () const
- acs_longInt getCondensationCounter () const
- acs_longInt getEndoCondensationCounter () const
- acs_longInt getCpxFormCounter () const
- acs_longInt getCpxDissCounter () const
- acs_longInt getOverallLoadedMolsCounter () const
- acs_longInt getSpontDissCounter () const
- acs_longInt getSpontAssCounter () const
- acs_int getTotNumberOfChargedMols ()
- void showGlobalParameter ()
- void printInitialCondition ()
- void printAllSpeciesIdAndSequence ()
- void printGillespieStructure ()
- void printNutrientsAndProbability ()
- void printAllChargeMols ()
- bool createInitialMoleculesPopulationFromFileSTD (string tmpSpeciesFilePath)
- bool createInitialReactionsLayerFromFileSTD (string tmpSpeciesFilePath)
- bool createInitialCatalysisLayerFromFileSTD (string tmpCatalysisFilePath)
- bool createInfluxLayersFromFileSTD (string tmpInfluxFilePath)
- bool createNrgBooleanFunctionsFromFileSTD (string tmpBoolNrgFilePath)
- bool createInitialMoleculesPopulationFromSpecificFileSTD (string tmpSpeciesFilePath, acs_int tmpActGEN, acs_int tmpActSIM)
- bool createInitialReactionsLayerFromSpecificFileSTD (string tmpReactionsFilePath, acs_int tmpActGEN, acs_int tmpActSIM)
- bool createInitialCatalysisLayerFromSpecificFileSTD (string tmpCatalysisFilePath, acs_int tmpActGEN, acs_int tmpActSIM)
- void nutrientsAmountsFixing ()
- acs_int computeSngSpeciesRctsNumber (acs_longInt tmpTotalNumberOfReactions, MTRand &tmpRndDoubleGen)

- acs_int selectWhetherCleavageOrCond (MTRand &tmp__RndDoubleGen)
- bool createReactionsForThisSpecies (acs_longInt tmpsID, acs_int tmpReactionsForThisSpecies, MTRand &tmp_RndDoubleGen, vector< acs_longInt > &tmpIDOfCandidateSpecies, acs_int tmpRctCreationType)
- bool updateReactions (acs_longInt tmpIDtoUpdate, acs_longInt tmpNewSpecies, acs_int tmpRctType, vector< acs_longInt > &tmp_AlreadyEvaluatedSpeciesVector, MTRand &tmp_RndDoubleGen)
- acs_double createDiffusionRenforcement (acs_double tmpDiffEnh, acs_int tmpNewSpeciesLength)
- bool setSolubility (acs_int tmpNewSpeciesLength, MTRand &tmpRndDoubleGen)
- acs_longInt returnPosSpeciesAlreadyPresent (string tmpNewSequence)
- acs_longInt returnPosReactionAlreadyPresent (acs_int tmpReactionType, acs_longInt tmpIds_I, acs_longInt tmpIds_II, acs_longInt tmpIds_III)
- bool checkIfTheReactionIsAlreadyCatalyzedByThisSpecies (acs_longInt tmpSPeciesID, acs_longInt tmpIdReaction)
- bool performOPTGillespieComputation (MTRand &tmpRndDoubleGen, clock_t &tmpTimeElapsed, acs_int tmpActGEN, acs_int tmpActSIM, acs_int tmpActSTEP, string tmpStoringPath)
- bool performReaction (acs_longInt reaction_u, MTRand &tmp_RndDoubleGen, acs_int tmp_ActGEN, acs_int tmp_ActSIM, acs_int tmp_ActSTEP, string tmp_StoringPath)
- bool newSpeciesEvaluationIII (acs_longInt tmpNewSpecies, MTRand &tmp___RndDoubleGen)
- bool complexEvaluation (string tmpComplex, MTRand &tmp___RndDoubleGen, acs_int tmpCuttingPnt, acs_longInt tmpCatalyst_ID, acs_longInt tmpCatID, acs_longInt tmpSubstrate_ID, acs_longInt tmpSecSub_ID, bool tmpCpxType)
- acs_double computeSinglGilScore (acs_longInt tmpAmountI, acs_double tmpDifI, acs_int tmpSolI, acs_longInt tmpAmountII, acs_double tmpDifII, acs_int tmpSolII, acs_double tmpK, bool tmpSameMol)
- void performSingleGilleSpieIntroduction (acs_longInt tmpAmountI, acs_longInt tmpAmountII, acs_longInt tmpIDI, acs_longInt tmpIDII, acs_longInt tmpIDCatalysis, acs_int tmp__rctType, acs_longInt tmpMol_I, acs_longInt tmpMol_II, acs_longInt tmpMol_III, acs_longInt tmpMol_IV, acs_int tmpNRGDirection, acs_longInt tmpRctID, bool tmpSameSpeciesControl)
- void changeVolume (acs_int tmpTimeSinceLastReaction)
- void incNumberOfSpecies ()
- void decNumberOfSpecies (acs_int tmpID)
- void incNumberOfMols ()
- void decNumberOfMols ()
- void incNumberOfCpx ()
- void decNumberOfCpx (acs_int tmpID)
- void incNumberOfCpxMols ()
- void decNumberOfCpxMols ()
- void incNumberOfNewSpecies (acs_int tmpID)
- void decNumberOfNewSpecies (acs_int tmpID)
- void incNumberOfNewMols (acs_int tmpID)
- void decNumberOfNewMols (acs_int tmpID)
- void decMolSpeciesProcedure (acs_int tmp_ID)
- void decCpxProcedure (acs_int tmp_ID)
- void incMolProcedure (acs_int tmp_ID)
- void incSpeciesProcedure (acs_int tmp_ID)
- void unchargeMolProcess (acs_int tmp_ID)
- void incCleavageCounter ()
- void incEndoCleavageCounter ()
- void incCondensationCounter ()
- void incEndoCondensationCounter ()
- void incCpxFormCounter ()
- void incCpxDissCounter ()
- void incOverallLoadedMolsCounter ()
- void decOverallLoadedMolsCounter ()
- void incSpontDissCounter ()
- void incSpontAssCounter ()
- void resetCleavageCounter ()

- void resetEndoCleavageCounter ()
- void resetCondensationCounter ()
- void resetEndoCondensationCounter ()
- void resetOverallLoadedMolsCounter ()
- void resetCpxFormCounter ()
- void resetCpxDissCounter ()
- void resetSpontDissCounter ()
- void resetSpontAssCounter ()
- void resetReactionsCounter ()
- bool addChargeMolToList (acs_int tmpSpeciesID)
- bool removeChargeMolFromList (acs_int tmpSpeciesID)
- void clearAllStructures ()
- void resetConcentrationToInitialConditions ()
- void storeInitialStructures ()
- bool performRefill (acs_double tmpTimeSinceTheLastInFlux, acs_double tmpMinimalTimeForOneMols, MTRand &tmp__RndDoubleGen)
- bool performMoleculesEfflux (acs_double tmpTimeInterval, MTRand &tmp_RndDoubleGen)
- bool performDETMoleculesCharging (acs_double tmpTimeInterval, MTRand &tmp_RndDoubleGen)
- bool performDETComplexDissociation (acs_double tmpTimeInterval, MTRand &tmp_RndDoubleGen)
- void setActualTime (acs_double tmpActualTime)
- void updateSpeciesAges ()
- void increaseAttempts ()
- bool performCondensation (acs_longInt tmpCatalyst, acs_longInt tmpSubstrate, acs_longInt tmpProduct, acs_longInt tmpComplex, acs_longInt tmpIdReaction, acs_longInt tmpIdCatalysis, MTRand &tmp__RndDoubleGen)
- bool perform_endo_Condensation (acs_longInt tmpCatalyst, acs_longInt tmpSubstrate, acs_longInt tmpProduct, acs_longInt tmpComplex, acs_int tmpNRGside, acs_longInt tmpIdReaction, acs_longInt tmpIdCatalysis, MTRand &tmp__RndDoubleGen)
- bool performCleavage (acs_longInt tmpSubstrate, acs_longInt tmpProduct_I, acs_longInt tmpProduct_II, acs_longInt tmpIdReaction, acs_longInt tmpIdCatalysis, MTRand &tmp__RndDoubleGen)
- bool perform_endo_Cleavage (acs_longInt tmpSubstrate, acs_longInt tmpProduct_I, acs_longInt tmpProduct_II, acs_int tmpNRGside, acs_longInt tmpIdReaction, acs_longInt tmpIdCatalysis, MTRand &tmp__RndDoubleGen)
- bool performComplexFormation (acs_longInt tmpCatalyst, acs_longInt tmpSubstrate, acs_longInt tmpCatID, acs_longInt tmpSecSub, MTRand &tmp__RndDoubleGen)
- bool perform_endo_ComplexFormation (acs_longInt tmpCatalyst, acs_longInt tmpSubstrate, acs_longInt tmpCatID, acs_longInt tmpSecSub, acs_int tmpNRGSide, MTRand &tmp__RndDoubleGen)
- bool performComplexDissociation (acs_longInt tmpComplex, acs_longInt tmpCatalyst, acs_longInt tmpSubstrate, MTRand &tmp__RndDoubleGen)
- bool performSpontaneousCondensation (acs_longInt tmpReaction, MTRand &tmp__RndDoubleGen)
- bool performSpontaneousCleavage (acs_longInt tmpReaction, MTRand &tmp__RndDoubleGen)
- bool performMoleculeEfflux (acs_longInt tmpSpecies, MTRand &tmp__RndDoubleGen)
- bool performEnergyEfflux (MTRand &tmp__RndDoubleGen)
- bool structureCoherenceCheckUp ()
- bool notInverseReactionAlreadyCatalyzed (acs_int tmpRct, acs_longInt tmpID_I, acs_longInt tmpID_II)
- bool checkIfOnlyMutualCatalysis (acs_int tmpCat, acs_int tmpCandidateProduct)
- bool checkAvailability (acs_longInt tmpMI, acs_longInt tmpMII, acs_longInt tmpQI, acs_longInt tmpQII)
- void inserSubListInSpecies ()
- void showSubListInSpecies ()
- void showGillEngagementInSpecies ()
- bool saveConfigurationFileSTD (string tmpStoringPath)
- bool saveInfluxStructureSTD (string tmpStoringPath)
- bool saveNrgBoolFncStructureSTD (string tmpStoringPath)
- string zeroBeforeStringNumberSTD (acs_int tmpTotN, acs_int tmpCurrentN)
- bool saveSpeciesStructureSTD (acs_int tmpCurrentGen, acs_int tmpCurrentSim, acs_int tmpCurrentStep, string tmpStoringPath)

- bool saveReactionsStructureSTD (acs_int tmpCurrentGen, acs_int tmpCurrentSim, acs_int tmpCurrentStep, string tmpStoringPath)
- bool saveCatalysisStructureSTD (acs_int tmpCurrentGen, acs_int tmpCurrentSim, acs_int tmpCurrentStep, string tmpStoringPath)
- bool saveTimesSTD (acs_int tmpCurrentGen, acs_int tmpCurrentSim, acs_int tmpCurrentStep, string tmpStoringPath)
- bool saveReactionsParametersSTD (acs_int tmp__CurrentGen, acs_int tmp__CurrentSim, acs_int tmp__CurrentStep, string tmp__StoringPath, acs_int tmpRctType, acs_longInt tmpCat, acs_longInt tmpMol_I, acs_longInt tmpMol_II, acs_longInt tmpMol_III)
- bool saveLivingSpeciesIDSTD (acs_int tmp__CurrentGen, acs_int tmp__CurrentSim, acs_int tmp__CurrentStep, string tmp__StoringPath)
- bool saveLivingSpeciesAmountSTD (acs_int tmp__CurrentGen, acs_int tmp__CurrentSim, string tmp__StoringPath)
- bool saveLivingSpeciesConcentrationSTD (acs_int tmp__CurrentGen, acs_int tmp__CurrentSim, string tmp__StoringPath)
- bool devStd ()
- bool entropy ()

12.3.1 Detailed Description

environment class

Author

Alessandro Filisetti

Version

2.4

Date

2010-06-10 ∗^∗^∗^∗^∗

This class contains environmental proprieties and methods; within this class all things occur

Author

Alessandro Filisetti

Version

0.2

Date

2011-12-15

Definition at line 20 of file environment.h.

12.3.2 Constructor & Destructor Documentation

12.3.2.1 environment::environment (string *tmpInitialPath*)

TEMPLATE DEACLARATION AND DEFINITION Test environment costructor

Version

2.4

Date

2010-06-27

Parameters

tmpRndDouble-Gen	randomGenerator reference Environment Constructor

Version

1.0

Parameters

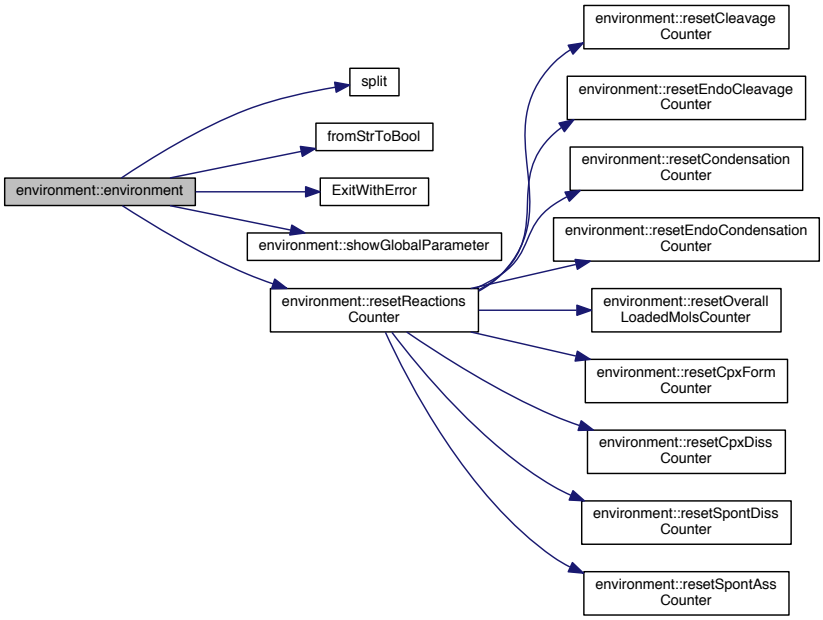
tmpInitialPath	

Date

2013/07/04

Definition at line 80 of file environment.cpp.

Here is the call graph for this function:



12.3.2.2 environment::~environment ( ) [inline]

Definition at line 148 of file environment.h.

12.3.3 Member Function Documentation

12.3.3.1 bool environment::addChargeMolToList (acs_int *tmpSpeciesID*)

Perform vector unchargedIDlist update adding a new charge molecule vector unchargedIDlist and cumUncharged-AmountList are involved

Version

1.0

Date

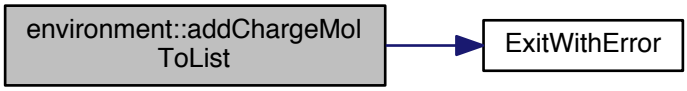
2010-10-10

Parameters

acs_int	tmpSpeciesID Specie to charge

Definition at line 4313 of file environment.cpp.

Here is the call graph for this function:



12.3.3.2 void environment::changeVolume (acs_int *tmpTimeSinceLastReaction*)

Change volume function 1.0

Date

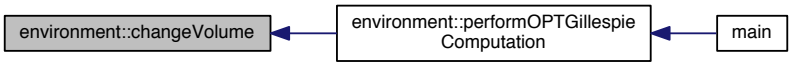
2013/07/17

Author

Alessandro Filisetti

Definition at line 6157 of file environment.cpp.

Here is the caller graph for this function:



12.3.3.3 bool environment::checkAvailability (acs_longInt *tmpMI*, acs_longInt *tmpMII*, acs_longInt *tmpQI*, acs_longInt *tmpQII*)

Return TRUE if there are sufficient molecules for the reaction. It is used for the reaction in which catalyst and substrate are the same molecules

Version

1.0

Date

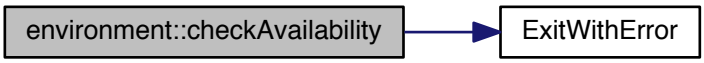
2011.07.25

Parameters

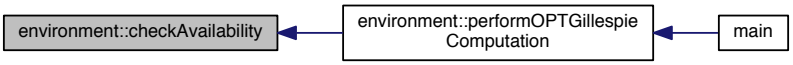
tmpMI	
tmpMII	
tmpQI	
tmpQII	

Definition at line 2635 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.4 bool environment::checkIfOnlyMutualCatalysis (acs_int *tmpCat*, acs_int *tmpCandidateProduct*)

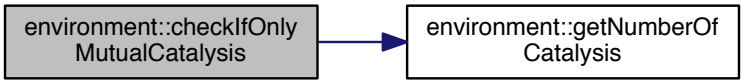
This function return false if the tmpCandidateProduct is a catalyst of tmpCat

Version

1.0 - last update 2009/10/08 - build 009

Definition at line 1965 of file environment.cpp.

Here is the call graph for this function:



12.3.3.5 bool environment::checkIfTheReactionIsAlreadyCatalyzedByThisSpecies (acs_longInt *tmpSPeciesID*, acs_longInt *tmpIdReaction*)

If the reaction is not new this function checks if the reactions has been already catalysed by this species

Version

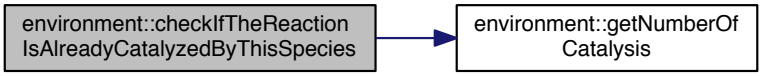
1.0

Parameters

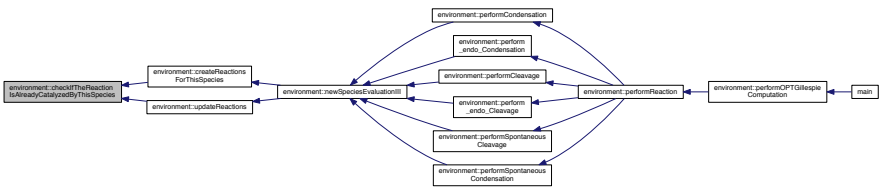
acs_int	tmpSPeciesID catalyst ID
tmpIdReaction	reaction ID

Definition at line 1937 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.6 void environment::clearAllStructures ( )

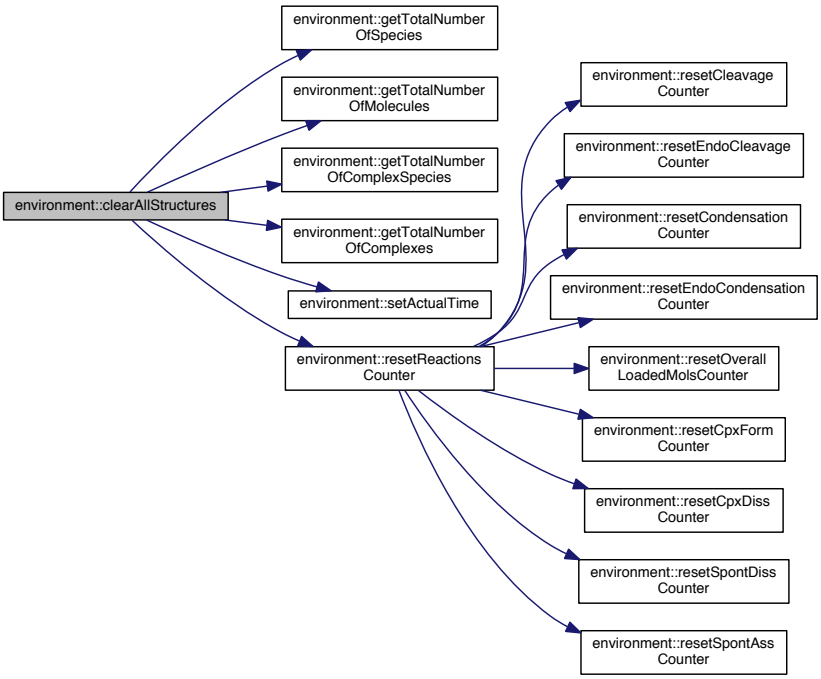
Clear all structures after each simulation

Version

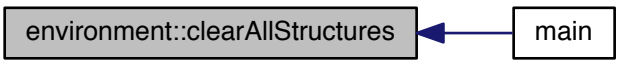
1.0

Definition at line 6405 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.7 bool environment::complexEvaluation (string *tmpComplex*, MTRand & *tmp___RndDoubleGen*, acs_int *tmpCuttingPnt*, acs_longInt *tmpCatalyst_ID*, acs_longInt *tmpSubstrate_ID*, acs_longInt *tmpCatID*, acs_longInt *tmpSecSub_ID*, bool *tmpCpxType*)

Complex evaluation

Version

1.1

Date

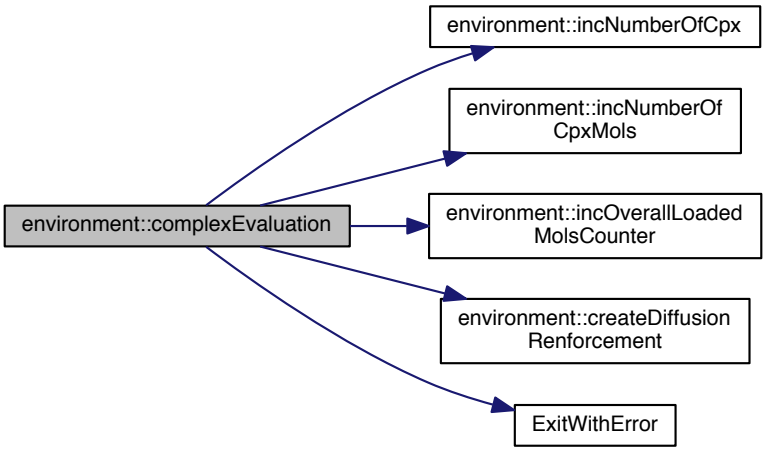
2010-06-04

Parameters

string	tmpNewSpecies New species sequence to evaluate
MTRand&	tmp___RndDoubleGen random number generator
tmpCuttingPnt	Complex cutting point
tmpCatalyst_ID	catalyst ID
tmpSubstrate_ID	substrate ID
tmp_catalysisID	catalysis ID
tmpCpxType	ENDOERGONIC or ESOERGONIC

Definition at line 6025 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

12.3.3.8 acs_double environment::computeSinglGilScore (acs_longInt *tmpAmountI*, acs_double *tmpDifI*, acs_int *tmpSolI*, acs_longInt *tmpAmountII*, acs_double *tmpDifII*, acs_int *tmpSolII*, acs_double *tmpK*, bool *tmpSameMol*)

Compute a single gillespie score according to the amount and peoprieties of the species involved

Version

1.0

Date

20110214

Definition at line 3288 of file environment.cpp.

Here is the caller graph for this function:

12.3.3.9 acs_int environment::computeSngSpeciesRctsNumber (acs_longInt *tmpTotalNumberOfReactions,* MTRand & *tmpRndDoubleGen*)

Initial molecule population creation. If the number of species stored in the configuration file is grater than the possible number of species according to the alphabet and maximum length all species up to the MAX length will be created

Version

1.0

Parameters

MTRand&	tmpRndDoubleGen initial layer initialization

Version

1.0

Parameters

MTRand&	tmpRndDoubleGen Compute number of reaction catalysd by a catalyst according to the total number of reactions and reactions probabilities

Version

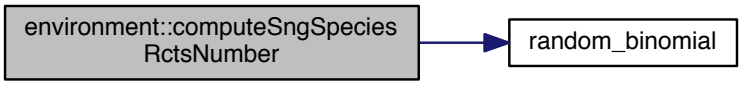
1.0

Parameters

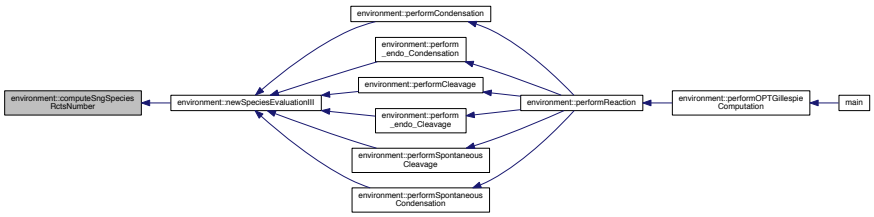
acs_int	tmpTotalNumberOfReactions Total number of conceivable reactions
acs_double	tmpRctsProb reaction probability

Definition at line 481 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.10 acs_double environment::createDiffusionRenforcement (acs_double *tmpDiffEnh*, acs_int *tmpNewSpeciesLength*)

Create the diffusion constant renforcement according to the species length

Version

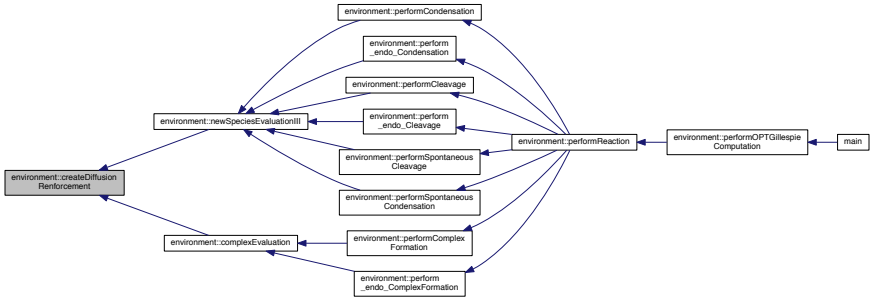
1.0

Parameters

tmpDiffEnh	diffusion enhancement from parameters
MTRand&	tmp_RndDoubleGen random number generator
tmpNewSpecies-Length	Lenght of the species

Definition at line 1819 of file environment.cpp.

Here is the caller graph for this function:



12.3.3.11 bool environment::createInfluxLayersFromFileSTD (string *tmpInfluxFilePath*)

Create influx layer from file C++ libraries

Version

1.0

Parameters

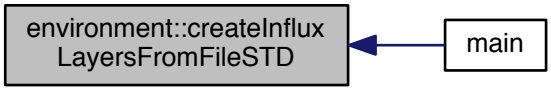
string	tmpInfluxFilePath file path

Date

20130702

Definition at line 1572 of file environment.cpp.

Here is the caller graph for this function:



12.3.3.12 bool environment::createInitialCatalysisLayerFromFileSTD (string *tmpCatalysisFilePath*)

Catalysis from file using standard C++ libraries

Version

1.0

Parameters

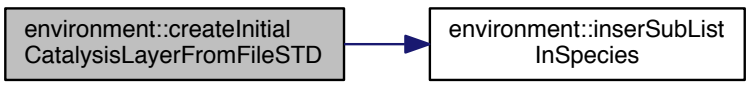
string	tmpSpeciesFilePath file path

Date

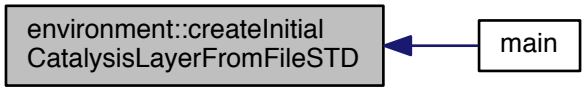
20130702

Definition at line 1724 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.13 bool environment::createInitialCatalysisLayerFromSpecificFileSTD (string *tmpCatalysisFilePath,* acs_int *tmpActGEN,* acs_int *tmpActSIM*)

catalysis from file using standard C++ libraries

Version

1.0

Parameters

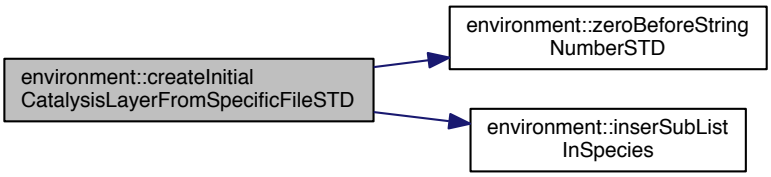
string	tmpSpeciesFilePath file path

Date

20130702

Definition at line 1766 of file environment.cpp.

Here is the call graph for this function:



12.3.3.14 bool environment::createInitialMoleculesPopulationFromFileSTD (string *tmpSpeciesFilePath*)

Initial molecule population creation from file using standard C++ libraries

Version

1.0

Parameters

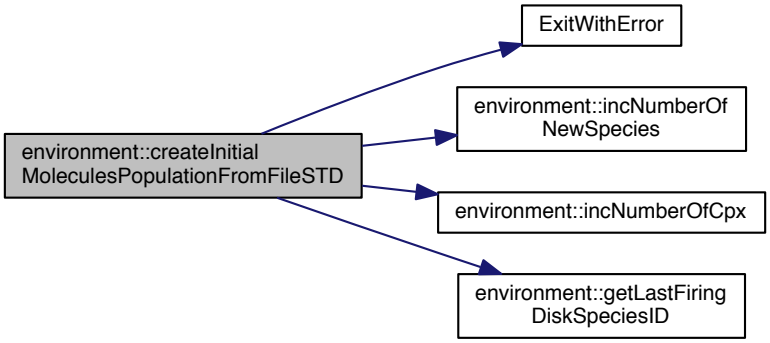
string	tmpSpeciesFilePath file path

Date

20130702

Definition at line 1249 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

12.3.3.15 bool environment::createInitialMoleculesPopulationFromSpecificFileSTD (string *tmpSpeciesFilePath*, acs_int *tmpActGEN*, acs_int *tmpActSIM*)

Initial molecule population creation from file

Version

1.0

Parameters

QString	tmpSpeciesFilePath file path Initial molecule population creation. Species are uploaed from a SPECIFIC file created using actual generation and simuation

Version

1.0

Parameters

QString	tmpSpeciesFilePath file path Initial molecule population creation. Species are uploaed from a SPECIFIC file created using actual generation and simuation (standard C++ libraries)

Version

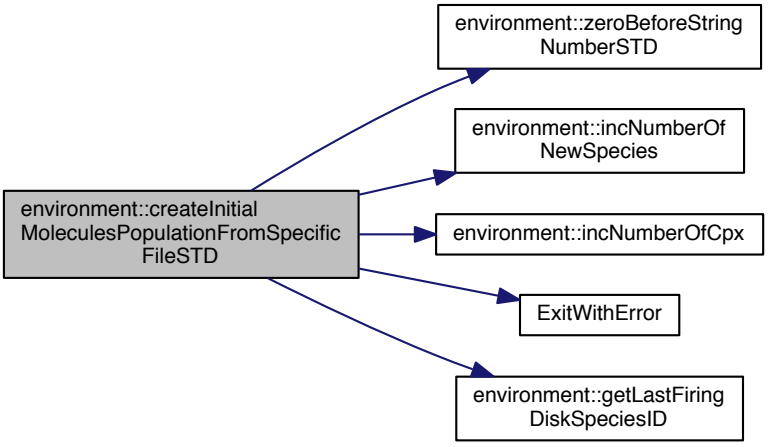
1.0

Parameters

QString	tmpSpeciesFilePath file path

Definition at line 1478 of file environment.cpp.

Here is the call graph for this function:



12.3.3.16 bool environment::createInitialReactionsLayerFromFileSTD (string *tmpSpeciesFilePath*)

Reactions from file using standard C++ libraries

Version

1.0

Parameters

string	tmpSpeciesFilePath file path

Date

20130702

Definition at line 1635 of file environment.cpp.

Here is the caller graph for this function:

12.3.3.17 bool environment::createInitialReactionsLayerFromSpecificFileSTD (string *tmpReactionsFilePath*, acs_int *tmpActGEN*, acs_int *tmpActSIM*)

Reactions from file using standard C++ libraries

Version

1.0

Parameters

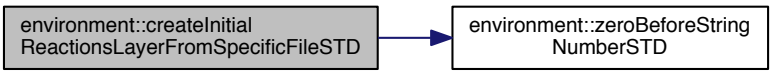
string	tmpSpeciesFilePath file path

Date

20130702

Definition at line 1674 of file environment.cpp.

Here is the call graph for this function:



12.3.3.18 bool environment::createNrgBooleanFunctionsFromFileSTD (string *tmpBoolNrgFilePath*)

load energy boolean function (in decimal format) - Standard C++

Version

1.0

Parameters

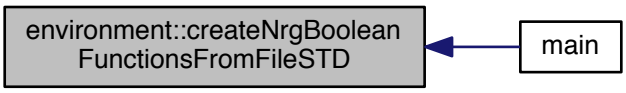
string	tmpBoolNrgFilePath file path

Date

20130702

Definition at line 1604 of file environment.cpp.

Here is the caller graph for this function:



12.3.3.19 bool environment::createReactionsForThisSpecies (acs_longInt *tmpsID*, acs_int *tmpReactionsForThisSpecies*, MTRand & *tmp_RndDoubleGen*, vector< acs_longInt > & *tmpIDOfCandidateSpecies*, acs_int *tmpRctCreationType*)

Creation of all the reactions related to one specific species

Version

1.1

Date

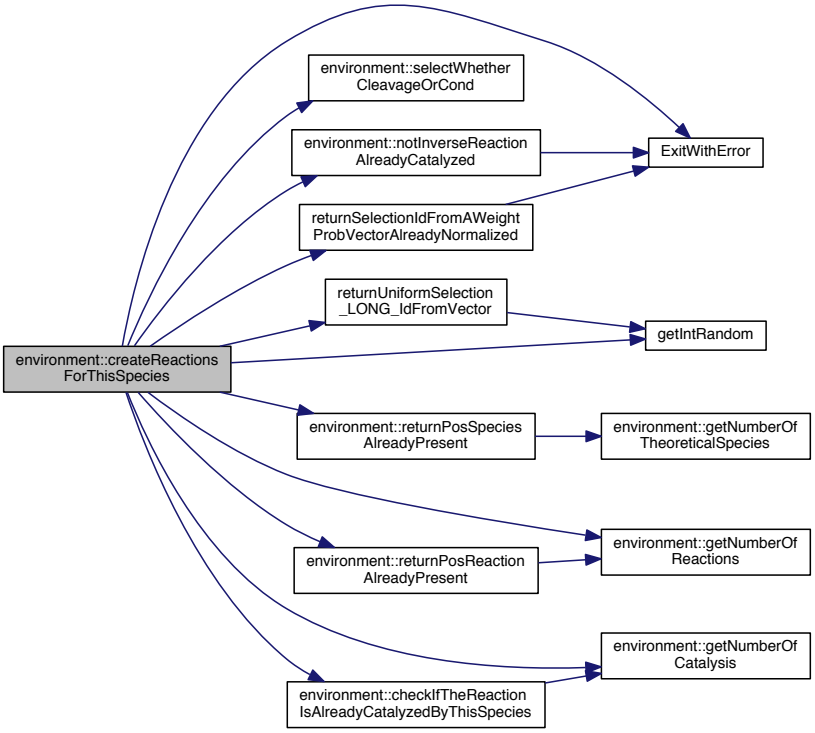
2011/07/07

Parameters

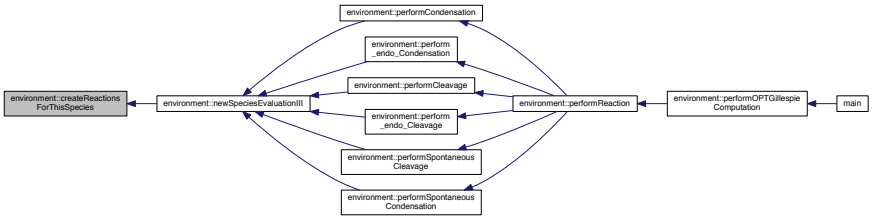
acs_longInt	tmpsID species vector ID
acs_int	tmpReactionsForThisSpecies number of reactions to create for this species
MTRand&	tmp_RndDoubleGen random number generator
vector<acs_longInt>&	tmpIDOfCandidateSpecies ID of the species avalaible for the reaction
acs_int	tmpRctCreationType NEWREACTION or UPGRADEREACTIONS

Definition at line 525 of file environment.cpp.

Here is the call graph for this function:



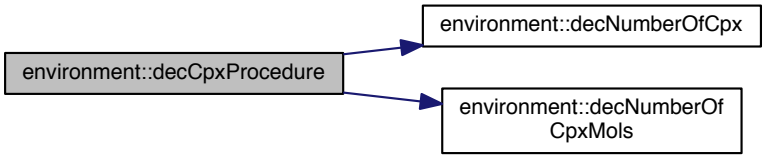
Here is the caller graph for this function:



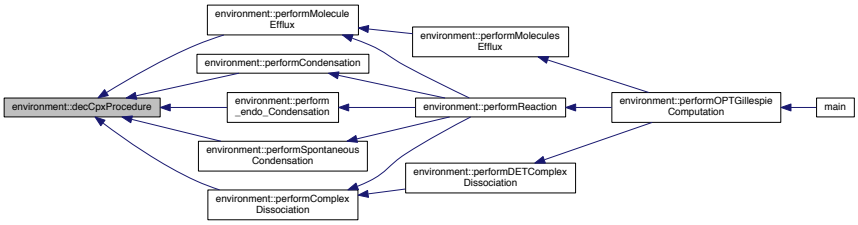
12.3.3.20 void environment::decCpxProcedure (acs_int *tmp_ID*) [inline]

Definition at line 298 of file environment.h.

Here is the call graph for this function:



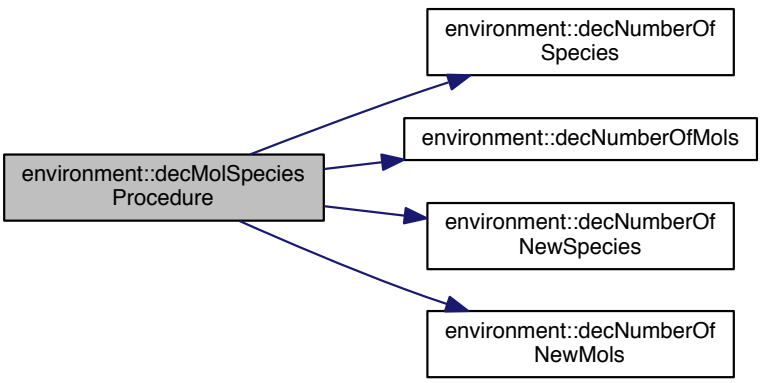
Here is the caller graph for this function:



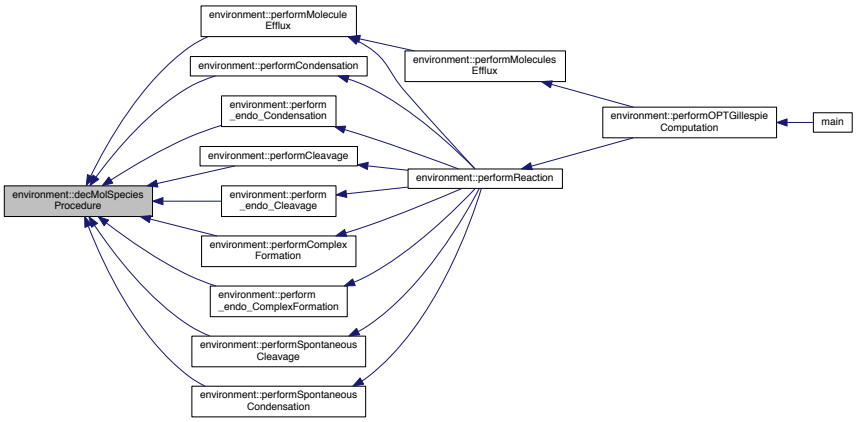
12.3.3.21 void environment::decMolSpeciesProcedure (acs_int *tmp_ID*) [inline]

Definition at line 297 of file environment.h.

Here is the call graph for this function:



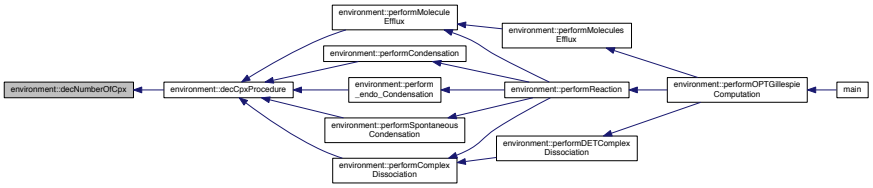
Here is the caller graph for this function:



12.3.3.22 void environment::decNumberOfCpx (acs_int *tmpID*) [inline]

Definition at line 288 of file environment.h.

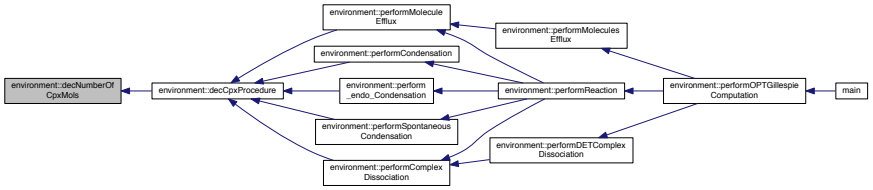
Here is the caller graph for this function:



12.3.3.23 void environment::decNumberOfCpxMols ( ) [inline]

Definition at line 290 of file environment.h.

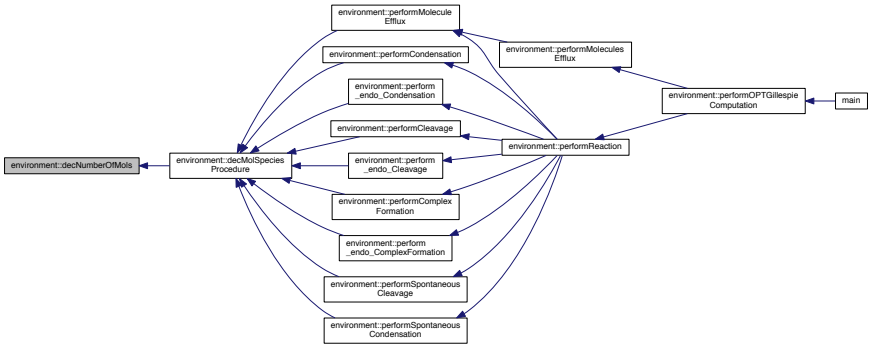
Here is the caller graph for this function:



12.3.3.24 void environment::decNumberOfMols ( ) [inline]

Definition at line 286 of file environment.h.

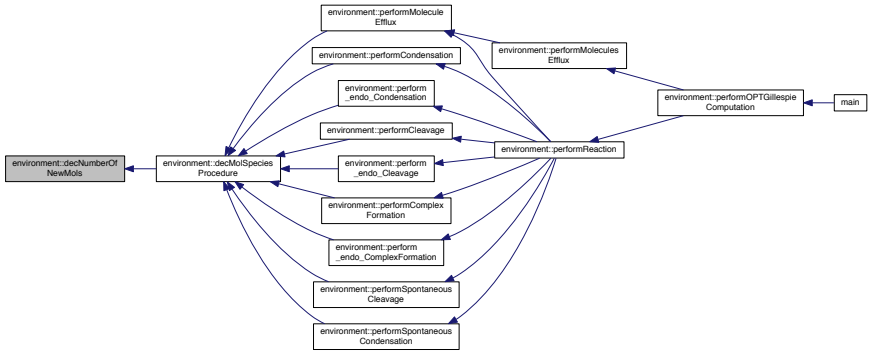
Here is the caller graph for this function:



12.3.3.25 void environment::decNumberOfNewMols (acs_int *tmpID*) [inline]

Definition at line 295 of file environment.h.

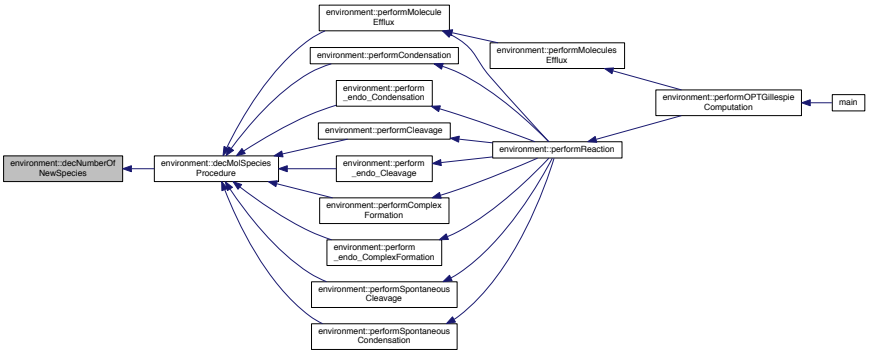
Here is the caller graph for this function:



12.3.3.26 void environment::decNumberOfNewSpecies (acs_int *tmpID*) [inline]

Definition at line 293 of file environment.h.

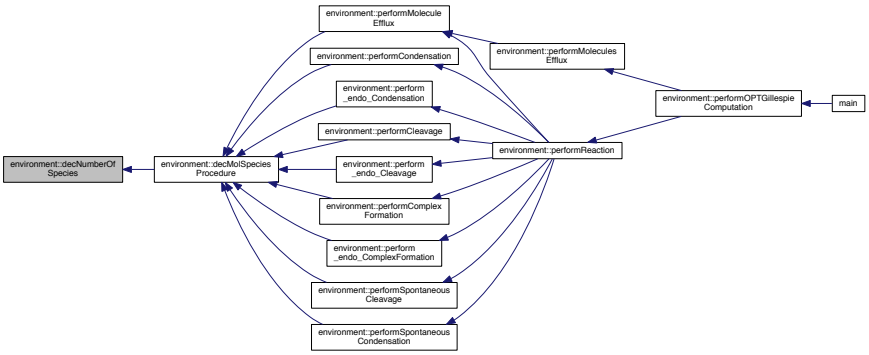
Here is the caller graph for this function:



12.3.3.27 void environment::decNumberOfSpecies (acs_int *tmpID*) [inline]

Definition at line 284 of file environment.h.

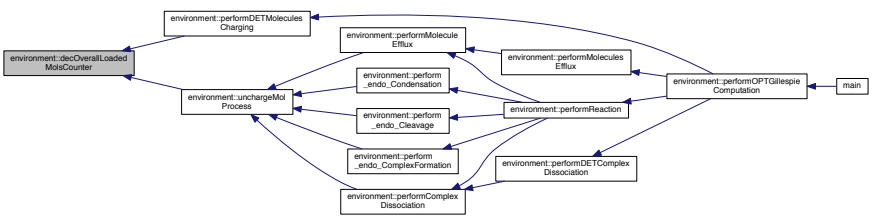
Here is the caller graph for this function:



12.3.3.28 void environment::decOverallLoadedMolsCounter ( ) [inline]

Definition at line 311 of file environment.h.

Here is the caller graph for this function:



12.3.3.29 bool environment::devStd ( )

Definition at line 7370 of file environment.cpp.

Here is the caller graph for this function:

12.3.3.30 bool environment::entropy ( )

Definition at line 7388 of file environment.cpp.

Here is the caller graph for this function:

12.3.3.31 acs_double environment::getActualTime ( ) const [inline]

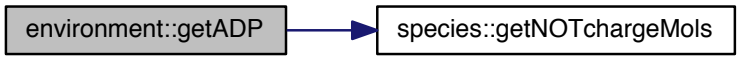
Definition at line 153 of file environment.h.

Here is the caller graph for this function:

12.3.3.32 acs_int environment::getADP ( ) const [inline]

Definition at line 169 of file environment.h.

Here is the call graph for this function:



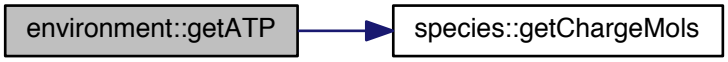
12.3.3.33 string environment::getAlphabet ( ) const [inline]

Definition at line 207 of file environment.h.

12.3.3.34 acs_int environment::getATP ( ) const [inline]

Definition at line 170 of file environment.h.

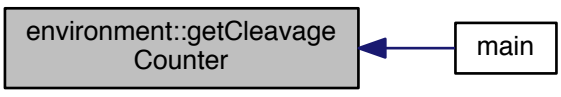
Here is the call graph for this function:



12.3.3.35 acs_longInt environment::getCleavageCounter ( ) const [inline]

Definition at line 226 of file environment.h.

Here is the caller graph for this function:



12.3.3.36 acs_double environment::getCleavageKC ( ) const [inline]

Definition at line 195 of file environment.h.

12.3.3.37 acs_double environment::getCleavProb ( ) const [inline]

Definition at line 165 of file environment.h.

12.3.3.38 acs_double environment::getComplexDegKC ( ) const [inline]

Definition at line 198 of file environment.h.

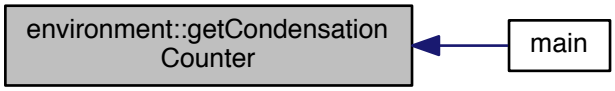
12.3.3.39 acs_double environment::getComplexKC ( ) const [inline]

Definition at line 196 of file environment.h.

12.3.3.40 acs_longInt environment::getCondensationCounter ( ) const [inline]

Definition at line 228 of file environment.h.

Here is the caller graph for this function:



12.3.3.41 acs_double environment::getCondensationKC ( ) const [inline]

Definition at line 197 of file environment.h.

12.3.3.42 acs_longInt environment::getCpxDissCounter ( ) const [inline]

Definition at line 231 of file environment.h.

Here is the caller graph for this function:

12.3.3.43 acs_longInt environment::getCpxFormCounter ( ) const [inline]

Definition at line 230 of file environment.h.

Here is the caller graph for this function:

12.3.3.44 acs_int environment::getCurrentAttempts ( ) const [inline]

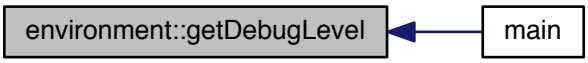
Definition at line 158 of file environment.h.

Here is the caller graph for this function:

12.3.3.45 int environment::getDebugLevel ( ) const [inline]

Definition at line 218 of file environment.h.

Here is the caller graph for this function:



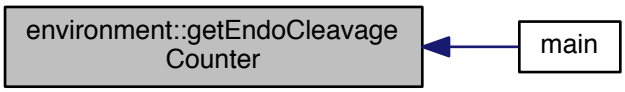
12.3.3.46 acs_double environment::getDiffusionContribute ( ) const [inline]

Definition at line 204 of file environment.h.

12.3.3.47 acs_longInt environment::getEndoCleavageCounter ( ) const [inline]

Definition at line 227 of file environment.h.

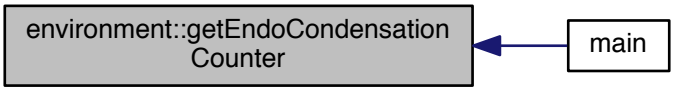
Here is the caller graph for this function:



12.3.3.48 acs_longInt environment::getEndoCondensationCounter ( ) const [inline]

Definition at line 229 of file environment.h.

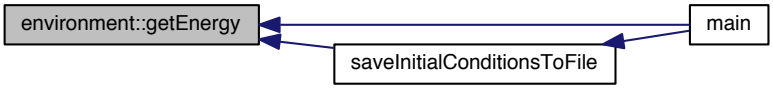
Here is the caller graph for this function:



12.3.3.49 acs_int environment::getEnergy ( ) const [inline]

Definition at line 167 of file environment.h.

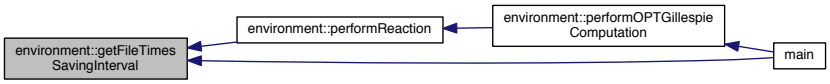
Here is the caller graph for this function:



12.3.3.50 acs_double environment::getFileTimesSavingInterval ( ) const [inline]

Definition at line 160 of file environment.h.

Here is the caller graph for this function:



12.3.3.51 acs_double environment::getgillespieEntropy ( ) const [inline]

Definition at line 180 of file environment.h.

Here is the caller graph for this function:

12.3.3.52 acs_double environment::getGillespieMean ( ) const [inline]

Definition at line 178 of file environment.h.

Here is the caller graph for this function:

12.3.3.53 acs_double environment::getgillespieSD ( ) const [inline]

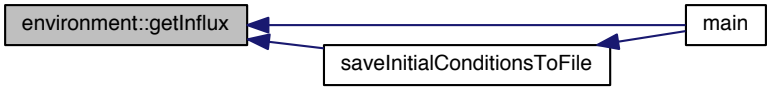
Definition at line 179 of file environment.h.

Here is the caller graph for this function:

12.3.3.54 acs_double environment::getInflux ( ) const [inline]

Definition at line 205 of file environment.h.

Here is the caller graph for this function:



12.3.3.55 acs_double environment::getK_spont_ass ( ) const [inline]

Definition at line 193 of file environment.h.

12.3.3.56 acs_double environment::getK_spont_diss ( ) const [inline]

Definition at line 192 of file environment.h.

12.3.3.57 acs_double environment::getKass ( ) const [inline]

Definition at line 187 of file environment.h.

12.3.3.58 acs_double environment::getKcpx ( ) const [inline]

Definition at line 188 of file environment.h.

12.3.3.59 acs_double environment::getKcpxDiss ( ) const [inline]

Definition at line 189 of file environment.h.

12.3.3.60 acs_double environment::getKdiss ( ) const [inline]

Definition at line 186 of file environment.h.

12.3.3.61 acs_double environment::getKirrad ( ) const [inline]

Definition at line 191 of file environment.h.

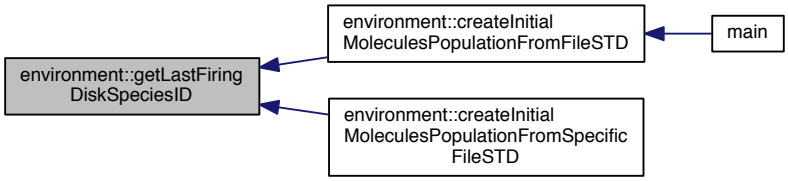
12.3.3.62 acs_double environment::getKnrg ( ) const [inline]

Definition at line 190 of file environment.h.

12.3.3.63 acs_int environment::getLastFiringDiskSpeciesID ( ) const [inline]

Definition at line 161 of file environment.h.

Here is the caller graph for this function:



12.3.3.64 acs_int environment::getMAXattempts ( ) const [inline]

Definition at line 157 of file environment.h.

Here is the caller graph for this function:

12.3.3.65 acs_double environment::getMAXhours ( ) const [inline]

Definition at line 156 of file environment.h.

Here is the caller graph for this function:

12.3.3.66 acs_int environment::getMaxLOut ( ) const [inline]

Definition at line 200 of file environment.h.

12.3.3.67 acs_int environment::getMaxNonCatalyticLength ( ) const [inline]

Definition at line 163 of file environment.h.

12.3.3.68 acs_double environment::getMoleculeDecayKC ( ) const [inline]

Definition at line 199 of file environment.h.

12.3.3.69 vector<species> environment::getMoleculesPopulation ( ) const [inline]

Definition at line 210 of file environment.h.

12.3.3.70 acs_longInt environment::getMols ( ) const [inline]

Definition at line 171 of file environment.h.

Here is the caller graph for this function:

12.3.3.71 acs_longInt environment::getNcpx ( ) const [inline]

Definition at line 175 of file environment.h.

Here is the caller graph for this function:

12.3.3.72 acs_longInt environment::getNcpxMols ( ) const [inline]

Definition at line 176 of file environment.h.

Here is the caller graph for this function:

12.3.3.73 acs_longInt environment::getNewMols ( ) const [inline]

Definition at line 172 of file environment.h.

Here is the caller graph for this function:

12.3.3.74 acs_int environment::getNgen ( ) const [inline]

Definition at line 151 of file environment.h.

Here is the caller graph for this function:

12.3.3.75 acs_longInt environment::getNnewSpecies ( ) const [inline]

Definition at line 174 of file environment.h.

Here is the caller graph for this function:

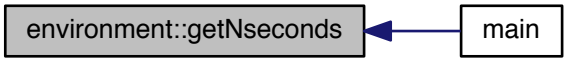
12.3.3.76 acs_int environment::getNreactions ( ) const [inline]

Definition at line 155 of file environment.h.

12.3.3.77 acs_double environment::getNseconds ( ) const [inline]

Definition at line 154 of file environment.h.

Here is the caller graph for this function:



12.3.3.78 acs_int environment::getNsim ( ) const [inline]

Definition at line 152 of file environment.h.

Here is the caller graph for this function:

12.3.3.79 acs_longInt environment::getNspecies ( ) const [inline]

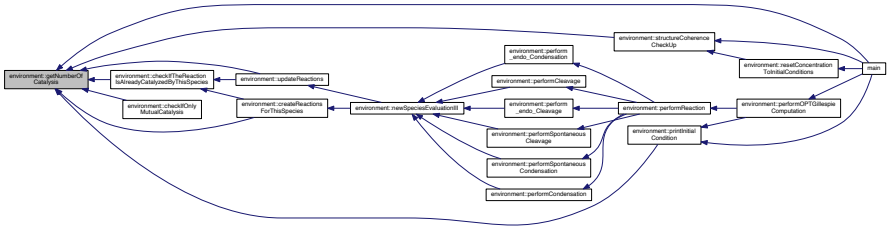
Definition at line 173 of file environment.h.

Here is the caller graph for this function:

12.3.3.80 acs_longInt environment::getNumberOfCatalysis ( ) const [inline]

Definition at line 221 of file environment.h.

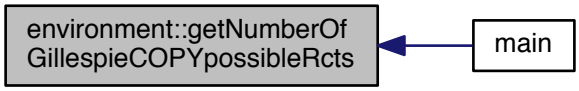
Here is the caller graph for this function:



12.3.3.81 acs_longInt environment::getNumberOfGillespieCOPYpossibleRcts ( ) const [inline]

Definition at line 222 of file environment.h.

Here is the caller graph for this function:



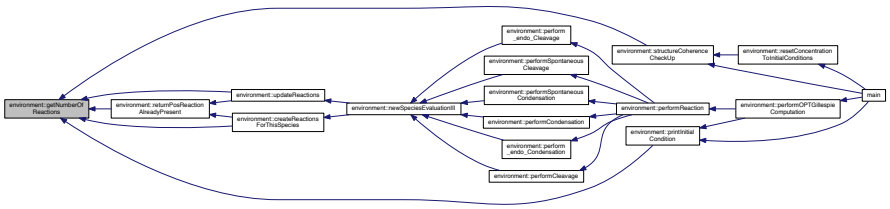
12.3.3.82 acs_longInt environment::getNumberOfGillespiePossibleRcts ( ) const [inline]

Definition at line 223 of file environment.h.

12.3.3.83 acs_longInt environment::getNumberOfReactions ( ) const [inline]

Definition at line 220 of file environment.h.

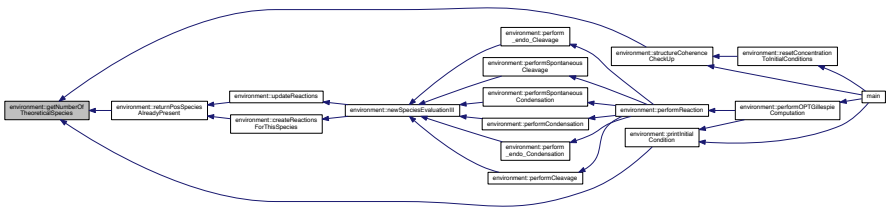
Here is the caller graph for this function:



12.3.3.84 acs_longInt environment::getNumberOfTheoreticalSpecies ( ) const [inline]

Definition at line 219 of file environment.h.

Here is the caller graph for this function:



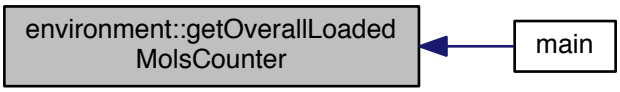
12.3.3.85 acs_double environment::getOverallConcentration ( ) const [inline]

Definition at line 162 of file environment.h.

12.3.3.86 acs_longInt environment::getOverallLoadedMolsCounter ( ) const [inline]

Definition at line 232 of file environment.h.

Here is the caller graph for this function:



12.3.3.87 acs_double environment::getRandomSeed ( ) const [inline]

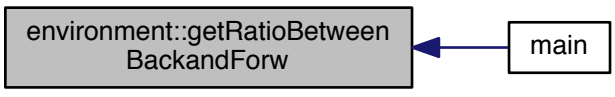
Definition at line 209 of file environment.h.

Here is the caller graph for this function:

12.3.3.88 acs_double environment::getRatioBetweenBackandForw ( ) const [inline]

Definition at line 182 of file environment.h.

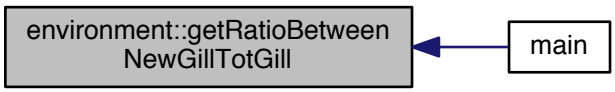
Here is the caller graph for this function:



12.3.3.89 acs_double environment::getRatioBetweenNewGillTotGill ( ) const [inline]

Definition at line 181 of file environment.h.

Here is the caller graph for this function:



12.3.3.90 acs_double environment::getRatioSpeciesEnergizable ( ) const [inline]

Definition at line 168 of file environment.h.

12.3.3.91 acs_double environment::getRctProb ( ) const [inline]

Definition at line 164 of file environment.h.

12.3.3.92 vector<reactions> environment::getReactionsLayer ( ) const [inline]

Definition at line 217 of file environment.h.

12.3.3.93 acs_double environment::getRefillInterval ( ) const [inline]

Definition at line 206 of file environment.h.

12.3.3.94 bool environment::getReverseReactions ( ) const [inline]

Definition at line 166 of file environment.h.

12.3.3.95 acs_int environment::getSolubilityThreshold ( ) const [inline]

Definition at line 201 of file environment.h.

12.3.3.96 acs_longInt environment::getSpontAssCounter ( ) const [inline]

Definition at line 234 of file environment.h.

Here is the caller graph for this function:

12.3.3.97 acs_longInt environment::getSpontDissCounter ( ) const [inline]

Definition at line 233 of file environment.h.

Here is the caller graph for this function:

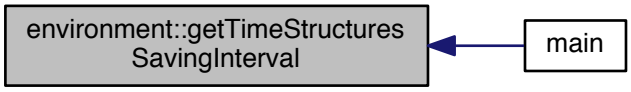
12.3.3.98 bool environment::getSystemExpFlag ( ) const [inline]

Definition at line 183 of file environment.h.

12.3.3.99 acs_double environment::getTimeStructuresSavingInterval ( ) const [inline]

Definition at line 159 of file environment.h.

Here is the caller graph for this function:



12.3.3.100 acs_longInt environment::getTotalNumberOfComplexes ( )

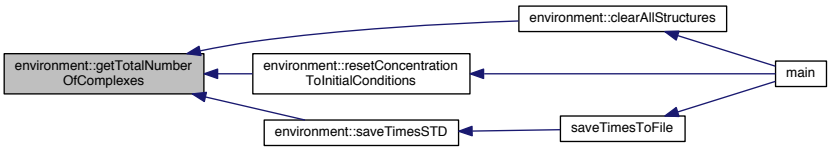
This Function returns the total amount of MOLECULES in the tmpSpeciesvector The file is saved in the directory indicated as a second parameter in the run command

Parameters

*vector<species>**	tmpSpeciesvector pointer to tmpSpeciesvector

Definition at line 3578 of file environment.cpp.

Here is the caller graph for this function:



12.3.3.101 acs_longInt environment::getTotalNumberOfComplexSpecies ( )

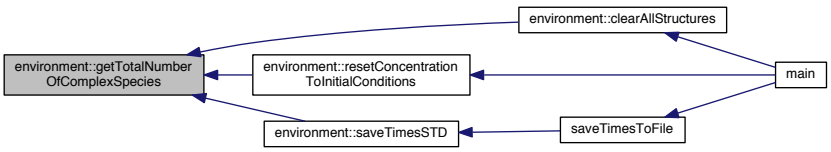
This Function returns the total amount of COMPLEX SPECIES in the tmpSpeciesvector The file is saved in the directory indicated as a second parameter in the run command

Parameters

*vector<species>**	tmpSpeciesvector pointer to tmpSpeciesvector

Definition at line 3557 of file environment.cpp.

Here is the caller graph for this function:



12.3.3.102 acs_longInt environment::getTotalNumberOfMolecules ( )

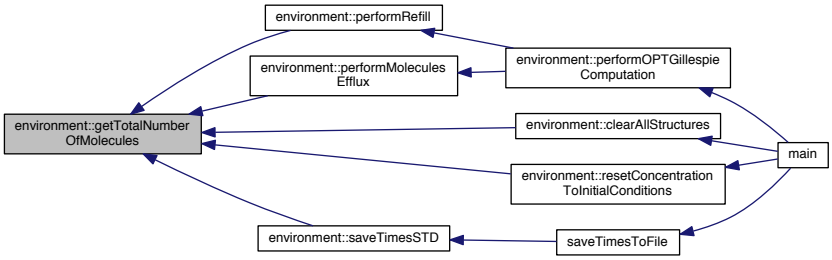
This Function returns the total amount of MOLECULES in the tmpSpeciesvector The file is saved in the directory indicated as a second parameter in the run command

Parameters

*vector<species>**	tmpSpeciesvector pointer to tmpSpeciesvector

Definition at line 3533 of file environment.cpp.

Here is the caller graph for this function:



12.3.3.103 acs_longInt environment::getTotalNumberOfMonomers ( )

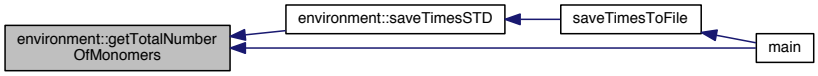
This Function returns the total amount of MONOMERS in the tmpSpeciesvector The file is saved in the directory indicated as a second parameter in the run command

Parameters

*vector<species>**	tmpSpeciesvector pointer to tmpSpeciesvector

Definition at line 3599 of file environment.cpp.

Here is the caller graph for this function:



12.3.3.104 acs_longInt environment::getTotalNumberOfPossibleCatalysts ( )

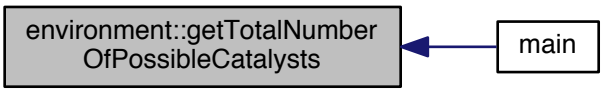
This Function returns the total amount of MOLECULES in the tmpSpeciesvector The file is saved in the directory indicated as a second parameter in the run command

Parameters

*vector<species>**	tmpSpeciesvector pointer to tmpSpeciesvector

Definition at line 3510 of file environment.cpp.

Here is the caller graph for this function:



12.3.3.105 acs_longInt environment::getTotalNumberOfSpecies ( )

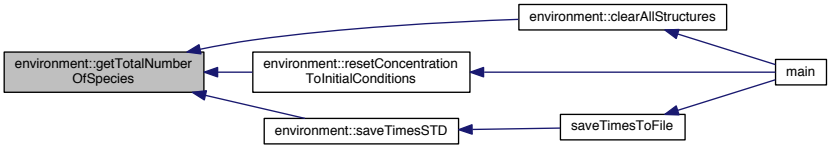
This Function returns the total amount of MOLECULES in the tmpSpeciesvector The file is saved in the directory indicated as a second parameter in the run command

Parameters

*vector<species>**	tmpSpeciesvector pointer to tmpSpeciesvector

Definition at line 3486 of file environment.cpp.

Here is the caller graph for this function:

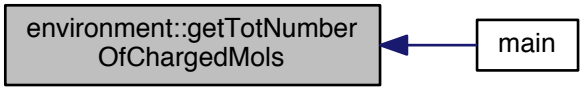


12.3.3.106 acs_int environment::getTotNumberOfChargedMols ( )

Get the total number of charged molecules

Definition at line 3619 of file environment.cpp.

Here is the caller graph for this function:



12.3.3.107 acs_double environment::getVolume ( ) const [inline]

Definition at line 208 of file environment.h.

Here is the caller graph for this function:

12.3.3.108 void environment::incCleavageCounter ( ) [inline]

Definition at line 304 of file environment.h.

Here is the caller graph for this function:

12.3.3.109 void environment::incCondensationCounter ( ) [inline]

Definition at line 306 of file environment.h.

Here is the caller graph for this function:

12.3.3.110 void environment::incCpxDissCounter ( ) [inline]

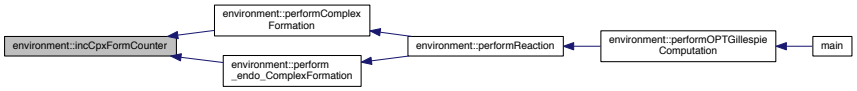
Definition at line 309 of file environment.h.

Here is the caller graph for this function:

12.3.3.111 void environment::incCpxFormCounter ( ) [inline]

Definition at line 308 of file environment.h.

Here is the caller graph for this function:



12.3.3.112 void environment::incEndoCleavageCounter ( ) [inline]

Definition at line 305 of file environment.h.

Here is the caller graph for this function:

12.3.3.113 void environment::incEndoCondensationCounter ( ) [inline]

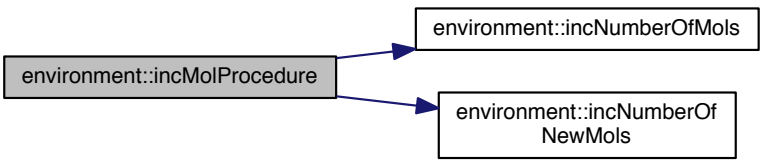
Definition at line 307 of file environment.h.

Here is the caller graph for this function:

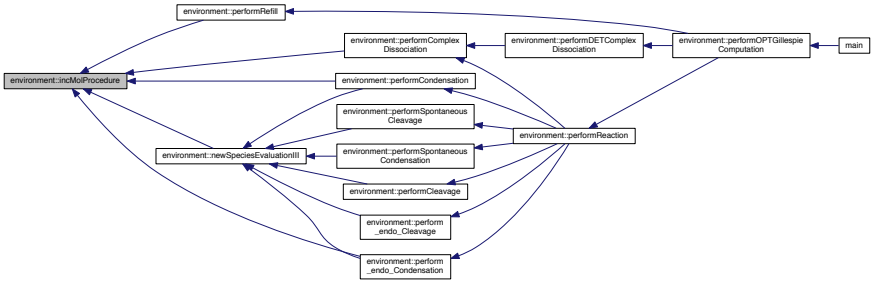
12.3.3.114 void environment::incMolProcedure (acs_int *tmp_ID*) [inline]

Definition at line 299 of file environment.h.

Here is the call graph for this function:



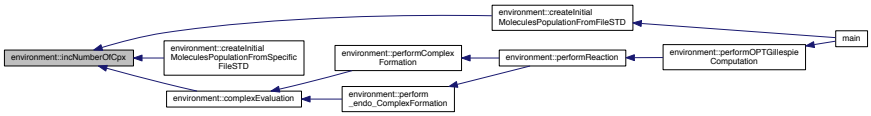
Here is the caller graph for this function:



12.3.3.115 void environment::incNumberOfCpx ( ) [inline]

Definition at line 287 of file environment.h.

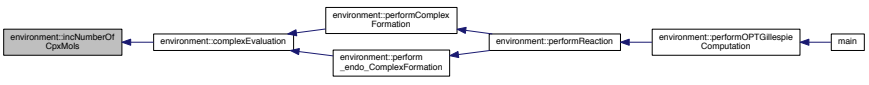
Here is the caller graph for this function:



12.3.3.116 void environment::incNumberOfCpxMols ( ) [inline]

Definition at line 289 of file environment.h.

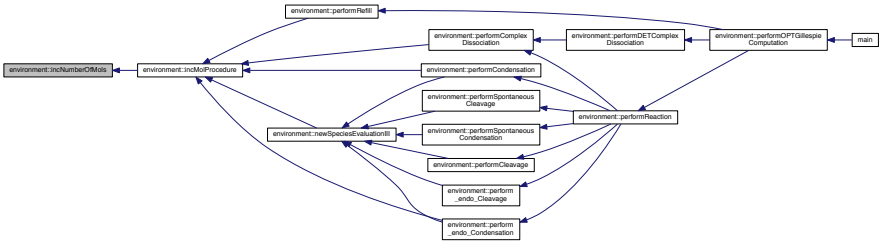
Here is the caller graph for this function:



12.3.3.117 void environment::incNumberOfMols ( ) [inline]

Definition at line 285 of file environment.h.

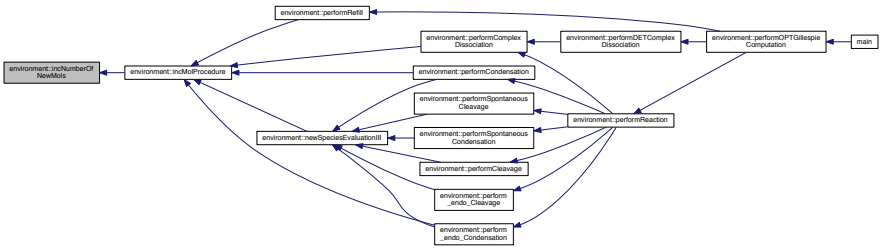
Here is the caller graph for this function:



12.3.3.118 void environment::incNumberOfNewMols (acs_int *tmpID*) [inline]

Definition at line 294 of file environment.h.

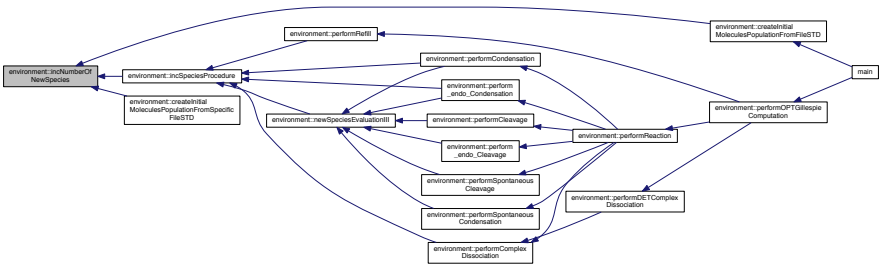
Here is the caller graph for this function:



12.3.3.119 void environment::incNumberOfNewSpecies (acs_int *tmpID*) [inline]

Definition at line 292 of file environment.h.

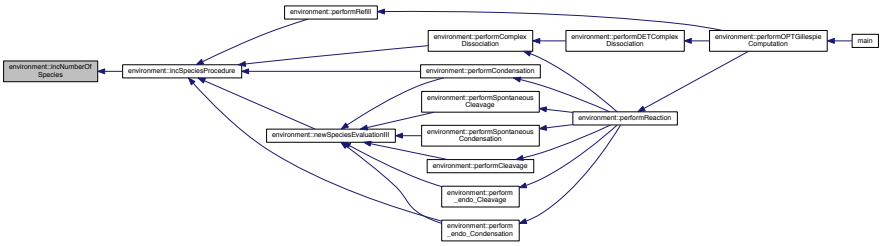
Here is the caller graph for this function:



12.3.3.120 void environment::incNumberOfSpecies ( ) [inline]

Definition at line 283 of file environment.h.

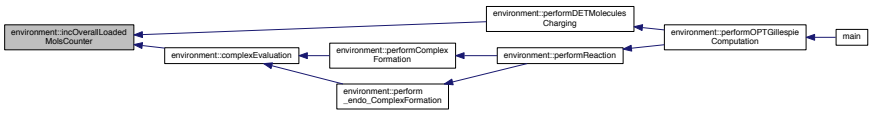
Here is the caller graph for this function:



12.3.3.121 void environment::incOverallLoadedMolsCounter ( ) [inline]

Definition at line 310 of file environment.h.

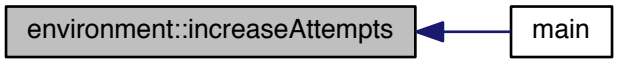
Here is the caller graph for this function:



12.3.3.122 void environment::increaseAttempts ( ) [inline]

Definition at line 346 of file environment.h.

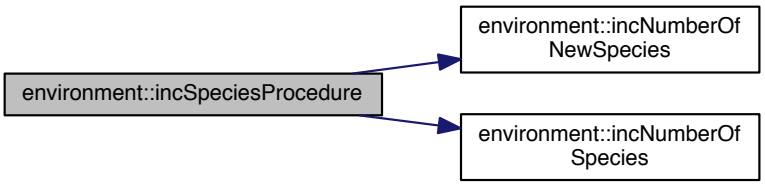
Here is the caller graph for this function:



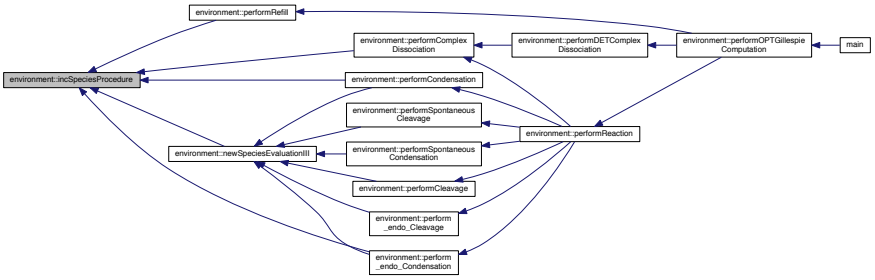
12.3.3.123 void environment::incSpeciesProcedure (acs_int *tmp_ID*) [inline]

Definition at line 300 of file environment.h.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.124 void environment::incSpontAssCounter ( ) [inline]

Definition at line 313 of file environment.h.

Here is the caller graph for this function:

12.3.3.125 void environment::incSpontDissCounter ( ) [inline]

Definition at line 312 of file environment.h.

Here is the caller graph for this function:

12.3.3.126 void environment::inserSubListInSpecies ( )

Insert substrate reactions information in species if complex

Version

1.0

Date

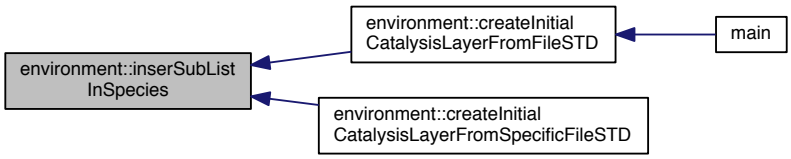
20130917

Author

Alessandro Filisetti

Definition at line 2462 of file environment.cpp.

Here is the caller graph for this function:



12.3.3.127 bool environment::newSpeciesEvaluationIII (acs_longInt *tmpNewSpecies,* MTRand & *tmp___RndDoubleGen*)

Evaluate new species (Optimized function with new species reactions update process)

Version

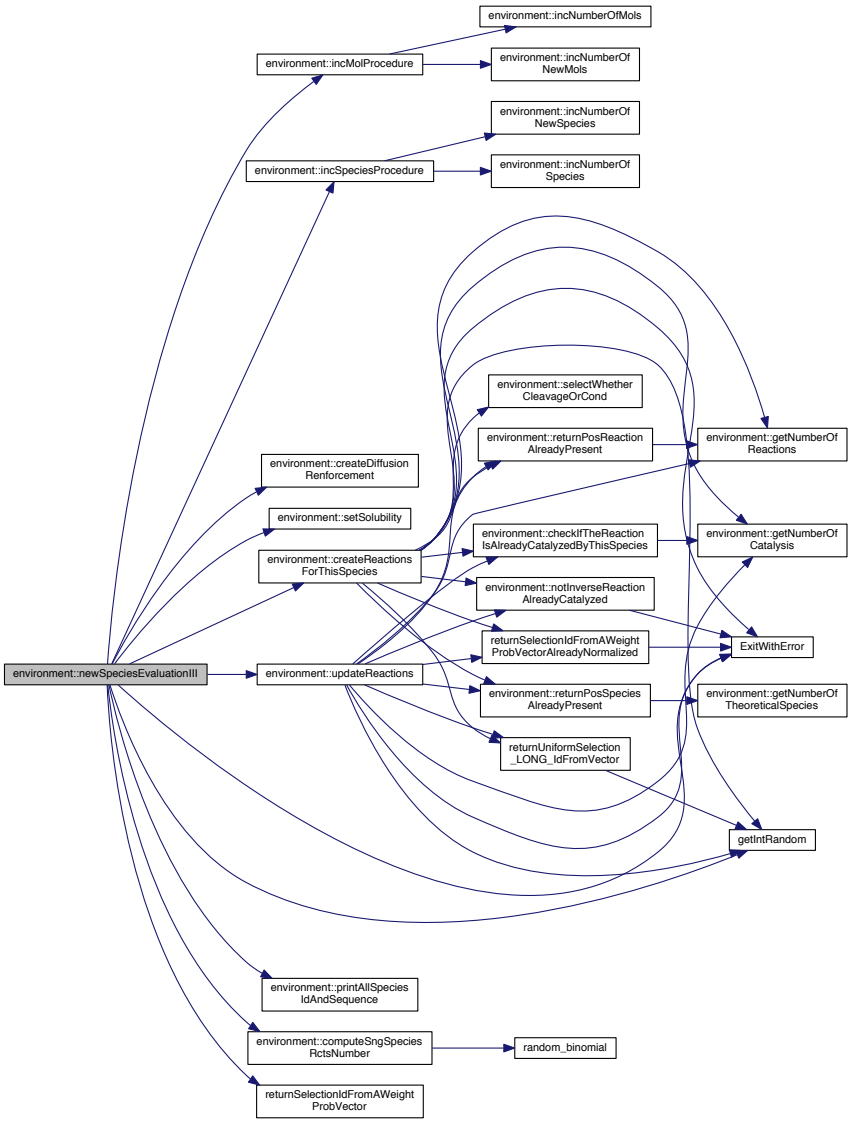
3.0

Parameters

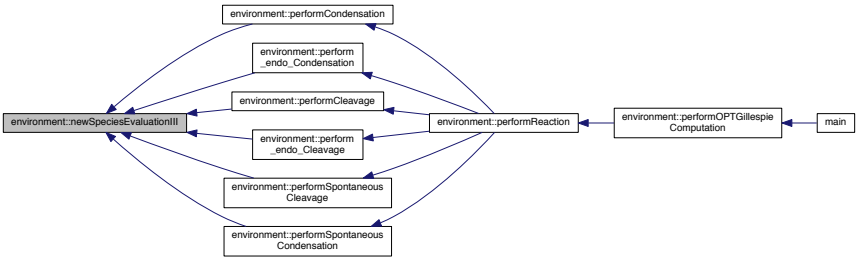
acs_int	tmpNewSpecies New species ID to evaluate
MTRand&	tmp___RndDoubleGen random number generator

Definition at line 5851 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.128 bool environment::notInverseReactionAlreadyCatalyzed (acs_int *tmpRct,* acs_longInt *tmpID_I,* acs_longInt *tmpID_II*)

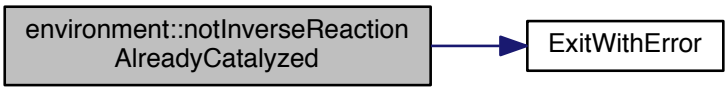
Check if the reaction catalyze both one reaction and the inverted one

Version

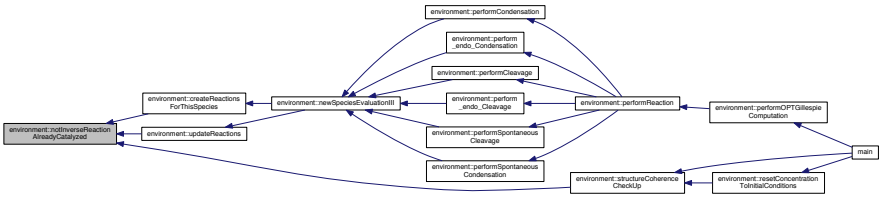
1.0

Definition at line 2578 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.129 void environment::nutrientsAmountsFixing ( )

Nutrients amount fixing process. The amount of nutrients has to be fixed according to the initial theoretical distribution

Version

1.0

Definition at line 3685 of file environment.cpp.

12.3.3.130 bool environment::perform_endo_Cleavage (acs_longInt *tmpSubstrate*, acs_longInt *tmpProduct_I*, acs_longInt *tmpProduct_II*, acs_int *tmpNRGside*, acs_longInt *tmpIdReaction*, acs_longInt *tmpIdCatalysis*, MTRand & *tmp__RndDoubleGen*)

Perform ENDO_CLEAVAGE reaction

Version

1.2

Date

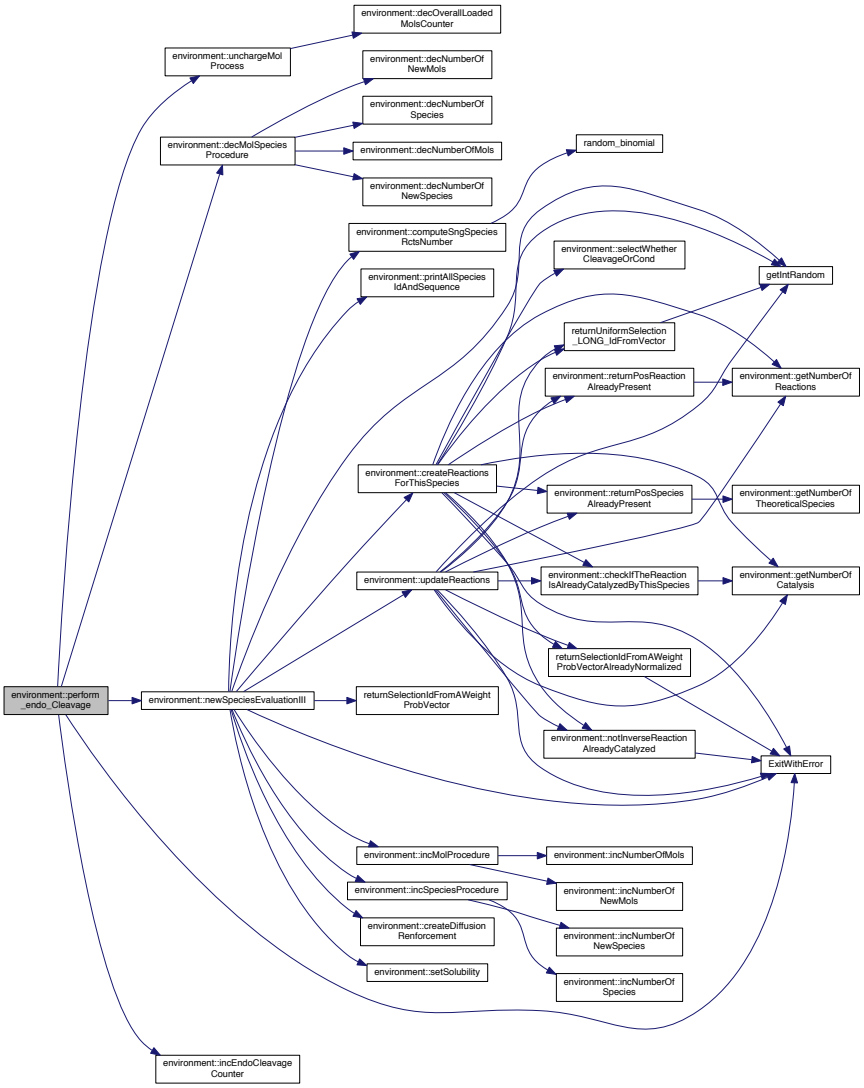
2010.11.08

Parameters

acs_longInt	tmpSubstrate Substrate ID
acs_longInt	tmpProduct_I Product 1 ID
acs_longInt	tmpProduct_II Product 2 ID
acs_int	tmpNrgTarget Energy Target –> catalyst loaded, substrate loaded or both
tmpIdReaction	Rections ID
tmpIdCatalysis	Catalysis ID
MTRand&	tmp__RndDoubleGen random number generator

Definition at line 5258 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

12.3.3.131 bool environment::perform_endo_ComplexFormation (acs_longInt *tmpCatalyst*, acs_longInt *tmpSubstrate*, acs_longInt *tmpCatID*, acs_longInt *tmpSecSub*, acs_int *tmpNRGSide*, MTRand & *tmp__RndDoubleGen*)

Perform ENDO COMPLEX FORMATION reaction

Version

1.2

Date

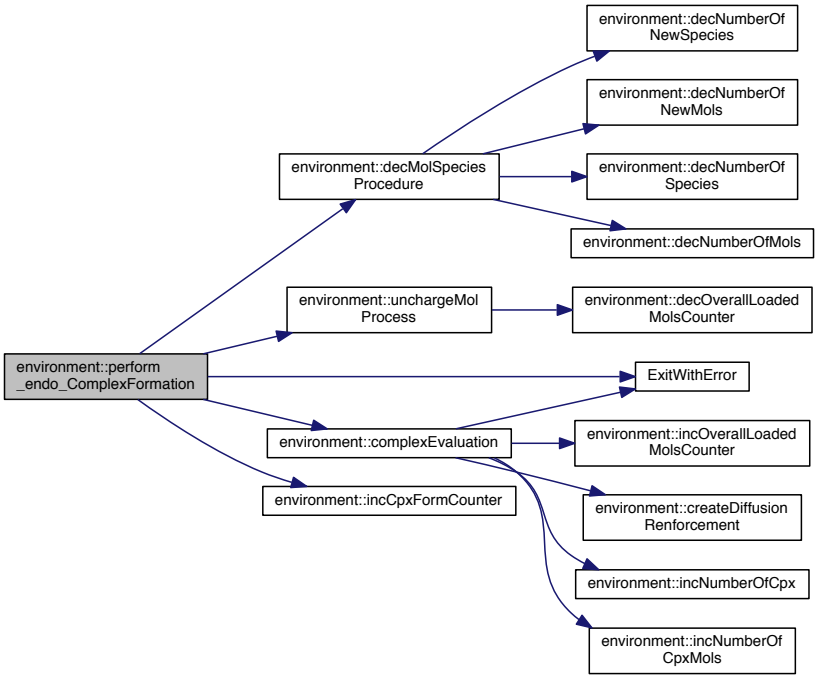
2011.04.13

Parameters

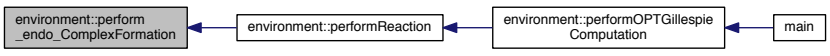
acs_longInt	tmpCatalyst Catalyst ID
acs_longInt	tmpSubstrate Substrate ID Catalysis ID
MTRand&	tmp__RndDoubleGen random generator

Definition at line 5467 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.132 bool environment::perform_endo_Condensation (acs_longInt *tmpCatalyst*, acs_longInt *tmpSubstrate*, acs_longInt *tmpProduct*, acs_longInt *tmpComplex*, acs_int *tmpNRGside*, acs_longInt *tmpIdReaction*, acs_longInt *tmpIdCatalysis*, MTRand & *tmp__RndDoubleGen*)

Perform ENDO CONDENSATION reaction

Version

1.2

Date

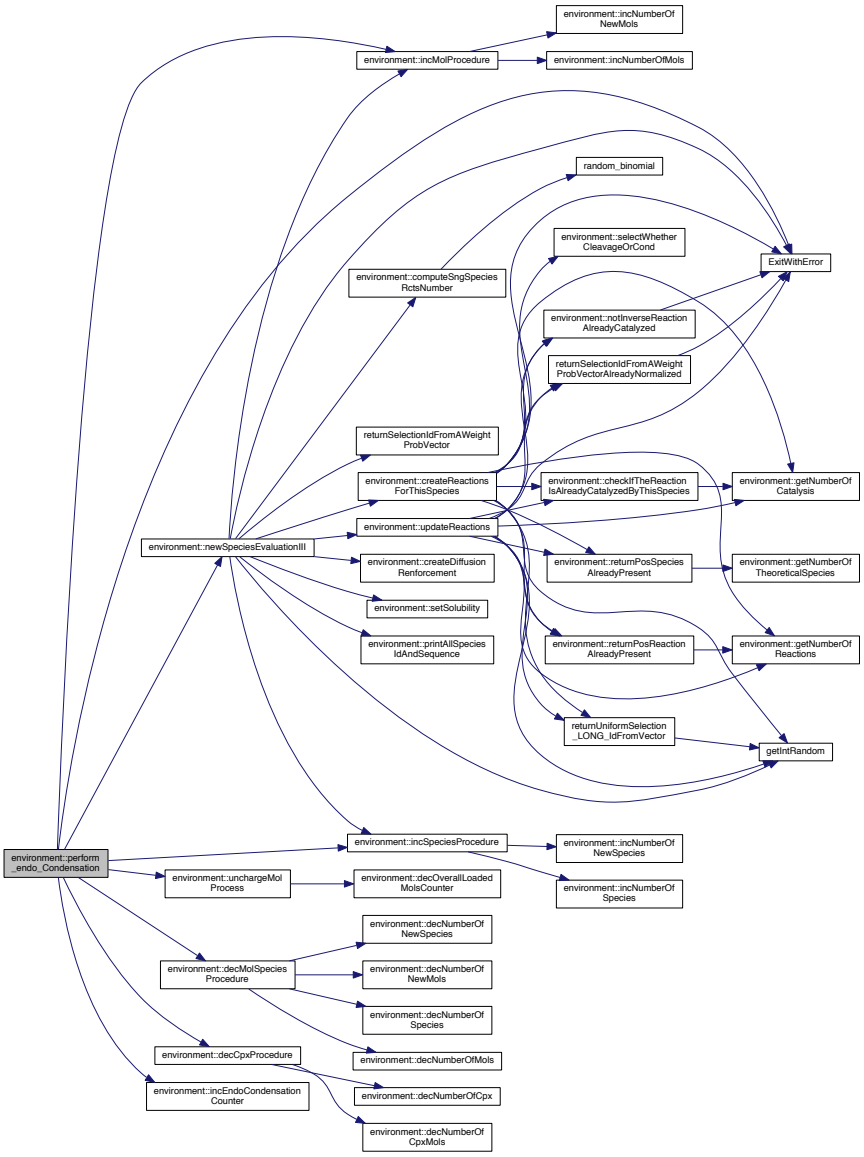
2011-02-12

Parameters

tmpCatalyst	Catalyst (bound in the complex) ID
tmpSubstrate	second (or first, depend on the condensation type) substrate ID
tmpProduct	product ID
acs_longInt	tmpComplex Complex ID
acs_longInt	tmpSubstrate Substrate ID
acs_longInt	tmpProduct Product ID
MTRand&	tmp__RndDoubleGen random generator

Definition at line 5015 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

12.3.3.133 bool environment::performCleavage (acs_longInt *tmpSubstrate,* acs_longInt *tmpProduct_I,* acs_longInt *tmpProduct_II,* acs_longInt *tmpIdReaction,* acs_longInt *tmpIdCatalysis,* MTRand & *tmp__RndDoubleGen*)

Perform CLEAVAGE reaction

Version

1.2

Date

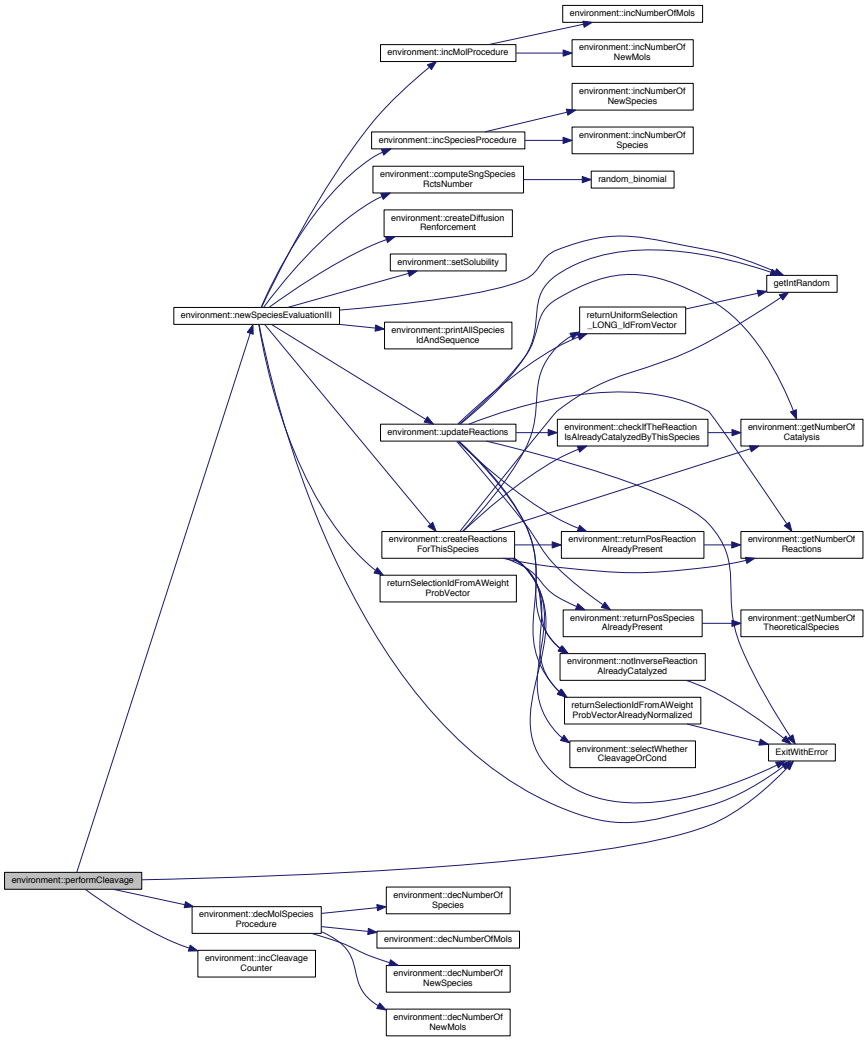
2011.02.12

Parameters

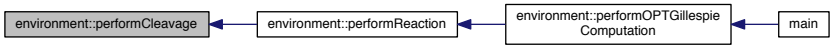
acs_longInt	tmpSubstrate Substrate ID
acs_longInt	tmpProduct_I Product 1 ID
acs_longInt	tmpProduct_II Product 2 ID
tmpIdReaction	Rections ID
tmpIdCatalysis	Catalysis ID
MTRand&	tmp__RndDoubleGen random number generator

Definition at line 5151 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.134 bool environment::performComplexDissociation (acs_longInt *tmpComplex*, acs_longInt *tmpCatalyst*, acs_longInt *tmpSubstrate*, MTRand & *tmp__RndDoubleGen*)

Perform COMPLEX DISASSOCIATION reaction

Version

1.1

Date

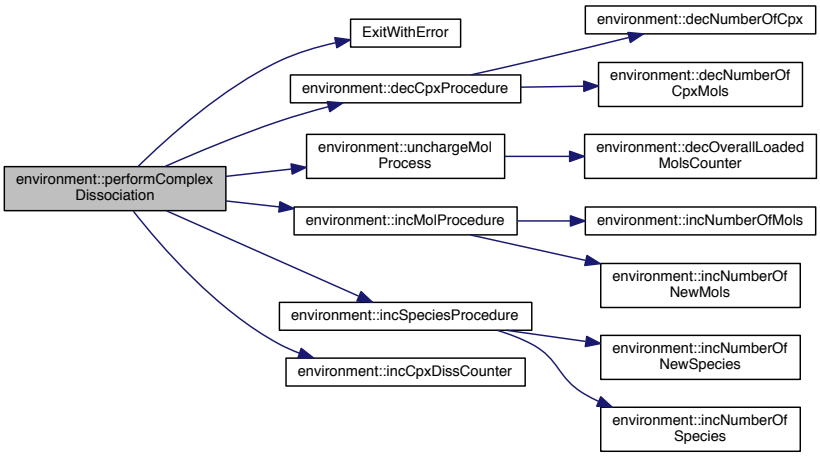
2010.06.08

Parameters

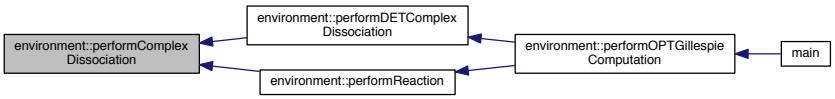
acs_longInt	tmpComplex Complex ID
acs_longInt	tmpCatalyst Catalyst ID
acs_longInt	tmpSubstrate Substrate ID
MTRand&	tmp__RndDoubleGen random generator

Definition at line 5565 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.135 bool environment::performComplexFormation (acs_longInt *tmpCatalyst*, acs_longInt *tmpSubstrate*, acs_longInt *tmpCatID*, acs_longInt *tmpSecSub*, MTRand & *tmp__RndDoubleGen*)

Perform COMPLEX FORMATION reaction

Version

1.2

Date

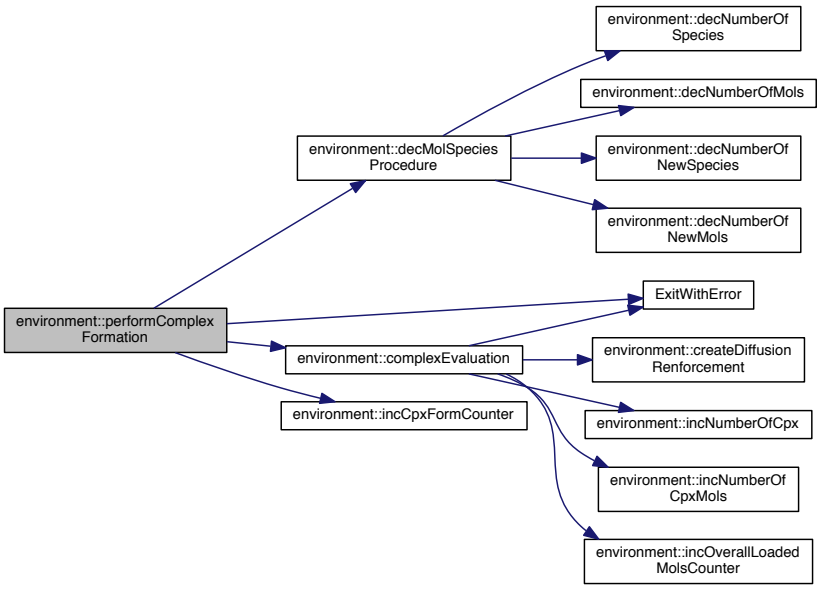
2011.02.13

Parameters

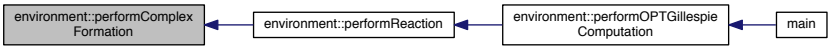
acs_longInt	tmpCatalyst Catalyst ID
acs_longInt	tmpSubstrate Substrate ID Catalysis ID
MTRand&	tmp__RndDoubleGen random generator

Definition at line 5379 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.136 bool environment::performCondensation (acs_longInt *tmpCatalyst,* acs_longInt *tmpSubstrate,* acs_longInt *tmpProduct,* acs_longInt *tmpComplex,* acs_longInt *tmpIdReaction,* acs_longInt *tmpIdCatalysis,* MTRand & *tmp__RndDoubleGen*)

Perform CONDENSATION reaction

Version

1.2

Date

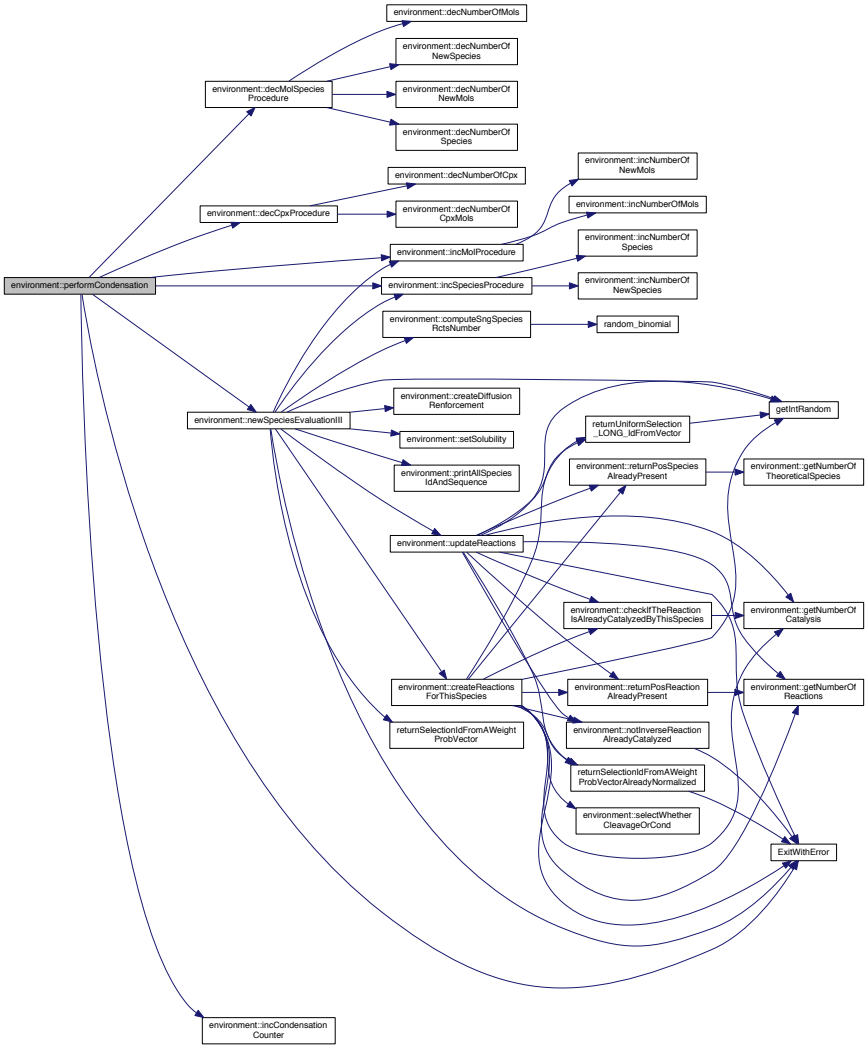
2011-02-12

Parameters

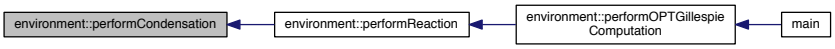
acs_longInt	tmpComplex Complex ID
acs_longInt	tmpSubstrate Substrate ID
acs_longInt	tmpProduct Product ID
MTRand&	tmp__RndDoubleGen random generator

Definition at line 4898 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.137 bool environment::performDETComplexDissociation (acs_double *tmpTimeInterval,* MTRand & *tmp_RndDoubleGen*)

This function perform the deterministic dissociation of the complex

Version

1.0

Date

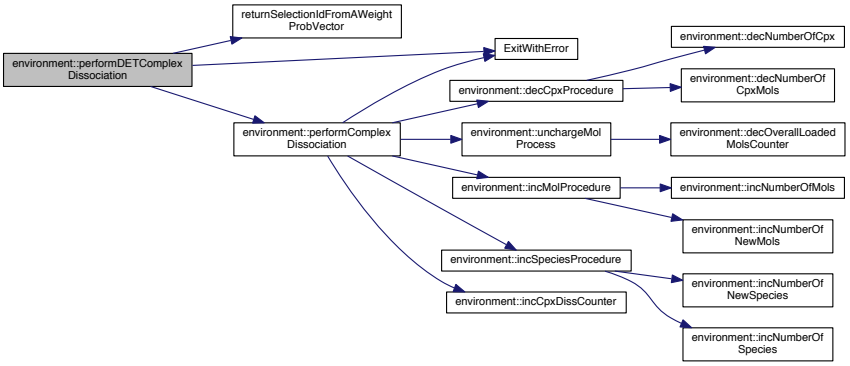
2013-09-18

Parameters

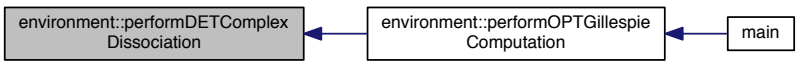
tmpTimeInterval	time elapsed since the last reaction
MTrand	&tmp_RndDoubleGen random number generator

Definition at line 3809 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.138 bool environment::performDETMoleculesCharging (acs_double *tmpTimeInterval,* MTRand & *tmp_RndDoubleGen*)

This function perform the pseudo-deterministic molecules charging process

Version

2.6

Date

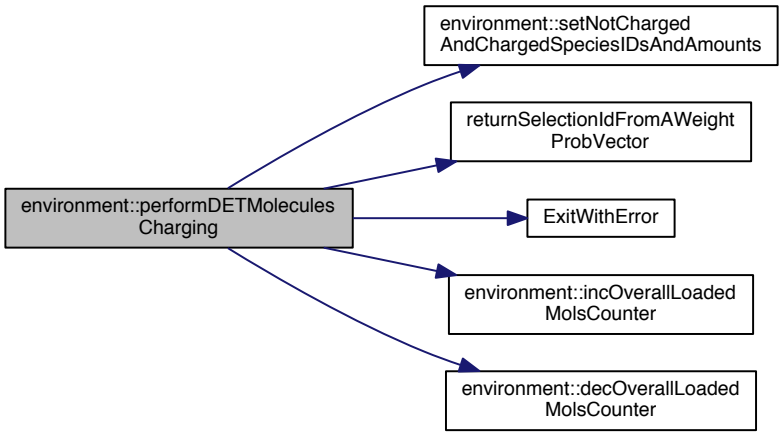
2011-02-24

Parameters

tmpTimeInterval	time elapsed since the last reaction
MTrand	&tmp_RndDoubleGen random number generator

Definition at line 3937 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

12.3.3.139 bool environment::performEnergyEfflux (MTRand & *tmp_RndDoubleGen*)

Perform ENERGY EFFLUX reaction

Version

2.4.1

Date

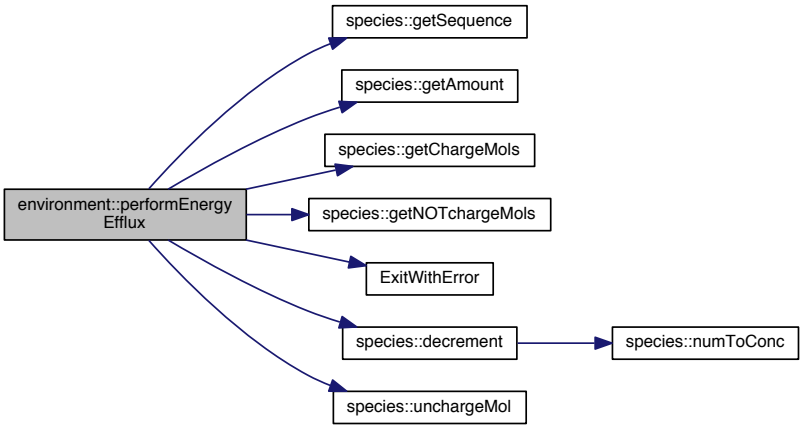
2010-06-27

Parameters

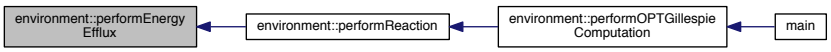
MTRand&	tmp__RndDoubleGen random number generator

Definition at line 4611 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.140 bool environment::performMoleculeEfflux (acs_longInt *tmpSpecies,* MTRand & *tmp__RndDoubleGen*)

Perform MOLECULE EFFLUX reaction

Version

2.5.1

Date

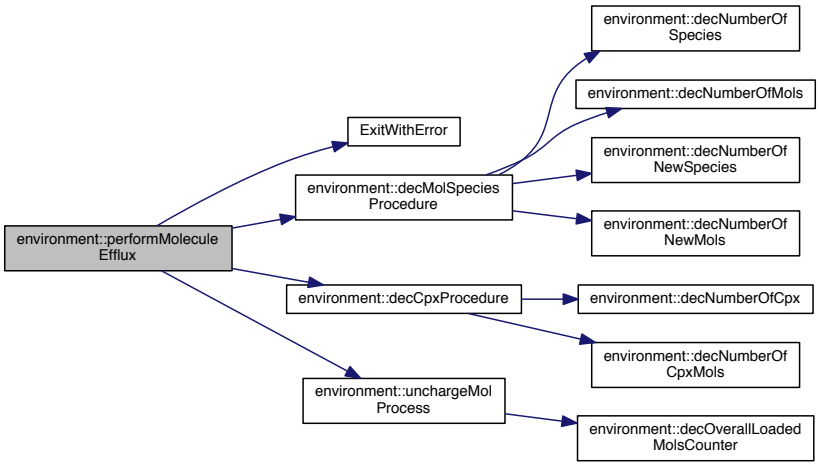
2010-06-27

Parameters

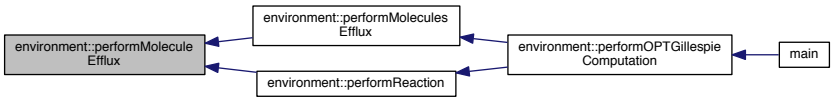
acs_longInt	tmpSpecies Species ID
MTRand&	tmp__RndDoubleGen random generator

Definition at line 4531 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.141 bool environment::performMoleculesEfflux (acs_double *tmpTimeInterval,* MTRand & *tmp_RndDoubleGen*)

This function perform the deterministic molecules efflux process

Version

2.5

Date

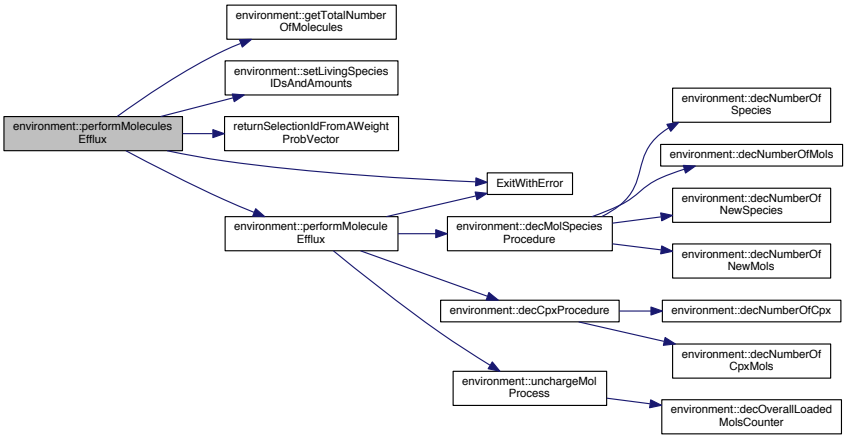
2010-06-25

Parameters

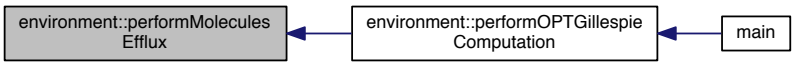
tmpTimeInterval	time elapsed since the last reaction
MTrand	&tmp_RndDoubleGen random number generator

Definition at line 4369 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.142 bool environment::performOPTGillespieComputation (MTRand & *tmpRndDoubleGen,* clock_t & *tmpTimeElapsed,* acs_int *tmpActGEN,* acs_int *tmpActSIM,* acs_int *tmpActSTEP,* string *tmpStoringPath*)

Perform all the gillespie algorithm procedure (OPTIMIZED VERSION)

Version

1.0

Date

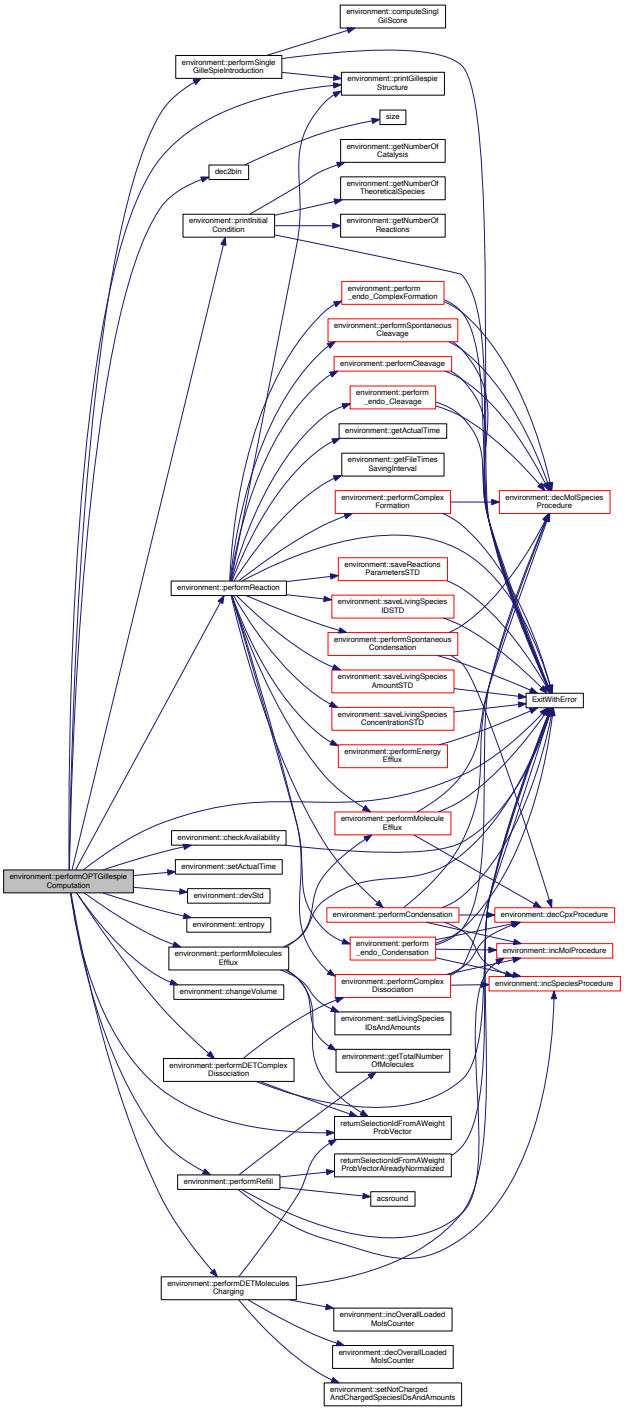
2013.09.17

Parameters

tmpRndDouble-Gen	random numbers generator
tmpTimeElapsed	Computation time elapsed from the T0
tmpActGEN	actual generation
tmpActSIM	actual simulation
tmpActSTEP	actual step (reaction)
tmpStoringPath	path where results are stored

Definition at line 2668 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

12.3.3.143 bool environment::performReaction (acs_longInt *reaction_u,* MTRand & *tmp_RndDoubleGen,* acs_int *tmp_ActGEN,* acs_int *tmp_ActSIM,* acs_int *tmp_ActSTEP,* string *tmp_StoringPath*)

Perform the reaction after the Gillespie computation

Version

1.1

Date

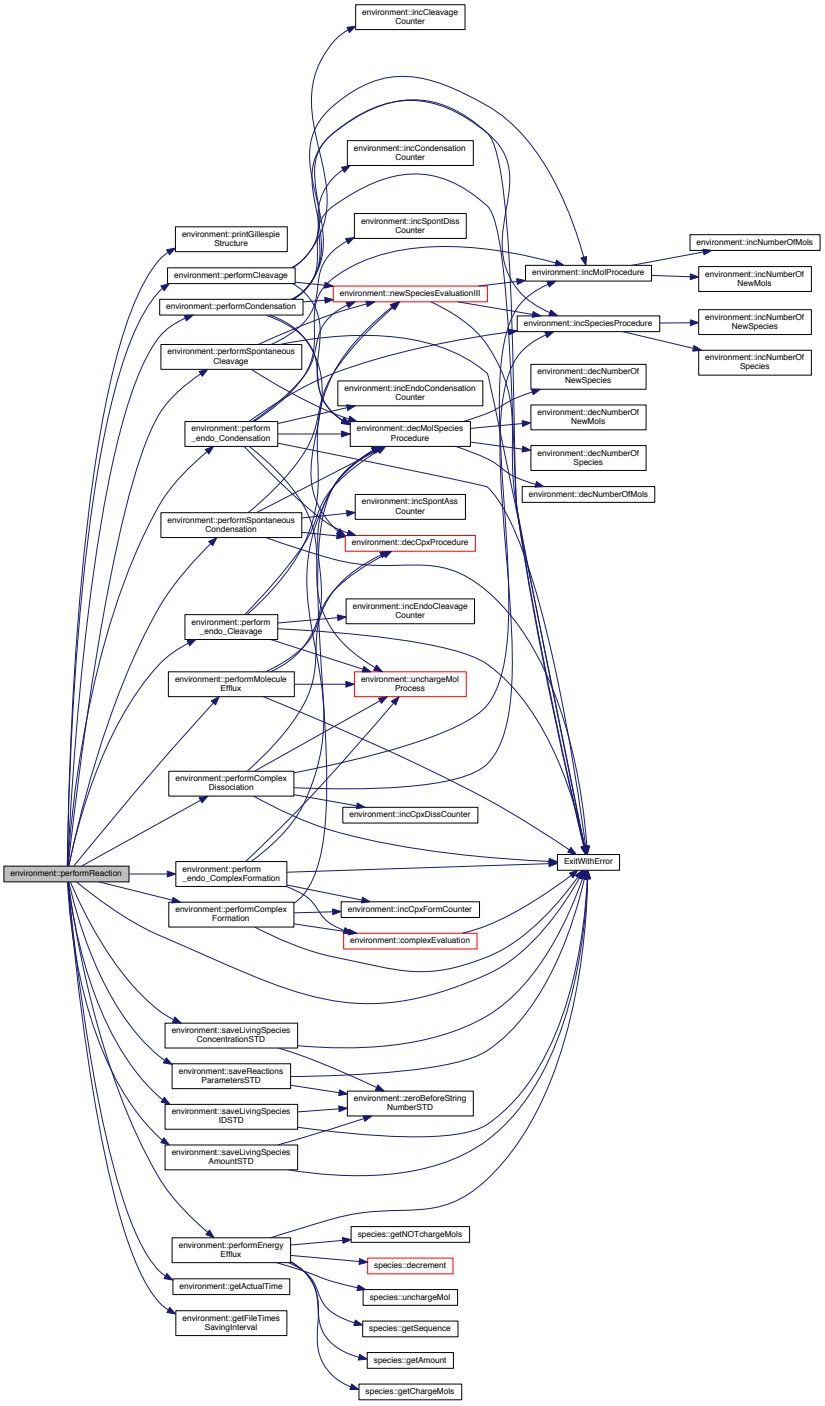
2013-10-29

Parameters

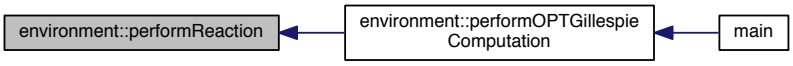
acs_longInt	reaction_u reaction ID in Gillespie structure
MTRand&	tmp_RndDoubleGen Random number generator
acs_int	tmp_ActGEN Current generation
acs_int	tmp_ActSIM Current generation
acs_int	tmp_ActSTEP Current step (reaction)
string	Storing directory

Definition at line 4668 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.144 bool environment::performRefill (acs_double *tmpTimeSinceTheLastInFlux*, acs_double *tmpMinimalTimeForOneMols*, MTRand & *tmp__RndDoubleGen*)

Perform Refill according to the time interval and the total amount of refill

Version

2.4

Date

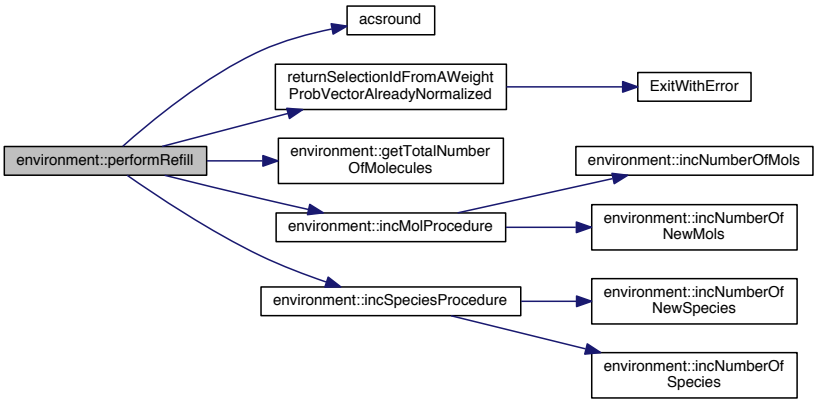
2010.06.10

Parameters

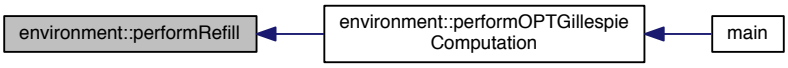
tmpTimeSinceTheLastInFlux	time elapsed since the last influx of at least one molcolule
tmpMinimalTimeForOneMols	time necessary to feed to the system one molecule
tmp__RndDoubleGen	random number generator

Definition at line 3718 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.145 void environment::performSingleGilleSpieIntroduction (acs_longInt *tmpAmountI*, acs_longInt *tmpAmountII*, acs_longInt *tmpIDI*, acs_longInt *tmpIDII*, acs_longInt *tmpIDCatalysis*, acs_int *tmp__rctType*, acs_longInt *tmpMol_I*, acs_longInt *tmpMol_II*, acs_longInt *tmpMol_III*, acs_longInt *tmpMol_IV*, acs_int *tmp_NRGDirection*, acs_longInt *tmpRctID*, bool *tmpSameSpeciesControl*)

Compute and introduct a single Gillespie entry within the Gillespie Structure

Version

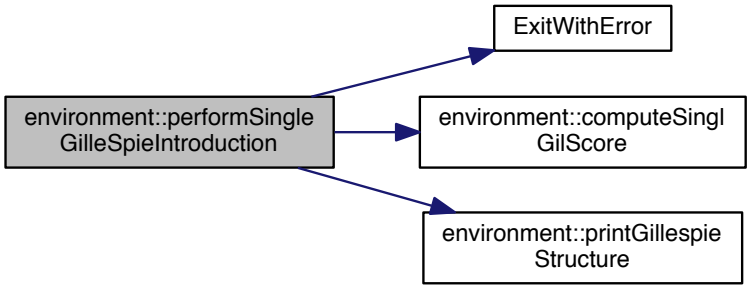
1.0

Date

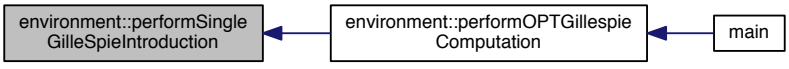
20110222

Definition at line 3314 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.146 bool environment::performSpontaneousCleavage (acs_longInt *tmpReaction*, MTRand & *tmp__RndDoubleGen*)

Perform SPONTANEOUS CLEAVAGE reaction

Version

1.0

Date

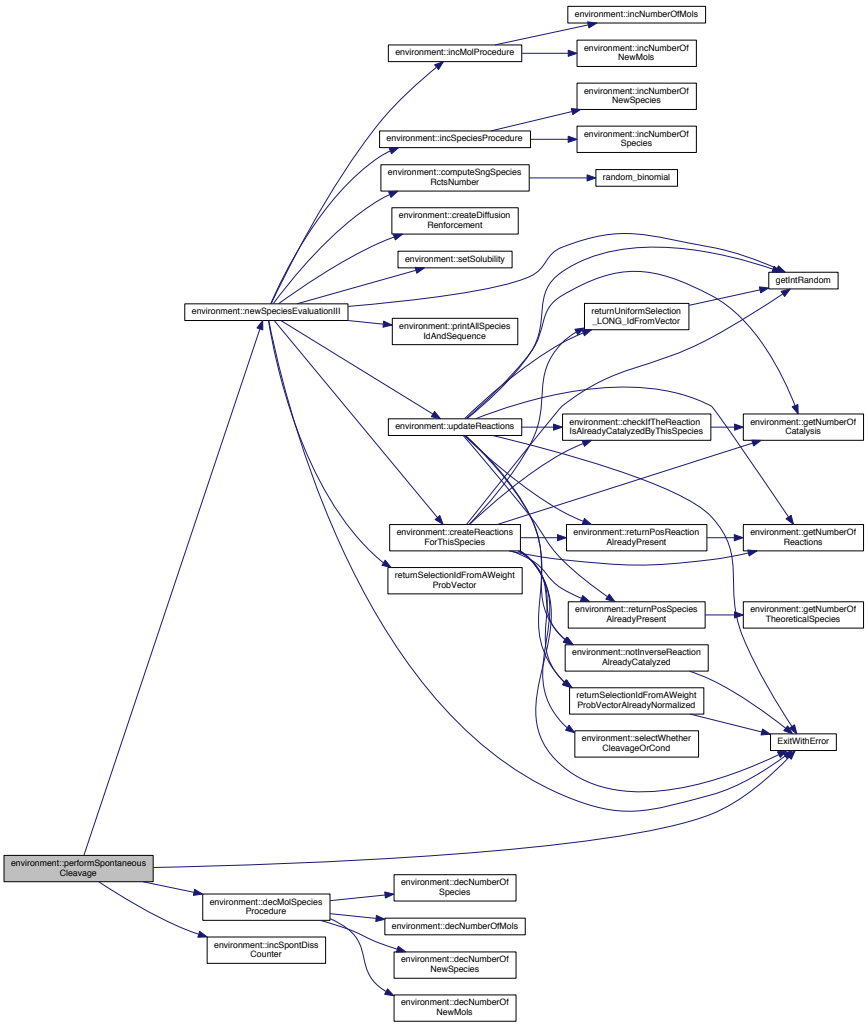
2013.10.28

Parameters

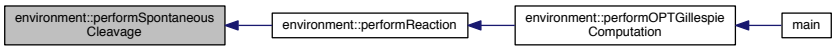
acs_longInt	tmpReaction Reaction ID
MTRand&	tmp__RndDoubleGen random generator

Definition at line 5657 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.147 bool environment::performSpontaneousCondensation (acs_longInt *tmpReaction,* MTRand & *tmp__RndDoubleGen*)

Perform SPONTANEOUS CONDENSATION reaction

Version

1.0

Date

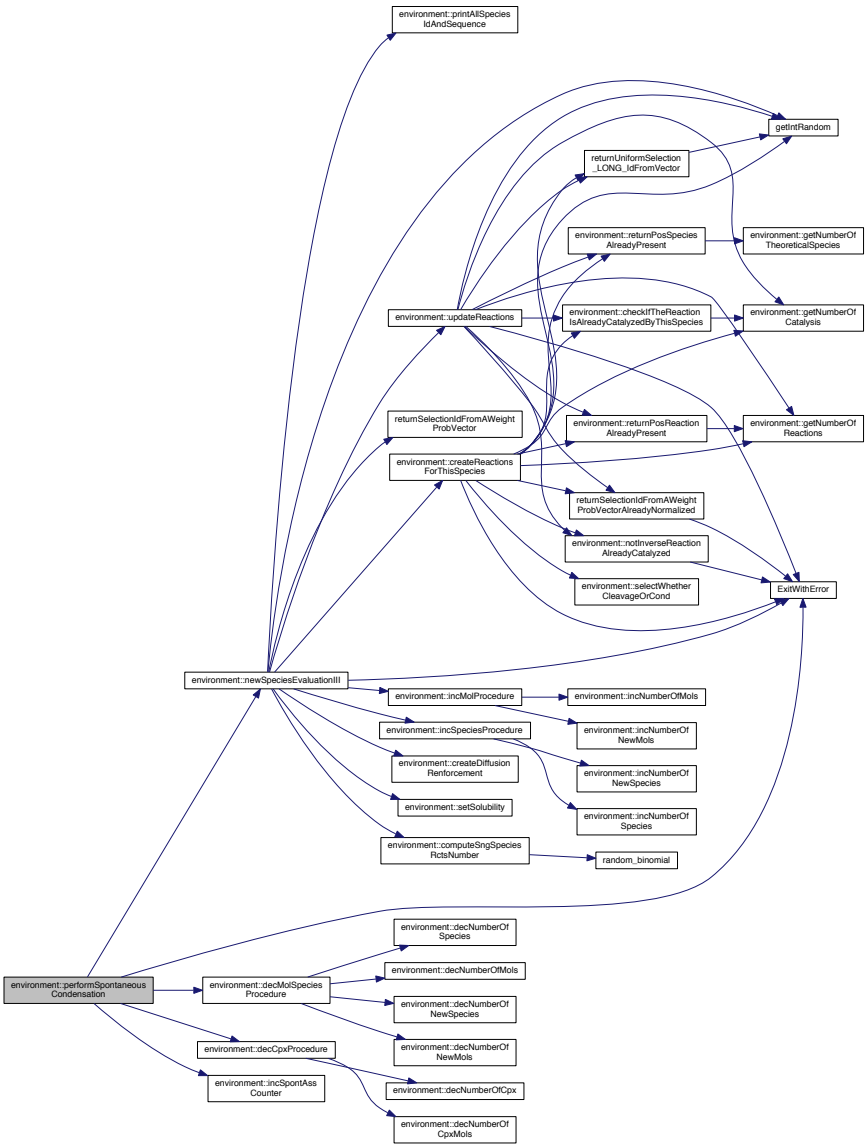
2013-10-28

Parameters

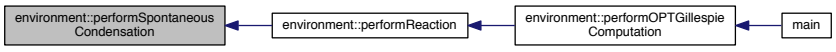
acs_longInt	tmpReaction Reaction ID
MTRand&	tmp__RndDoubleGen random generator

Definition at line 5727 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.148 void environment::printAllChargeMols ( )

Print all the energized molecules 1.0

Date

2010-11-14

Definition at line 6381 of file environment.cpp.

Here is the call graph for this function:

12.3.3.149 void environment::printAllSpeciesIdAndSequence ( )

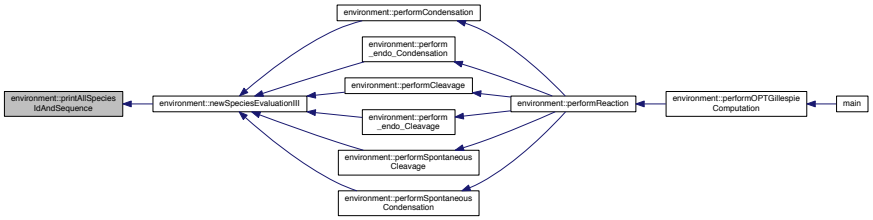
Show all the species with their ID

Version

1.0

Definition at line 6325 of file environment.cpp.

Here is the caller graph for this function:



12.3.3.150 void environment::printGillespieStructure ( )

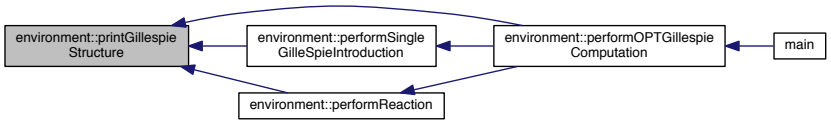
Show all the Gillespie Structure

Version

1.0

Definition at line 6353 of file environment.cpp.

Here is the caller graph for this function:



12.3.3.151 void environment::printInitialCondition ( )

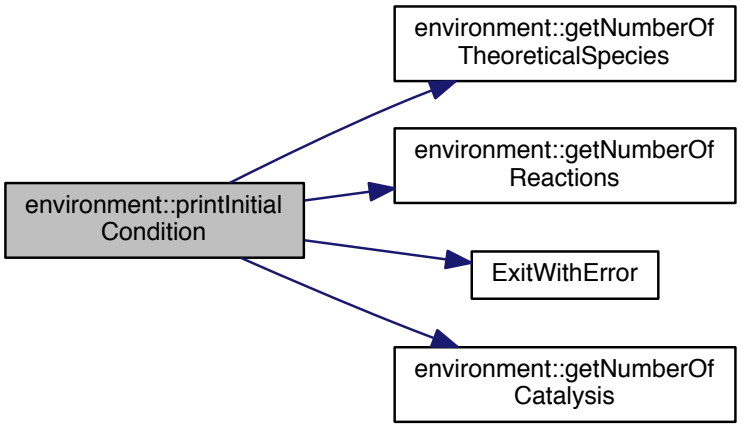
Show all initial species in table format

Version

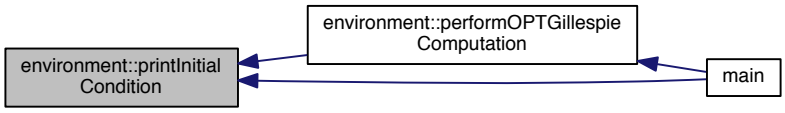
1.0

Definition at line 6267 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.152 void environment::printNutrientsAndProbability ( )

This Function populates two vectors containing all the nutrients and all the probabilities for the influx selected species from the firing disk according to the max length of the influx This function print to monitor the content of the vectors nutrientsForInflux and nutrientsProb2BeSelected

Version

1.0

Date

2010-05-17

Definition at line 3660 of file environment.cpp.

12.3.3.153 bool environment::removeChargeMolFromList (acs_int *tmpSpeciesID*)

Perform vector unchargedIDlist update removing a new charge molecule vectors unchargedIDlist and cumUnchargedAmountList are involved

Version

1.0

Date

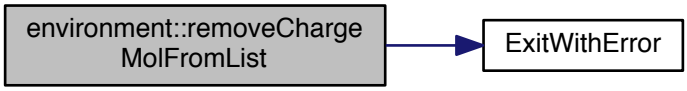
2010-10-10

Parameters

acs_int	tmpSpeciesID Specie to uncharge

Definition at line 4265 of file environment.cpp.

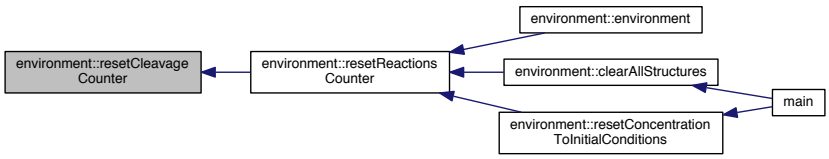
Here is the call graph for this function:



12.3.3.154 void environment::resetCleavageCounter ( ) [inline]

Definition at line 315 of file environment.h.

Here is the caller graph for this function:



12.3.3.155 void environment::resetConcentrationToInitialConditions ( )

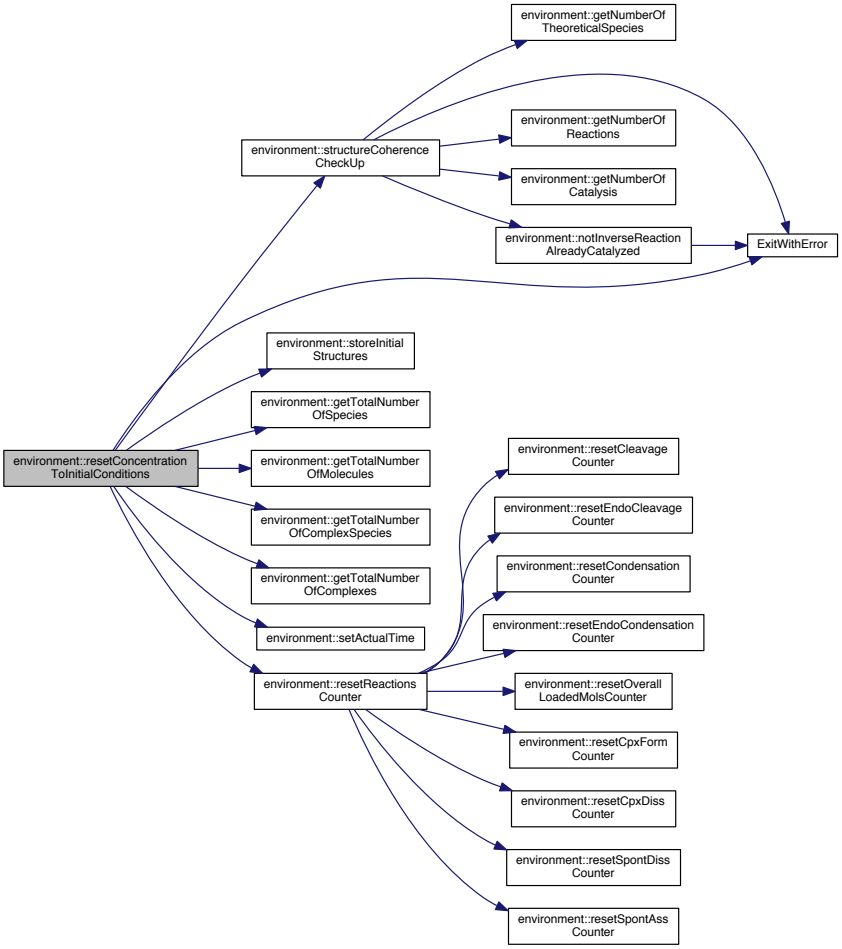
set the concentrations to the initial values and reset internal statistics and counter

Version

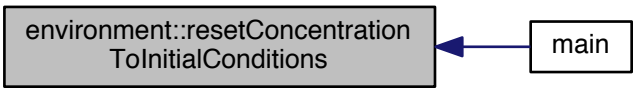
1.0

Definition at line 6472 of file environment.cpp.

Here is the call graph for this function:



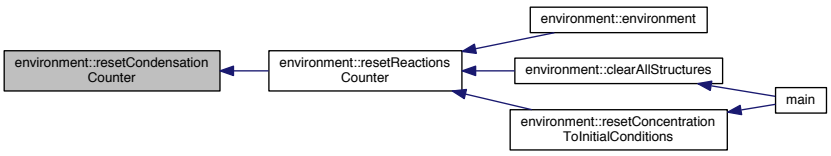
Here is the caller graph for this function:



12.3.3.156 void environment::resetCondensationCounter ( ) [inline]

Definition at line 317 of file environment.h.

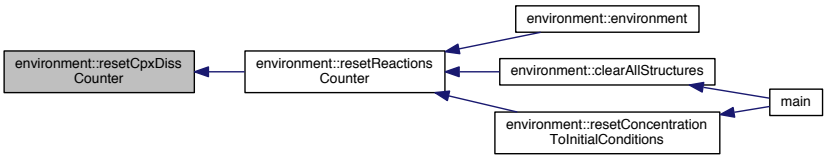
Here is the caller graph for this function:



12.3.3.157 void environment::resetCpxDissCounter ( ) [inline]

Definition at line 321 of file environment.h.

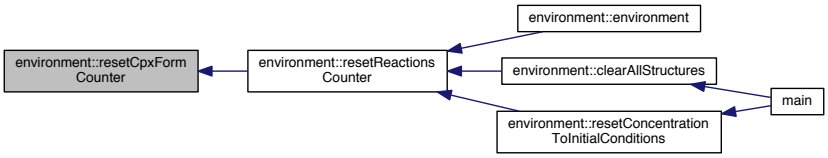
Here is the caller graph for this function:



12.3.3.158 void environment::resetCpxFormCounter ( ) [inline]

Definition at line 320 of file environment.h.

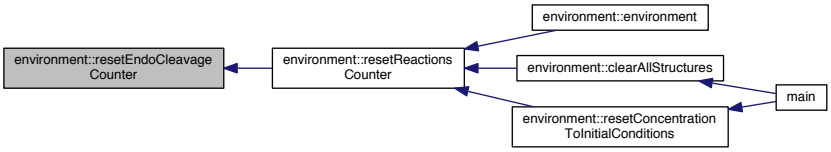
Here is the caller graph for this function:



12.3.3.159 void environment::resetEndoCleavageCounter ( ) [inline]

Definition at line 316 of file environment.h.

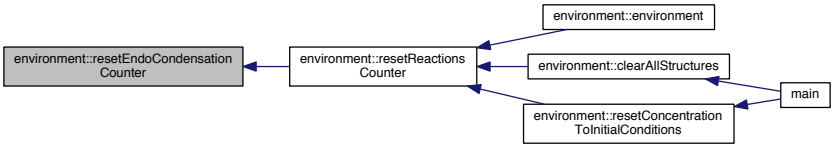
Here is the caller graph for this function:



12.3.3.160 void environment::resetEndoCondensationCounter ( ) [inline]

Definition at line 318 of file environment.h.

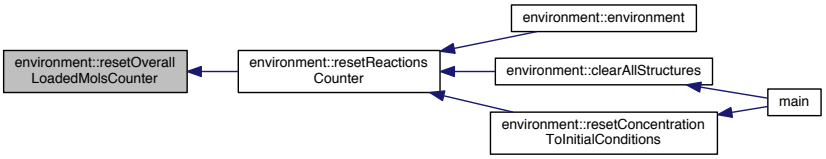
Here is the caller graph for this function:



12.3.3.161 void environment::resetOverallLoadedMolsCounter ( ) [inline]

Definition at line 319 of file environment.h.

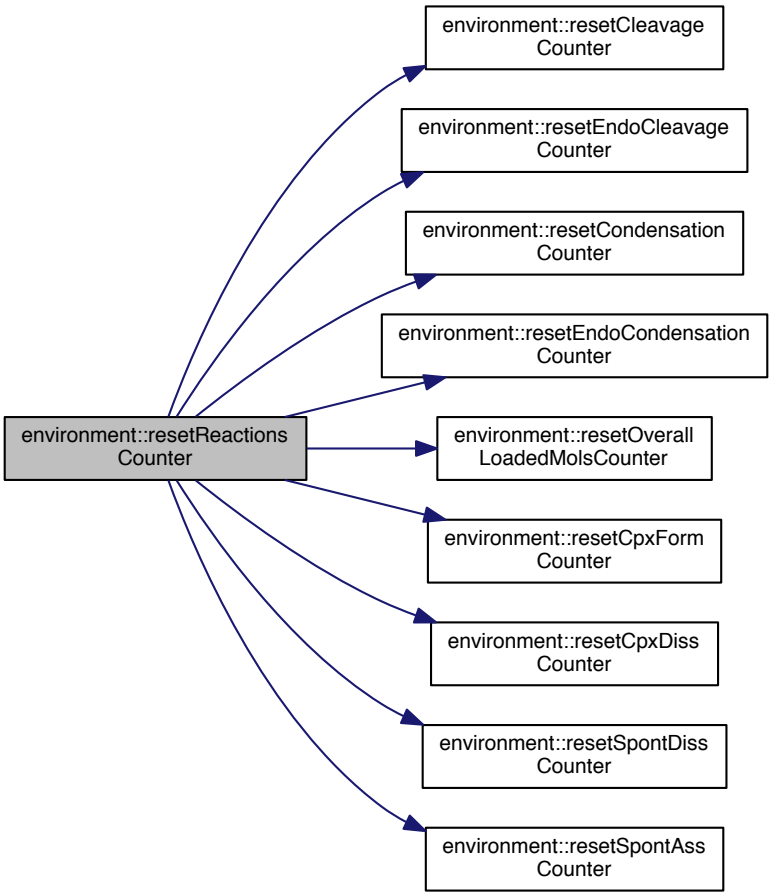
Here is the caller graph for this function:



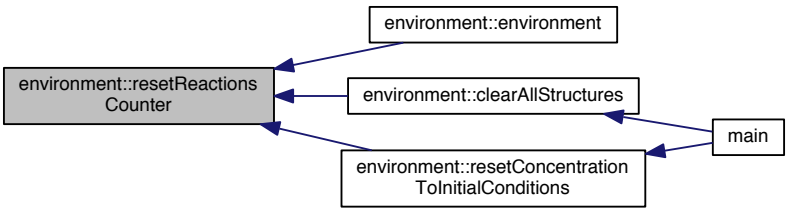
12.3.3.162 void environment::resetReactionsCounter ( ) [inline]

Definition at line 325 of file environment.h.

Here is the call graph for this function:



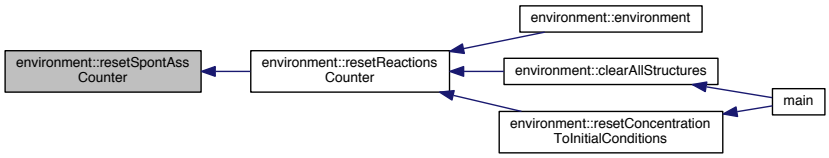
Here is the caller graph for this function:



12.3.3.163 void environment::resetSpontAssCounter ( ) [inline]

Definition at line 323 of file environment.h.

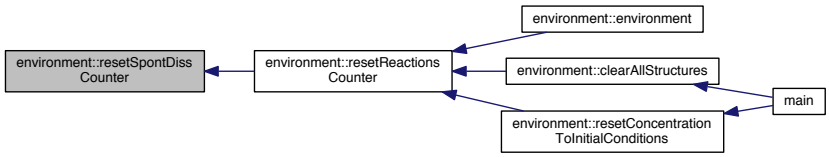
Here is the caller graph for this function:



12.3.3.164 void environment::resetSpontDissCounter ( ) [inline]

Definition at line 322 of file environment.h.

Here is the caller graph for this function:



12.3.3.165 acs_longInt environment::returnPosReactionAlreadyPresent (acs_int *tmpReactionType,* acs_longInt *tmpIds_I,* acs_longInt *tmpIds_II,* acs_longInt *tmpIds_III*)

Return the reaction ID if the reaction is already present, otherwise it returns the new reaction ID

Version

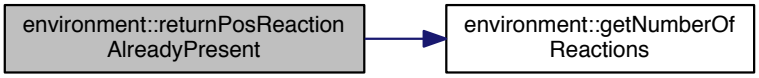
1.0

Parameters

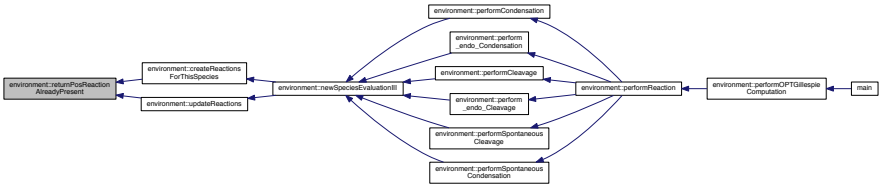
acs_int	tmpReactionType rection type (cleavage or condensation)
acs_longInt	tmpIds_I species I ID
acs_longInt	tmpIds_II species II ID
acs_longInt	tmpIds_III species III ID

Definition at line 1904 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.166 acs_longInt environment::returnPosSpeciesAlreadyPresent (string *tmpNewSequence*)

This functions returns the ID of the species whether this one is already present, otherwise the new ID is created as the number of elements in the species vector

Version

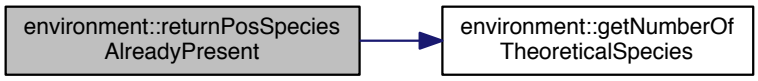
1.0

Parameters

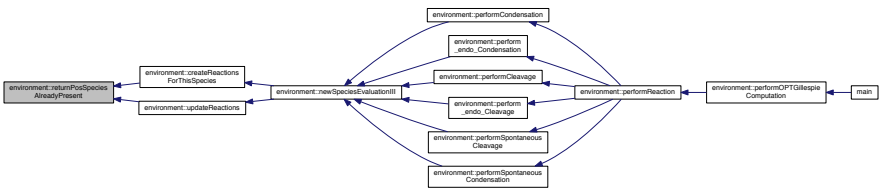
tmpNew-Sequence	New sequence to evaluate
MTRand&	tmp_RndDoubleGen random number generator
tmpNewSpecies-Length	Lenght of the species

Definition at line 1874 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.167 bool environment::saveCatalysisStructureSTD (acs_int *tmpCurrentGen,* acs_int *tmpCurrentSim,* acs_int *tmpCurrentStep,* string *tmpStoringPath*)

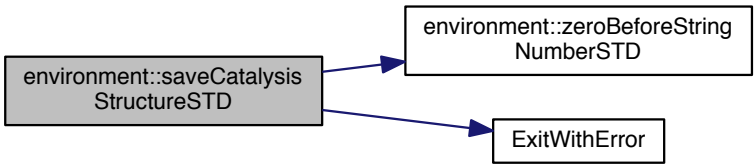
Save the catalysis structures in a file named catalysis_[currentSims]_[currentStep].csv. This is file is equal to the acs_catalysis.csv input file. Standard C++ The file is saved in the directory indicated as a second parameter in the run command

Version

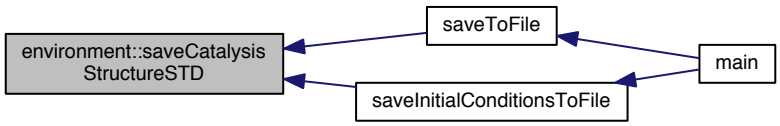
1.0

Definition at line 6978 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.168 bool environment::saveConfigurationFileSTD (string *tmpStoringPath*)

Save a file with the configuration parameters

Version

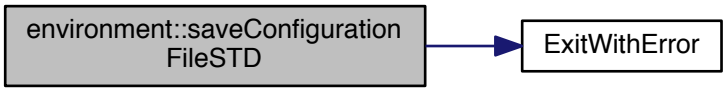
1.0

Date

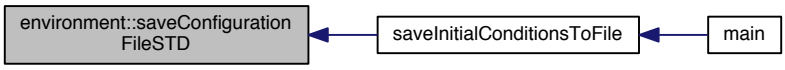
2013/07/03

Definition at line 6613 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.169 bool environment::saveInfluxStructureSTD (string *tmpStoringPath*)

Save influx structure on file, standard C++

Parameters

bool	saveInfluxStructure(QString tmpStoringPath);

Version

1.0

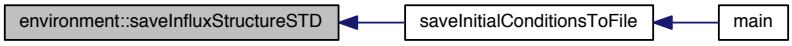
Date

2013-047-03

Definition at line 6783 of file environment.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:



12.3.3.170 bool environment::saveLivingSpeciesAmountSTD (acs_int *tmp__CurrentGen,* acs_int *tmp__CurrentSim,* string *tmp__StoringPath*)

Save living species total AMOUNT in a file named livingAmount_[CurrentGen]_[currentSim].csv. The file is saved in the directory indicated as a second parameter in the run command - Standard C++

Version

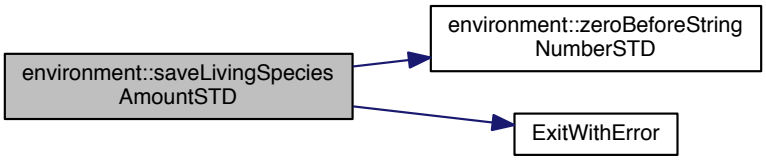
1.0

Date

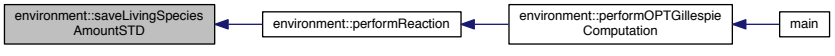
2013/07/03

Definition at line 7242 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.171 bool environment::saveLivingSpeciesConcentrationSTD (acs_int *tmp__CurrentGen,* acs_int *tmp__CurrentSim,* string *tmp__StoringPath*)

Save living species total CONCENTRATION in a file named livingAmount_[CurrentGen]_[currentSim].csv. The file is saved in the directory indicated as a second parameter in the run command

Version

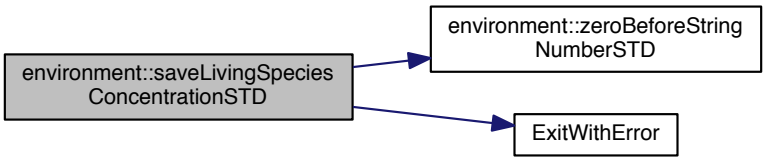
1.0

Date

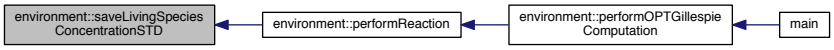
2013/07/03

Definition at line 7308 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.172 bool environment::saveLivingSpeciesIDSTD (acs_int *tmp__CurrentGen,* acs_int *tmp__CurrentSim,* acs_int *tmp__CurrentStep,* string *tmp__StoringPath*)

Save living species in a file named living_species_[currentSim].csv. Standard C++ The file is saved in the directory indicated as a second parameter in the run command

Version

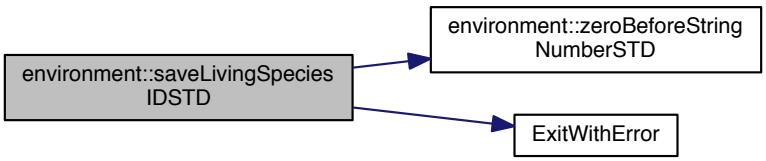
1.0

Date

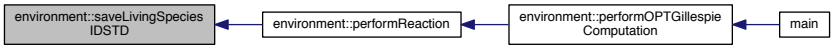
2013/07/03

Definition at line 7181 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.173 bool environment::saveNrgBoolFncStructureSTD (string *tmpStoringPath*)

Save Energetic Boolean Function on file, standard C++

Parameters

bool	saveInfluxStructure(QString tmpStoringPath);

Version

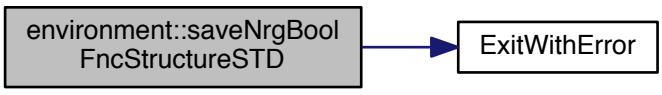
1.0

Date

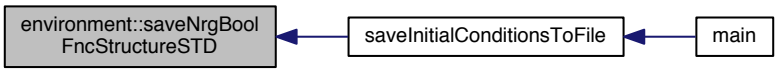
2013-047-03

Definition at line 6821 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.174 bool environment::saveReactionsParametersSTD (acs_int *tmp__CurrentGen,* acs_int *tmp__CurrentSim,* acs_int *tmp__CurrentStep,* string *tmp__StoringPath,* acs_int *tmpRctType,* acs_longInt *tmpCat,* acs_longInt *tmpMol_I,* acs_longInt *tmpMol_II,* acs_longInt *tmpMol_III*)

Save the reactions parameters in a file named reactions_parameters_[currentSim].csv. The file is saved in the directory indicated as a second parameter in the run command

Version

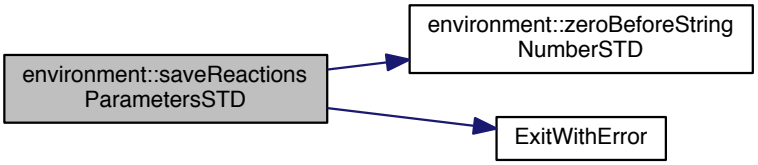
1.0

Date

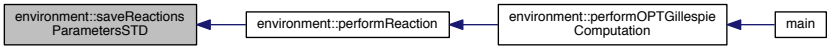
2013/07/03

Definition at line 7122 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.175 bool environment::saveReactionsStructureSTD (acs_int *tmpCurrentGen,* acs_int *tmpCurrentSim,* acs_int *tmpCurrentStep,* string *tmpStoringPath*)

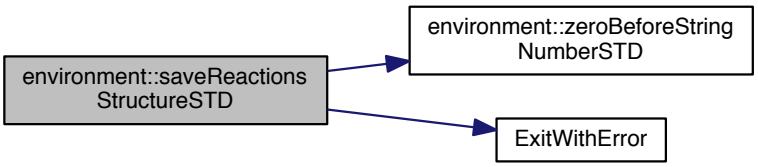
Save the reactions structures in a file named reactions_[currentSims]_[currentStep].csv. This is file is equal to the acs_reactions.csv input file. The file is saved in the directory indicated as a second parameter in the run command

Version

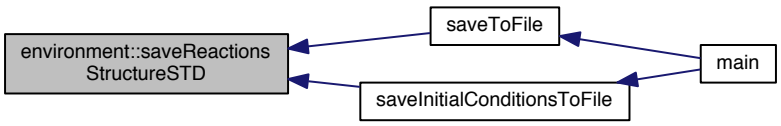
1.0

Definition at line 6924 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.176 bool environment::saveSpeciesStructureSTD (acs_int *tmpCurrentGen*, acs_int *tmpCurrentSim*, acs_int *tmpCurrentStep*, string *tmpStoringPath*)

Save the species structures in a file named species_[currentSims]_[currentStep].csv. This is file is equal to the acs_species.csv input file. C++ standard The file is saved in the directory indicated as a second parameter in the run command

Version

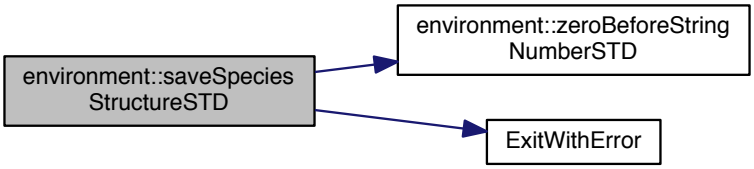
1.0

Date

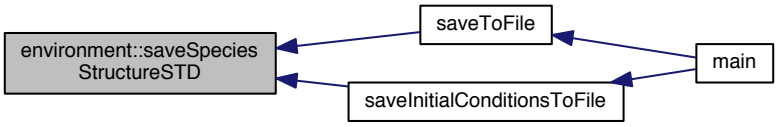
2013/07/03

Definition at line 6859 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.177 bool environment::saveTimesSTD (acs_int *tmpCurrentGen,* acs_int *tmpCurrentSim,* acs_int *tmpCurrentStep,* string *tmpStoringPath*)

Save the reactions times in a file named times_[currentSim].csv. Standard C++ The file is saved in the directory indicated as a second parameter in the run command

Version

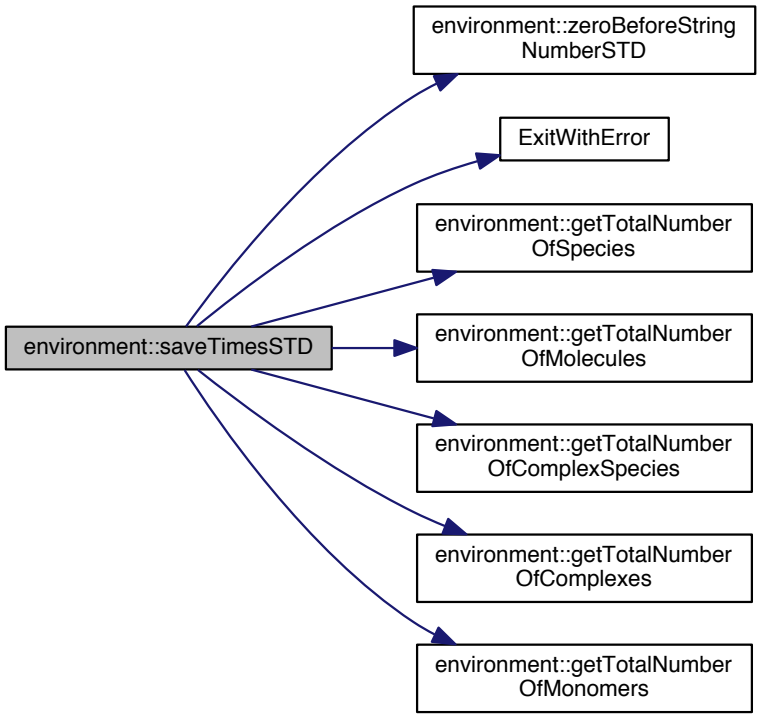
1.0

Date

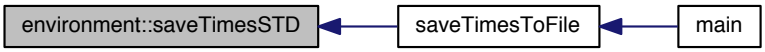
2013/07/03

Definition at line 7035 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.178 acs_int environment::selectWhetherCleavageOrCond (MTRand & *tmp__RndDoubleGen*)

Select reaction type (cleavage or condensation) according to the cleavage probability

Version

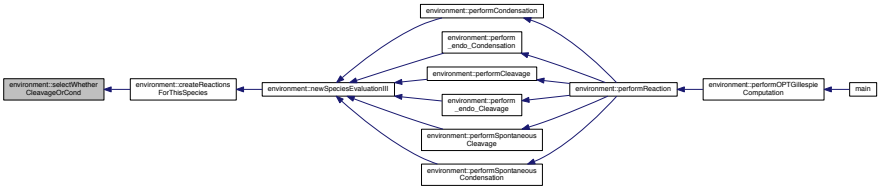
1.0

Parameters

acs_int	tmpTotalNumberOfReactions Total number of conceivable reactions
acs_double	tmpRctsProb reaction probability

Definition at line 1227 of file environment.cpp.

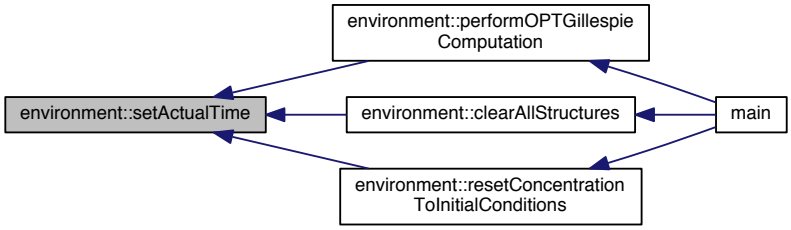
Here is the caller graph for this function:



12.3.3.179 void environment::setActualTime (acs_double *tmpActualTime*) [inline]

Definition at line 343 of file environment.h.

Here is the caller graph for this function:



12.3.3.180 void environment::setLivingSpeciesIDsAndAmounts ( )

This function populates to lists, the first containing all the living species ID and the second one containing the cumulative number of living species

Version

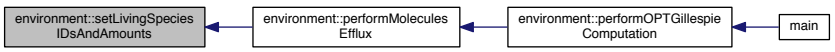
2.5.2

Date

2010-11-11

Definition at line 4176 of file environment.cpp.

Here is the caller graph for this function:



12.3.3.181 void environment::setNotChargedAndChargedSpeciesIDsAndAmounts ( )

This function populates to lists, the first one contains the IDs of the uncharged mols and the second one contains the cumulative amount of uncharged molecules

Version

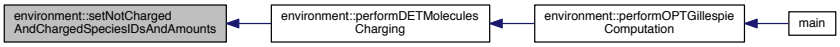
2.5.3

Date

2011-02-22

Definition at line 4217 of file environment.cpp.

Here is the caller graph for this function:



12.3.3.182 bool environment::setSolubility (acs_int *tmpNewSpeciesLength,* MTRand & *tmpRndDoubleGen*)

Create the precipitation constant renforcement according to the species length

Version

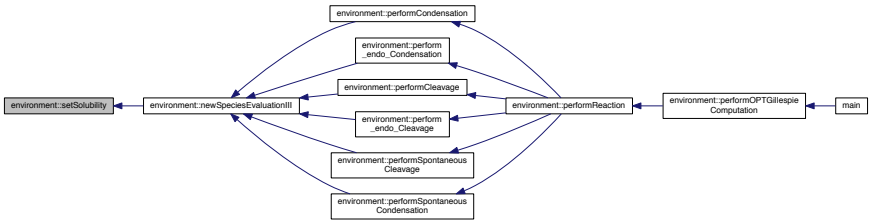
2.5.1

Parameters

tmpPreEnh	precipitation enhancement from parameters
MTRand&	tmp_RndDoubleGen random number generator
tmpNewSpecies-Length	Lenght of the species

Definition at line 1840 of file environment.cpp.

Here is the caller graph for this function:



12.3.3.183 void environment::showGillEngagementInSpecies ( )

Function to show the Gillespie engagement of each species.

Version

: 1.0

Date

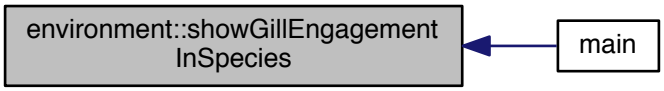
: 20131031

Author

: Alessandro Filisetti

Definition at line 2559 of file environment.cpp.

Here is the caller graph for this function:



12.3.3.184 void environment::showGlobalParameter ( )

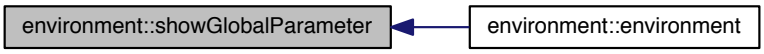
Shows all parameters uploaded from the configuration file

Version

1.0

Definition at line 6174 of file environment.cpp.

Here is the caller graph for this function:



12.3.3.185 void environment::showSubListInSpecies ( )

Function to show the substrates list of the complex species

Definition at line 2523 of file environment.cpp.

12.3.3.186 void environment::storeInitialStructures ( )

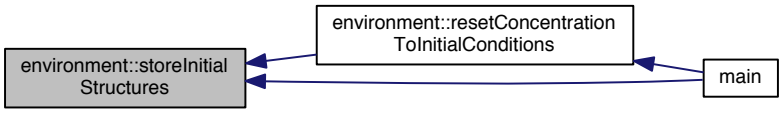
Store initial structures into storing variables

Version

1.0

Definition at line 6567 of file environment.cpp.

Here is the caller graph for this function:



12.3.3.187 bool environment::structureCoherenceCheckUp ( )

This function check the coherence of the data structures. IF IT FAULTS THE SYSTEM RETURNS ERROR!!! IT IS VERY RECOMMENDED RUN THIS FUNCTION ALWAYS AFTER THE INITIALIZATION

Version

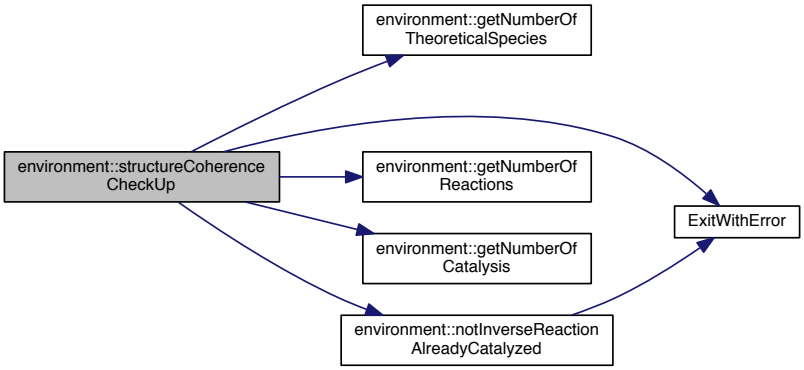
1.1

Date

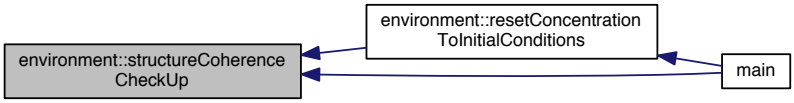
2011-04-13

Definition at line 2004 of file environment.cpp.

Here is the call graph for this function:



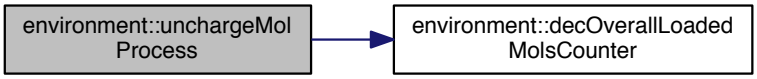
Here is the caller graph for this function:



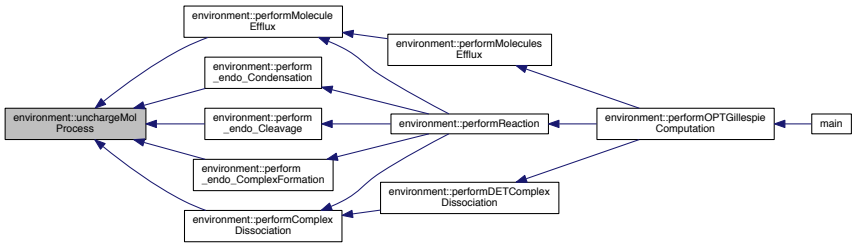
12.3.3.188 void environment::unchargeMolProcess (acs_int *tmp_ID*) [inline]

Definition at line 302 of file environment.h.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.189 bool environment::updateReactions (acs_longInt *tmpIDtoUpdate,* acs_longInt *tmpNewSpecies,* acs_int *tmpRctType,* vector< acs_longInt > & *tmp_AlreadyEvaluatedSpeciesVector,* MTRand & *tmp_RndDoubleGen*)

Old Species Reactions and Catalysis update process

Version

1.0

Date

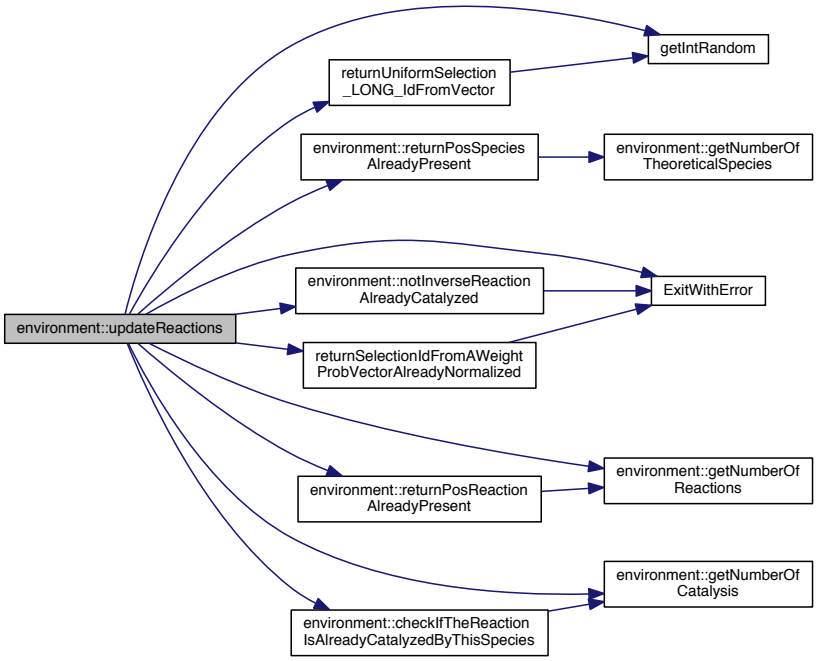
2013/07/30

Parameters

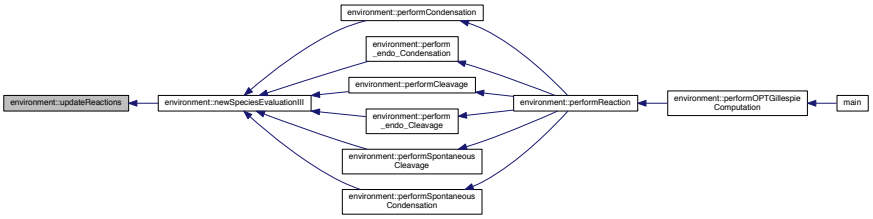
acs_longInt	tmpIDtoUpdate ID of the species to update
acs_longInt	ID of the new species involved in the reactions to update
acs_int	tmpRctType Reaction type, namely cleavage or condensation
MTRand&	tmp____RndDoubleGen Random number generator

Definition at line 941 of file environment.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.3.190 void environment::updateSpeciesAges ( )

Update the species age

Version

1.0

Definition at line 6125 of file environment.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

12.3.3.191 string environment::zeroBeforeStringNumberSTD (acs_int *tmpTotN,* acs_int *tmpCurrentN*)

This function creates a chain of zero as STRING according to tmpTotN and tmpCurrent N in order to make possible a sorting (e.g. tmpTotN = 1000, tmpCurrentN = 3, return 0003

Version

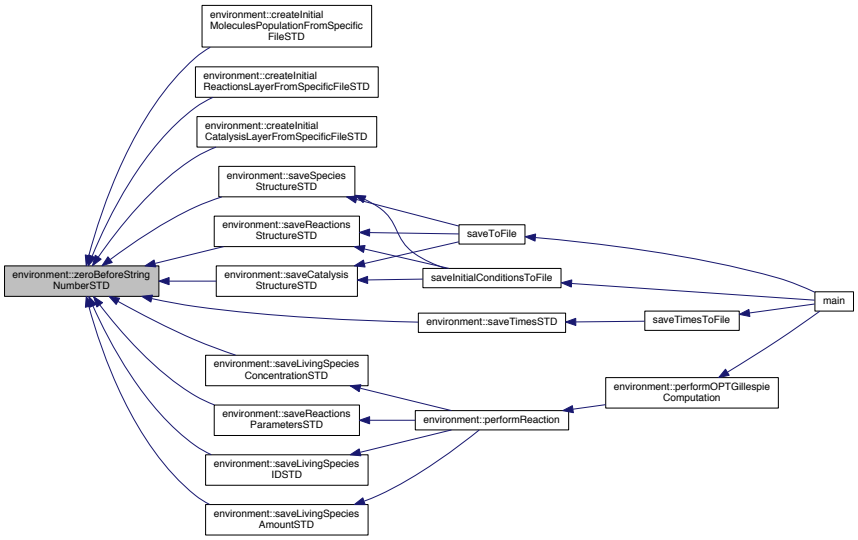
1.0

Parameters

acs_int	tmpTotN Total N
acs_int	tmpCurrentN current N

Definition at line 6590 of file environment.cpp.

Here is the caller graph for this function:



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

- /Users/alessandrofilisetti/Documents/GIT/carness/environment.h
- /Users/alessandrofilisetti/Documents/GIT/carness/environment.cpp

12.4 gillespie Class Reference

`#include <gillespie.h>`

Public Member Functions

- gillespie ()
- gillespie (acs_longInt tmpIDU, acs_int tmpIdReactionType, acs_double tmpScore, acs_longInt tmpMolI, acs_longInt tmpMolII, acs_longInt tmpMolIII, acs_longInt tmpMolIV, acs_longInt tmpIdReaction, acs_longInt tmpIdCatalysis)
- gillespie (acs_longInt tmpIDU, acs_int tmpIdReactionType, acs_double tmpScore, acs_longInt tmpMolI, acs_longInt tmpMolII, acs_longInt tmpMolIII, acs_longInt tmpMolIV, acs_int tmpNRGside, acs_longInt tmpIdReaction, acs_longInt tmpIdCatalysis)
- ~gillespie ()
- acs_longInt getID () const
- acs_int getIdReactionType () const
- acs_double getScore () const

- acs_longInt getMolI () const
- acs_longInt getMolII () const
- acs_longInt getMolIII () const
- acs_longInt getMolIV () const
- acs_longInt getReactionID () const
- acs_longInt getCatalysisID () const
- acs_int getNRGside () const

12.4.1 Detailed Description

Author

Alessandro Filisetti

Version

0.1

Date

2009-04-21

Definition at line 12 of file gillespie.h.

12.4.2 Constructor & Destructor Documentation

12.4.2.1 gillespie::gillespie ( )

12.4.2.2 gillespie::gillespie (acs_longInt *tmpIDU,* acs_int *tmpIdReactionType,* acs_double *tmpScore,* acs_longInt *tmpMolI,* acs_longInt *tmpMolII,* acs_longInt *tmpMolIII,* acs_longInt *tmpMolIV,* acs_longInt *tmpIdReaction,* acs_longInt *tmpIdCatalysis*)

Gillespie Obj constructor

Version

1.0

Parameters

tmpRndDouble-Gen	randomGenerator reference

Definition at line 21 of file gillespie.cpp.

12.4.2.3 gillespie::gillespie (acs_longInt *tmpIDU,* acs_int *tmpIdReactionType,* acs_double *tmpScore,* acs_longInt *tmpMolI,* acs_longInt *tmpMolII,* acs_longInt *tmpMolIII,* acs_longInt *tmpMolIV,* acs_int *tmpNRGside,* acs_longInt *tmpIdReaction,* acs_longInt *tmpIdCatalysis*)

Definition at line 38 of file gillespie.cpp.

12.4.2.4 gillespie::~gillespie ( ) [inline]

Definition at line 33 of file gillespie.h.

12.4.3 Member Function Documentation

12.4.3.1 acs_longInt gillespie::getCatalysisID ( ) const [inline]

Definition at line 44 of file gillespie.h.

12.4.3.2 acs_longInt gillespie::getID ( ) const [inline]

Definition at line 36 of file gillespie.h.

12.4.3.3 acs_int gillespie::getIdReactionType ( ) const [inline]

Definition at line 37 of file gillespie.h.

12.4.3.4 acs_longInt gillespie::getMolI ( ) const [inline]

Definition at line 39 of file gillespie.h.

12.4.3.5 acs_longInt gillespie::getMolII ( ) const [inline]

Definition at line 40 of file gillespie.h.

12.4.3.6 acs_longInt gillespie::getMolIII ( ) const [inline]

Definition at line 41 of file gillespie.h.

12.4.3.7 acs_longInt gillespie::getMolIV ( ) const [inline]

Definition at line 42 of file gillespie.h.

12.4.3.8 acs_int gillespie::getNRGside ( ) const [inline]

Definition at line 45 of file gillespie.h.

12.4.3.9 acs_longInt gillespie::getReactionID ( ) const [inline]

Definition at line 43 of file gillespie.h.

12.4.3.10 acs_double gillespie::getScore ( ) const [inline]

Definition at line 38 of file gillespie.h.

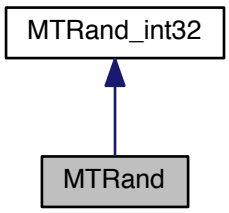
The documentation for this class was generated from the following files:

- /Users/alessandrofilisetti/Documents/GIT/carness/gillespie.h
- /Users/alessandrofilisetti/Documents/GIT/carness/gillespie.cpp

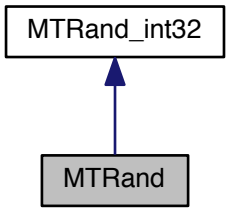
12.5 MTRand Class Reference

`#include <mtrand.h>`

Inheritance diagram for MTRand:



Collaboration diagram for MTRand:



Public Member Functions

- MTRand ()
- MTRand (unsigned long seed)
- MTRand (const unsigned long ∗seed, int size)
- ∼MTRand ()
- double operator() ()

Additional Inherited Members

12.5.1 Detailed Description

Definition at line 97 of file mtrand.h.

12.5.2 Constructor & Destructor Documentation

12.5.2.1 MTRand::MTRand ( ) [inline]

Definition at line 99 of file mtrand.h.

12.5.2.2 MTRand::MTRand (unsigned long *seed*) [inline]

Definition at line 100 of file mtrand.h.

12.5.2.3 MTRand::MTRand (const unsigned long * *seed*, int *size*) [inline]

Definition at line 101 of file mtrand.h.

12.5.2.4 MTRand::~MTRand ( ) [inline]

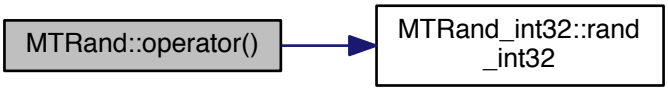
Definition at line 102 of file mtrand.h.

12.5.3 Member Function Documentation

12.5.3.1 double MTRand::operator() ( ) [inline]

Definition at line 103 of file mtrand.h.

Here is the call graph for this function:



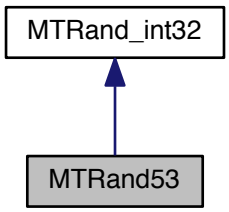
The documentation for this class was generated from the following file:

- /Users/alessandrofilisetti/Documents/GIT/carness/mtrand.h

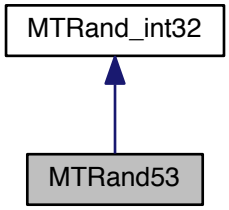
12.6 MTRand53 Class Reference

```
#include <mtrand.h>
```

Inheritance diagram for MTRand53:



Collaboration diagram for MTRand53:



Public Member Functions

- MTRand53 ()
- MTRand53 (unsigned long seed)
- MTRand53 (const unsigned long *seed, int size)
- ~MTRand53 ()
- double operator() ()

Additional Inherited Members

12.6.1 Detailed Description

Definition at line 139 of file mtrand.h.

12.6.2 Constructor & Destructor Documentation

12.6.2.1 MTRand53::MTRand53 ( ) [inline]

Definition at line 141 of file mtrand.h.

12.6.2.2 MTRand53::MTRand53 (unsigned long *seed*) [inline]

Definition at line 142 of file mtrand.h.

12.6.2.3 MTRand53::MTRand53 (const unsigned long * *seed*, int *size*) [inline]

Definition at line 143 of file mtrand.h.

12.6.2.4 MTRand53::~MTRand53 ( ) [inline]

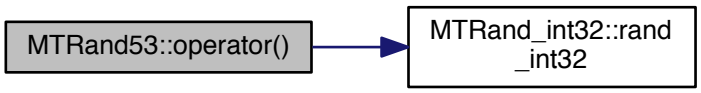
Definition at line 144 of file mtrand.h.

12.6.3 Member Function Documentation

12.6.3.1 double MTRand53::operator() ( ) [inline]

Definition at line 145 of file mtrand.h.

Here is the call graph for this function:



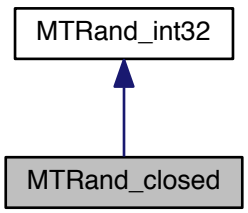
The documentation for this class was generated from the following file:

- /Users/alessandrofilisetti/Documents/GIT/carness/mtrand.h

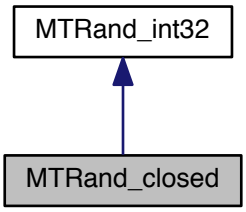
12.7 MTRand_closed Class Reference

`#include <mtrand.h>`

Inheritance diagram for MTRand_closed:



Collaboration diagram for MTRand_closed:



Public Member Functions

- MTRand_closed ()
- MTRand_closed (unsigned long seed)
- MTRand_closed (const unsigned long ∗seed, int size)
- ∼MTRand_closed ()
- double operator() ()

Additional Inherited Members

12.7.1 Detailed Description

Definition at line 111 of file mtrand.h.

12.7.2 Constructor & Destructor Documentation

12.7.2.1 MTRand_closed::MTRand_closed ( ) [inline]

Definition at line 113 of file mtrand.h.

12.7.2.2 MTRand_closed::MTRand_closed (unsigned long *seed*) [inline]

Definition at line 114 of file mtrand.h.

12.7.2.3 MTRand_closed::MTRand_closed (const unsigned long ∗ *seed*, int *size*) [inline]

Definition at line 115 of file mtrand.h.

12.7.2.4 MTRand_closed::∼MTRand_closed ( ) [inline]

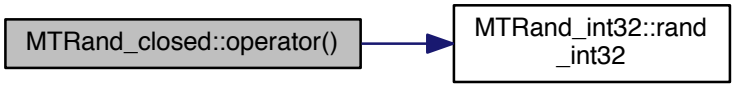
Definition at line 116 of file mtrand.h.

12.7.3 Member Function Documentation

12.7.3.1 double MTRand_closed::operator() ( ) [inline]

Definition at line 117 of file mtrand.h.

Here is the call graph for this function:



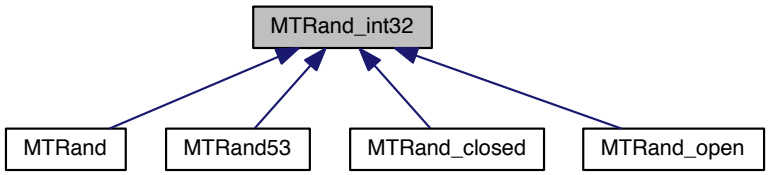
The documentation for this class was generated from the following file:

- /Users/alessandrofilisetti/Documents/GIT/carness/mtrand.h

12.8 MTRand_int32 Class Reference

`#include <mtrand.h>`

Inheritance diagram for MTRand_int32:



Public Member Functions

- MTRand_int32 ()
- MTRand_int32 (unsigned long s)
- MTRand_int32 (const unsigned long *array, int size)
- void seed (unsigned long)
- void seed (const unsigned long *, int size)
- unsigned long operator() ()
- virtual ~MTRand_int32 ()

Protected Member Functions

- unsigned long rand_int32 ()

12.8.1 Detailed Description

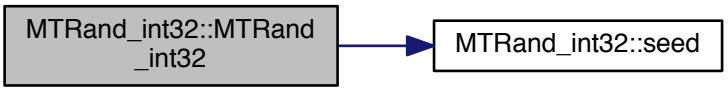
Definition at line 48 of file mtrand.h.

12.8.2 Constructor & Destructor Documentation

12.8.2.1 MTRand_int32::MTRand_int32 ( ) [inline]

Definition at line 51 of file mtrand.h.

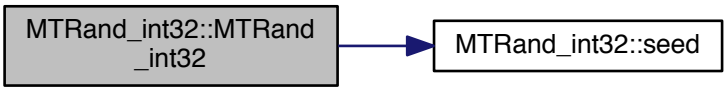
Here is the call graph for this function:



12.8.2.2 MTRand_int32::MTRand_int32 (unsigned long *s*) [inline]

Definition at line 53 of file mtrand.h.

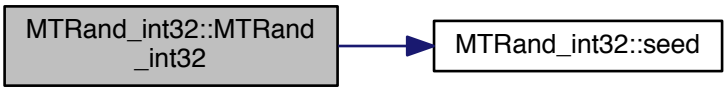
Here is the call graph for this function:



12.8.2.3 MTRand_int32::MTRand_int32 (const unsigned long * *array*, int *size*) [inline]

Definition at line 55 of file mtrand.h.

Here is the call graph for this function:



12.8.2.4 virtual MTRand_int32::~MTRand_int32 ( ) [inline], [virtual]

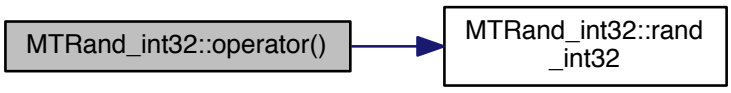
Definition at line 62 of file mtrand.h.

12.8.3 Member Function Documentation

12.8.3.1 unsigned long MTRand_int32::operator() ( ) [inline]

Definition at line 60 of file mtrand.h.

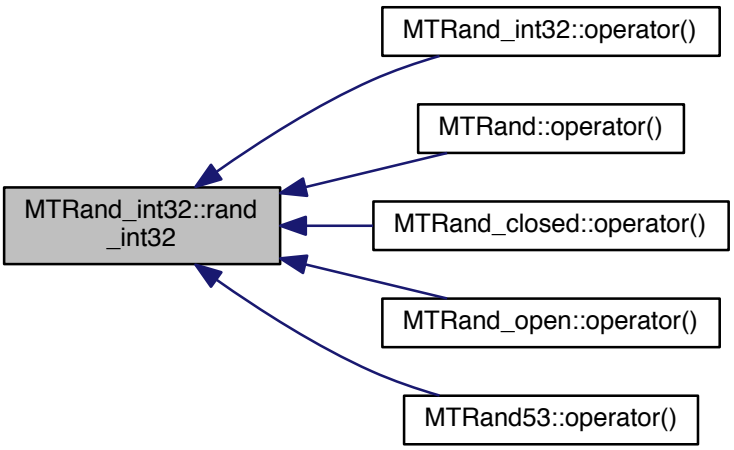
Here is the call graph for this function:



12.8.3.2 unsigned long MTRand_int32::rand_int32 ( ) [inline], [protected]

Definition at line 85 of file mtrand.h.

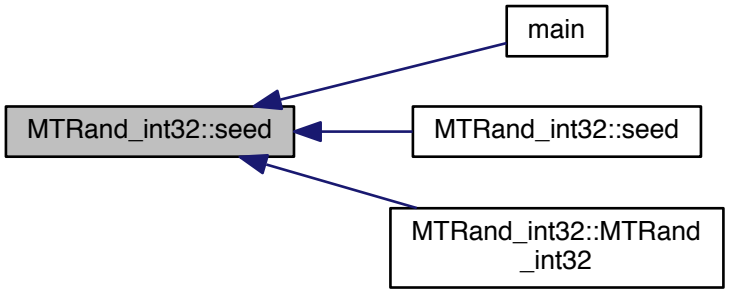
Here is the caller graph for this function:



12.8.3.3 void MTRand_int32::seed (unsigned long *s*)

Definition at line 23 of file mtrand.cpp.

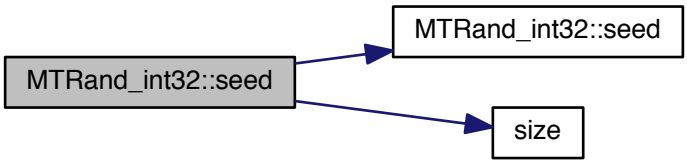
Here is the caller graph for this function:



12.8.3.4 void MTRand_int32::seed (const unsigned long * *array*, int *size*)

Definition at line 35 of file mtrand.cpp.

Here is the call graph for this function:



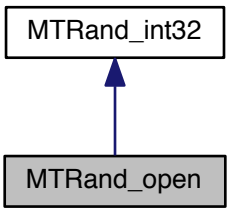
The documentation for this class was generated from the following files:

- /Users/alessandrofilisetti/Documents/GIT/carness/mtrand.h
- /Users/alessandrofilisetti/Documents/GIT/carness/mtrand.cpp

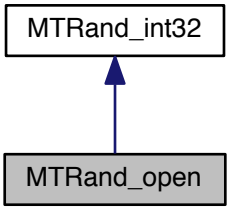
12.9 MTRand_open Class Reference

`#include <mtrand.h>`

Inheritance diagram for MTRand_open:



Collaboration diagram for MTRand_open:



Public Member Functions

- MTRand_open ()
- MTRand_open (unsigned long seed)
- MTRand_open (const unsigned long *seed, int size)
- ~MTRand_open ()
- double operator() ()

Additional Inherited Members

12.9.1 Detailed Description

Definition at line 125 of file mtrand.h.

12.9.2 Constructor & Destructor Documentation

12.9.2.1 MTRand_open::MTRand_open ( ) [inline]

Definition at line 127 of file mtrand.h.

12.9.2.2 MTRand_open::MTRand_open (unsigned long *seed*) [inline]

Definition at line 128 of file mtrand.h.

12.9.2.3 MTRand_open::MTRand_open (const unsigned long * *seed*, int *size*) [inline]

Definition at line 129 of file mtrand.h.

12.9.2.4 MTRand_open::~MTRand_open ( ) [inline]

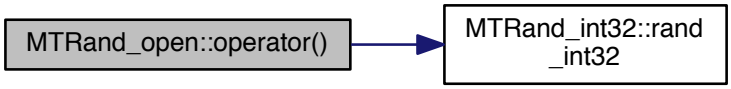
Definition at line 130 of file mtrand.h.

12.9.3 Member Function Documentation

12.9.3.1 double MTRand_open::operator() ( ) [inline]

Definition at line 131 of file mtrand.h.

Here is the call graph for this function:



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

- /Users/alessandrofilisetti/Documents/GIT/carness/mtrand.h

12.10 reactions Class Reference

`#include <reactions.h>`

Public Member Functions

- reactions (acs_longInt tmpID, acs_int tmpType, acs_longInt tmpM_I, acs_longInt tmpM_II, acs_longInt tmpM_III, acs_int tmpEvents, acs_int tmpEnergyType, acs_double tmpKspont)

  Constructor.
- ~reactions ()
- acs_longInt getID () const
- acs_int getType () const
- acs_longInt getSpecies_I () const
- acs_longInt getSpecies_II () const
- acs_longInt getSpecies_III () const
- acs_int getEvents () const
- acs_int getEnergyType () const

- acs_double getKspont () const
- void updateTotEvents ()
- void resetEventsCounter ()

12.10.1 Detailed Description

Definition at line 17 of file reactions.h.

12.10.2 Constructor & Destructor Documentation

12.10.2.1 reactions::reactions (acs_longInt *tmpID*, acs_int *tmpType*, acs_longInt *tmpM_I*, acs_longInt *tmpM_II*, acs_longInt *tmpM_III*, acs_int *tmpEvents*, acs_int *tmpEnergyType*, acs_double *tmpKspont*)

Constructor.

Parameters

tmpID	reaction identificator
tmpType	condensation or cleavage
tmpM_I	product (if condensation) or substrates (if cleavage)
tmpM_II	product (if cleavage) or substrates (if condensation)
tmpM_III	product (if cleavage) or substrates (if condensation)
tmpKeq	equilibrium constant

Definition at line 20 of file reactions.cpp.

12.10.2.2 reactions::~reactions ( ) [inline]

Definition at line 33 of file reactions.h.

12.10.3 Member Function Documentation

12.10.3.1 acs_int reactions::getEnergyType ( ) const [inline]

Definition at line 42 of file reactions.h.

12.10.3.2 acs_int reactions::getEvents ( ) const [inline]

Definition at line 41 of file reactions.h.

12.10.3.3 acs_longInt reactions::getID ( ) const [inline]

Definition at line 36 of file reactions.h.

12.10.3.4 acs_double reactions::getKspont ( ) const [inline]

Definition at line 43 of file reactions.h.

12.10.3.5 acs_longInt reactions::getSpecies_I ( ) const [inline]

Definition at line 38 of file reactions.h.

12.10.3.6 acs_longInt reactions::getSpecies_II ( ) const [inline]

Definition at line 39 of file reactions.h.

12.10.3.7 acs_longInt reactions::getSpecies_III ( ) const [inline]

Definition at line 40 of file reactions.h.

12.10.3.8 acs_int reactions::getType ( ) const [inline]

Definition at line 37 of file reactions.h.

12.10.3.9 void reactions::resetEventsCounter ( ) [inline]

Definition at line 47 of file reactions.h.

12.10.3.10 void reactions::updateTotEvents ( ) [inline]

Definition at line 46 of file reactions.h.

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

- /Users/alessandrofilisetti/Documents/GIT/carness/reactions.h
- /Users/alessandrofilisetti/Documents/GIT/carness/reactions.cpp

12.11 species Class Reference

This class contains declarations of the species class.

`#include <species.h>`

Public Member Functions

- species ()

  < *New species constructor (IN AMOUNT)*
- species (acs_longInt tmpID, string tmpSequence, acs_longInt tmpAmount, acs_double tmpDiffusionEnh, acs_int tmpSoluble, acs_double tmpComplexDegEnh, acs_int tmpComplexCuttingPoint, acs_int tmpEvalueted, acs_double tmpVolume, acs_double tmpK_phospho, acs_int tmpEnergizable, acs_double tmpInflux_rate, acs_int tmpMaxLOut)

  New species constructor (IN CONCENTRATION)
- species (acs_longInt tmpID, string tmpSequence, acs_double tmpConcentration, acs_double tmpDiffusionEnh, acs_int tmpSoluble, acs_double tmpComplexDegEnh, acs_int tmpComplexCuttingPoint, acs_int tmpEvalueted, acs_double tmpVolume, acs_double tmpK_phospho, acs_int tmpEnergizable, acs_double tmpInflux_rate, acs_int tmpMaxLOut)

  New species constructor in case of species structure file upload (IN AMOUNT)
- species (acs_longInt tmpID, string tmpSequence, acs_longInt tmpAmount, acs_double tmpDiffusionEnh, acs_int tmpSoluble, acs_double tmpComplexDegEnh, acs_int tmpComplexCuttingPoint, acs_int tmpEvalueted, acs_double tmpAge, acs_int tmpReborns, acs_double tmpVolume, acs_longInt tmpNotUsedCatID, acs_longInt tmpNotUsedSubID, acs_double tmpK_phospho, acs_int tmpEnergizable, acs_double tmpInflux_rate, acs_int tmpMaxLOut)

  New species constructor in case of species structure file upload (IN CONCENTRATION)

- species (acs_longInt tmpID, string tmpSequence, acs_double tmpConcentration, acs_double tmpDiffusionEnh, acs_int tmpSoluble, acs_double tmpComplexDegEnh, acs_int tmpComplexCuttingPoint, acs_int tmpEvalueted, acs_double tmpAge, acs_int tmpReborns, acs_double tmpVolume, acs_longInt tmpNotUsedCatID, acs_longInt tmpNotUsedSubID, acs_double tmpK_phospho, acs_double tmpKLoadConc, acs_int tmpEnergizable, acs_double tmpInflux_rate, acs_int tmpMaxLOut)

  New random species constructor.
- species (acs_longInt tmpID, string tmpSequence, acs_longInt tmpAmount, acs_double tmpDiffusionEnh, acs_int tmpSoluble, acs_double tmpComplexProb, acs_double tmpMaxComplexDegKinetic, MTRand &tmp_RandomGenerator, acs_double tmpVolume, acs_double tmpK_phospho, acs_int tmpEnergizable)

  new Complex species constructor
- species (acs_longInt tmpID, string tmpSequence, acs_double tmpDiffusionEnh, acs_int tmpSoluble, acs_double tmpMaxComplexDegKinetic, acs_int tmpCuttingPoint, MTRand &tmp_RandomGenerator, acs_longInt tmpCatalyst_ID, acs_longInt tmpSubstrate_ID, acs_double tmpVolume, acs_double tmpK_phospho, acs_int tmpEnergizable)

  This constructor is used to create a molecular complex.
- ~species ()
- acs_longInt getID () const
- string getSequence () const
- acs_int getSequenceLength () const
- acs_longInt getAmount () const
- acs_longInt getNOTchargeMols () const
- acs_longInt getChargeMols () const
- acs_double getConcentration () const
- acs_double getLoadedConcentration (acs_double tmpVolume)
- acs_double getAge () const
- acs_int getReborns () const
- acs_double getDiffusionEnh () const
- acs_int getSolubility () const
- acs_double getComplexDegEnh () const
- acs_int getComplexCutPnt () const
- acs_int getEvaluated () const
- acs_longInt getCatalyst_ID () const
- acs_longInt getSubstrate_ID () const
- acs_double getK_phospho () const
- acs_int getEnergizable () const
- bool getConcentrationFixed () const
- acs_double getFirstConcentration () const
- acs_int getLastSpeciesEvaluated () const
- vector< acs_longInt > getSecSubList () const
- vector< acs_double > getSec_k_SubList () const
- vector< acs_longInt > getCatalysisIfCpx () const
- acs_int getSecSubListSize () const
- acs_longInt getSecSubListID (acs_int tmpID) const
- acs_double getSec_k_SubListID (acs_int tmpID) const
- acs_longInt getCatalysisIfCpxID (acs_int tmpID) const
- void increment (acs_double tmpVolume)
- void specificIncrement (acs_int tmpIncrement, acs_double tmpVolume)
- void setAmount (acs_int tmpAmount, acs_double tmpVolume)
- void setConcentration (acs_double tmpConc, acs_double tmpVolume)
- void decrement (acs_double tmpVolume)
- bool setChargeMols (acs_int tmpMolsToCharge)
- bool setSpecificChargeMols (acs_int tmpMolsToCharge)
- bool chargeMol ()
- bool unchargeMol ()
- void setEvaluated ()

- void setDiffusion (acs_double tmpDiff)
- void setSolubility (acs_int tmpSol)
- void setKphospho (acs_double tmpKphospho)
- void setNewAge (acs_double tmpLastTimeInterval)
- void rebornsIncrement ()
- void concToNum (acs_double tmpVolume)
- void numToConc (acs_double tmpVolume)
- void resetAge ()
- void resetReborns ()
- void resetToInitConc (acs_double tmpVolume)
- void setLastSpeciesEvaluated (acs_int tmpID)
- void insertSecSub (acs_longInt tmpID, acs_double tmpK, acs_longInt tmpCat)
- void insertGillID (acs_longInt tmpID)
- acs_longInt getGillIDpos (acs_longInt tmpID) const
- void showGillEngagement ()

12.11.1 Detailed Description

This class contains declarations of the species class.

class species

Authors

Alessandro Filisetti

Version

1.1 questa modifica è di prova per subversion

Created by Alessandro Filisetti on 19/02/09. Copyright 2009 European Centre for Living Technology. All rights reserved. Test paxelito SVN

Definition at line 18 of file species.h.

12.11.2 Constructor & Destructor Documentation

12.11.2.1 species::species ( )

< New species constructor (IN AMOUNT)

This class containing the declaration of the species.

class species

Authors

Alessandro Filisetti

Version

0.1

Created by Alessandro Filisetti on 19/02/09. Copyright 2009 European Centre for Living Technology. All rights reserved.Default constructor

Definition at line 16 of file species.cpp.

12.11.2.2 species::species (acs_longInt *tmpID,* string *tmpSequence,* acs_longInt *tmpAmount,* acs_double *tmpDiffusionEnh,* acs_int *tmpSoluble,* acs_double *tmpComplexDegEnh,* acs_int *tmpComplexCuttingPoint,* acs_int *tmpEvalueted,* acs_double *tmpVolume,* acs_double *tmpK_phospho,* acs_int *tmpEnergizable,* acs_double *tmpInflux_rate,* acs_int *tmpMaxLOut*)

New species constructor (IN CONCENTRATION)

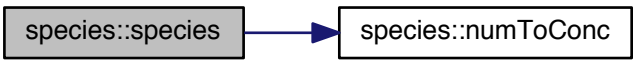
This constructor is used each time a new species is created (AMOUNT BASED)

Parameters

tmpID	species identificator
tmpSequence	species sequence (e.g. ABABAABABA)
tmpAmount	species initial amount
tmpDiffusionEnh	Diffusion enhancement degree
tmpSoluble	1 if the species is soluble, 0 otherwise
tmpComplex-DegEnh	complex dissociation kinetic constant
tmpComplex-CuttingPoint	complex cutting point (catalyst-substrate)
tmpEvalueted	This parameter indicates whether the species has been already evalutad (i.e. all the catalysis of the species are instantiated)
tmpVolume	the volume is necessary to convert numbers in concentrations
tmpK_phospho	phosphorilation kinetic constant (in case of energy based simulations)
tmpEnergizable	this is a flag indicating whether or not the species is energizable

Definition at line 53 of file species.cpp.

Here is the call graph for this function:



12.11.2.3 species::species (acs_longInt *tmpID,* string *tmpSequence,* acs_double *tmpConcentration,* acs_double *tmpDiffusionEnh,* acs_int *tmpSoluble,* acs_double *tmpComplexDegEnh,* acs_int *tmpComplexCuttingPoint,* acs_int *tmpEvalueted,* acs_double *tmpVolume,* acs_double *tmpK_phospho,* acs_int *tmpEnergizable,* acs_double *tmpInflux_rate,* acs_int *tmpMaxLOut*)

New species constructor in case of species structure file upload (IN AMOUNT)

This constructor is used each time a new species is created (CONCENTRATION BASED)

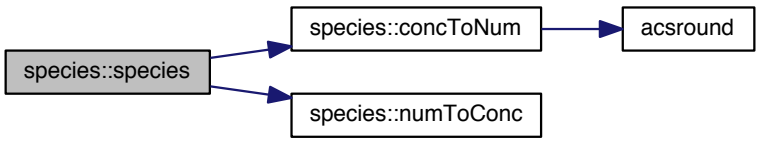
Parameters

tmpID	species identificator
tmpSequence	species sequence (e.g. ABABAABABA)
tmp-Concentration	species initial concentration

tmpDiffusionEnh	Diffusion enhancement degree
tmpSoluble	1 if the species is soluble, 0 otherwise
tmpComplex-DegEnh	complex dissociation kinetic constant
tmpComplex-CuttingPoint	complex cutting point (catalyst-substrate)
tmpEvalueted	This parameter indicates whether the species has been already evalutad (i.e. all the catalysis of the species are instantiated)
tmpVolume	the volume is necessary to convert concentrations in numbers
tmpK_phospho	phosphorilation kinetic constant (in case of energy based simulations)
tmpEnergizable	this is a flag indicating whether or not the species is energizable

Definition at line 99 of file species.cpp.

Here is the call graph for this function:



12.11.2.4 species::species (acs_longInt *tmpID*, string *tmpSequence*, acs_longInt *tmpAmount*, acs_double *tmpDiffusionEnh*, acs_int *tmpSoluble*, acs_double *tmpComplexDegEnh*, acs_int *tmpComplexCuttingPoint*, acs_int *tmpEvalueted*, acs_double *tmpAge*, acs_int *tmpReborns*, acs_double *tmpVolume*, acs_longInt *tmpNotUsedCatID*, acs_longInt *tmpNotUsedSubID*, acs_double *tmpK_phospho*, acs_int *tmpEnergizable*, acs_double *tmpInflux_rate*, acs_int *tmpMaxLOut*)

New species constructor in case of species structure file upload (IN CONCENTRATION)

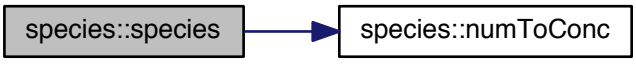
This constructor is used when a new species is uploaded from file (TOTAL AMOUNT BASED)

Parameters

tmpID	species identificator
tmpSequence	species sequence (e.g. ABABAABABA)
tmpAmount	species initial amount of molecules
tmpReactions_-constant	????

Definition at line 138 of file species.cpp.

Here is the call graph for this function:



12.11.2.5 species::species (acs_longInt *tmpID*, string *tmpSequence*, acs_double *tmpConcentration*, acs_double *tmpDiffusionEnh*, acs_int *tmpSoluble*, acs_double *tmpComplexDegEnh*, acs_int *tmpComplexCuttingPoint*, acs_int *tmpEvalueted*, acs_double *tmpAge*, acs_int *tmpReborns*, acs_double *tmpVolume*, acs_longInt *tmpNotUsedCatID*, acs_longInt *tmpNotUsedSubID*, acs_double *tmpK_phospho*, acs_double *tmpKLoadConc*, acs_int *tmpEnergizable*, acs_double *tmpInflux_rate*, acs_int *tmpMaxLOut*)

New random species constructor.

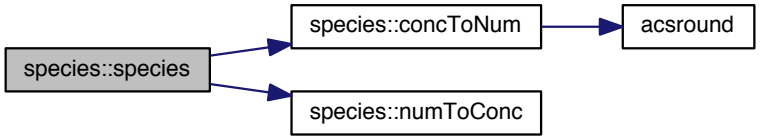
This constructor is used when a new species is uploaded from file (CONCENTRATION BASED)

Parameters

tmpID	species identificator
tmpSequence	species sequence (e.g. ABABAABABA)
tmpAmount	species initial amount of molecules
tmpReactions_constant	????

Definition at line 180 of file species.cpp.

Here is the call graph for this function:



12.11.2.6 species::species (acs_longInt *tmpID*, string *tmpSequence*, acs_longInt *tmpAmount*, acs_double *tmpDiffusionEnh*, acs_int *tmpSoluble*, acs_double *tmpComplexProb*, acs_double *tmpMaxComplexDegKinetic*, MTRand & *tmp_RandomGenerator*, acs_double *tmpVolume*, acs_double *tmpK_phospho*, acs_int *tmpEnergizable*)

new Complex species constructor

This constructor is used when a species is randomly created (!!! NOT USED NOW)

Version

0.1 (8 parameters)

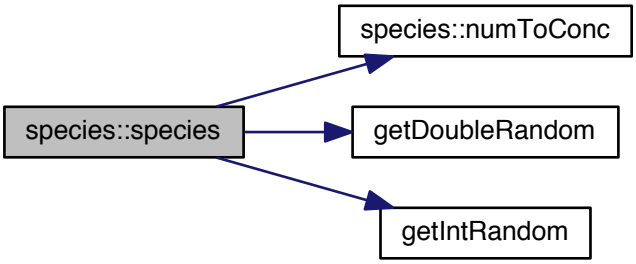
Parameters

tmpID	species identificator
tmpSequence	species sequence (e.g. ABABAABABA)
tmpAmount	species initial amount of molecules
acs_double	tmpDiffusionEnh Diffusione enhancement parameter
acs_double	tmpPrecipitationEnh Precipitation Enhancement parameters
acs_double	tmpComplexProb Probability to be a complex

acs_double	tmpMaxComplexDegKinetic max complex degradation constant
MTRand&	tmp_RandomGenerator random generator

Definition at line 228 of file species.cpp.

Here is the call graph for this function:



12.11.2.7 species::species (acs_longInt *tmpID*, string *tmpSequence*, acs_double *tmpDiffusionEnh*, acs_int *tmpSoluble*, acs_double *tmpMaxComplexDegKinetic*, acs_int *tmpCuttingPoint*, MTRand & *tmp_RandomGenerator*, acs_longInt *tmpCatalyst_ID*, acs_longInt *tmpSubstrate_ID*, acs_double *tmpVolume*, acs_double *tmpK_phospho*, acs_int *tmpEnergizable*)

This constructor is used to create a molecular complex.

Version

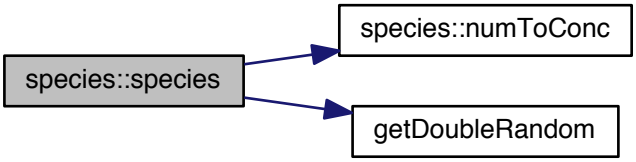
0.1 (10 paramters)

Parameters

tmpID	species identificator
tmpSequence	species sequence (e.g. ABABAABABA)
acs_double	tmpDiffusionEnh Diffusione enhancement parameter
acs_double	tmpPrecipitationEnh Precipitation Enhancement parameters
acs_double	tmpComplexProb Probability to be a complex
acs_double	tmpMaxComplexDegKinetic max complex degradation constant
MTRand&	tmp_RandomGenerator random generator
acs_int	tmpCatalyst_ID Catalyst ID
acs_int	tmpSubstrate_ID substrate ID

Definition at line 274 of file species.cpp.

Here is the call graph for this function:



12.11.2.8 species::~species ( ) [inline]

Definition at line 83 of file species.h.

12.11.3 Member Function Documentation

12.11.3.1 bool species::chargeMol ( )

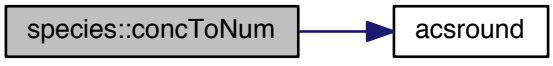
to charge molecules

Definition at line 350 of file species.cpp.

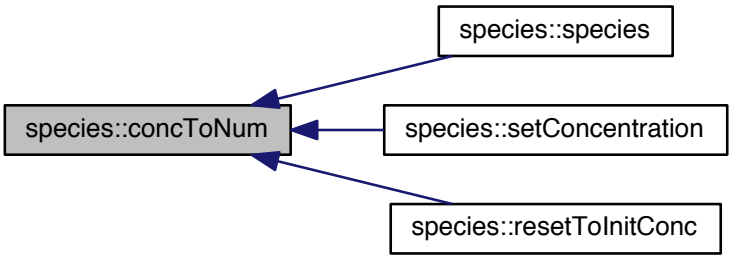
12.11.3.2 void species::concToNum (acs_double *tmpVolume*) [inline]

Definition at line 136 of file species.h.

Here is the call graph for this function:



Here is the caller graph for this function:



12.11.3.3 void species::decrement (acs_double *tmpVolume*)

Function to decrement the total number of molecules belonging to this species

Definition at line 313 of file species.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

12.11.3.4 acs_double species::getAge ( ) const [inline]

Definition at line 94 of file species.h.

12.11.3.5 acs_longInt species::getAmount ( ) const [inline]

Definition at line 89 of file species.h.

Here is the caller graph for this function:

12.11.3.6 vector<acs_longInt> species::getCatalysisIfCpx ( ) const [inline]

Definition at line 110 of file species.h.

12.11.3.7 acs_longInt species::getCatalysisIfCpxID (acs_int *tmpID*) const [inline]

Definition at line 114 of file species.h.

12.11.3.8 acs_longInt species::getCatalyst_ID ( ) const [inline]

Definition at line 101 of file species.h.

12.11.3.9 acs_longInt species::getChargeMols ( ) const [inline]

Definition at line 91 of file species.h.

Here is the caller graph for this function:

12.11.3.10 acs_int species::getComplexCutPnt ( ) const [inline]

Definition at line 99 of file species.h.

12.11.3.11 acs_double species::getComplexDegEnh ( ) const [inline]

Definition at line 98 of file species.h.

12.11.3.12 acs_double species::getConcentration ( ) const [inline]

Definition at line 92 of file species.h.

12.11.3.13 bool species::getConcentrationFixed ( ) const [inline]

Definition at line 105 of file species.h.

12.11.3.14 acs_double species::getDiffusionEnh ( ) const [inline]

Definition at line 96 of file species.h.

12.11.3.15 acs_int species::getEnergizable ( ) const [inline]

Definition at line 104 of file species.h.

12.11.3.16 acs_int species::getEvaluated ( ) const [inline]

Definition at line 100 of file species.h.

12.11.3.17 acs_double species::getFirstConcentration ( ) const [inline]

Definition at line 106 of file species.h.

12.11.3.18 acs_longInt species::getGillIDpos (acs_longInt *tmpID*) const [inline]

Definition at line 145 of file species.h.

12.11.3.19 acs_longInt species::getID ( ) const [inline]

Definition at line 86 of file species.h.

12.11.3.20 acs_double species::getK_phospho ( ) const [inline]

Definition at line 103 of file species.h.

12.11.3.21 acs_int species::getLastSpeciesEvaluated ( ) const [inline]

Definition at line 107 of file species.h.

12.11.3.22 acs_double species::getLoadedConcentration (acs_double *tmpVolume*)

return the concentration of the loaded molecules

Definition at line 377 of file species.cpp.

12.11.3.23 acs_longInt species::getNOTchargeMols ( ) const [inline]

Definition at line 90 of file species.h.

Here is the caller graph for this function:

12.11.3.24 acs_int species::getReborns ( ) const [inline]

Definition at line 95 of file species.h.

12.11.3.25 vector<acs_double> species::getSec_k_SubList ( ) const [inline]

Definition at line 109 of file species.h.

12.11.3.26 acs_double species::getSec_k_SubListID (acs_int *tmpID*) const [inline]

Definition at line 113 of file species.h.

12.11.3.27 vector<acs_longInt> species::getSecSubList ( ) const [inline]

Definition at line 108 of file species.h.

12.11.3.28 acs_longInt species::getSecSubListID (acs_int *tmpID*) const [inline]

Definition at line 112 of file species.h.

12.11.3.29 acs_int species::getSecSubListSize ( ) const [inline]

Definition at line 111 of file species.h.

12.11.3.30 string species::getSequence ( ) const [inline]

Definition at line 87 of file species.h.

Here is the caller graph for this function:

12.11.3.31 acs_int species::getSequenceLength ( ) const [inline]

Definition at line 88 of file species.h.

12.11.3.32 acs_int species::getSolubility ( ) const [inline]

Definition at line 97 of file species.h.

12.11.3.33 acs_longInt species::getSubstrate_ID ( ) const [inline]

Definition at line 102 of file species.h.

12.11.3.34 void species::increment (acs_double *tmpVolume*)

Function to increment the total number of molecules belonging to this species

Definition at line 302 of file species.cpp.

Here is the call graph for this function:

12.11.3.35 void species::insertGillID (acs_longInt *tmpID*) [inline]

Definition at line 144 of file species.h.

12.11.3.36 void species::insertSecSub (acs_longInt *tmpID,* acs_double *tmpK,* acs_longInt *tmpCat*)

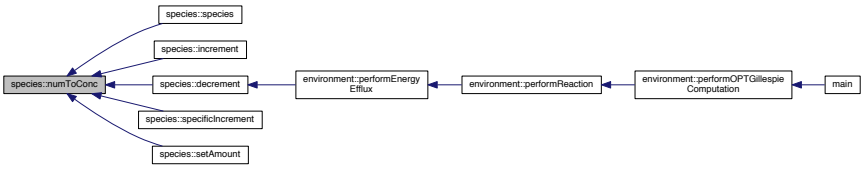
to insert the second substrate - and k reaction - to the species (complexes list)

Definition at line 386 of file species.cpp.

12.11.3.37 void species::numToConc (acs_double *tmpVolume*) [inline]

Definition at line 137 of file species.h.

Here is the caller graph for this function:



12.11.3.38 void species::rebornsIncrement ( ) [inline]

Definition at line 134 of file species.h.

12.11.3.39 void species::resetAge ( ) [inline]

Definition at line 139 of file species.h.

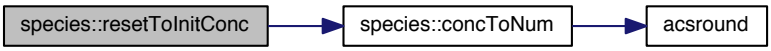
12.11.3.40 void species::resetReborns ( ) [inline]

Definition at line 140 of file species.h.

12.11.3.41 void species::resetToInitConc (acs_double *tmpVolume*) [inline]

Definition at line 141 of file species.h.

Here is the call graph for this function:



12.11.3.42 void species::setAmount (acs_int *tmpAmount,* acs_double *tmpVolume*) [inline]

Definition at line 120 of file species.h.

Here is the call graph for this function:

12.11.3.43 bool species::setChargeMols (acs_int *tmpMolsToCharge*)

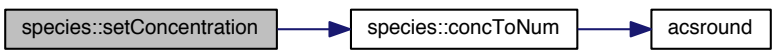
to charge a specific number of molecules

Definition at line 337 of file species.cpp.

12.11.3.44 void species::setConcentration (acs_double *tmpConc,* acs_double *tmpVolume*) [inline]

Definition at line 121 of file species.h.

Here is the call graph for this function:



12.11.3.45 void species::setDiffusion (acs_double *tmpDiff*) [inline]

Definition at line 130 of file species.h.

12.11.3.46 void species::setEvaluated ( ) [inline]

Definition at line 129 of file species.h.

12.11.3.47 void species::setKphospho (acs_double *tmpKphospho*) [inline]

Definition at line 132 of file species.h.

12.11.3.48 void species::setLastSpeciesEvaluated (acs_int *tmpID*) [inline]

Definition at line 142 of file species.h.

12.11.3.49 void species::setNewAge (acs_double *tmpLastTimeInterval*) [inline]

Definition at line 133 of file species.h.

12.11.3.50 void species::setSolubility (acs_int *tmpSol*) [inline]

Definition at line 131 of file species.h.

12.11.3.51 bool species::setSpecificChargeMols (acs_int *tmpMolsToCharge*)

to charge a specific number of molecules

Definition at line 324 of file species.cpp.

12.11.3.52 void species::showGillEngagement ( )

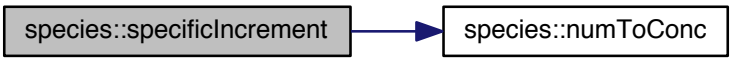
Function to show the species gillespie engagement

Definition at line 406 of file species.cpp.

12.11.3.53 void species::specificIncrement (acs_int *tmpIncrement,* acs_double *tmpVolume*) [inline]

Definition at line 119 of file species.h.

Here is the call graph for this function:



12.11.3.54 bool species::unchargeMol ( )

to uncharge molecules

Definition at line 365 of file species.cpp.

Here is the caller graph for this function:

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

- /Users/alessandrofilisetti/Documents/GIT/carness/species.h
- /Users/alessandrofilisetti/Documents/GIT/carness/species.cpp

Chapter 13

File Documentation

13.1 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/old/allTimesAnalysis.m File Reference

Functions

- params do not prompt figures (0)
- OUT disp ('|--------------------------------|')
- disp ('|-All times analysis...|')

Variables

- function [out]
- params figureVisible = prompt figures (1)
- params deltaT = Delta T
- params totT = total time of the simulation
- currentDir = cd()

13.1.1 Function Documentation

13.1.1.1 disp ('|--------------------------------|')

13.1.1.2 disp ('|-All times analysis...|')

13.1.1.3 params do not prompt figures (0)

13.1.2 Variable Documentation

13.1.2.1 currentDir = cd()

Definition at line 17 of file allTimesAnalysis.m.

13.1.2.2 params deltaT = Delta T

Definition at line 9 of file allTimesAnalysis.m.

13.1.2.3 params figureVisible = prompt figures (1)

Definition at line 8 of file allTimesAnalysis.m.

13.1.2.4 function[out]

Initial value:

```
= allTimesAnalysis(params)
% function [out] = allTimesAnalysis(params)
%

%
% INPUT
%    params.tmpPath = simulations path
```

Definition at line 1 of file allTimesAnalysis.m.

13.1.2.5 params totT = total time of the simulation

Definition at line 10 of file allTimesAnalysis.m.

13.2 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/old/bufferedFluxAnalysis.py File Reference

Namespaces

- bufferedFluxAnalysis

Variables

- list bufferedFluxAnalysis.StrPath = sys.argv[1]
- list bufferedFluxAnalysis.tmpResFold = sys.argv[2]
- tuple bufferedFluxAnalysis.today = dt.date.today()
- string bufferedFluxAnalysis.simF = StrPath+'/'
- tuple bufferedFluxAnalysis.tmpDirs = sort(glob.glob(simF))
- tuple bufferedFluxAnalysis.newdir = os.path.join(os.curdir, '0_statistics')
- tuple bufferedFluxAnalysis.matrixTimeLITE = np.zeros((101,len(tmpDirs)))
- tuple bufferedFluxAnalysis.matrixFluxLITE = np.zeros((101,len(tmpDirs)))
- tuple bufferedFluxAnalysis.matrixAbsLITE = np.zeros((101,len(tmpDirs)))
- tuple bufferedFluxAnalysis.matrixExpLITE = np.zeros((101,len(tmpDirs)))
- int bufferedFluxAnalysis.tmpDirsCnt = 0
- tuple bufferedFluxAnalysis.speciesFiles = sort(glob.glob('species_*'))
- tuple bufferedFluxAnalysis.fidSpecies = open(speciesFiles[0], 'r')
- int bufferedFluxAnalysis.ok = 0
- list bufferedFluxAnalysis.fluxIndexes = []
- list bufferedFluxAnalysis.fluxLengths = []
- tuple bufferedFluxAnalysis.index = int(tmpID)
- tuple bufferedFluxAnalysis.speciesSeq = len(tmpSeq)
- tuple bufferedFluxAnalysis.concFixed = int(tmpConcFixed)
- tuple bufferedFluxAnalysis.fileslist = sort(glob.glob('reactions_parameters_*'))
- int bufferedFluxAnalysis.rctParFileNum = 1
- int bufferedFluxAnalysis.rctID = 1
- list bufferedFluxAnalysis.totFluxDyn = []

- list bufferedFluxAnalysis.absorbedBricks = []
- int bufferedFluxAnalysis.tempAbsorbedBricks = 0
- list bufferedFluxAnalysis.expelledBricks = []
- int bufferedFluxAnalysis.tempExpelledBricks = 0
- list bufferedFluxAnalysis.totTimes = []
- list bufferedFluxAnalysis.totFluxDyn_LITE = []
- list bufferedFluxAnalysis.absorbedBricks_LITE = []
- list bufferedFluxAnalysis.expelledBricks_LITE = []
- list bufferedFluxAnalysis.totTimes_LITE = []
- tuple bufferedFluxAnalysis.fid = open(file, 'r')
- int bufferedFluxAnalysis.okmonitor = 1
- int bufferedFluxAnalysis.oksaveLite = 0
- int bufferedFluxAnalysis.deltaRct = 0
- tuple bufferedFluxAnalysis.reaction = int(tmpReaction)
- tuple bufferedFluxAnalysis.rtime = float(tmpTime)
- tuple bufferedFluxAnalysis.cc = int(tmpcc)
- tuple bufferedFluxAnalysis.cat = int(tmpCat)
- tuple bufferedFluxAnalysis.mol_I = int(tmpMol_I)
- tuple bufferedFluxAnalysis.mol_II = int(tmpMol_II)
- tuple bufferedFluxAnalysis.mol_III = int(tmpMol_III)
- tuple bufferedFluxAnalysis.loadedMolsConc = float(tmpLoadedMolsConc)
- tuple bufferedFluxAnalysis.loadedMols = int(tmpLoadedMols)
- tuple bufferedFluxAnalysis.gillMean = float(tmpGillMean)
- tuple bufferedFluxAnalysis.gillSD = float(tmpGillSD)
- tuple bufferedFluxAnalysis.gillEntropy = float(tmpGillEntropy)
- tuple bufferedFluxAnalysis.savingMatrix = np.zeros((len(totTimes),4))
- tuple bufferedFluxAnalysis.tmpDirSplit = tmpDir.split("/")
- string bufferedFluxAnalysis.cmdFileName = StrPath+'0_statistics/0_fluxDynamics_'
- tuple bufferedFluxAnalysis.cmdFileFid = open(cmdFileName, 'a')

13.3 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/old/concAnalysis.m File Reference

Functions

- ID ho raggiunto la fine del file if isempty (itmp)
- fid,'%s', 1 fscanf ()
- Species sequence concentrazione (indice)
- Dissociation Kinetic Constant bindpnt (indice)
- Binding point evaluated (indice)
- Species Age (in seconds) itmp =fscanf(fid,'%d',1)
- fixed Concentration if (evaluated(indice)==1)&&(bindpnt(indice)
- else speciesLENvec (indice)=0
- Species sequence overallConcMatrix (j, indice)
- Dissociation Kinetic Constant tmpbindpnt (indice)
- Binding point tmpevaluated (indice)
- fixed Concentration if (tmpevaluated(indice)==1)&&(tmpbindpnt(indice)
- coss (j+1)
- angles (j+1)
- overallConcMatrix (end,:)
- cd (strcat(params.tmpPath,'/0_statistics')) = size(overallConcMatrix)
- if exist (fileConcAllName,'file') delete(fileConcAllName)
- fclose (fidC)

- cd (strcat('../', thisSimFolder))
- INIT lengths analysis file (maxL, meanL, medianL) lenghtAnalysis
- lenghtAnalysis (end,:)
- Create vector containing all the coseno (nSpecies files%-1) from both the previous file and the first file coss
- lenghtAnalysis (j,:)
- fclose (fid)
- end disp ('Save concentrations...')
- :,(and(evaluated >0, bindpnt==0 overallConcMatrix ()

Variables

- function [out]
- on each single species file currentDir = cd()
- if nargin< 1params.path=currentDir();params.figureVisible=0;endcd(params.tmpPath);disp('|----------|');disp('|-Concentration Analysis...|');disp('|----------|');if∼isdir('0_statistics') mkdir('0_statistics');end%READ ALL THE DIRECTORY CONTAINING SIMULATIONSsearch=strcat('∗', params.simFolder,'∗');simDirs=dir(search);%For each folder the necessary computations are performedfor i=1:length(simDirs)%Go into the results folderif isdir(strcat(simDirs(i).name)) cd(strcat(simDirs(i).name,'/res'));thisSimFolder=strcat(simDirs(i).name,'/res');%Read configuration fileconfParams=readParameters();%File Containing all Timestimes=0:confParams.timeStructuresSavingInterval:confParams.nSeconds;speciesFiles=dir('species_ ∗');%speciesFiles=speciesFiles(1:end-1);to comment if species_2%does not exist%speciesFiles=speciesFiles(1:length(speciesFiles)-1);%For each species file, from the last to the firstnFile=1;[nSpeciesFile, r]=size(speciesFiles);for j=nSpeciesFile:-1:1%FROM VILLANI MARCOfid=fopen(speciesFiles(j).name,'r');%apro il primo filedisp(sprintf('Processing dir%s, file%s,%d/%d', simDirs(i).name, speciesFiles(j).name, nSpeciesFile-j, nSpeciesFile));%leggo gli oggetti che ci sono in ogni riga-alcuni li memorizzoindice=1;%definisco il parametro di controlo"continua"while indice >
- ID __pad0__
- end
- if nFile
- Total concentration of the species itmp =fscanf(fid,'%d',1)
- Precipitation flag ftmp =fscanf(fid,'%f',1)
- Phosphorilation Kinetic constant
- Phosphorilation Kinetic f
- end indice =indice+1
- else per ora ho memorizzato
  solo nome specie e concentrazione
- if nFile Compute matrix indicators tmpcos = (overallConcMatrix(j,:)∗overallConcMatrix(j+1,:)')/(norm(overallConcMatrix(j,:))∗norm(overallConcMatrix(j+1,:)))
- strZero = zeroBeforeStrNum(i,length(simDirs))
- fileConcAllName = strcat('0_concentrations_ALL_',strZero,int2str(i),'.csv')
- end fidC =fopen(fileConcAllName,'a')
- for a
- speciesLENvec = speciesLENvec(speciesLENvec > 0)
- Create vector containing all the angles = zeros(1,nSpeciesFile)
- cossFromInit = zeros(1,nSpeciesFile)
- anglesFromInit = zeros(1,nSpeciesFile)
- pause
- Clear temp variable used to store species
- Clear temp variable used to
  store binding points and
  evaluation flags for each file
  clear tmpspecie
- clear tmpbindpnt
- clear tmpevaluated
- for k
- overallConcMatrixEval = overallConcMatrix(:,evaluated>0)

13.3.1 Function Documentation

13.3.1.1 Species Age (in *seconds*) =fscanf(fid,'%d',1)

13.3.1.2 angles (j+ *1*)

13.3.1.3 Dissociation Kinetic Constant bindpnt (indice)

13.3.1.4 MOVING INTO THE STATISTIC FOLDER cd (strcat(params.tmpPath,'/0_statistics')) = size(overallConcMatrix)

13.3.1.5 cd (strcat('../', thisSimFolder))

13.3.1.6 Species sequence concentrazione (indice)

13.3.1.7 Create vector containing all the coseno (nSpecies files%- *1*)

13.3.1.8 coss (j+ *1*)

13.3.1.9 end disp ('Save concentrations...')

13.3.1.10 Binding point evaluated (indice)

13.3.1.11 if exist (fileConcAllName , 'file')

13.3.1.12 fclose (fidC)

13.3.1.13 fclose (fid)

13.3.1.14 INIT lengths analysis file (maxL , meanL , medianL)

13.3.1.15 fid s fscanf ( ) [virtual]

13.3.1.16 fixed Concentration if (evaluated(indice) ==1)

13.3.1.17 fixed Concentration if (tmpevaluated(indice) ==1)

13.3.1.18 ID ho raggiunto la fine del file if isempty (itmp)

13.3.1.19 lenghtAnalysis (end , :)

13.3.1.20 lenghtAnalysis (j , :)

13.3.1.21 Species sequence overallConcMatrix (j , indice)

13.3.1.22 overallConcMatrix (end , :)

13.3.1.23 :,(and(evaluated>0,bindpnt==0 overallConcMatrix ( ) [virtual]

13.3.1.24 else speciesLENvec (indice) [pure virtual]

13.3.1.25 Dissociation Kinetic Constant tmpbindpnt (indice)

13.3.1.26 Binding point tmpevaluated (indice)

13.3.2 Variable Documentation

13.3.2.1 ID __pad0__

Definition at line 69 of file concAnalysis.m.

13.3.2.2 for a

Initial value:

```
=1:indice-1
                    fprintf(fidC,'%s \t',specie(a).nome)
```

Definition at line 149 of file concAnalysis.m.

13.3.2.3 angles = zeros(1,nSpeciesFile)

Definition at line 159 of file concAnalysis.m.

13.3.2.4 anglesFromInit = zeros(1,nSpeciesFile)

Definition at line 164 of file concAnalysis.m.

13.3.2.5 else per ora ho memorizzato solo nome specie e concentrazione

Definition at line 101 of file concAnalysis.m.

13.3.2.6 Phosphorilation Kinetic constant

Definition at line 90 of file concAnalysis.m.

13.3.2.7 cossFromInit = zeros(1,nSpeciesFile)

Definition at line 163 of file concAnalysis.m.

13.3.2.8 on each single species file currentDir = cd()

Definition at line 10 of file concAnalysis.m.

13.3.2.9 end

Definition at line 73 of file concAnalysis.m.

13.3.2.10 Phosphorilation Kinetic f

Definition at line 90 of file concAnalysis.m.

13.3.2.11 fidC =fopen(fileConcAllName,'a')

Definition at line 148 of file concAnalysis.m.

13.3.2.12 fileConcAllName = strcat('0_concentrations_ALL_',strZero,int2str(i),'.csv')

Definition at line 144 of file concAnalysis.m.

13.3.2.13 Phosphorilation Kinetic Charged Molecules Concentration ftmp =fscanf(fid,'%f',1)

Definition at line 82 of file concAnalysis.m.

13.3.2.14 function[out]

Initial value:

```
= concAnalysis(params)
% function [out] = concAnalysis(params)
%
% INPUT
% params.tmpPath = tmpPath of the simulations root
%
% This function performs statistics on each single simulation
```

Definition at line 1 of file concAnalysis.m.

13.3.2.15 end indice =indice+1

Definition at line 99 of file concAnalysis.m.

13.3.2.16 Catalyst ID itmp =fscanf(fid,'%d',1)

Definition at line 80 of file concAnalysis.m.

13.3.2.17 end end case favorire quelle corte con una scale free di esponente gamma k

Initial value:

```
= 1 : ss
            fprintf(fidC,' \n')
```

Definition at line 186 of file concAnalysis.m.

13.3.2.18 if nargin< 1params.path=currentDir();params.figureVisible=0;endcd(params.tmpPath);disp('|--------

----------|');disp('|-Concentration Analysis...|');disp('|----------|');if∼isdir('0_-

statistics')mkdir('0_statistics');end%READ ALL THE DIRECTORY CONTAINING SIMULATIONSsearch=strcat('∗',

params.simFolder,'∗');simDirs=dir(search);%For each folder the necessary computations are performedfor

i=1:length(simDirs)%Go into the results folderif isdir(strcat(simDirs(i).name))cd(strcat(simDirs(i).name,'/res'));this-

SimFolder=strcat(simDirs(i).name,'/res');%Read configuration fileconfParams=readParameters();%File Containing

all Timestimes=0:confParams.timeStructuresSavingInterval:confParams.nSeconds;speciesFiles=dir('species_

∗');%speciesFiles=speciesFiles(1:end-1);to comment if species_2%does not exist%speciesFiles=species-

Files(1:length(speciesFiles)-1);%For each species file, from the last to the firstnFile=1;[nSpeciesFile,

r]=size(speciesFiles);for j=nSpeciesFile:-1:1%FROM VILLANI MARCOfid=fopen(speciesFiles(j).name,'r');%apro il

primo filedisp(sprintf('Processing dir%s, file%s,%d/%d', simDirs(i).name, speciesFiles(j).name, nSpeciesFile-j,

nSpeciesFile));%leggo gli oggetti che ci sono in ogni riga-alcuni li memorizzoindice=1;%definisco il parametro di

controlo"continua"while indice >

Initial value:

```
=1 % For each species
                itmp=fscanf(fid,'%d',1)
```

Definition at line 65 of file concAnalysis.m.

13.3.2.19 end if nFile

Initial value:

```
== 1
                    % per ora ho memorizzato solo nome specie e concentrazione
```

Definition at line 75 of file concAnalysis.m.

13.3.2.20 overallConcMatrixEval = overallConcMatrix(:,evaluated>0)

Definition at line 195 of file concAnalysis.m.

13.3.2.21 pause

Definition at line 169 of file concAnalysis.m.

13.3.2.22 Clear temp variable used to store species

Definition at line 172 of file concAnalysis.m.

13.3.2.23 clear speciesLENvec = speciesLENvec(speciesLENvec > 0)

Definition at line 157 of file concAnalysis.m.

13.3.2.24 strZero = zeroBeforeStrNum(i,length(simDirs))

Definition at line 143 of file concAnalysis.m.

13.3.2.25 clear tmpbindpnt

Definition at line 174 of file concAnalysis.m.

13.3.2.26 if nFile Compute matrix indicators tmpcos = (overallConcMatrix(j,:)∗overallConcMatrix(j+1,-:)')/(norm(overallConcMatrix(j,:))∗norm(overallConcMatrix(j+1,:)))

Definition at line 132 of file concAnalysis.m.

13.3.2.27 clear tmpevaluated

Definition at line 174 of file concAnalysis.m.

13.3.2.28 Clear temp variable used to store binding points and evaluation flags for each file clear tmpspecie

Definition at line 172 of file concAnalysis.m.

13.4 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/old/fromWithin2Between.py File Reference

Namespaces

- fromWithin2Between

Functions

- def fromWithin2Between.zeroBeforeStrNum

Variables

- tuple fromWithin2Between.zeroSIM = zeroBeforeStrNum(i,numSim)
- tuple fromWithin2Between.zeroGEN = zeroBeforeStrNum(j,numGen)
- string fromWithin2Between.folderNew = "s_"
- tuple fromWithin2Between.resdir = os.path.join(os.curdir, "res")

13.5 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/old/garbageSearch.m File Reference

Functions

- function garbageSearch (param) currentDir
- cd (param.tmpPath)
- disp ('|----------|')
- disp ('|-Garbage analysis is started|')
- READ ALL THE DIRECTORY
  CONTAINING SIMULATIONS if ∼isdir ('0_statistics') mkdir('0_statistics')
- cd ('0_statistics')
- fclose (fid)
- cd ('../')
- disp (strcat('|-Folder: ', simDirs(x).name,'/res...'))
- if conc && sValues (5)
- speciesMatrix (continua, 2)
- break end reactionsMatrix (continua, 1:6)
- cleavagesMatrix (cleavages, 1:6)
- condensationsMatrix (condensations, 1:6)
- end catalysisMatrix (continua, 1:4)
- garbageMatrix (gRows, 1:2)
- fprintf (fid,'NET%d\n\n', x)
- fprintf (fid,'%f\n\n', garbageMatrix(i, 2))
- end cd (currentDir)

Variables

- end tmpFolder = strcat('∗',param.simFolder,'∗')
- simDirs = dir(tmpFolder)
- fid = fopen('ALL_garbage.txt','w')
- for x
- creating species matrix sFiles = dir('species∗')

- speciesFile = sFiles(length(sFiles)).name
- continua = 1
- speciesMatrix = 0
- while continua index = fscanf(fid,'%d',1)
- stop = isempty(index)
- break end name = fscanf(fid,'%s',1)
- conc = fscanf(fid,'%f',1)
- sValues = fscanf(fid,'%f',12)
- creating reactions matrix rFiles = dir('reactions*')
- reactionsFile = rFiles(length(rFiles)-1).name
- reactionsMatrix = 0
- while continua rValues = fscanf(fid,'%d',7)
- dividing into condensations
  and cleavages condensations = 0
- cleavages = 0
- cleavagesMatrix = 0
- condensationsMatrix = 0
- for i
- creating catalysis matrix cFiles = dir('catalysis*')
- catalysisFile = cFiles(length(cFiles)).name
- catalysisMatrix = 0
- while continua cValues = fscanf(fid,'%f',7)
- break
- identifying garbage [rsm csm] = size(speciesMatrix)
- gRows = 0
- garbageMatrix = 0
- check2 = ismember(speciesMatrix(i,1),condensationsMatrix(:,4:5))
- check3 = ismember(speciesMatrix(i,1),catalysisMatrix(:,2))
- if check1
- else [r c] = size(garbageMatrix)
- clear r
- clear c

13.5.1 Function Documentation

13.5.1.1 end catalysisMatrix (continua , 1:4)

13.5.1.2 end cd (param. *tmpPath*)

13.5.1.3 cd ('0_statistics')

13.5.1.4 cd ('../')

13.5.1.5 end cd (currentDir)

13.5.1.6 cleavagesMatrix (cleavages , 1:6)

13.5.1.7 condensationsMatrix (condensations , 1:6)

13.5.1.8 disp ('|----------')

13.5.1.9 disp ('|-Garbage analysis is started|')

13.5.1.10 disp (strcat('|-Folder: ', simDirs(x).name,'/res...'))

13.5.1.11 fclose (fid)

13.5.1.12 fprintf (fid , 'NET%d\n\n' , x)

13.5.1.13 fprintf (fid , '%f\n\n' , garbageMatrix(i, 2))

13.5.1.14 garbageMatrix (gRows , 1:2)

13.5.1.15 function garbageSearch (param)

13.5.1.16 break end reactionsMatrix (continua , 1:6)

13.5.1.17 speciesMatrix (continua , 2)

13.5.1.18 if conc&& sValues (5)

13.5.1.19 READ ALL THE DIRECTORY CONTAINING SIMULATIONS if ∼isdir ('0_statistics')

13.5.2 Variable Documentation

13.5.2.1 break

Definition at line 98 of file garbageSearch.m.

13.5.2.2 clear c

Initial value:

```
= 1 : length(numScc) % for each ACS (if present)
                                    if numScc(c) > 1 % IF the ACS contains more than one
      species
                                        inSCCFlag = 0
```

Definition at line 138 of file garbageSearch.m.

13.5.2.3 catalysisFile = cFiles(length(cFiles)).name

Definition at line 89 of file garbageSearch.m.

13.5.2.4 catalysisMatrix = 0

Definition at line 92 of file garbageSearch.m.

13.5.2.5 creating catalysis matrix cFiles = dir('catalysis∗')

Definition at line 88 of file garbageSearch.m.

13.5.2.6 if check1

Initial value:

```
== 0 && check2 == 0 && check3 == 0
            gRows = gRows+1
```

Definition at line 113 of file garbageSearch.m.

13.5.2.7 check2 = ismember(speciesMatrix(i,1),condensationsMatrix(:,4:5))

Definition at line 111 of file garbageSearch.m.

13.5.2.8 check3 = ismember(speciesMatrix(i,1),catalysisMatrix(:,2))

Definition at line 112 of file garbageSearch.m.

13.5.2.9 cleavages = 0

Definition at line 72 of file garbageSearch.m.

13.5.2.10 cleavagesMatrix = 0

Definition at line 74 of file garbageSearch.m.

13.5.2.11 conc = fscanf(fid,'%f',1)

Definition at line 42 of file garbageSearch.m.

13.5.2.12 else condensations = 0

Definition at line 71 of file garbageSearch.m.

13.5.2.13 condensationsMatrix = 0

Definition at line 75 of file garbageSearch.m.

13.5.2.14 continua = 1

Definition at line 32 of file garbageSearch.m.

13.5.2.15 while continua cValues = fscanf(fid,'%f',7)

Definition at line 94 of file garbageSearch.m.

13.5.2.16 else[r c] = size(garbageMatrix)

Definition at line 128 of file garbageSearch.m.

13.5.2.17 fid = fopen('ALL_garbage.txt','w')

Definition at line 20 of file garbageSearch.m.

13.5.2.18 identifying garbage[rsm csm] = size(speciesMatrix)

Definition at line 106 of file garbageSearch.m.

13.5.2.19 if garbageMatrix = 0

Definition at line 108 of file garbageSearch.m.

13.5.2.20 gRows = 0

Definition at line 107 of file garbageSearch.m.

13.5.2.21 end ********** T A R T **********for i

Initial value:

```
= 1:rrm
        if reactionsMatrix(i,2) == 1
            cleavages = cleavages+1
```

Definition at line 76 of file garbageSearch.m.

13.5.2.22 else index = fscanf(fid,'%d',1)

Definition at line 35 of file garbageSearch.m.

13.5.2.23 break end name = fscanf(fid,'%s',1)

Definition at line 41 of file garbageSearch.m.

13.5.2.24 clear r

Definition at line 137 of file garbageSearch.m.

13.5.2.25 reactionsFile = rFiles(length(rFiles)-1).name

Definition at line 54 of file garbageSearch.m.

13.5.2.26 reactionsMatrix = 0

Definition at line 57 of file garbageSearch.m.

13.5.2.27 creating reactions matrix rFiles = dir('reactions*')

Definition at line 53 of file garbageSearch.m.

13.5.2.28 while continua rValues = fscanf(fid,'%d',7)

Definition at line 59 of file garbageSearch.m.

13.5.2.29 creating species matrix sFiles = dir('species*')

Definition at line 29 of file garbageSearch.m.

13.5.2.30 simDirs = dir(tmpFolder)

Definition at line 17 of file garbageSearch.m.

13.5.2.31 speciesFile = sFiles(length(sFiles)).name

Definition at line 30 of file garbageSearch.m.

13.5.2.32 speciesMatrix = 0

Definition at line 33 of file garbageSearch.m.

13.5.2.33 if stop = isempty(index)

Definition at line 36 of file garbageSearch.m.

13.5.2.34 sValues = fscanf(fid,'%f',12)

Definition at line 43 of file garbageSearch.m.

13.5.2.35 end tmpFolder = strcat('*',param.simFolder,'*')

Definition at line 16 of file garbageSearch.m.

13.5.2.36 for x

Initial value:

```
= 1:length(simDirs)
    cd(strcat(simDirs(x).name,'/res'))
```

Definition at line 24 of file garbageSearch.m.

13.6 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/old/generalConcentrationOverThreshold.m File Reference

Functions

- cd (params.tmpPath)
- if ∼isdir ('0_statistics') mkdir('0_statistics')
- if exist (outFilenName,'file') delete(outFilenName)
- fprintf (fidFINAL,'\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\n',..
'Reaction Probability','Energy Concentration','Tot over Threshold','Real ACSs','in ACSs',... 'Leaves','by Chain','Overlap','Autocatalysis','in ACSs(W)','Leaves(W)','by Chain(W)','Overlap(W)','Autocatalysis(W)',... '[ACS]','[Leaves]','[Chains]','[Self]','Endo Cond','Cond','Endo Cleavage','Cleavage','Wasted species')
- fclose (fidFINAL)
- For each folder the necessary computations are performed disp ('**********
- disp ('*NEW SIMULATION ANALYSIS PROCESS')
- Go into the results folder cd (strcat(simDirs(IDF).name,'/res'))

- end If there are species over
  theshold network analysis is
  performed disp ('|-Graph Creation')
- if ∼isdir (strcat('../../', folderCat)) mkdir(strcat('../../'
- end if ∼isdir (strcat('../../', folderSub)) mkdir(strcat('../../'
- disp (strcat('|-File ', rcsFiles(rfileID).name, 'processing...'))
- while ischar (rlineb) if ischar(rlineb)%Format lines if isstrprop(rline(end)
- while rline (end)
- end gillTimeSeries (rlineID,:)
- If the time is righe save
  iGraph structures to file disp (sprintf('|-%s|Reaction%d-time:%6.4f-Save structures', simDirs(IDF).name, reaction, rtime))
- saveGraphToFile (folderCat, reaction, rtime, confparams.nReactions, graph, filextPre)
- saveGraphSUBToFile (folderSub, reaction, rtime, confparams.nReactions, graphSUB, filextPre)
- end If the prompt time is
  righe a message on the screen
  indicating the reaction number
  and the time is shown if and ((rtime > rctIDshowNoSave ∗analysisTimeIntervalNoSave),(printTemporalMessage==1)) msg
- disp (msg)
- :, 4 timeInterval ()
- graph (:, 5)
- graphSUB (:, 5)
- end end if (cc==0)||(cc
- end INFLUX ECONOMY ANALYSIS if sum (influx==mol_I) >
- params params Substrate (If different from 1) end else%CAT-> PRO, Otherwise if the reaction is already present its parameters are updatedif sum(and((graph(:, 1)==cat), graph(:, 2)==mol_I))==1position=and((graph(:, 1)==cat), graph(:, 2)==mol_I)
- graph (position,:)
- :, 2 mol_I ()
- graphSUB (position,:)
- :, 2 mol_II ()
- end fclose (fid)
- cd ('../../0_statistics')
- e.g.sim 1 of 10 sims, strZero
  will be '0'in order to create
  a%file named XXX_01_XXX
  strZero=zeroBeforeStrNum(IDF,
  length(simDirs names ()
- grid on set (gca,'fontsize', 15,'fontname','times')
- xlabel ('Time', 'Interpreter', 'latex', 'fontsize', 15)
- ylabel ('Gillespie Mean', 'Interpreter', 'latex', 'fontsize', 15)
- eval (['print-depsc ', fileName])
- saveas (figure1, fileName)
- ylabel ('Gillespie SD', 'Interpreter', 'latex', 'fontsize', 15)
- saveas (figure2, fileName)
- ylabel ('Entropy', 'Interpreter', 'latex', 'fontsize', 15)
- saveas (figure3, fileName)
- ylabel ('New species Probability', 'Interpreter', 'latex', 'fontsize', 15)
- saveas (figure4, fileName)
- ylabel ('Flux Molecules Balance', 'Interpreter', 'latex', 'fontsize', 15)
- saveas (figure5, fileName)
- ylabel ('Ratio of backward reactions', 'Interpreter', 'latex', 'fontsize', 15)
- saveas (figure6, fileName)
- ylabel ('Flux Molecules', 'Interpreter', 'latex', 'fontsize', 15)

- legend ('Added','Removed')
- saveas (figure7, fileName)
- NET ANALYSIS disp ('|-*******NETWORK ANALYSIS *******')
- disp ('|-Strongly connected components analysis... ')
- disp (sprintf('<> SCC n.%d', c))
- end end disp (sprintf('|-Number of ACS:%d', realSccs))
- disp (sprintf('|-Number of ACS(length 1):%d', self))
- disp (sprintf('|-Species over threshold:%d', length(IDsOverThreshold)))
- if ~isempty (IDsOverThreshold) for idot
- if ~isempty (incomingNodes) wasteSpeciesFLAG
- if sum (find(scc==c)==IDsOverThreshold(idot))%If the species belong to the ACS inSCCFlag
- disp (fprintf('\t Within Acs%d->%d#%d-[%d]%d%6.4f', incomingNodes(innode), IDsOverThreshold(idot), weightToDistribute,...IDsOverThreshold(idot), IDsOverThreshold(idot), concVec(idot)))
- disp (fprintf('\t From Acs%d->%d#%d-[%d]%d%6.4f', incomingNodes(innode), IDsOverThreshold(idot), weightToDistribute,...IDsOverThreshold(idot), IDsOverThreshold(idot), concVec(idot)))
- end end end end if self if sum (selfID==idot) > 0 wasteSpeciesFLAG
- end end disp (fprintf('\t\t<> Number of Structural Autocatalytic set of molecules:%d', realSccs))
- disp (fprintf('\t\t<>---'))
- disp (fprintf('\t\t<> Species over threshold produced by a CHAIN:%d', prod_chain))
- if realSccs disp (fprintf('\t\t<> Species over threshold produced INTO an ACS:%d', prod_inSCC))
- disp (fprintf('\t\t<> Species over threshold produced INTO an ACS(weigthed):%d', prod_inSCC_weight))
- disp (fprintf('\t\t<> Species over threshold produced BY an ACS:%d', prod_bySCC))
- disp (fprintf('\t\t<> Species over threshold produced BY an ACS(weigthed):%d', prod_bySCC_weight))
- disp (fprintf('\t\t<> Species over threshold produced by an overlap:%d', prod_overlap))
- disp (fprintf('\t\t<> Species over threshold produced by an overlap(weigthed):%d', prod_overlap_weight))
- disp (fprintf('\t\t<> Species over threshold produced by itself:%d', autocatalysis))
- disp (fprintf('\t\t<> Species over threshold produced by itself(weigthed):%d', self_loop_weight))
- disp (fprintf('\t\t<> Concentration in ACSs:%6.4f', conc_inSCC))
- disp (fprintf('\t\t<> Concentration in ACSs leaves:%6.4f', conc_bySCC))
- disp (fprintf('\t\t<> Concentration in chains:%6.4f', conc_chain))
- disp (fprintf('\t\t<> Concentration of autocatalyst:%6.4f', conc_selfCat))
- disp (fprintf('\t\t<> Number of endo condensations:%6.4f', endo_condensation_counter))
- disp (fprintf('\t\t<> Number of condensations:%6.4f', condensation_counter))
- disp (fprintf('\t\t<> Number of endo cleavages:%6.4f', endo_cleavage_counter))
- disp (fprintf('\t\t<> Number of cleavages:%6.4f', cleavage_counter))
- fprintf (fidFINAL,'\n%s\t%6.4f\t%6.4f\t%6.4f\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%6.-4f\t%6.4f\t%6.4f\t%6.4f\t%d\t%d\t%d\t%d\t%d\t%d',...simDirs(IDF).name, rct, ecc, idOt, realSccs, prod_inSCC, prod_bySCC, prod_chain, prod_overlap, autocatalysis, prod_inSCC_weight,...prod_bySCC_weight, prod_chain_weight, prod_overlap_weight, self_loop_weight, conc_inSCC, conc_bySCC, conc_chain, conc_selfCat, endo_condensation_counter,...condensation_counter, endo_cleavage_counter, cleavage_counter, wasteSpecies)
- disp (fprintf('\n ANALYSIS of the SIMULATION%s IS FINISHED\n', simDirs(IDF).name))
- cd (currentDir)
- tmpl, tmpL strZero ()
- cd ('./res')
- end function saveGraphToFile (tmpDir, tmpRct, tmprTime, tmpRcts, tmpGraph, tmpFilextPre) currentDir
- cd (strcat('../../', tmpDir))
- if exist (outFname,'file') delete(outFname)
- end fclose (fid1)
- end function saveGraphSUBToFile (tmpDir, tmpRct, tmprTime, tmpRcts, tmpGraph, tmpFilextPre) currentDir
- while ischar (tline) tline
- if isequal (tline(1:param-1),'nSeconds') confparams.nSeconds
- elseif isequal (tline(1:param-1),'nGEN') confparams.nGEN
- elseif isequal (tline(1:param-1),'nSIM') confparams.nSIM
- elseif isequal (tline(1:param-1),'nReactions') confparams.nReactions

- elseif isequal (tline(1:param-1),'timeStructuresSavingInterval') confparams.timeStructuresSavingInterval
- elseif isequal (tline(1:param-1),'reactionProbability') confparams.reactionProbability
- elseif isequal (tline(1:param-1),'energy') confparams.energy
- elseif isequal (tline(1:param-1),'ECConcentration') confparams.ECConcentration = str2num(tline(param+1-:length(tline)))
- elseif isequal (tline(1:param-1),'influx_rate') confparams.influx_rate
- elseif isequal (tline(1:param-1),'maxLOut') confparams.maxLOut
- elseif isequal (tline(1:param-1),'volume') confparams.volume
- end end fclose (fidConf)

Variables

- function out
- params threshold = 0
- params decayTime = 100
- params simFolder = 'K_cpx5_rete_n_'
- params tmpRctFileToLoad = ''
- params tmpRctSUBFileToLoad = ''
- params figureVisible = 0
- params interval = 10
- end Set current date and
  current directory currentDate = date()
- currentDir = cd()
- end CREATE HEADER ROW IN THE
  FINAL OUTCOMES FILES outFilenName = strcat(currentDate, '_' , int2str(params.threshold), '_convOver-Threshold_', int2str(params.decayTime), '.csv')
- end fidFINAL = fopen(outFilenName,'w')
- Come back to the original
  folder READ ALL THE DIRECTORY
  CONTAINING SIMULATIONS search = strcat('*',params.simFolder,'*')
- simDirs = dir(search)
- File Containing all Times times = 0:params.deltaT:params.totT
- IDsimFOLDER = 1
- for IDF
- Read file conf confparams = readParameters()
- Select Times and reactions files speciesFiles = dir('species_1*')
- if confparams energy< 2 nrg=1;else
  nrg=0;end%analysisTimeInterval
  is 1/10 of the total decay
  time analysisTimeInterval=params.interval;analysisTimeIntervalNoSave=confparams.nSeconds./params.-decayTime;%load
  incoming flux if
  confparams.influx_rate > influx = loadInflux()'
- folderCat = strcat('__0_iGraph_CAT_', int2str(params.decayTime))
- folderSub = strcat('__0_iGraph_SUB_', int2str(params.decayTime))
- end ID file species counter fileSpeciesID = 1
- Compute species over threshold IDsOverThreshold = tmpID(and(tmpConc>params.threshold,tmpCpx-Cut==0))
- concVec = tmpConc
- reaction parameters files rcsFiles = dir('*reactions_parameters*')
- filextPre = strcat('_',zeroBeforeStrNum(IDsimFOLDER, length(simDirs)),int2str(IDsimFOLDER))
- for rfileID
- condensation_counter = 0
- endo_cleavage_counter = 0
- cleavage_counter = 0

- fid =fopen(rcsFiles(rfileID).name,'r')
- apro il primo file initialize
  useful variables previousTime = 0
- rctIDshow = 1
- rctIDshowNoSave = 1
- rctID = 1
- rline = fgetl(fid)
- rlineb = rline
- rlineID = 1
- FOR EACH REACTION fluxEconomy = 0
- fluxPlus = 0
- fluxMinus = 0
- fluxEconomyArray = []
- while cntrl
- Craete different variables reaction = rline(1)
- rtime = rline(2)
- cc = rline(3)
- cat = rline(4)
- mol_I = rline(5)
- mol_II = rline(6)
- mol_III = rline(7)
- loadedMolsConc = rline(8)
- loadedMols = rline(9)
- gillMean = rline(10)
- gillSD = rline(11)
- gillEntropy = rline(12)
- newSpeciesProb = rline(13)
- ratioBackForward = rline(14)
- end if nrg
- printTemporalMessage = 1
- end update time intervals timeInterval = rtime - previousTime
- graph = graph(graph(:,5)>0,:)
- graphSUB = graphSUB(graphSUB(:,5)>0,:)
- else endo_condensation_counter = endo_condensation_counter + 1
- currentFolder = cd()
- fileName = strcat('25_gilleMean_',strZero,int2str(IDF),'.eps')
- figure1 = gcf
- figure2 = gcf
- figure3 = gcf
- figure4 = gcf
- figure5 = gcf
- figure6 = gcf
- figure7 = gcf
- SAVE THE GILLESPIE MATRXI ON FILE filename = strcat('0_gillespie_',strZero,int2str(IDF),'.txt')
- fluxEconomyMatrix = [gillTimeSeries(:,1), fluxEconomyArray]
- Computing actual number od
  strongly connected components from = graph(:,1)+1
- to = graph(:,2)+1
- prod_inSCC = 0
- prod_chain = 0
- prod_bySCC = 0
- prod_overlap = 0
- sccID = 0
- autocatalysis = 0
- prod_inSCC_weight = 0

- prod_chain_weight = 0
- prod_bySCC_weight = 0
- prod_overlap_weight = 0
- self_loop_weight = 0
- conc_inSCC = 0
- conc_chain = 0
- conc_bySCC = 0
- conc_selfCat = 0
- wasteSpecies = 0
- catSparse = sparse(from,to,true,max(max(from,to)),max(max(from,to)))
- numScc = histc(scc,1:max(scc))
- realSccs = 0
- for c
- alreadyAdded_leaves = 0
- alreadyAdded_chain = 0
- tmpProd_chain = 0
- incomingNodes = graph(graph(:,2)==IDsOverThreshold(idot),1)
- tempProd_chain_weight = 0
- wasteSpeciesFLAG = 0
- for innode
- noInAcs = 1
- if alreadyAdded_ACS
- end end Reactions to
  distribuite in the different
  nature of the reactions weightToDistribute = graph(and((graph(:,1) == incomingNodes(innode)),(graph(:,2) == IDsOverThreshold(idot))),6)
- if inSCCFlag == 1 % If the node is in an ACS
- else Otherwise it has been
  produced by an ACS
- end end If the species
  concentration but the species
  is not produced by other species
- rct = confparams.reactionProbability
- ecc = confparams.ECConcentration
- idOt = length(IDsOverThreshold)
- clear nrgTimeSeries
- clear gillTimeSeries = [rtime,gillMean,gillSD,gillEntropy,newSpeciesProb]
- nZeros = length(num2str(tmpL)) - length(num2str(tmpl))
- if nZeros for p
- tmpStrZeros = zeroBeforeStrNum(tmpRct, tmpRcts)
- outFname = strcat('_iGraph_CAT',tmpFilextPre,'_',tmpStrZeros,num2str(tmpRct),'_',num2str(tmprTime),'.csv')
- end fid1 = fopen(outFname,'a')
- for j
- end function [N, ids]
- ids = graph(graph(:,1)==graph(:,2),1)
- tline = fgets(fidConf)
- param = findstr(tline,'=')

13.6.1 Function Documentation

13.6.1.1 end If the prompt time is righe a message on the screen indicating the reaction number and the time is shown if and ((rtime > rctIDshowNoSave ∗analysisTimeIntervalNoSave) , (printTemporalMessage==1))

13.6.1.2 cd (params. *tmpPath*)

13.6.1.3 Go into the results folder cd (strcat(simDirs(IDF).name,'/res'))

13.6.1.4 cd ('../../0_statistics')

13.6.1.5 cd (currentDir)

13.6.1.6 cd ('./res')

13.6.1.7 cd (strcat('../../', tmpDir))

13.6.1.8 For each folder the necessary computations are performed disp ('**********')

13.6.1.9 disp ('∗NEW SIMULATION ANALYSIS PROCESS')

13.6.1.10 end If there are species over theshold network analysis is performed disp ('|-Graph Creation')

13.6.1.11 disp (strcat('|-File ', rcsFiles(rfileID).name, 'processing...'))

13.6.1.12 If the time is righe save iGraph structures to file disp (sprintf('|-%s|Reaction%d-time:%6.4f-Save structures', simDirs(IDF).name, reaction, rtime))

13.6.1.13 disp (msg)

13.6.1.14 NET ANALYSIS disp ('|-∗∗∗∗∗∗∗NETWORK ANALYSIS ∗∗∗∗∗∗∗')

13.6.1.15 disp ('|-Strongly connected components analysis... ')

13.6.1.16 disp (sprintf('<> SCC n.%d', c))

13.6.1.17 end end disp (sprintf('|-Number of ACS:%d', realSccs))

13.6.1.18 disp (sprintf('|-Number of ACS(length 1):%d', self))

13.6.1.19 disp (sprintf('|-Species over threshold:%d', length(IDsOverThreshold)))

13.6.1.20 disp (fprintf('\t Within Acs%d->%d#%d-[%d]%d%6.4f', incomingNodes(innode), IDsOverThreshold(idot), weightToDistribute,...IDsOverThreshold(idot), IDsOverThreshold(idot), concVec(idot)))

13.6.1.21 disp (fprintf('\t From Acs%d->%d#%d-[%d]%d%6.4f', incomingNodes(innode), IDsOverThreshold(idot), weightToDistribute,...IDsOverThreshold(idot), IDsOverThreshold(idot), concVec(idot)))

13.6.1.22 end end disp (fprintf('\t\t<> Number of Structural Autocatalytic set of molecules:%d', realSccs))

13.6.1.23 end disp (fprintf('\t\t<>----------'))

13.6.1.24 disp (fprintf('\t\t<> Species over threshold produced by a CHAIN:%d', prod_chain))

13.6.1.25 if realSccs disp (fprintf('\t\t<> Species over threshold produced INTO an ACS:%d', prod_inSCC))

13.6.1.26 disp (fprintf('\t\t<> Species over threshold produced INTO an ACS(weigthed):%d', prod_inSCC_weight))

13.6.1.27 disp (fprintf('\t\t<> Species over threshold produced BY an ACS:%d', prod_bySCC))

13.6.1.28 disp (fprintf('\t\t<> Species over threshold produced BY an ACS(weigthed):%d', prod_bySCC_weight))

13.6.1.29 disp (fprintf('\t\t<> Species over threshold produced by an overlap:%d', prod_overlap))

13.6.1.30 disp (fprintf('\t\t<> Species over threshold produced by an overlap(weigthed):%d', prod_overlap_weight))

13.6.1.31 disp (fprintf('\t\t<> Species over threshold produced by itself:%d', autocatalysis))

13.6.1.32 disp (fprintf('\t\t<> Species over threshold produced by itself(weigthed):%d', self_loop_weight))

13.6.1.33 disp (fprintf('\t\t<> Concentration in ACSs:%6.4f', conc_inSCC))

13.6.1.34 disp (fprintf('\t\t<> Concentration in ACSs leaves:%6.4f', conc_bySCC))

13.6.1.35 disp (fprintf('\t\t<> Concentration in chains:%6.4f', conc_chain))

13.6.1.36 disp (fprintf('\t\t<> Concentration of autocatalyst:%6.4f', conc_selfCat))

13.6.1.37 disp (fprintf('\t\t<> Number of endo condensations:%6.4f', endo_condensation_counter))

13.6.1.38 disp (fprintf('\t\t<> Number of condensations:%6.4f', condensation_counter))

13.6.1.39 disp (fprintf('\t\t<> Number of endo cleavages:%6.4f', endo_cleavage_counter))

13.6.1.40 disp (fprintf('\t\t<> Number of cleavages:%6.4f', cleavage_counter))

13.6.1.41 disp (fprintf('\n ANALYSIS of the SIMULATION%s IS FINISHED\n', simDirs(IDF).name))

13.6.1.42 eval ( )

13.6.1.43 if exist (outFilenName , 'file')

13.6.1.44 if exist (outFname , 'file')

13.6.1.45 fclose (fidFINAL)

13.6.1.46 end fclose (fid)

13.6.1.47 end fclose (fid1)

13.6.1.48 end end fclose (fidConf)

13.6.1.49 fprintf (fidFINAL , '\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\n'
, ... 'Reaction Probability' , 'Energy Concentration' , 'Tot over Threshold' , 'Real ACSs' , 'in ACSs' , ... 'Leaves' , 'by
Chain' , 'Overlap' , 'Autocatalysis' , 'in ACSs(W)' , 'Leaves(W)' , 'by Chain(W)' , 'Overlap(W)' , 'Autocatalysis(W)' , ...
" *[ACS],* " *[Leaves],* " *[Chains],* " *[Self],* 'Endo Cond' , 'Cond' , 'Endo Cleavage' , 'Cleavage' , 'Wasted species')

13.6.1.50 fprintf (fidFINAL , '\n%s\t%6.4f\t%6.4f\t%6.4f\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%6.-4f\t%6.4f\t%6.4f\t%6.4f\t%d\t%d\t%d\t%d\t%d' , ...simDirs(IDF). *name,* rct , ecc , idOt , realSccs , prod_inSCC , prod_bySCC , prod_chain , prod_overlap , autocatalysis , prod_inSCC_weight , *prod_bySCC_weight,* prod_chain_weight , prod_overlap_weight , self_loop_weight , conc_inSCC , conc_bySCC , conc_chain , conc_selfCat , endo_condensation_counter , *condensation_counter,* endo_cleavage_counter , cleavage_counter , wasteSpecies)

13.6.1.51 end gillTimeSeries (rlineID , :)

13.6.1.52 graph (: , 5)

13.6.1.53 graph (position , :)

13.6.1.54 graphSUB (: , 5)

13.6.1.55 graphSUB (position , :)

13.6.1.56 end end if (cc ==0)

13.6.1.57 while ischar (rlineb)

13.6.1.58 while ischar (tline)

13.6.1.59 if isequal (tline(1:param-1) , 'nSeconds')

13.6.1.60 elseif isequal (tline(1:param-1) , 'nGEN')

13.6.1.61 elseif isequal (tline(1:param-1) , 'nSIM')

13.6.1.62 elseif isequal (tline(1:param-1) , 'nReactions')

13.6.1.63 elseif isequal (tline(1:param-1) , 'timeStructuresSavingInterval')

13.6.1.64 elseif isequal (tline(1:param-1) , 'reactionProbability')

13.6.1.65 elseif isequal (tline(1:param-1) , 'energy')

13.6.1.66 elseif isequal (tline(1:param-1) , 'ECConcentration') = str2num(tline(param+1:length(tline)))

13.6.1.67 elseif isequal (tline(1:param-1) , 'influx_rate')

13.6.1.68 elseif isequal (tline(1:param-1) , 'maxLOut')

13.6.1.69 elseif isequal (tline(1:param-1) , 'volume')

13.6.1.70 legend ('Added' , 'Removed')

13.6.1.71 :,2 mol_I ( ) [virtual]

13.6.1.72 :,2 mol_II ( ) [virtual]

13.6.1.73 e.g. sim 1 of 10 sims, strZero will be '0' in order to create a % file named XXX_01_XXX strZero = zeroBeforeStrNum(IDF,length(simDirs names ( ) [virtual]

13.6.1.74 while rline (end)

13.6.1.75 saveas (figure1 , fileName)

13.6.1.76 saveas (figure2 , fileName)

13.6.1.77 saveas (figure3 , fileName)

13.6.1.78 saveas (figure4 , fileName)

13.6.1.79 saveas (figure5 , fileName)

13.6.1.80 saveas (figure6 , fileName)

13.6.1.81 saveas (figure7 , fileName)

13.6.1.82 saveGraphSUBToFile (folderSub , reaction , rtime , confparams. *nReactions,* graphSUB , filextPre)

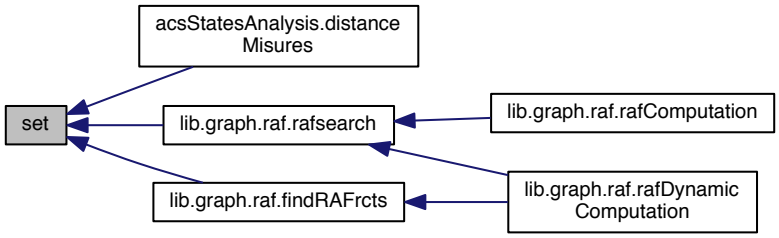
13.6.1.83 **end function** saveGraphSUBToFile (tmpDir , tmpRct , tmprTime , tmpRcts , tmpGraph , tmpFilextPre)

13.6.1.84 saveGraphToFile (folderCat , reaction , rtime , confparams. *nReactions,* graph , filextPre)

13.6.1.85 **end function** saveGraphToFile (tmpDir , tmpRct , tmprTime , tmpRcts , tmpGraph , tmpFilextPre)

13.6.1.86 grid on set (gca , 'fontsize' , 15 , 'fontname' , 'times')

Here is the caller graph for this function:



13.6.1.87 tmpl, tmpL strZero ( ) [virtual]

13.6.1.88 params params Substrate (If different from *1*) -> PRO, Otherwise if the reaction is already present its parameters are updatedif sum(and((graph(:, 1)==cat), graph(:, 2)==mol_l))==1position=and((graph(:, 1)==cat), graph(:, 2)==mol_l)

13.6.1.89 **end if** sum (influx *=*=mol_l)

Initial value:

```
= 1
                        fluxEconomy = fluxEconomy + 1
```

Here is the caller graph for this function:

13.6.1.90 if sum (find(scc==c) ==IDsOverThreshold (idot))

13.6.1.91 end end end end if self if sum (selfID ==idot)

13.6.1.92 :,4 timeInterval ( ) [virtual]

13.6.1.93 xlabel ('Time' , 'Interpreter' , 'latex' , 'fontsize' , 15)

13.6.1.94 ylabel ('Gillespie Mean' , 'Interpreter' , 'latex' , 'fontsize' , 15)

13.6.1.95 ylabel ('Gillespie SD' , 'Interpreter' , 'latex' , 'fontsize' , 15)

13.6.1.96 ylabel ('Entropy' , 'Interpreter' , 'latex' , 'fontsize' , 15)

13.6.1.97 ylabel ('New species Probability' , 'Interpreter' , 'latex' , 'fontsize' , 15)

13.6.1.98 ylabel ('Flux Molecules Balance' , 'Interpreter' , 'latex' , 'fontsize' , 15)

13.6.1.99 ylabel ('Ratio of backward reactions' , 'Interpreter' , 'latex' , 'fontsize' , 15)

13.6.1.100 ylabel ('Flux Molecules' , 'Interpreter' , 'latex' , 'fontsize' , 15)

13.6.1.101 if ∼isdir ('0_statistics')

13.6.1.102 if ∼isdir (strcat('../../', folderCat))

13.6.1.103 end if ∼isdir (strcat('../../', folderSub))

13.6.1.104 if ∼isempty (IDsOverThreshold)

13.6.1.105 if ∼isempty (incomingNodes)

13.6.2 Variable Documentation

13.6.2.1 else Otherwise it has been produced by an ACS

Definition at line 567 of file generalConcentrationOverThreshold.m.

13.6.2.2 alreadyAdded_ACS

Initial value:

```
== 0 % Add concentration to conc of ACSs
                                                conc_inSCC =
      conc_inSCC + concVec(idot)
```

Definition at line 551 of file generalConcentrationOverThreshold.m.

13.6.2.3 if alreadyAdded_chain = 0

Definition at line 533 of file generalConcentrationOverThreshold.m.

13.6.2.4 else Otherwise it has been produced by an so it is a first layer leaf if alreadyAdded_leaves = 0

Definition at line 532 of file generalConcentrationOverThreshold.m.

13.6.2.5 autocatalysis = 0

Definition at line 501 of file generalConcentrationOverThreshold.m.

13.6.2.6 if realSccs If there are ACS for c

Initial value:

```
= 1 : length(numScc)
                if numScc(c) > 1
                    sccID = sccID + 1
```

Definition at line 518 of file generalConcentrationOverThreshold.m.

13.6.2.7 cat = rline(4)

Definition at line 162 of file generalConcentrationOverThreshold.m.

13.6.2.8 catSparse = sparse(from,to,true,max(max(from,to)),max(max(from,to)))

Definition at line 513 of file generalConcentrationOverThreshold.m.

13.6.2.9 params params end end end else if cc = rline(3)

Definition at line 161 of file generalConcentrationOverThreshold.m.

13.6.2.10 else cleavage_counter = 0

Definition at line 131 of file generalConcentrationOverThreshold.m.

13.6.2.11 while cntrl

Definition at line 156 of file generalConcentrationOverThreshold.m.

13.6.2.12 conc_bySCC = 0

Definition at line 509 of file generalConcentrationOverThreshold.m.

13.6.2.13 conc_chain = 0

Definition at line 508 of file generalConcentrationOverThreshold.m.

13.6.2.14 conc_inSCC = 0

Definition at line 507 of file generalConcentrationOverThreshold.m.

13.6.2.15 conc_selfCat = 0

Definition at line 510 of file generalConcentrationOverThreshold.m.

13.6.2.16 concVec = tmpConc

Definition at line 117 of file generalConcentrationOverThreshold.m.

13.6.2.17 condensation_counter = 0

Definition at line 129 of file generalConcentrationOverThreshold.m.

13.6.2.18 end function confparams = readParameters()

Definition at line 71 of file generalConcentrationOverThreshold.m.

13.6.2.19 end Set current date and current directory currentDate = date()

Definition at line 21 of file generalConcentrationOverThreshold.m.

13.6.2.20 currentDir = cd()

Definition at line 22 of file generalConcentrationOverThreshold.m.

13.6.2.21 currentFolder = cd()

Definition at line 352 of file generalConcentrationOverThreshold.m.

13.6.2.22 params params decayTime = 100

Definition at line 12 of file generalConcentrationOverThreshold.m.

13.6.2.23 ecc = confparams.ECConcentration

Definition at line 640 of file generalConcentrationOverThreshold.m.

13.6.2.24 endo_cleavage_counter = 0

Definition at line 130 of file generalConcentrationOverThreshold.m.

13.6.2.25 else endo_condensation_counter = endo_condensation_counter + 1

Definition at line 227 of file generalConcentrationOverThreshold.m.

13.6.2.26 fid =fopen(rcsFiles(rfileID).name,'r')

Definition at line 135 of file generalConcentrationOverThreshold.m.

13.6.2.27 end fid1 = fopen(outFname,'a')

Definition at line 696 of file generalConcentrationOverThreshold.m.

13.6.2.28 end fidFINAL = fopen(outFilenName,'w')

Definition at line 38 of file generalConcentrationOverThreshold.m.

13.6.2.29 figure1 = gcf

Definition at line 374 of file generalConcentrationOverThreshold.m.

13.6.2.30 figure2 = gcf

Definition at line 390 of file generalConcentrationOverThreshold.m.

13.6.2.31 figure3 = gcf

Definition at line 406 of file generalConcentrationOverThreshold.m.

13.6.2.32 figure4 = gcf

Definition at line 422 of file generalConcentrationOverThreshold.m.

13.6.2.33 figure5 = gcf

Definition at line 438 of file generalConcentrationOverThreshold.m.

13.6.2.34 figure6 = gcf

Definition at line 454 of file generalConcentrationOverThreshold.m.

13.6.2.35 figure7 = gcf

Definition at line 471 of file generalConcentrationOverThreshold.m.

13.6.2.36 if params figureVisible = 0

Definition at line 16 of file generalConcentrationOverThreshold.m.

13.6.2.37 fileName = strcat('25_gilleMean_',strZero,int2str(IDF),'.eps')

Definition at line 372 of file generalConcentrationOverThreshold.m.

13.6.2.38 SAVE THE GILLESPIE MATRXI ON FILE filename = strcat('0_gillespie_',strZero,int2str(IDF),'.txt')

Definition at line 476 of file generalConcentrationOverThreshold.m.

13.6.2.39 fileSpeciesID = 1

Definition at line 107 of file generalConcentrationOverThreshold.m.

13.6.2.40 filextPre = strcat('_',zeroBeforeStrNum(IDsimFOLDER, length(simDirs)),int2str(IDsimFOLDER))

Definition at line 122 of file generalConcentrationOverThreshold.m.

13.6.2.41 fluxEconomy = 0

Definition at line 148 of file generalConcentrationOverThreshold.m.

13.6.2.42 params params end end end end fluxEconomyArray = [ ]

Definition at line 151 of file generalConcentrationOverThreshold.m.

13.6.2.43 fluxEconomyMatrix = [gillTimeSeries(:,1), fluxEconomyArray]

Definition at line 479 of file generalConcentrationOverThreshold.m.

13.6.2.44 fluxMinus = 0

Definition at line 150 of file generalConcentrationOverThreshold.m.

13.6.2.45 fluxPlus = 0

Definition at line 149 of file generalConcentrationOverThreshold.m.

13.6.2.46 if folderCat = strcat('__0_iGraph_CAT_', int2str(params.decayTime))

Definition at line 95 of file generalConcentrationOverThreshold.m.

13.6.2.47 end if folderSub = strcat('__0_iGraph_SUB_', int2str(params.decayTime))

Definition at line 96 of file generalConcentrationOverThreshold.m.

13.6.2.48 Computing actual number od strongly connected components from = graph(:,1)+1

Definition at line 493 of file generalConcentrationOverThreshold.m.

13.6.2.49 end function[N, ids]

Initial value:

```
= self_loops(graph)
    %Compute self-loop
    N = sum(graph(:,1)==graph(:,2))
```

Definition at line 722 of file generalConcentrationOverThreshold.m.

13.6.2.50 gillEntropy = rline(12)

Definition at line 170 of file generalConcentrationOverThreshold.m.

13.6.2.51 gillMean = rline(10)

Definition at line 168 of file generalConcentrationOverThreshold.m.

13.6.2.52 gillSD = rline(11)

Definition at line 169 of file generalConcentrationOverThreshold.m.

13.6.2.53 gillTimeSeries = [rtime,gillMean,gillSD,gillEntropy,newSpeciesProb]

Definition at line 649 of file generalConcentrationOverThreshold.m.

13.6.2.54 else graph = graph(graph(:,5)>0,:)

Definition at line 214 of file generalConcentrationOverThreshold.m.

13.6.2.55 else graphSUB = graphSUB(graphSUB(:,5)>0,:)

Definition at line 220 of file generalConcentrationOverThreshold.m.

13.6.2.56 for IDF

Initial value:

```
=1:length(simDirs)

    if isdir(strcat(simDirs(IDF).name))

        disp(sprintf('|- Processing dir %s',simDirs(IDF).name))
```

Definition at line 61 of file generalConcentrationOverThreshold.m.

13.6.2.57 idOt = length(IDsOverThreshold)

Definition at line 641 of file generalConcentrationOverThreshold.m.

13.6.2.58 ids = graph(graph(:,1)==graph(:,2),1)

Definition at line 725 of file generalConcentrationOverThreshold.m.

13.6.2.59 IDsimFOLDER = 1

Definition at line 59 of file generalConcentrationOverThreshold.m.

13.6.2.60 Compute species over threshold IDsOverThreshold = tmpID(and(tmpConc>params.threshold,tmpCpx-Cut==0))

Definition at line 116 of file generalConcentrationOverThreshold.m.

13.6.2.61 incomingNodes = graph(graph(:,2)==IDsOverThreshold(idot),1)

Definition at line 535 of file generalConcentrationOverThreshold.m.

13.6.2.62 controllo che non ci siano cicli nell influx = loadInflux()'

Definition at line 88 of file generalConcentrationOverThreshold.m.

13.6.2.63 for innode

Initial value:

```
= 1 : length(incomingNodes)
                            alreadyAdded_ACS = 0
```

Definition at line 541 of file generalConcentrationOverThreshold.m.

13.6.2.64 if inSCCFlag == 1 % If the node is in an ACS

Definition at line 561 of file generalConcentrationOverThreshold.m.

13.6.2.65 params interval = 10

Definition at line 17 of file generalConcentrationOverThreshold.m.

13.6.2.66 clear j

Initial value:

```
= 1 : r
        fprintf(fid1,'%d\t%d\t%d\t%6.4f\t%6.4f\t%d\n',tmpGraph(j,1),tmpGraph(
      j,2),tmpGraph(j,3),tmpGraph(j,4),tmpGraph(j,5),tmpGraph(j,6))
```

Definition at line 698 of file generalConcentrationOverThreshold.m.

13.6.2.67 loadedMols = rline(9)

Definition at line 167 of file generalConcentrationOverThreshold.m.

13.6.2.68 loadedMolsConc = rline(8)

Definition at line 166 of file generalConcentrationOverThreshold.m.

13.6.2.69 mol_I = rline(5)

Definition at line 163 of file generalConcentrationOverThreshold.m.

13.6.2.70 params params end if mol_II = rline(6)

Definition at line 164 of file generalConcentrationOverThreshold.m.

13.6.2.71 mol_III = rline(7)

Definition at line 165 of file generalConcentrationOverThreshold.m.

13.6.2.72 newSpeciesProb = rline(13)

Definition at line 171 of file generalConcentrationOverThreshold.m.

13.6.2.73 end end end end end If both the species over threshold and the incoming node are not belonging to an ACS if noInAcs = 1

Definition at line 544 of file generalConcentrationOverThreshold.m.

13.6.2.74 end if nrg

Initial value:

```
== 1
                    nrgTimeSeries(rlineID,:) = [rtime,loadedMolsConc,loadedMols]
```

Definition at line 175 of file generalConcentrationOverThreshold.m.

13.6.2.75 clear nrgTimeSeries

Definition at line 648 of file generalConcentrationOverThreshold.m.

13.6.2.76 numScc = histc(scc,1:max(scc))

Definition at line 515 of file generalConcentrationOverThreshold.m.

13.6.2.77 nZeros = length(num2str(tmpL)) - length(num2str(tmpl))

Definition at line 673 of file generalConcentrationOverThreshold.m.

13.6.2.78 Come back to the original folder end end out

Initial value:

```
= generalConcentrationOverThreshold(params)

% param.threshold              # Theshold above which the concentration has
      to be
% param.decayTime              # Reaction decay time
% param.simFolder              # Subtring identifycating the folders containing sims
% param.tmpRctFileToLoad       # Reaction file to load
% param.tmpRctSUBFileToLoad    # Reaction SUB file to load
```

```
if nargin < 1
    params.tmpPath = '~/Documents/simChiara/variaK_cpx/K_cpx_05'
```

Definition at line 1 of file generalConcentrationOverThreshold.m.

13.6.2.79 end CREATE HEADER ROW IN THE FINAL OUTCOMES FILES outFilenName = strcat(currentDate, '_' , int2str(params.threshold), '_convOverThreshold_', int2str(params.decayTime), '.csv')

Definition at line 34 of file generalConcentrationOverThreshold.m.

13.6.2.80 outFname = strcat('_iGraph_CAT',tmpFilextPre,'_',tmpStrZeros,num2str(tmpRct),'_',num2str(tmprTime),'.csv')

Definition at line 692 of file generalConcentrationOverThreshold.m.

13.6.2.81 if nZeros for p

Initial value:

```
=1:nZeros
            strZero = strcat(strZero,'0')
```

Definition at line 675 of file generalConcentrationOverThreshold.m.

13.6.2.82 param = findstr(tline,'=')

Definition at line 736 of file generalConcentrationOverThreshold.m.

13.6.2.83 previousTime = 0

Definition at line 138 of file generalConcentrationOverThreshold.m.

13.6.2.84 printTemporalMessage = 1

Definition at line 182 of file generalConcentrationOverThreshold.m.

13.6.2.85 end prod_bySCC = 0

Definition at line 498 of file generalConcentrationOverThreshold.m.

13.6.2.86 prod_bySCC_weight = 0

Definition at line 504 of file generalConcentrationOverThreshold.m.

13.6.2.87 prod_chain = 0

Definition at line 497 of file generalConcentrationOverThreshold.m.

13.6.2.88 prod_chain_weight = 0

Definition at line 503 of file generalConcentrationOverThreshold.m.

13.6.2.89 if then it has been produced within an ACS prod_inSCC = 0

Definition at line 496 of file generalConcentrationOverThreshold.m.

13.6.2.90 prod_inSCC_weight = 0

Definition at line 502 of file generalConcentrationOverThreshold.m.

13.6.2.91 end Compute the overlap between the different counter prod_overlap = 0

Definition at line 499 of file generalConcentrationOverThreshold.m.

13.6.2.92 prod_overlap_weight = 0

Definition at line 505 of file generalConcentrationOverThreshold.m.

13.6.2.93 ratioBackForward = rline(14)

Definition at line 172 of file generalConcentrationOverThreshold.m.

13.6.2.94 reaction parameters files rcsFiles = dir('∗reactions_parameters∗')

Definition at line 120 of file generalConcentrationOverThreshold.m.

13.6.2.95 rct = confparams.reactionProbability

Definition at line 639 of file generalConcentrationOverThreshold.m.

13.6.2.96 rctID = 1

Definition at line 141 of file generalConcentrationOverThreshold.m.

13.6.2.97 rctIDshow = 1

Definition at line 139 of file generalConcentrationOverThreshold.m.

13.6.2.98 rctIDshowNoSave = 1

Definition at line 140 of file generalConcentrationOverThreshold.m.

13.6.2.99 Craete different variables reaction = rline(1)

Definition at line 159 of file generalConcentrationOverThreshold.m.

13.6.2.100 realSccs = 0

Definition at line 517 of file generalConcentrationOverThreshold.m.

13.6.2.101 for rfileID

Initial value:

```
=1:length(rcsFiles)
            % read species file

            %Initialize reaction type counter
            endo_condensation_counter = 0
```

Definition at line 124 of file generalConcentrationOverThreshold.m.

13.6.2.102 end rline = fgetl(fid)

Definition at line 143 of file generalConcentrationOverThreshold.m.

13.6.2.103 rlineb = rline

Definition at line 144 of file generalConcentrationOverThreshold.m.

13.6.2.104 Update rlineID rlineID = 1

Definition at line 145 of file generalConcentrationOverThreshold.m.

13.6.2.105 rtime = rline(2)

Definition at line 160 of file generalConcentrationOverThreshold.m.

13.6.2.106 sccID = 0

Definition at line 500 of file generalConcentrationOverThreshold.m.

13.6.2.107 Come back to the original folder READ ALL THE DIRECTORY CONTAINING SIMULATIONS search = strcat('*',params.simFolder,'*')

Definition at line 48 of file generalConcentrationOverThreshold.m.

13.6.2.108 self_loop_weight = 0

Definition at line 506 of file generalConcentrationOverThreshold.m.

13.6.2.109 simDirs = dir(search)

Definition at line 49 of file generalConcentrationOverThreshold.m.

13.6.2.110 params simFolder = 'K_cpx5_rete_n_'

Definition at line 13 of file generalConcentrationOverThreshold.m.

13.6.2.111 end end If the species concentration but the species is not produced by other species

Definition at line 602 of file generalConcentrationOverThreshold.m.

13.6.2.112 LOAD FIRST SPECIES FILE speciesFiles = dir('species_1*')

Definition at line 74 of file generalConcentrationOverThreshold.m.

13.6.2.113 tempProd_chain_weight = 0

Definition at line 537 of file generalConcentrationOverThreshold.m.

13.6.2.114 params threshold = 0

Definition at line 11 of file generalConcentrationOverThreshold.m.

13.6.2.115 : timeInterval = rtime - previousTime

Definition at line 201 of file generalConcentrationOverThreshold.m.

13.6.2.116 File Containing all Times times = 0:params.deltaT:params.totT

Definition at line 52 of file generalConcentrationOverThreshold.m.

13.6.2.117 tline = fgets(fidConf)

Definition at line 732 of file generalConcentrationOverThreshold.m.

13.6.2.118 tmpProd_chain = 0

Definition at line 534 of file generalConcentrationOverThreshold.m.

13.6.2.119 params tmpRctFileToLoad = ''

Definition at line 14 of file generalConcentrationOverThreshold.m.

13.6.2.120 params tmpRctSUBFileToLoad = ''

Definition at line 15 of file generalConcentrationOverThreshold.m.

13.6.2.121 tmpStrZeros = zeroBeforeStrNum(tmpRct, tmpRcts)

Definition at line 691 of file generalConcentrationOverThreshold.m.

13.6.2.122 to = graph(:,2)+1

Definition at line 494 of file generalConcentrationOverThreshold.m.

13.6.2.123 wasteSpecies = 0

Definition at line 511 of file generalConcentrationOverThreshold.m.

13.6.2.124 end end If the species concentration but the species is not produced by other so it is waste if wasteSpeciesFLAG = 0

Definition at line 538 of file generalConcentrationOverThreshold.m.

13.6.2.125 end end Reactions to distribuite in the different nature of the reactions weightToDistribute = graph(and((graph(:,1) == incomingNodes(innode)),(graph(:,2) == IDsOverThreshold(idot))),6)

Definition at line 558 of file generalConcentrationOverThreshold.m.

13.7 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/old/KillSpam.m File Reference

Functions

- function KillSpam (param) currentDir
- cd (param.tmpPath)
- disp ('|----------|')
- disp ('|-Killer and Spammer analysis is started|')
- READ ALL THE DIRECTORY
  CONTAINING SIMULATIONS if ~isdir ('0_statistics') mkdir('0_statistics')
- cd ('0_statistics')
- fclose (fid)
- cd ('../')
- disp (strcat('|-Folder: ', simDirs(x).name,'/res...'))
- break end reactionsMatrix (continua, 1:6)
- cleavagesMatrix (cleavages, 1:6)
- condensationsMatrix (condensations, 1:6)
- end catalysisMatrix (continua, 1:4)
- outliersMatrix (outRows, 1:6)
- if ((outliersMatrix(i, 2)==1)&&(check2==1 &&check3==1))||(outliersMatrix(i
- killersMatrix (killRows, 1:6)
- killerCatalysts (KCRows, 1)
- killerCatalysts (KCRows, 2)
- killerCatalysts (KCRows, 3)
- id KSRows ()
- if outliersMatrix (i, 2)
- KSMatrix (KSRows, 1:6)
- KSMatrix (i,:)
- KSCatalysts (KSCRows, 1)
- KSCatalysts (KSCRows, 2)
- KSCatalysts (KSCRows, 3)
- if ((outliersMatrix(i, 2)==1)&&(check2==0 &&check3==0))||(outliersMatrix(i
- SpammersMatrix (SpamRows, 1:6)
- if SpamRows SpammersMatrix (:, 7)
- SpammersMatrix (i,:)
- else SpammersMatrix (i, 7)
- spammerCatalysts (KSCRows, 1)
- spammerCatalysts (KSCRows, 2)
- spammerCatalysts (KSCRows, 3)
- fprintf (fid,'NET%d\n\n', x)
- fprintf (fid,'KILLERS\n\n')
- end fprintf (fid,'\n\n')

- fprintf (fid,'Catalysts\n\n')
- fprintf (fid,'%d\t', killerCatalysts(i, 2))
- fprintf (fid,'%d\n\n', killerCatalysts(i, 3))
- end fprintf (fid,'KILLERS-SPAMMERS\n\n')
- fprintf (fid,'%d\t', KSCatalysts(KSCRows, 2))
- fprintf (fid,'%d\n\n', KSCatalysts(KSCRows, 3))
- end fprintf (fid,'SPAMMERS\n\n')
- fprintf (fid,'%d\t', spammerCatalysts(SCRows, 2))
- fprintf (fid,'%d\n\n', spammerCatalysts(SCRows, 3))
- end cd (currentDir)

Variables

- end tmpFolder = strcat('*',param.simFolder,'*')
- simDirs = dir(tmpFolder)
- fid = fopen('ALL_results_TS1.txt','w')
- for x
- blocked = 0:param.lastSpecies
- creating reactions matrix rFiles = dir('reactions*')
- reactionsFile = rFiles(length(rFiles)-1).name
- continua = 1
- reactionsMatrix = 0
- while continua rValues = fscanf(fid,'%d',7)
- stop = isempty(rValues)
- dividing into condensations
  and cleavages condensations = 0
- cleavages = 0
- cleavagesMatrix = 0
- condensationsMatrix = 0
- for i
- creating catalysis matrix cFiles = dir('catalysis*')
- catalysisFile = cFiles(length(cFiles)).name
- catalysisMatrix = 0
- while continua cValues = fscanf(fid,'%f',7)
- break
- identifying outliers reactionsCounters = reactionsMatrix(:,6)
- outRows = 0
- outliersMatrix = 0
- if possible
- identifying killers killRows = 0
- killersMatrix = 0
- check2 = ismember(outliersMatrix(i,4),blocked)
- check3 = ismember(outliersMatrix(i,5),blocked)
- killerCatalysts = 0
- KCRows = 0
- for j
- KSMatrix = 0
- check4 = xor(check2,check3)
- check6 = ismember(KSMatrix(i,5),cleavagesMatrix(:,3))
- check7 = ismember(KSMatrix(i,5),catalysisMatrix(:,2))
- if check5
- checkKSM = isempty(KSMatrix)
- KSCatalysts = 0
- KSCRows = 0

- end identifying spammers SpamRows = 0
- SpammersMatrix = 0
- end checkSM = isempty(SpammersMatrix)
- spammersCatalysts = 0
- SCRows = 0
- checkKM = isempty(killersMatrix)
- else [r c] = size(killersMatrix)
- clear r
- clear c

13.7.1 Function Documentation

13.7.1.1 **end** catalysisMatrix (**continua** , 1:4)

13.7.1.2 cd (param. *tmpPath*)

13.7.1.3 cd ('0_statistics')

13.7.1.4 cd ('../')

13.7.1.5 **end** cd (**currentDir**)

13.7.1.6 cleavagesMatrix (**cleavages** , 1:6)

13.7.1.7 condensationsMatrix (**condensations** , 1:6)

13.7.1.8 disp ('|----------|')

13.7.1.9 disp ('|-Killer **and** Spammer **analysis** is started|')

13.7.1.10 disp (strcat('|-Folder: ', **simDirs(x).name**,'/res...'))

13.7.1.11 fclose (**fid**)

13.7.1.12 fprintf (**fid** , 'NET%d\n\n' , **x**)

13.7.1.13 fprintf (**fid** , 'KILLERS\n\n')

13.7.1.14 **end** fprintf (**fid** , '\n\n')

13.7.1.15 fprintf (**fid** , 'Catalysts\n\n')

13.7.1.16 fprintf (**fid** , '%d\t' , **killerCatalysts(i, 2)**)

13.7.1.17 fprintf (**fid** , '%d\n\n' , **killerCatalysts(i, 3)**)

13.7.1.18 **end** fprintf (**fid** , 'KILLERS-SPAMMERS\n\n')

13.7.1.19 fprintf (**fid** , '%d\t' , **KSCatalysts(KSCRows, 2)**)

13.7.1.20 fprintf (**fid** , '%d\n\n' , **KSCatalysts(KSCRows, 3)**)

13.7.1.21 **end** fprintf (**fid** , 'SPAMMERS\n\n')

13.7.1.22 fprintf (**fid** , '%d\t' , **spammerCatalysts(SCRows, 2)**)

13.7.1.23 fprintf (fid , '%d\n\n' , spammerCatalysts(SCRows, 3))

13.7.1.24 if ((outliersMatrix(i, 2)==1)&&(check2==1 &&check3==1))

13.7.1.25 if ((outliersMatrix(i, 2)==1)&&(check2==0 &&check3==0))

13.7.1.26 killerCatalysts (KCRows , 1)

13.7.1.27 killerCatalysts (KCRows , 2)

13.7.1.28 killerCatalysts (KCRows , 3)

13.7.1.29 killersMatrix (killRows , 1:6)

13.7.1.30 function KillSpam (param)

13.7.1.31 KSCatalysts (KSCRows , 1)

13.7.1.32 KSCatalysts (KSCRows , 2)

13.7.1.33 KSCatalysts (KSCRows , 3)

13.7.1.34 KSMatrix (KSRows , 1:6)

13.7.1.35 KSMatrix (i , :)

13.7.1.36 id KSRows ( ) [virtual]

13.7.1.37 outliersMatrix (outRows , 1:6)

13.7.1.38 if outliersMatrix (i , 2)

13.7.1.39 break end reactionsMatrix (continua , 1:6)

13.7.1.40 spammerCatalysts (KSCRows , 1)

13.7.1.41 spammerCatalysts (KSCRows , 2)

13.7.1.42 spammerCatalysts (KSCRows , 3)

13.7.1.43 SpammersMatrix (SpamRows , 1:6)

13.7.1.44 if SpamRows SpammersMatrix (: , 7)

13.7.1.45 SpammersMatrix (i , :)

13.7.1.46 else SpammersMatrix (i , 7)

13.7.1.47 READ ALL THE DIRECTORY CONTAINING SIMULATIONS if ~isdir ('0_statistics')

13.7.2 Variable Documentation

13.7.2.1 blocked = 0:param.lastSpecies

Definition at line 27 of file KillSpam.m.

13.7.2.2 break

Definition at line 74 of file KillSpam.m.

13.7.2.3 clear c

Definition at line 273 of file KillSpam.m.

13.7.2.4 catalysisFile = cFiles(length(cFiles)).name

Definition at line 65 of file KillSpam.m.

13.7.2.5 catalysisMatrix = 0

Definition at line 68 of file KillSpam.m.

13.7.2.6 creating catalysis matrix cFiles = dir('catalysis*')

Definition at line 64 of file KillSpam.m.

13.7.2.7 check2 = ismember(outliersMatrix(i,4),blocked)

Definition at line 106 of file KillSpam.m.

13.7.2.8 check3 = ismember(outliersMatrix(i,5),blocked)

Definition at line 107 of file KillSpam.m.

13.7.2.9 check4 = xor(check2,check3)

Definition at line 139 of file KillSpam.m.

13.7.2.10 if check5

Initial value:

```
== 1 || check6 == 1 || check7 == 1
                    KSRows = KSRows-1
```

Definition at line 152 of file KillSpam.m.

13.7.2.11 check6 = ismember(KSMatrix(i,5),cleavagesMatrix(:,3))

Definition at line 150 of file KillSpam.m.

13.7.2.12 check7 = ismember(KSMatrix(i,5),catalysisMatrix(:,2))

Definition at line 151 of file KillSpam.m.

13.7.2.13 if checkKM = isempty(killersMatrix)

Definition at line 259 of file KillSpam.m.

13.7.2.14 if checkKSM = isempty(KSMatrix)

Definition at line 168 of file KillSpam.m.

13.7.2.15 if checkSM = isempty(SpammersMatrix)

Definition at line 234 of file KillSpam.m.

13.7.2.16 cleavages = 0

Definition at line 48 of file KillSpam.m.

13.7.2.17 cleavagesMatrix = 0

Definition at line 50 of file KillSpam.m.

13.7.2.18 else condensations = 0

Definition at line 47 of file KillSpam.m.

13.7.2.19 condensationsMatrix = 0

Definition at line 51 of file KillSpam.m.

13.7.2.20 continua = 1

Definition at line 32 of file KillSpam.m.

13.7.2.21 while continua cValues = fscanf(fid,'%f',7)

Definition at line 70 of file KillSpam.m.

13.7.2.22 else[r c] = size(killersMatrix)

Definition at line 263 of file KillSpam.m.

13.7.2.23 fid = fopen('ALL_results_TS1.txt','w')

Definition at line 20 of file KillSpam.m.

13.7.2.24 end clear i

Initial value:

```
= 1:rrm
        if reactionsMatrix(i,2) == 1
            cleavages = cleavages+1
```

Definition at line 52 of file KillSpam.m.

13.7.2.25 clear j

Initial value:

```
= 1:rcm
                if killerReaction == catalysisMatrix(j,3) &&
      catalysisMatrix(j,4) > 0
                    KCRows = KCRows+1
```

Definition at line 120 of file KillSpam.m.

13.7.2.26 KCRows = 0

Definition at line 117 of file KillSpam.m.

13.7.2.27 killerCatalysts = 0

Definition at line 116 of file KillSpam.m.

13.7.2.28 killersMatrix = 0

Definition at line 103 of file KillSpam.m.

13.7.2.29 if killRows = 0

Definition at line 102 of file KillSpam.m.

13.7.2.30 KSCatalysts = 0

Definition at line 171 of file KillSpam.m.

13.7.2.31 KSCRows = 0

Definition at line 172 of file KillSpam.m.

13.7.2.32 KSMatrix = 0

Definition at line 135 of file KillSpam.m.

13.7.2.33 outliersMatrix = 0

Definition at line 91 of file KillSpam.m.

13.7.2.34 outRows = 0

Definition at line 90 of file KillSpam.m.

13.7.2.35 if possible

Initial value:

```
== 1
            outRows = outRows+1
```

Definition at line 94 of file KillSpam.m.

13.7.2.36 clear r

Definition at line 272 of file KillSpam.m.

13.7.2.37 identifying outliers reactionsCounters = reactionsMatrix(:,6)

Definition at line 83 of file KillSpam.m.

13.7.2.38 reactionsFile = rFiles(length(rFiles)-1).name

Definition at line 30 of file KillSpam.m.

13.7.2.39 reactionsMatrix = 0

Definition at line 33 of file KillSpam.m.

13.7.2.40 creating reactions matrix rFiles = dir('reactions∗')

Definition at line 29 of file KillSpam.m.

13.7.2.41 while continua rValues = fscanf(fid,'%d',7)

Definition at line 35 of file KillSpam.m.

13.7.2.42 SCRows = 0

Definition at line 238 of file KillSpam.m.

13.7.2.43 simDirs = dir(tmpFolder)

Definition at line 17 of file KillSpam.m.

13.7.2.44 spammersCatalysts = 0

Definition at line 237 of file KillSpam.m.

13.7.2.45 SpammersMatrix = 0

Definition at line 190 of file KillSpam.m.

13.7.2.46 end identifying spammers SpamRows = 0

Definition at line 189 of file KillSpam.m.

13.7.2.47 if stop = isempty(rValues)

Definition at line 36 of file KillSpam.m.

13.7.2.48 end tmpFolder = strcat('*',param.simFolder,'*')

Definition at line 16 of file KillSpam.m.

13.7.2.49 for x

Initial value:

```
= 1:length(simDirs)
    cd(strcat(simDirs(x).name,'/res'))
```

Definition at line 24 of file KillSpam.m.

13.8 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/old/KSSearch.m File Reference

Functions

- break end reactionsMatrix (continua, 1:6)
- end fclose (fid)
- cleavagesMatrix (i, 1:6)
- condensationsMatrix (i, 1:6)
- break end catalysisMatrix (continua, 1:7)
- outliersMatrix (outRows, 1:6)
- if ((outliersMatrix(i, 2)==0)&&(check2==1 &&check3==1))||(outliersMatrix(i
- killersMatrix (killRows, 1:6)
- id KSRows ()
- if outliersMatrix (i, 2)
- KSMatrix (KSRows, 1:6)
- KSMatrix (i,:)
- if ((outliersMatrix(i, 2)==0)&&(check2==0 &&check3==0))||(outliersMatrix(i
- SpammersMatrix (SpamRows, 1:6)
- if SpamRows SpammersMatrix (:, 7)
- SpammersMatrix (i,:)
- else SpammersMatrix (i, 7)

Variables

- blocked = 0:5
- creating reactions matrix rFiles = dir('reactions*')
- reactionsFile = rFiles(length(rFiles)-1).name
- fid = fopen(reactionsFile)
- continua = 1

- reactionsMatrix = 0
- while continua rValues = fscanf(fid,'%d',7)
- stop = isempty(rValues)
- dividing into condensations
  and cleavages condensations = 0
- cleavages = 0
- cleavagesMatrix = 0
- condensationsMatrix = 0
- for i
- creating catalysis matrix cFiles = dir('catalysis∗')
- catalysisFile = cFiles(length(cFiles)).name
- catalysisMatrix = 0
- while continua cValues = fscanf(fid,'%f',7)
- identifying outliers reactionsCounters = sort(reactionsMatrix(:,6))'
- fpc = ceil(rrm/20∗19)
- outliersCounter = reactionsCounters(fpc:rrm)
- outRows = 0
- outliersMatrix = 0
- if possible
- identifying killers killRows = 0
- killersMatrix = 0
- check2 = ismember(outliersMatrix(i,4),blocked)
- check3 = ismember(outliersMatrix(i,5),blocked)
- KSMatrix = 0
- check4 = xor(check2,check3)
- check6 = ismember(KSMatrix(i,5),cleavagesMatrix(:,3))
- check7 = ismember(KSMatrix(i,5),catalysisMatrix(:,2))
- if check5
- identifying spammers SpamRows = 0
- SpammersMatrix = 0

13.8.1 Function Documentation

13.8.1.1 break end catalysisMatrix (continua , 1:7)

13.8.1.2 cleavagesMatrix (i , 1:6)

13.8.1.3 condensationsMatrix (i , 1:6)

13.8.1.4 end fclose (fid)

13.8.1.5 if ((outliersMatrix(i, 2)==0)&&(check2==1 &&check3==1))

13.8.1.6 if ((outliersMatrix(i, 2)==0)&&(check2==0 &&check3==0))

13.8.1.7 killersMatrix (killRows , 1:6)

13.8.1.8 KSMatrix (KSRows , 1:6)

13.8.1.9 KSMatrix (i , :)

13.8.1.10 id KSRows ( ) [virtual]

13.8.1.11 outliersMatrix (outRows , 1:6)

13.8.1.12 if outliersMatrix (i , 2)

13.8.1.13 break end reactionsMatrix (continua , 1:6)

13.8.1.14 SpammersMatrix (SpamRows , 1:6)

13.8.1.15 if SpamRows SpammersMatrix (: , 7)

13.8.1.16 SpammersMatrix (i , :)

13.8.1.17 else SpammersMatrix (i , 7)

13.8.2 Variable Documentation

13.8.2.1 blocked = 0:5

Definition at line 1 of file KSSearch.m.

13.8.2.2 catalysisFile = cFiles(length(cFiles)).name

Definition at line 40 of file KSSearch.m.

13.8.2.3 catalysisMatrix = 0

Definition at line 43 of file KSSearch.m.

13.8.2.4 creating catalysis matrix cFiles = dir('catalysis*')

Definition at line 39 of file KSSearch.m.

13.8.2.5 check2 = ismember(outliersMatrix(i,4),blocked)

Definition at line 82 of file KSSearch.m.

13.8.2.6 check3 = ismember(outliersMatrix(i,5),blocked)

Definition at line 83 of file KSSearch.m.

13.8.2.7 check4 = xor(check2,check3)

Definition at line 97 of file KSSearch.m.

13.8.2.8 if check5

Initial value:

```
== 1 || check6 == 1 || check7 == 1
                KSRows = KSRows-1
```

Definition at line 110 of file KSSearch.m.

13.8.2.9 check6 = ismember(KSMatrix(i,5),cleavagesMatrix(:,3))

Definition at line 108 of file KSSearch.m.

13.8.2.10 check7 = ismember(KSMatrix(i,5),catalysisMatrix(:,2))

Definition at line 109 of file KSSearch.m.

13.8.2.11 cleavages = 0

Definition at line 23 of file KSSearch.m.

13.8.2.12 cleavagesMatrix = 0

Definition at line 25 of file KSSearch.m.

13.8.2.13 else condensations = 0

Definition at line 22 of file KSSearch.m.

13.8.2.14 condensationsMatrix = 0

Definition at line 26 of file KSSearch.m.

13.8.2.15 continua = 1

Definition at line 7 of file KSSearch.m.

13.8.2.16 while continua cValues = fscanf(fid,'%f',7)

Definition at line 45 of file KSSearch.m.

13.8.2.17 fid = fopen(reactionsFile)

Definition at line 6 of file KSSearch.m.

13.8.2.18 fpc = ceil(rrm/20∗19)

Definition at line 58 of file KSSearch.m.

13.8.2.19 end end end clear i

Initial value:

```
= 1:rrm
    if reactionsMatrix(i,2) == 0
        cleavages = cleavages+1
```

Definition at line 27 of file KSSearch.m.

13.8.2.20 killersMatrix = 0

Definition at line 79 of file KSSearch.m.

13.8.2.21 identifying killers killRows = 0

Definition at line 78 of file KSSearch.m.

13.8.2.22 KSMatrix = 0

Definition at line 93 of file KSSearch.m.

13.8.2.23 outliersCounter = reactionsCounters(fpc:rrm)

Definition at line 59 of file KSSearch.m.

13.8.2.24 outliersMatrix = 0

Definition at line 67 of file KSSearch.m.

13.8.2.25 outRows = 0

Definition at line 66 of file KSSearch.m.

13.8.2.26 if possible

Initial value:

```
== 1
        outRows = outRows+1
```

Definition at line 70 of file KSSearch.m.

13.8.2.27 identifying outliers reactionsCounters = sort(reactionsMatrix(:,6))'

Definition at line 57 of file KSSearch.m.

13.8.2.28 reactionsFile = rFiles(length(rFiles)-1).name

Definition at line 5 of file KSSearch.m.

13.8.2.29 reactionsMatrix = 0

Definition at line 8 of file KSSearch.m.

13.8.2.30 creating reactions matrix rFiles = dir('reactions*')

Definition at line 4 of file KSSearch.m.

13.8.2.31 while continua rValues = fscanf(fid,'%d',7)

Definition at line 10 of file KSSearch.m.

13.8.2.32 SpammersMatrix = 0

Definition at line 129 of file KSSearch.m.

13.8.2.33 identifying spammers SpamRows = 0

Definition at line 128 of file KSSearch.m.

13.8.2.34 if stop = isempty(rValues)

Definition at line 11 of file KSSearch.m.

13.9 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/old/KSSearchLauncher.m File Reference

Functions

- fclose (fid)
- cd (currentDir)
- fprintf (fid,'NET%d\n\n', x)
- fprintf (fid,'KILLERS\n\n')
- end fprintf (fid,'\n\n')
- fprintf (fid,'KILLERS-SPAMMERS\n\n')
- fprintf (fid,'SPAMMERS\n\n')

Variables

- currentDir = cd()
- simDirs = dir('sim_*')
- fid = fopen('results.txt','w')
- for x
- KSSearch
- checkKM = isempty(killersMatrix)
- else [r c] = size(killersMatrix)
- for i
- clear j
- clear r
- clear c
- checkKSM = isempty(KSMatrix)
- checkSM = isempty(SpammersMatrix)

13.9.1 Function Documentation

13.9.1.1 cd (currentDir)

13.9.1.2 fclose (fid)

13.9.1.3 fprintf (fid , 'NET%d\n\n' , x)

13.9.1.4 fprintf (fid , 'KILLERS\n\n')

13.9.1.5 end fprintf (fid , '\n\n')

13.9.1.6 fprintf (fid , 'KILLERS-SPAMMERS\n\n')

13.9.1.7 fprintf (fid , 'SPAMMERS\n\n')

13.9.2 Variable Documentation

13.9.2.1 clear c

Definition at line 27 of file KSSearchLauncher.m.

13.9.2.2 if checkKM = isempty(killersMatrix)

Definition at line 12 of file KSSearchLauncher.m.

13.9.2.3 if checkKSM = isempty(KSMatrix)

Definition at line 29 of file KSSearchLauncher.m.

13.9.2.4 if checkSM = isempty(SpammersMatrix)

Definition at line 46 of file KSSearchLauncher.m.

13.9.2.5 currentDir = cd()

Definition at line 1 of file KSSearchLauncher.m.

13.9.2.6 else[r c] = size(killersMatrix)

Definition at line 16 of file KSSearchLauncher.m.

13.9.2.7 fid = fopen('results.txt','w')

Definition at line 3 of file KSSearchLauncher.m.

13.9.2.8 end end clear i

Initial value:

```
= 1:r
            for j = 1:c
                fprintf(fid,'%d\t',killersMatrix(i,j))
```

Definition at line 17 of file KSSearchLauncher.m.

13.9.2.9 clear j

Definition at line 25 of file KSSearchLauncher.m.

13.9.2.10 KSSearch

Definition at line 7 of file KSSearchLauncher.m.

13.9.2.11 clear r

Definition at line 26 of file KSSearchLauncher.m.

13.9.2.12 simDirs = dir('sim_*')

Definition at line 2 of file KSSearchLauncher.m.

13.9.2.13 for x

Initial value:

```
= 1:length(simDirs)
    cd(strcat(simDirs(x).name,'/res'))
```

Definition at line 5 of file KSSearchLauncher.m.

13.10 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/old/overallStats.m File Reference

Functions

- function overallStats () close all clear all p.tmpAnalysis
- Used in KillSpan it represents
  the last influx species LATEST
  FILES ANALYSIS if p tmpAnalysis (1)
- end if p tmpAnalysis (2)
- end if p tmpAnalysis (3)
- end if p tmpAnalysis (4)
- if p tmpAnalysis (6)
- if exist ('MATLAB2R.Rout','file') delete('MATLAB2R.Rout')
- eval (['!R CMD BATCH--no-save--no-restore--slave"--args sf="',scriptAnalDir,'" ls=',num2str(p.lastSpecies),...
  ' sss="',p.simFolder,'" gfs="',gfs,'" sd="',p.tmpPath,'"'"MATLAB2R.R'])
- cd (crtDir)

Variables

- p tmpPath = '/Users/alessandrofilisetti/Documents/results/VILLANI/'
- p figureVisible = 0
- p simFolder = 'ACS2'
- p lastSpecies = 2

- Used in KillSpan analysis
- p threshold = 0
- p tmpRctFileToLoad = ''
- p tmpRctSUBFileToLoad = ''
- end scriptAnalDir = cd()
- gfs = strcat('_',num2str(p.decayTime))

13.10.1 Function Documentation

13.10.1.1 cd (crtDir)

13.10.1.2 eval ( )

13.10.1.3 if exist ('MATLAB2R.Rout' , 'file')

13.10.1.4 function overallStats ( )

13.10.1.5 Used in KillSpan it represents the last influx species LATEST FILES ANALYSIS if p tmpAnalysis (1)

13.10.1.6 end if p tmpAnalysis (2)

13.10.1.7 end if p tmpAnalysis (3)

13.10.1.8 end if p tmpAnalysis (4)

13.10.1.9 end Need p tmpAnalysis (6)

Initial value:

```
== 1
    % Using an R-Cran script this part is a little bit different
    crtDir = cd()
```

13.10.2 Variable Documentation

13.10.2.1 Used in KillSpan analysis

Definition at line 10 of file overallStats.m.

13.10.2.2 p figureVisible = 0

Definition at line 8 of file overallStats.m.

13.10.2.3 gfs = strcat('_',num2str(p.decayTime))

Definition at line 50 of file overallStats.m.

13.10.2.4 p lastSpecies = 2

Definition at line 10 of file overallStats.m.

13.10.2.5 end scriptAnalDir = cd()

Definition at line 49 of file overallStats.m.

13.10.2.6 p simFolder = 'ACS2'

Definition at line 9 of file overallStats.m.

13.10.2.7 p threshold = 0

Definition at line 36 of file overallStats.m.

13.10.2.8 concAnalysis(params) clear all close all tmpPath = '/Users/alessandrofilisetti/Documents/results/VILLANI/'

Definition at line 7 of file overallStats.m.

13.10.2.9 p tmpRctFileToLoad = ''

Definition at line 37 of file overallStats.m.

13.10.2.10 p tmpRctSUBFileToLoad = ''

Definition at line 38 of file overallStats.m.

13.11 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/old/readParameters.m File Reference

Functions

- while ischar (tline) tline
- if isequal (tline(1:param-1),'nSeconds') confparams.nSeconds
- elseif isequal (tline(1:param-1),'nGEN') confparams.nGEN
- elseif isequal (tline(1:param-1),'nSIM') confparams.nSIM
- elseif isequal (tline(1:param-1),'nReactions') confparams.nReactions
- elseif isequal (tline(1:param-1),'timeStructuresSavingInterval') confparams.timeStructuresSavingInterval
- elseif isequal (tline(1:param-1),'fileTimesSaveInterval') confparams.fileTimesSaveInterval
- elseif isequal (tline(1:param-1),'reactionProbability') confparams.reactionProbability
- elseif isequal (tline(1:param-1),'energy') confparams.energy
- elseif isequal (tline(1:param-1),'ECConcentration') confparams.ECConcentration
- elseif isequal (tline(1:param-1),'influx_rate') confparams.influx_rate
- elseif isequal (tline(1:param-1),'maxLOut') confparams.maxLOut
- elseif isequal (tline(1:param-1),'volume') confparams.volume
- end end fclose (fidConf)

Variables

- function confparams
- tline = fgets(fidConf)
- param = findstr(tline,'=')

13.11.1 Function Documentation

13.11.1.1 end end fclose (fidConf)

13.11.1.2 while ischar (tline)

13.11.1.3 if isequal (tline(1:param-1) , 'nSeconds')

13.11.1.4 elseif isequal (tline(1:param-1) , 'nGEN')

13.11.1.5 elseif isequal (tline(1:param-1) , 'nSIM')

13.11.1.6 elseif isequal (tline(1:param-1) , 'nReactions')

13.11.1.7 elseif isequal (tline(1:param-1) , 'timeStructuresSavingInterval')

13.11.1.8 elseif isequal (tline(1:param-1) , 'fileTimesSaveInterval')

13.11.1.9 elseif isequal (tline(1:param-1) , 'reactionProbability')

13.11.1.10 elseif isequal (tline(1:param-1) , 'energy')

13.11.1.11 elseif isequal (tline(1:param-1) , 'ECConcentration')

13.11.1.12 elseif isequal (tline(1:param-1) , 'influx_rate')

13.11.1.13 elseif isequal (tline(1:param-1) , 'maxLOut')

13.11.1.14 elseif isequal (tline(1:param-1) , 'volume')

13.11.2 Variable Documentation

13.11.2.1 function confparams

Initial value:

```
= readParameters()

    fidConf=fopen('acsm2s.conf','r')
```

Definition at line 1 of file readParameters.m.

13.11.2.2 param = findstr(tline,'=')

Definition at line 9 of file readParameters.m.

13.11.2.3 tline = fgets(fidConf)

Definition at line 5 of file readParameters.m.

13.12 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/old/resetForNewSimulations.py File Reference

Namespaces

- resetForNewSimulations

Functions

- def resetForNewSimulations.zeroBeforeStrNum

Variables

- int resetForNewSimulations.foldersSIMS = 10
- int resetForNewSimulations.foldersREP = 10
- tuple resetForNewSimulations.zerosSIMS = zeroBeforeStrNum(i,foldersSIMS)
- tuple resetForNewSimulations.zerosREPS = zeroBeforeStrNum(j,foldersREP)
- string resetForNewSimulations.folderName = "s_inv_1e-2_"
- string resetForNewSimulations.folderNew = "s_inv_1e-1_"
- tuple resetForNewSimulations.resdir = os.path.join(os.curdir, "res")
- tuple resetForNewSimulations.crtSimFolder = os.path.join(StrTo,folderName)
- tuple resetForNewSimulations.fileDest = os.path.join(StrTo,folderNew,"_acsinflux.csv")
- tuple resetForNewSimulations.speciesFiles = sorted(glob.glob("species_1_∗"))
- tuple resetForNewSimulations.mod = open("_acsspecies.csv")
- int resetForNewSimulations.id = 0
- tuple resetForNewSimulations.linesplitted = line.split("\t")
- tuple resetForNewSimulations.file = open("_acsspecies.csv", "w")

13.13 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/old/somma.m File Reference

Functions

- function somma (N) close all a
- a (i)
- if r b (i)
- end c (i)

Variables

- b = zeros(1,N)
- c = zeros(1,N)
- for i

13.13.1 Function Documentation

13.13.1.1 a (i)

13.13.1.2 if r b (i)

13.13.1.3 end c (i)

13.13.1.4 function somma (N)

13.13.2 Variable Documentation

13.13.2.1 b = zeros(1,N)

Definition at line 7 of file somma.m.

13.13.2.2 c = zeros(1,N)

Definition at line 8 of file somma.m.

13.13.2.3 for i

Initial value:

```
= 1 : N
    r = rand()
```

Definition at line 10 of file somma.m.

13.14 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/old/stats.m File Reference

Functions

- function stats ()%clear all%close all%%params.path
- concAnalysis (params)%clear all%close all%%params.path

Variables

- params deltaT = 10
- params totT = 1000
- params showFig = 0
- params figureVisible = 0
- params threshold = 0
- params decayTime = 10
- params tmpResFold = 'res'
- params distinctiveSubStr = 'sim'
- params tmpIDsim = '5'
- params tmpRctFileToLoad = ''
- params tmpRctSUBFileToLoad = ''

13.14.1 Function Documentation

13.14.1.1 concAnalysis (params)

13.14.1.2 function stats ( )

13.14.2 Variable Documentation

13.14.2.1 params decayTime = 10

Definition at line 84 of file stats.m.

13.14.2.2 params deltaT = 10

Definition at line 7 of file stats.m.

13.14.2.3 params distinctiveSubStr = 'sim'

Definition at line 86 of file stats.m.

13.14.2.4 params figureVisible = 0

Definition at line 57 of file stats.m.

13.14.2.5 params showFig = 0

Definition at line 9 of file stats.m.

13.14.2.6 params threshold = 0

Definition at line 83 of file stats.m.

13.14.2.7 params tmpIDsim = '5'

Definition at line 87 of file stats.m.

13.14.2.8 params tmpRctFileToLoad = ''

Definition at line 88 of file stats.m.

13.14.2.9 params tmpRctSUBFileToLoad = ''

Definition at line 89 of file stats.m.

13.14.2.10 params tmpResFold = 'res'

Definition at line 85 of file stats.m.

13.14.2.11 params totT = 1000

Definition at line 8 of file stats.m.

13.15 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/old/timesAnalysis.m File Reference

Variables

- function [out]

- hence different scatterplot between different dimensions are performed Output out is a structure containing two variables

13.15.1 Variable Documentation

13.15.1.1 function[out]

Initial value:

```
= timesAnalysis(param)
% function [out] = timesAnalysis(tmpPath)
%
% This function collects all the last reactions contained in both the
      times and
% reactions_parameter files
```

Definition at line 1 of file timesAnalysis.m.

13.15.1.2 hence different scatterplot between different dimensions are performed Output out is a structure containing two variables

Definition at line 1 of file timesAnalysis.m.

13.16 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/old/timesAnalysis_PANINI.m File Reference

Variables

- function [out]

13.16.1 Variable Documentation

13.16.1.1 function[out]

Initial value:

```
= timesAnalysis(tmpPath)
% function [out] = timesAnalysis(tmpPath)
%
% This function collects all the last reactions contained in the times
      and
% reactions_parameter files and performs different scatterplot between different dimensions.
% Output out is a structure containing two variables:
% out.timesMatrix = all the last row of the times files
% out.cct = the correlation coefficients of the out.timesMatrix matrix
%
% INPUT
% tmpPath = path of the simulations root

currentDir = cd()
```

Definition at line 1 of file timesAnalysis_PANINI.m.

13.17 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/underDevelopment/acsAttractorAnalysis.py File Reference

Namespaces

- acsAttractorAnalysis

Functions

- def acsAttractorAnalysis.zeroBeforeStrNum

Variables

- tuple acsAttractorAnalysis.StrPath = os.path.abspath(StrPath)
- tuple acsAttractorAnalysis.today = dt.date.today()
- tuple acsAttractorAnalysis.tmpDirs = sort(os.listdir(StrPath))
- list acsAttractorAnalysis.allSortedSpecies = []
- list acsAttractorAnalysis.allConcentrations = []
- list acsAttractorAnalysis.allSortedSpeciesNOINFLUX = []
- list acsAttractorAnalysis.allConcentrationsNOINFLUX = []
- tuple acsAttractorAnalysis.totDirName = os.path.join(StrPath,tmpDir)
- tuple acsAttractorAnalysis.resDirPath = os.path.abspath(os.path.join("./", "res"))
- tuple acsAttractorAnalysis.numberOfGen = len(glob.glob(os.path.join(resDirPath,'times_*')))
- tuple acsAttractorAnalysis.strZeros = zeroBeforeStrNum(ngen, numberOfGen)
- string acsAttractorAnalysis.strSpecies = 'species_'
- tuple acsAttractorAnalysis.speciesFiles = sorted(glob.glob(os.path.join(resDirPath,strSpecies)))
- list acsAttractorAnalysis.speciesFile = speciesFiles[-1]
- tuple acsAttractorAnalysis.fidSpecies = open(speciesFile, 'r')
- list acsAttractorAnalysis.seq = []
- list acsAttractorAnalysis.overallConcList = []
- list acsAttractorAnalysis.overallConcListNOINFLUX = []
- int acsAttractorAnalysis.numberOfFolders = 0
- tuple acsAttractorAnalysis.pos = seq.index(key)
- tuple acsAttractorAnalysis.pos2 = seq.index(key2)
- tuple acsAttractorAnalysis.ANG_overallResMatrix = np.zeros((numberOfFolders,numberOfFolders))
- tuple acsAttractorAnalysis.ANG_overallResMatrixNOINFLUX = np.zeros((numberOfFolders,numberOfFolders))
- tuple acsAttractorAnalysis.HAM_overallResMatrix = np.zeros((numberOfFolders,numberOfFolders))
- tuple acsAttractorAnalysis.HAM_overallResMatrixNOINFLUX = np.zeros((numberOfFolders,numberOfFolders))
- tuple acsAttractorAnalysis.EUC_overallResMatrix = np.zeros((numberOfFolders,numberOfFolders))
- tuple acsAttractorAnalysis.EUC_overallResMatrixNOINFLUX = np.zeros((numberOfFolders,numberOfFolders))
- tuple acsAttractorAnalysis.vecX = np.array(lx)
- tuple acsAttractorAnalysis.vecY = np.array(ly)
- tuple acsAttractorAnalysis.tmpCos = float(np.dot(vecX,vecY) / (np.linalg.norm(vecX) * np.linalg.norm(vecY)))
- int acsAttractorAnalysis.tmpHD = 0
- int acsAttractorAnalysis.tmpEU = 0
- string acsAttractorAnalysis.ndn = '_0_new_allStatResults'
- tuple acsAttractorAnalysis.newdirAllResults = os.path.join(os.curdir, ndn)
- string acsAttractorAnalysis.outFnameStat = 'speciesVector.csv'
- tuple acsAttractorAnalysis.saveFileStat = open(outFnameStat, 'w')
- string acsAttractorAnalysis.strTypes = ''
- int acsAttractorAnalysis.cnt = 0

13.18 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/underDevelopment/acsAttractorAnalysisInTime.py File Reference

Namespaces

- acsAttractorAnalysisInTime

Functions

- def acsAttractorAnalysisInTime.zeroBeforeStrNum

Variables

- tuple acsAttractorAnalysisInTime.StrPath = os.path.abspath(StrPath)
- tuple acsAttractorAnalysisInTime.today = dt.date.today()
- tuple acsAttractorAnalysisInTime.tmpDirs = sort(os.listdir(StrPath))
- list acsAttractorAnalysisInTime.allSortedSpecies = []
- list acsAttractorAnalysisInTime.allConcentrations = []
- list acsAttractorAnalysisInTime.allSortedSpeciesNOINFLUX = []
- list acsAttractorAnalysisInTime.allConcentrationsNOINFLUX = []
- tuple acsAttractorAnalysisInTime.totDirName = os.path.join(StrPath,tmpDir)
- tuple acsAttractorAnalysisInTime.resDirPath = os.path.abspath(os.path.join("./", "res"))
- tuple acsAttractorAnalysisInTime.numberOfGen = len(glob.glob(os.path.join(resDirPath,'times_*')))
- tuple acsAttractorAnalysisInTime.strZeros = zeroBeforeStrNum(ngen, numberOfGen)
- string acsAttractorAnalysisInTime.strSpeciesZero = 'species_'
- tuple acsAttractorAnalysisInTime.speciesFilesZero = sorted(glob.glob(os.path.join(resDirPath,strSpeciesZero)))
- string acsAttractorAnalysisInTime.strSpecies = 'species_'
- tuple acsAttractorAnalysisInTime.speciesFiles = sorted(glob.glob(os.path.join(resDirPath,strSpecies)))
- list acsAttractorAnalysisInTime.speciesFile = speciesFiles[timeFileID]
- tuple acsAttractorAnalysisInTime.fidSpecies = open(speciesFile, 'r')
- list acsAttractorAnalysisInTime.seq = []
- list acsAttractorAnalysisInTime.overallConcList = []
- list acsAttractorAnalysisInTime.overallConcListNOINFLUX = []
- int acsAttractorAnalysisInTime.numberOfFolders = 0
- tuple acsAttractorAnalysisInTime.pos = seq.index(key)
- tuple acsAttractorAnalysisInTime.pos2 = seq.index(key2)
- tuple acsAttractorAnalysisInTime.ANG_overallResMatrix = np.zeros((numberOfFolders,numberOfFolders))
- tuple acsAttractorAnalysisInTime.ANG_overallResMatrixNOINFLUX = np.zeros((numberOfFolders,numberOfFolders))
- tuple acsAttractorAnalysisInTime.HAM_overallResMatrix = np.zeros((numberOfFolders,numberOfFolders))
- tuple acsAttractorAnalysisInTime.HAM_overallResMatrixNOINFLUX = np.zeros((numberOfFolders,numberOfFolders))
- tuple acsAttractorAnalysisInTime.EUC_overallResMatrix = np.zeros((numberOfFolders,numberOfFolders))
- tuple acsAttractorAnalysisInTime.EUC_overallResMatrixNOINFLUX = np.zeros((numberOfFolders,numberOfFolders))
- tuple acsAttractorAnalysisInTime.vecX = np.array(lx)
- tuple acsAttractorAnalysisInTime.vecY = np.array(ly)
- tuple acsAttractorAnalysisInTime.tmpCos = float(np.dot(vecX,vecY) / (np.linalg.norm(vecX) * np.linalg.norm(vecY)))
- int acsAttractorAnalysisInTime.tmpHD = 0
- int acsAttractorAnalysisInTime.tmpEU = 0
- string acsAttractorAnalysisInTime.ndn = '_0_new_allStatResults'

- tuple acsAttractorAnalysisInTime.newdirAllResults = os.path.join(os.curdir, ndn)
- tuple acsAttractorAnalysisInTime.tmpZeroSaving = zeroBeforeStrNum(timeFileID, tmpNOF)
- string acsAttractorAnalysisInTime.outFnameStat = 'speciesVector'
- tuple acsAttractorAnalysisInTime.saveFileStat = open(outFnameStat, 'w')
- string acsAttractorAnalysisInTime.strTypes = ''
- int acsAttractorAnalysisInTime.cnt = 0

13.19 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/underDevelopment/acsBufferedFluxes.py File Reference

Namespaces

- acsBufferedFluxes

Functions

- def acsBufferedFluxes.zeroBeforeStrNum

Variables

- tuple acsBufferedFluxes.parser
- tuple acsBufferedFluxes.args = parser.parse_args()
- tuple acsBufferedFluxes.strPath = os.path.abspath(args.strPath)
- tuple acsBufferedFluxes.tmpDirs = sort(os.listdir(StrPath))
- int acsBufferedFluxes._CONDENSATION_ = 0
- int acsBufferedFluxes._CLEAVAGE_ = 1
- int acsBufferedFluxes._ENDOCONDENSATION_ = 7
- int acsBufferedFluxes._ENDOCLEAVAGE_ = 6
- int acsBufferedFluxes._SPONTCONDENSATION_ = 10
- int acsBufferedFluxes._SPONTCLEAVAGE_ = 11
- int acsBufferedFluxes.chemistry = 1
- string acsBufferedFluxes.ndn = '_0_new_allStatResults'
- tuple acsBufferedFluxes.newdirAllResults = os.path.join(StrPath, ndn)
- tuple acsBufferedFluxes.totDirName = os.path.join(StrPath,tmpDir)
- tuple acsBufferedFluxes.resDirPath = os.path.abspath(os.path.join("./", "res"))
- tuple acsBufferedFluxes.numberOfGen = len(glob.glob(os.path.join(resDirPath,'times_*')))
- string acsBufferedFluxes.tmpFluxFile = 'fluxDynamics_'
- tuple acsBufferedFluxes.tmpSpeciesStatsSummaryNameFID = open(tmpFluxFile, 'w')
- tuple acsBufferedFluxes.strZeros = zeroBeforeStrNum(ngen, numberOfGen)
- string acsBufferedFluxes.strSpeciesZero = 'species_'
- tuple acsBufferedFluxes.speciesFilesZero = sorted(glob.glob(os.path.join(resDirPath,strSpeciesZero)))
- string acsBufferedFluxes.strSpecies = 'species_'
- tuple acsBufferedFluxes.speciesFiles = sorted(glob.glob(os.path.join(resDirPath,strSpecies)))
- list acsBufferedFluxes.lastfilespecies = speciesFiles[-1]
- tuple acsBufferedFluxes.fidSpecies = open(lastfilespecies, 'r')
- list acsBufferedFluxes.flux_seq = []
- tuple acsBufferedFluxes.counters = np.zeros((nSpecies,(len(seq)*2)+7))
- string acsBufferedFluxes.strRctPar = 'reactions_parameters_'
- tuple acsBufferedFluxes.fidRctPar = open(strRctPar, 'r')
- int acsBufferedFluxes.totIN = 0
- int acsBufferedFluxes.totOUT = 0
- int acsBufferedFluxes.deltaIO = 0

- int acsBufferedFluxes.totBIN = 0
- int acsBufferedFluxes.totBOUT = 0
- tuple acsBufferedFluxes.rctTime = int(tmpRctT)
- tuple acsBufferedFluxes.rctType = int(tmpRctType)
- tuple acsBufferedFluxes.cat = int(tmpCat)
- tuple acsBufferedFluxes.S1 = int(tmpS1)
- tuple acsBufferedFluxes.S2 = int(tmpS2)
- tuple acsBufferedFluxes.S3 = int(tmpS3)
- string acsBufferedFluxes.tmpFileName = 'speciesStats_'
- tuple acsBufferedFluxes.tmpFileNameFID = open(tmpFileName, 'w')
- int acsBufferedFluxes.ID = 0
- string acsBufferedFluxes.tmpStr = 'Total Number of Reactions\t\t\t\t'

13.20 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/underDevelopment/acsDynStatInTime.py File Reference

Namespaces

- acsDynStatInTime

Variables

- tuple acsDynStatInTime.parser
- tuple acsDynStatInTime.args = parser.parse_args()
- tuple acsDynStatInTime.strPath = os.path.abspath(args.strPath)
- string acsDynStatInTime.strSubStrKey = '*'
- tuple acsDynStatInTime.filesToAnal = sorted(glob.glob(os.path.join(strPath,strSubStrKey)))
- tuple acsDynStatInTime.lenFilesToAnal = len(filesToAnal)
- int acsDynStatInTime.numOfTraj = 0
- tuple acsDynStatInTime.x = np.array(map(list, np.loadtxt(fileToAnal, str)), float)
- acsDynStatInTime.xsize = x.shape
- list acsDynStatInTime.run4Chem = xsize[0]
- tuple acsDynStatInTime.y = np.zeros((lenFilesToAnal,numOfTraj))
- int acsDynStatInTime.pos = 0
- int acsDynStatInTime.chem = 1
- string acsDynStatInTime.tmpHead = ''
- string acsDynStatInTime.filename = "_arranged_"

13.21 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/underDevelopment/acsFromWim2Carness.py File Reference

Namespaces

- acsFromWim2Carness

Variables

- tuple acsFromWim2Carness.parser
- tuple acsFromWim2Carness.args = parser.parse_args()
- string acsFromWim2Carness.ndn = '_0_new_allStatResults'
- tuple acsFromWim2Carness.newdirAllResults = os.path.join(args.strOut, ndn)
- tuple acsFromWim2Carness.fname_initRafRes = os.path.join(newdirAllResults, '0_initRafAnalysis.csv')
- tuple acsFromWim2Carness.fname_initRafResLIST = os.path.join(newdirAllResults, '0_initRafAnalysisLIST.csv')
- tuple acsFromWim2Carness.fname_initRafResALL = os.path.join(newdirAllResults, '0_initRafAnalysisALL.csv')
- tuple acsFromWim2Carness.fid_initRafRes = open(fname_initRafRes, 'w')
- tuple acsFromWim2Carness.fid_initRafResLIST = open(fname_initRafResLIST, 'w')
- tuple acsFromWim2Carness.fid_initRafResALL = open(fname_initRafResALL, 'w')
- string acsFromWim2Carness.strToWrite = "Folder\tP\tAC\tM\tRAFsize\tClosure\tCats\tuRAF\n"
- tuple acsFromWim2Carness.f = open(args.wimFile)
- tuple acsFromWim2Carness.lines = f.readlines()
- int acsFromWim2Carness.procedure = 0
- list acsFromWim2Carness.speciesList = []
- list acsFromWim2Carness.foodList = []
- int acsFromWim2Carness.rct = 0
- tuple acsFromWim2Carness.l = line.split()
- tuple acsFromWim2Carness.catNums = len(l)
- tuple acsFromWim2Carness.cats = np.vstack([cats,(int(rct), int(speciesList.index(l[6+catNums-1][0:len(l[6+catNums-1])-1])), int(rct), int(0), int(5), int(25), int(5), int(1))])
- tuple acsFromWim2Carness.rafsets = raf.rafComputation(fid_initRafRes, fid_initRafResALL, fid_initRafResLIST, 'tmpDir', 0, rct/float(len(speciesList)), rcts, cats, foodList, 10)

13.22 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/underDevelopment/acsRAFanalysis.py File Reference

Namespaces

- acsRAFanalysis

Variables

- tuple acsRAFanalysis.parser
- tuple acsRAFanalysis.args = parser.parse_args()
- tuple acsRAFanalysis.strPath = os.path.abspath(args.strPath)
- int acsRAFanalysis._CLOSE_ = 0
- int acsRAFanalysis._PROTO_ = 1
- int acsRAFanalysis._CSTR_ = 2
- tuple acsRAFanalysis.conf = rf.readConfFile(strPath)
- tuple acsRAFanalysis.closure = dm.generateFluxList(strPath, sysType)
- tuple acsRAFanalysis.foodSet = deepcopy(closure)
- tuple acsRAFanalysis.rcts = rf.loadAllData(strPath,'_acsreactions.csv')
- tuple acsRAFanalysis.cats = rf.loadAllData(strPath,'_acscatalysis.csv')
- tuple acsRAFanalysis.RA = raf.RAcondition(strPath,closure,rcts,cats)
- tuple acsRAFanalysis.RAF = raf.Fcondition(strPath,closure,RA,rcts)
- tuple acsRAFanalysis.RAFlpre = len(RAF)
- list acsRAFanalysis.redRcts = rcts[RAF,:]
- int acsRAFanalysis.RAFlpost = 0
- tuple acsRAFanalysis.catalists = raf.findCatforRAF(cats, RAF, closure)

13.23 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/underDevelopment/acsSCCanalysis.py File Reference

Namespaces

- acsSCCanalysis

Functions

- def acsSCCanalysis.zeroBeforeStrNum
- def acsSCCanalysis.loadReactionGraph
- def acsSCCanalysis.loadSpecificReactionGraph
- def acsSCCanalysis.loadSpecificReactionSubGraph
- def acsSCCanalysis.saveGraphToFile
- def acsSCCanalysis.saveGraphSUBToFile
- def acsSCCanalysis.saveNrgToFile
- def acsSCCanalysis.saveGillToFile

Variables

- tuple acsSCCanalysis.StrPath = os.path.abspath(StrPath)
- tuple acsSCCanalysis.today = dt.date.today()
- tuple acsSCCanalysis.mswindows = (sys.platform == "win32")
- string acsSCCanalysis.cmdFileName = StrPath+'/'
- tuple acsSCCanalysis.cmdFileFid = open(cmdFileName, 'a')
- string acsSCCanalysis.strToWrite = "\tReaction Probability"
- int acsSCCanalysis.initilizeGraphStructure = 0
- tuple acsSCCanalysis.graph = loadSpecificReactionGraph()
- tuple acsSCCanalysis.graphSUB = loadSpecificReactionSubGraph()
- tuple acsSCCanalysis.tmpRctFileToLoadSplitString = tmpRctFileToLoad.split("_")
- tuple acsSCCanalysis.initRctId = int(tmpRctFileToLoadSplitString[len(tmpRctFileToLoadSplitString)-2])
- list acsSCCanalysis.initTempTime = tmpRctFileToLoadSplitString[len(tmpRctFileToLoadSplitString)-1]
- tuple acsSCCanalysis.initTime = float(initTempTime[0:len(initTempTime)-4])
- tuple acsSCCanalysis.tmpDirs = sort(os.listdir(StrPath))
- tuple acsSCCanalysis.totDirName = os.path.join(StrPath,tmpDir)
- tuple acsSCCanalysis.resDirPath = os.path.abspath(os.path.join("./", "res"))
- tuple acsSCCanalysis.numberOfGen = len(glob.glob(os.path.join(resDirPath,'times_*')))
- tuple acsSCCanalysis.strZeros = zeroBeforeStrNum(nGen, numberOfGen)
- string acsSCCanalysis.paramFile = "acsm2s.conf"
- int acsSCCanalysis.simFolder = 0
- tuple acsSCCanalysis.fid = open(paramFile, 'r')
- tuple acsSCCanalysis.strLine = line.split('=')
- tuple acsSCCanalysis.rp = float(strLine[1])
- tuple acsSCCanalysis.decayTime = int(strLine[1])
- tuple acsSCCanalysis.totTimes = int(strLine[1])
- tuple acsSCCanalysis.nrgType = int(strLine[1])
- tuple acsSCCanalysis.totalRcts = int(strLine[1])
- tuple acsSCCanalysis.nrgConc = float(strLine[1])
- tuple acsSCCanalysis.influx_rate = float(strLine[1])
- tuple acsSCCanalysis.maxLOut = float(strLine[1])
- int acsSCCanalysis._ANALTIMEINTERAVAL_ = totTimes/10
- int acsSCCanalysis._ANALTIMEINTERAVALNOSAVE_ = totTimes/100
- int acsSCCanalysis.nrg = 1

- tuple acsSCCanalysis.speciesInFlux = range(0,int(pow(2,(maxLOut+1)) - 2))
- tuple acsSCCanalysis.fidflux = open('_acsinflux.csv', 'r')
- tuple acsSCCanalysis.overThreshold = float(0)
- tuple acsSCCanalysis.overThresholdTOT = float(0)
- string acsSCCanalysis.tmpFilesToSearch = 'species_'
- tuple acsSCCanalysis.speciesFiles = sorted(glob.glob(os.path.join(resDirPath,tmpFilesToSearch)))
- list acsSCCanalysis.speciesFile = speciesFiles[-1]
- tuple acsSCCanalysis.fidSpecies = open(speciesFile, 'r')
- int acsSCCanalysis.ok = 0
- list acsSCCanalysis.IDsOverThreshold = []
- list acsSCCanalysis.concVec = []
- tuple acsSCCanalysis.index = int(tmpID)
- tuple acsSCCanalysis.conc = float(tmpConc)
- tuple acsSCCanalysis.cpxCut = int(tmpCpxCut)
- tuple acsSCCanalysis.age = float(tmpAge)
- acsSCCanalysis.realT = threshold
- string acsSCCanalysis.folderCat = '__0_iGraph_CAT_'
- string acsSCCanalysis.folderSub = '__0_iGraph_SUB_'
- tuple acsSCCanalysis.newdir = os.path.join(os.curdir, folderCat)
- tuple acsSCCanalysis.newdirSUB = os.path.join(os.curdir, folderSub)
- string acsSCCanalysis.filextPre = '_'
- string acsSCCanalysis.rctParamFileQ = 'reactions_parameters_'
- tuple acsSCCanalysis.rctParamFile = sorted(glob.glob(os.path.join(resDirPath,rctParamFileQ)))
- float acsSCCanalysis.rctIDshow = 1.0
- float acsSCCanalysis.rctIDshowNoSave = 1.0
- acsSCCanalysis.rctID = initRctId
- int acsSCCanalysis.previousTime = 0
- int acsSCCanalysis.endo_condensation_counter = 0
- int acsSCCanalysis.condensation_counter = 0
- int acsSCCanalysis.endo_cleavage_counter = 0
- int acsSCCanalysis.cleavage_counter = 0
- tuple acsSCCanalysis.nrgTimeSeries = np.array([[0,0,0]])
- tuple acsSCCanalysis.gillTimeSeries = np.array([[0,0,0,0]])
- int acsSCCanalysis.tmpFlagLastRctSaved = 0
- tuple acsSCCanalysis.reaction = int(tmpReaction)
- tuple acsSCCanalysis.rtime = float(tmpTime)
- tuple acsSCCanalysis.cc = int(tmpcc)
- tuple acsSCCanalysis.cat = int(tmpCat)
- tuple acsSCCanalysis.mol_I = int(tmpMol_I)
- tuple acsSCCanalysis.mol_II = int(tmpMol_II)
- tuple acsSCCanalysis.mol_III = int(tmpMol_III)
- tuple acsSCCanalysis.loadedMolsConc = float(tmpLoadedMolsConc)
- tuple acsSCCanalysis.loadedMols = int(tmpLoadedMols)
- tuple acsSCCanalysis.gillMean = float(tmpGillMean)
- tuple acsSCCanalysis.gillSD = float(tmpGillSD)
- tuple acsSCCanalysis.gillEntropy = float(tmpGillEntropy)
- tuple acsSCCanalysis.newSpeciesCreationProb = float(tmpNSCprob)
- tuple acsSCCanalysis.reverseProbability = float(tmpRevProb)
- int acsSCCanalysis.printTemporalMessage = 1
- acsSCCanalysis.timeInterval = rtime-previousTime
- tuple acsSCCanalysis.position = ((graph[:,0] == cat) & (graph[:,1] == mol_I))
- int acsSCCanalysis.realSccs = 0
- tuple acsSCCanalysis.Gcatpro = nx.DiGraph()
- tuple acsSCCanalysis.scc = nx.strongly_connected_components(Gcatpro)
- tuple acsSCCanalysis.sccN = nx.number_strongly_connected_components(Gcatpro)

- tuple acsSCCanalysis.selfLoops = Gcatpro.number_of_selfloops()
- tuple acsSCCanalysis.selfLoopsEgdes = Gcatpro.selfloop_edges()
- int acsSCCanalysis.prod_inSCC = 0
- int acsSCCanalysis.prod_chain = 0
- int acsSCCanalysis.prod_bySCC = 0
- int acsSCCanalysis.prod_overlap = 0
- int acsSCCanalysis.sccID = 0
- int acsSCCanalysis.autocatalysis = 0
- int acsSCCanalysis.prod_inSCC_weight = 0
- int acsSCCanalysis.prod_chain_weight = 0
- int acsSCCanalysis.prod_bySCC_weight = 0
- int acsSCCanalysis.prod_overlap_weight = 0
- int acsSCCanalysis.self_loop_weight = 0
- int acsSCCanalysis.conc_inSCC = 0
- int acsSCCanalysis.conc_chain = 0
- int acsSCCanalysis.conc_bySCC = 0
- int acsSCCanalysis.conc_overLap = 0
- int acsSCCanalysis.conc_selfCat = 0
- int acsSCCanalysis.wasteSpecies = 0
- int acsSCCanalysis.alreadyAdded_ACS = 0
- int acsSCCanalysis.alreadyAdded_leaves = 0
- int acsSCCanalysis.alreadyAdded_chain = 0
- int acsSCCanalysis.tmpProd_chain = 0
- tuple acsSCCanalysis.incomingNodes = Gcatpro.predecessors(IdsOT)
- int acsSCCanalysis.tempProd_chain_weight = 0
- int acsSCCanalysis.noInAcs = 1
- int acsSCCanalysis.inSCCFlag = 0
- list acsSCCanalysis.weightToDistribute = graph[((graph[:,0] == sngInNode) & (graph[:,1] == IdsOT)),5]
- tuple acsSCCanalysis.inDegreeMean = mean(Gcatpro.in_degree().values())
- tuple acsSCCanalysis.meanOverThreshold = float(overThreshold)

13.24 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/underDevelopment/acsSpeciesActivities.py File Reference

Namespaces

- acsSpeciesActivities

Functions

- def acsSpeciesActivities.zeroBeforeStrNum

Variables

- tuple acsSpeciesActivities.StrPath = os.path.abspath(StrPath)
- tuple acsSpeciesActivities.tmpDirs = sort(os.listdir(StrPath))
- int acsSpeciesActivities.chemistry = 1
- string acsSpeciesActivities.ndn = '_0_new_allStatResults'
- tuple acsSpeciesActivities.newdirAllResults = os.path.join(StrPath, ndn)
- tuple acsSpeciesActivities.newdirAllResultsInner = os.path.join(StrPath,'_0_new_allStatResults',ndn)
- tuple acsSpeciesActivities.totDirName = os.path.join(StrPath,tmpDir)
- tuple acsSpeciesActivities.resDirPath = os.path.abspath(os.path.join("./", "res"))

- tuple acsSpeciesActivities.numberOfGen = len(glob.glob(os.path.join(resDirPath,'times_*')))
- string acsSpeciesActivities.tmpSpeciesStatsSummaryName = 'speciesStatsSummary_'
- tuple acsSpeciesActivities.tmpSpeciesStatsSummaryNameFID = open(tmpSpeciesStatsSummaryName, 'w')
- string acsSpeciesActivities.tmpStr = '\n------- CHEMISTRY '
- tuple acsSpeciesActivities.strZeros = zeroBeforeStrNum(ngen, numberOfGen)
- string acsSpeciesActivities.strSpeciesZero = 'species_'
- tuple acsSpeciesActivities.speciesFilesZero = sorted(glob.glob(os.path.join(resDirPath,strSpeciesZero)))
- string acsSpeciesActivities.strSpecies = 'species_'
- tuple acsSpeciesActivities.speciesFiles = sorted(glob.glob(os.path.join(resDirPath,strSpecies)))
- list acsSpeciesActivities.lastfilespecies = speciesFiles[-1]
- tuple acsSpeciesActivities.fidSpecies = open(lastfilespecies, 'r')
- list acsSpeciesActivities.seq = []
- tuple acsSpeciesActivities.counters = np.zeros((nSpecies,3))
- string acsSpeciesActivities.strRctPar = 'reactions_parameters_'
- tuple acsSpeciesActivities.fidRctPar = open(strRctPar, 'r')
- tuple acsSpeciesActivities.rctType = int(tmpRctType)
- tuple acsSpeciesActivities.cat = int(tmpCat)
- tuple acsSpeciesActivities.S1 = int(tmpS1)
- tuple acsSpeciesActivities.S2 = int(tmpS2)
- tuple acsSpeciesActivities.S3 = int(tmpS3)
- string acsSpeciesActivities.tmpFileName = 'speciesStats_'
- tuple acsSpeciesActivities.tmpFileNameFID = open(tmpFileName, 'w')
- int acsSpeciesActivities.ID = 0

13.25 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/underDevelopment/acsStatesAnalysis.py File Reference

Namespaces

- acsStatesAnalysis

Functions

- def acsStatesAnalysis.zeroBeforeStrNum
- def acsStatesAnalysis.returnZeroSpeciesList
- def acsStatesAnalysis.distanceMisures

Variables

- tuple acsStatesAnalysis.today = dt.date.today()
- tuple acsStatesAnalysis.StrPath = os.path.abspath(StrPath)
- tuple acsStatesAnalysis.tmpDirs = sort(os.listdir(StrPath))
- string acsStatesAnalysis.currentDir = ''
- string acsStatesAnalysis.ndn = currentDir+'_0_new_allStatResults'
- tuple acsStatesAnalysis.newdirAllResults = os.path.join(os.curdir, ndn)
- tuple acsStatesAnalysis.previousFILE_FID = open('STAT_t_tminus_1.csv', 'w')
- tuple acsStatesAnalysis.previousNOINFLUX_FILE_FID = open('STAT_t_tminus_1_NOINFLUX.csv', 'w')
- tuple acsStatesAnalysis.startFILE_FID = open('STAT_t_start.csv', 'w')
- tuple acsStatesAnalysis.startNOINFLUX_FILE_FID = open('STAT_t_start_NOINFLUX.csv', 'w')
- tuple acsStatesAnalysis.HAM_previousFILE_FID = open('STAT_HAM_t_tminus_1.csv', 'w')

- tuple acsStatesAnalysis.HAM_previousNOINFLUX_FILE_FID = open('STAT_HAM_t_tminus_1_NOINFLUX.csv', 'w')
- tuple acsStatesAnalysis.HAM_startFILE_FID = open('STAT_HAM_t_start.csv', 'w')
- tuple acsStatesAnalysis.HAM_startNOINFLUX_FILE_FID = open('STAT_HAM_t_start_NOINFLUX.csv', 'w')
- tuple acsStatesAnalysis.EUC_previousFILE_FID = open('STAT_EUC_t_tminus_1.csv', 'w')
- tuple acsStatesAnalysis.EUC_previousNOINFLUX_FILE_FID = open('STAT_EUC_t_tminus_1_NOINFLUX.csv', 'w')
- tuple acsStatesAnalysis.EUC_startFILE_FID = open('STAT_EUC_t_start.csv', 'w')
- tuple acsStatesAnalysis.EUC_startNOINFLUX_FILE_FID = open('STAT_EUC_t_start_NOINFLUX.csv', 'w')
- tuple acsStatesAnalysis.ANG_middlePreviousFILE_FID = open('STAT_ANG_t_middle_NOINFLUX.csv', 'w')
- tuple acsStatesAnalysis.HAM_middlePreviousFILE_FID = open('STAT_HAM_t_middle_NOINFLUX.csv', 'w')
- tuple acsStatesAnalysis.EUC_middlePreviousFILE_FID = open('STAT_EUC_t_middle_NOINFLUX.csv', 'w')
- tuple acsStatesAnalysis.previousFILE_FID_group = open('STAT_t_tminus_1_group.csv', 'w')
- tuple acsStatesAnalysis.previousNOINFLUX_FILE_FID_group = open('STAT_t_tminus_1_NOINFLUX_group.csv', 'w')
- tuple acsStatesAnalysis.startFILE_FID_group = open('STAT_t_start_group.csv', 'w')
- tuple acsStatesAnalysis.startNOINFLUX_FILE_FID_group = open('STAT_t_start_NOINFLUX_group.csv', 'w')
- tuple acsStatesAnalysis.HAM_previousFILE_FID_group = open('STAT_HAM_t_tminus_1_group.csv', 'w')
- tuple acsStatesAnalysis.HAM_previousNOINFLUX_FILE_FID_group = open('STAT_HAM_t_tminus_1_NOINFLUX_group.csv', 'w')
- tuple acsStatesAnalysis.HAM_startFILE_FID_group = open('STAT_HAM_t_start_group.csv', 'w')
- tuple acsStatesAnalysis.HAM_startNOINFLUX_FILE_FID_group = open('STAT_HAM_t_start_NOINFLUX_group.csv', 'w')
- tuple acsStatesAnalysis.EUC_previousFILE_FID_group = open('STAT_EUC_t_tminus_1_group.csv', 'w')
- tuple acsStatesAnalysis.EUC_previousNOINFLUX_FILE_FID_group = open('STAT_EUC_t_tminus_1_NOINFLUX_group.csv', 'w')
- tuple acsStatesAnalysis.EUC_startFILE_FID_group = open('STAT_EUC_t_start_group.csv', 'w')
- tuple acsStatesAnalysis.EUC_startNOINFLUX_FILE_FID_group = open('STAT_EUC_t_start_NOINFLUX_group.csv', 'w')
- tuple acsStatesAnalysis.newSpecies_FID = open('STAT_GENERAL_newSpecies.csv', 'w')
- tuple acsStatesAnalysis.livingSpecies_FID = open('STAT_GENERAL_livingSpecies.csv', 'w')
- tuple acsStatesAnalysis.mols_FID = open('STAT_GENERAL_mols.csv', 'w')
- tuple acsStatesAnalysis.totMass_FID = open('STAT_GENERAL_overallMass.csv', 'w')
- tuple acsStatesAnalysis.totOverallMass_FID = open('STAT_GENERAL_overallTotMass.csv', 'w')
- tuple acsStatesAnalysis.complex_FID = open('STAT_GENERAL_complex.csv', 'w')
- tuple acsStatesAnalysis.complexMols_FID = open('STAT_GENERAL_complexMols.csv', 'w')
- tuple acsStatesAnalysis.evaluatedFID = open('STAT_GENERAL_evaluated.csv', 'w')
- tuple acsStatesAnalysis.zeroOneSpeciesFID = open('STAT_GENERAL_zeroOneSpecies.csv', 'w')
- tuple acsStatesAnalysis.biodeversityFID = open('STAT_GENERAL_bioDiversity.csv', 'w')
- int acsStatesAnalysis.validDir = 1
- tuple acsStatesAnalysis.totDirName = os.path.join(StrPath,tmpDir)
- tuple acsStatesAnalysis.resDirPath = os.path.abspath(os.path.join("./", "res"))
- tuple acsStatesAnalysis.numberOfGen = len(glob.glob(os.path.join(resDirPath,'times_*')))
- list acsStatesAnalysis.group_A_prev = []
- list acsStatesAnalysis.group_A_start = []
- list acsStatesAnalysis.group_A_prev_NI = []
- list acsStatesAnalysis.group_A_start_NI = []
- tuple acsStatesAnalysis.strZeros = zeroBeforeStrNum(ngen, numberOfGen)
- string acsStatesAnalysis.strSpeciesZero = 'species_'
- tuple acsStatesAnalysis.speciesFilesZero = sorted(glob.glob(os.path.join(resDirPath,strSpeciesZero)))
- string acsStatesAnalysis.strSpecies = 'species_'
- tuple acsStatesAnalysis.speciesFiles = sorted(glob.glob(os.path.join(resDirPath,strSpecies)))
- tuple acsStatesAnalysis.zeroList = returnZeroSpeciesList(speciesFiles[-1])
- tuple acsStatesAnalysis.speciesConcs = np.zeros((len(speciesFiles)+1,len(zeroList)))

- list acsStatesAnalysis.seqOLD = []
- list acsStatesAnalysis.seqSTART = []
- list acsStatesAnalysis.totMass = []
- list acsStatesAnalysis.seqMIDDLE_NOINFLUX = []
- int acsStatesAnalysis.oldNumberOfSpecies = 0
- tuple acsStatesAnalysis.fidSpecies = open(sngSpeciesFile, 'r')
- list acsStatesAnalysis.seq = []
- int acsStatesAnalysis.tmpMols = 0
- int acsStatesAnalysis.bioDivInd = 0
- acsStatesAnalysis.deltaNspecies = numberOfSpecies-oldNumberOfSpecies
- tuple acsStatesAnalysis.strtoW = str(deltaNspecies)
- tuple acsStatesAnalysis.tmpMisure = distanceMisures(seq, conc, seqOLD, concOLD, idS)
- list acsStatesAnalysis.seqOLDNOINFLUX = seqNOINFLUX[:]
- list acsStatesAnalysis.concOLD = conc[:]
- string acsStatesAnalysis.filename = "STAT_species_Concentrations_"

13.26 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/underDevelopment/init.py File Reference

Namespaces

- init

Variables

- tuple init.parser
- tuple init.args = parser.parse_args()
- string init.ndn = '_0_new_allStatResults'
- tuple init.newdirAllResults = os.path.join(args.strOut, ndn)
- tuple init.fname_initRafRes = os.path.join(newdirAllResults, '0_initRafAnalysis.csv')
- tuple init.fname_initRafResSUM = os.path.join(newdirAllResults, '0_initRafAnalysisSUM.csv')
- tuple init.fname_initRafResLIST = os.path.join(newdirAllResults, '0_initRafAnalysisLIST.csv')
- tuple init.fname_initRafResALL = os.path.join(newdirAllResults, '0_initRafAnalysisALL.csv')
- tuple init.fid_initRafRes = open(fname_initRafRes, 'w')
- tuple init.fid_initRafResSUM = open(fname_initRafResSUM, 'w')
- tuple init.fid_initRafResLIST = open(fname_initRafResLIST, 'w')
- tuple init.fid_initRafResALL = open(fname_initRafResALL, 'w')
- string init.strToWrite = "Folder\tP\tAC\tM\tRAFsize\tClosure\tCats\tuRAF\n"
- tuple init.foodList = range(args.lastFood+1)
- tuple init.avgCon = dn.rangeFloat(float(args.avgCon[0]), float(args.avgCon[1]), float(args.avgCon[2]))
- int init.raffound = 0
- list init.alphabet = ['A', 'B']
- list init.species = []
- int init.totSpecies = 2
- tuple init.totCleavage = sum(map(lambda x: len(x)-1,species))
- int init.totCond = totSpecies**2
- init.totRcts = totCleavage+totCond
- tuple init.rctToCat = int(round(totRcts * totSpecies * prob))
- int init.nCleavage = 0
- int init.nCondensa = 0
- tuple init.initSpeciesListLength = len(species)
- tuple init.conf = (1,1,2000,0,200000,0,0,2,args.lastFood,prob)

- int init.rctType = 1
- tuple init.molToCleav = ran.choice(species[len(alphabet):initSpeciesListLength-1])
- tuple init.cutPt = ran.randint(1,len(molToCleav)-1)
- list init.tmp1 = molToCleav[0:cutPt]
- tuple init.tmp1id = species.index(tmp1)
- init.find1 = True
- list init.tmp2 = molToCleav[cutPt:len(molToCleav)]
- tuple init.tmp2id = species.index(tmp2)
- tuple init.sub1 = ran.choice(species[:initSpeciesListLength-1])
- tuple init.idsub1 = species.index(sub1)
- tuple init.sub2 = ran.choice(species[:initSpeciesListLength-1])
- tuple init.idsub2 = species.index(sub2)
- init.prod = sub1+sub2
- tuple init.tmpprodid = species.index(prod)
- int init.catalyst = -1
- init.catFound = False
- tuple init.rafsets = raf.rafComputation(fid_initRafRes, fid_initRafResALL, fid_initRafResLIST, 'tmpDir', prob, averageConn, rcts, cats, foodList, maxlength)

13.27 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/underDevelopment/lib/-__init__.py File Reference

Namespaces

- lib

13.28 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/underDevelopment/lib/dyn/-__init__.py File Reference

Namespaces

- lib.dyn

13.29 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/underDevelopment/lib/graph/-__init__.py File Reference

Namespaces

- lib.graph

13.30 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/underDevelopment/lib/-IO/__init__.py File Reference

Namespaces

- lib.IO

13.31 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/underDevelopment/lib/dyn/dynamics File Reference

Namespaces

- lib.dyn.dynamics

Functions

- def lib.dyn.dynamics.generateFluxList
- def lib.dyn.dynamics.rangeFloat
- def lib.dyn.dynamics.fluxAnalysis

13.32 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/underDevelopment/lib/graph/networ File Reference

Namespaces

- lib.graph.network

Functions

- def lib.graph.network.removeRareRcts
- def lib.graph.network.fixCondensationReaction

13.33 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/underDevelopment/lib/graph/raf.py File Reference

Namespaces

- lib.graph.raf

Functions

- def lib.graph.raf.generateClosure
- def lib.graph.raf.RAcondition
- def lib.graph.raf.Fcondition
- def lib.graph.raf.findCatforRAF
- def lib.graph.raf.rafsearch
- def lib.graph.raf.rafComputation
- def lib.graph.raf.rafDynamicComputation
- def lib.graph.raf.findRAFrcts

13.34 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/underDevelopment/lib/-IO/readfiles.py File Reference

Namespaces

- lib.IO.readfiles

Functions

- def lib.IO.readfiles.readConfFile
- def lib.IO.readfiles.readInitConfFile
- def lib.IO.readfiles.readBufferedID
- def lib.IO.readfiles.readCSTRflux
- def lib.IO.readfiles.loadAllData
- def lib.IO.readfiles.zeroBeforeStrNum
- def lib.IO.readfiles.splitRctParsLine

13.35 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/underDevelopment/lib/-IO/writefiles.py File Reference

Namespaces

- lib.IO.writefiles

Functions

- def lib.IO.writefiles.write_init_raf_list
- def lib.IO.writefiles.write_init_raf_all

13.36 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/underDevelopment/main.py File Reference

Namespaces

- main

Variables

- tuple main.parser
- tuple main.args = parser.parse_args()
- tuple main.strPath = os.path.abspath(args.strPath)
- tuple main.tmpDirs = sort(os.listdir(strPath))
- string main.ndn = '_0_new_allStatResults'
- tuple main.newdirAllResults = os.path.join(strPath, ndn)
- int main._CLOSE_ = 0
- int main._PROTO_ = 1
- int main._CSTR_ = 2
- tuple main.fname_initRafRes = os.path.join(newdirAllResults, '0_initRafAnalysis.csv')
- tuple main.fname_initRafResLIST = os.path.join(newdirAllResults, '0_initRafAnalysisLIST.csv')
- tuple main.fname_initRafResALL = os.path.join(newdirAllResults, '0_initRafAnalysisALL.csv')
- tuple main.fid_initRafRes = open(fname_initRafRes, 'w')
- tuple main.fid_initRafResLIST = open(fname_initRafResLIST, 'w')
- tuple main.fid_initRafResALL = open(fname_initRafResALL, 'w')
- string main.strToWrite = "Folder\tP\tAC\tM\tRAFsize\tClosureSize\tCatsSize\tuRAF\n"
- tuple main.totDirName = os.path.join(strPath,tmpDir)
- tuple main.resDirPath = os.path.abspath(os.path.join("./", args.resFolder))
- tuple main.conf = readfiles.readConfFile(totDirName)

- tuple main.foodList = dm.generateFluxList(totDirName, sysType)
- tuple main.rcts = readfiles.loadAllData(totDirName,'_acsreactions.csv')
- tuple main.cats = readfiles.loadAllData(totDirName,'_acscatalysis.csv')
- tuple main.numberOfGen = len(glob.glob(os.path.join(resDirPath,'times_*')))
- tuple main.strZeros = readfiles.zeroBeforeStrNum(ngen, numberOfGen)
- string main.fName = 'RAF_structuresInTime_analysis_gen_'
- tuple main.fname_inTimeRafRes = os.path.join(newdirAllResults, fName)
- tuple main.fid_inTimeRafRes = open(fname_inTimeRafRes, 'w')
- main.potential = False
- string main.strRctZero = 'reactions_'
- string main.strCatZero = 'catalysis_'
- tuple main.rctFilesZero = sorted(glob.glob(os.path.join(resDirPath,strRctZero)))
- tuple main.catFilesZero = sorted(glob.glob(os.path.join(resDirPath,strCatZero)))
- string main.strRct = 'reactions_'
- string main.strCat = 'catalysis_'
- tuple main.rctFiles = sorted(glob.glob(os.path.join(resDirPath,strRct)))
- tuple main.catFiles = sorted(glob.glob(os.path.join(resDirPath,strCat)))
- list main.sngTime = conf[2]
- int main.actTime = 0
- list main.procrcts = rcts[rcts[:,5] > 0,:]
- list main.proccats = cats[cats[:,3] > 0,:]
- tuple main.R = raf.rafDynamicComputation(fid_inTimeRafRes, actTime, procrcts[:,0:5], proccats[:,0:5], foodList, potential, rcts, cats, debug=args.debug)
- tuple main.lastRct = readfiles.loadAllData(totDirName,rctFiles[-1])
- tuple main.lastCat = readfiles.loadAllData(totDirName,catFiles[-1])
- tuple main.fname_dynRafRes = os.path.join(newdirAllResults, fName)
- tuple main.fid_dynRafRes = open(fname_dynRafRes, 'w')
- string main.strRctPar = 'reactions_parameters_'
- tuple main.rctParamFile = sorted(glob.glob(os.path.join(resDirPath,strRctPar)))
- tuple main.fid = open(rctParamFile[0], 'r')
- int main.previousTime = 0
- main.decayTime = args.decay
- int main.condensation_counter = 0
- int main.endo_condensation_counter = 0
- int main.cleavage_counter = 0
- int main.endo_cleavage_counter = 0
- int main.nAnal = 1
- int main.rctCurrID = 0
- int main.catCurrID = 0
- main.timeInterval = rtime-previousTime
- tuple main.graph = network.removeRareRcts(graph,2,3,4,timeInterval)
- tuple main.graphSUB = network.removeRareRcts(graphSUB,2,3,4,timeInterval)
- tuple main.onrcts = network.removeRareRcts(onrcts,5,6,7,timeInterval)
- tuple main.oncats = network.removeRareRcts(oncats,3,4,5,timeInterval)
- tuple main.positionR = ((onrcts[:,1] == cc) & (onrcts[:,2] == mol_I) & (onrcts[:,3] == mol_II))
- tuple main.position = ((oncats[:,1] == cat) & (oncats[:,2] == onrcts[positionR,0]))

13.37 /Users/alessandrofilisetti/Documents/GIT/carness/_analysis/underDevelopment/prepareNewSim.py File Reference

Namespaces

- prepareNewSim

Functions

- def prepareNewSim.zeroBeforeStrNum

Variables

- tuple prepareNewSim.parser
- tuple prepareNewSim.args = parser.parse_args()
- tuple prepareNewSim.StrFrom = os.path.abspath(args.StrFrom)
- tuple prepareNewSim.StrTo = os.path.abspath(args.StrTo)
- tuple prepareNewSim.StrFileSpeciesToGetConc = os.path.abspath(args.FileSpeciesToGetConc)
- tuple prepareNewSim.origin = os.getcwd()
- int prepareNewSim._LASTSPECIES_ = 29
- tuple prepareNewSim._REVRCTS_ = int(args.revRct)
- tuple prepareNewSim._RATIOREV_ = int(args.k_revRct)
- float prepareNewSim._CLEAVAGE_ = 25.0
- float prepareNewSim._CONDENSATION_ = 50.0
- float prepareNewSim._COMPLEXFORM_ = 50.0
- tuple prepareNewSim._COMPLEXDISS_ = float(args.kDiss)
- tuple prepareNewSim._INITSPECIESCONC_ = float(args.singleInitConc)
- tuple prepareNewSim.fileDest = os.path.join(StrTo,"_acsinflux.csv")
- tuple prepareNewSim.sourceResFolder = os.path.join(StrFrom,"res")
- tuple prepareNewSim.lastSpeciesFile = sorted(glob.glob('species_∗'))
- tuple prepareNewSim.lastReactionsFile = sorted(glob.glob('reactions_1∗'))
- tuple prepareNewSim.lastCatalysisFile = sorted(glob.glob('catalysis_∗'))
- tuple prepareNewSim.mod = open("acsm2s.conf")
- int prepareNewSim.id = 0
- tuple prepareNewSim.linesplitted = line.split("=")
- tuple prepareNewSim.file = open("acsm2s.conf", "w")
- list prepareNewSim.concs = []
- tuple prepareNewSim.specFileLines = open(StrFileSpeciesToGetConc)
- tuple prepareNewSim.mod_rct = open("_acsreactions.csv")
- int prepareNewSim.flag = 0
- tuple prepareNewSim.catRct = linecache.getline('_acsreactions.csv', int(linesplitted[2])+1)
- tuple prepareNewSim.carRctSplit = catRct.split("\t")

13.38 /Users/alessandrofilisetti/Documents/GIT/carness/_matlabinitializator/crea_catalizzatori.m File Reference

Functions

- distribuendo le catalisi a
  caso viene una distribuzione
  uniforme switch (decisione_catalizzatori)%distribuzione dei catalizzatori 1 random 2 con distribuzione case 1
  k=0
- end catalizzatore (kk, 1)
- id catalizzatore catalizzatore (kk, 2)
- id specie catalizzatore (kk, 3)

Variables

- function [catalizzatore reazione specie_non_esistenti]
- numero_specie = length(firing_disk(:,1))
- for i
- end tot_cond = numero_specie^2
- tot_reaz =tot_cond+tot_cleav
- controllo while tot_reaz
  ∗reactionProbability
  ∗cleavageProbability tot_cleav reactionProbability = input('introduci la nuova reactionProbability \n')
- cleavageProbability = input('introduci la nuova cleavageProbability \n')
- end catalisi_reali = round(tot_reaz∗reactionProbability∗numero_specie)
- reazione = -9999
- specie_non_esistenti {1,1} =-9999
- else h = 0
- indice reazione kk = 0
- indice catalizzatore while
  catalisi_reali k = k+1
- numero_specie_da_togliere =0
- end numero_specie_da_togliere index = ceil(rand∗(numero_specie-numero_specie_da_togliere))+numero_specie_da_togliere
- if fino_a_che_lunghezza_i_polimeri_non_catalizzano == 0 %i monomeri non catalizzano 0

13.38.1 Function Documentation

13.38.1.1 end catalizzatore (kk , 1)

13.38.1.2 id catalizzatore catalizzatore (kk , 2)

13.38.1.3 id specie catalizzatore (kk , 3)

13.38.1.4 distribuendo le catalisi a caso viene una distribuzione uniforme switch (decisione_catalizzatori) [pure virtual]

13.38.2 Variable Documentation

13.38.2.1 if catalisi_reali = round(tot_reaz∗reactionProbability∗numero_specie)

Definition at line 21 of file crea_catalizzatori.m.

13.38.2.2 cleavageProbability = input('introduci la nuova cleavageProbability \n')

Definition at line 18 of file crea_catalizzatori.m.

13.38.2.3 if fino_a_che_lunghezza_i_polimeri_non_catalizzano == 0 %i monomeri non catalizzano 0

Definition at line 54 of file crea_catalizzatori.m.

13.38.2.4 function[catalizzatore reazione specie_non_esistenti]

Initial value:

```
=crea_catalizzatori (firing_disk, reactionProbability,
      decisione_catalizzatori,
      fino_a_che_lunghezza_i_polimeri_non_catalizzano,
      alphabet, cleavageProbability, specie_def)
%function [catalizzatore reazione specie_non_esistenti]=crea_catalizzatori (
      firing_disk, reactionProbability,
      decisione_catalizzatori,
      fino_a_che_lunghezza_i_polimeri_non_catalizzano,
      alphabet, cleavageProbability, specie_def)

tot_cleav = 0
```

Definition at line 1 of file crea_catalizzatori.m.

13.38.2.5 else h = 0

Definition at line 29 of file crea_catalizzatori.m.

13.38.2.6 for i

Initial value:

```
=1:numero_specie
    tot_cleav = tot_cleav + firing_disk(i,2)-1
```

Definition at line 7 of file crea_catalizzatori.m.

13.38.2.7 else index = ceil(rand∗(numero_specie-numero_specie_da_togliere))+numero_specie_da_togliere

Definition at line 49 of file crea_catalizzatori.m.

13.38.2.8 indice catalizzatore while catalisi_reali k = k+1

Definition at line 40 of file crea_catalizzatori.m.

13.38.2.9 kk = 0

Definition at line 36 of file crea_catalizzatori.m.

13.38.2.10 numero_specie = length(firing_disk(:,1))

Definition at line 5 of file crea_catalizzatori.m.

13.38.2.11 numero_specie_da_togliere =0

Definition at line 43 of file crea_catalizzatori.m.

13.38.2.12 controllo while tot_reaz∗ reactionProbability∗ cleavageProbability tot_cleav reactionProbability = input('introduci la nuova reactionProbability \n')

Definition at line 17 of file crea_catalizzatori.m.

13.38.2.13 reazione = -9999

Definition at line 25 of file crea_catalizzatori.m.

13.38.2.14 id reazione if rand< cleavageProbability%cleveage o condensazione??catalizzatore(kk, 4)=1%caso cleveage trovato=0 while trovato==0 index_reaz=ceil(rand ∗(numero_specie-length(alphabet)))+length(alphabet) punto_di_taglio=ceil(rand ∗(firing_disk(index_reaz, 2)-1)) reazione(k, 1)=k reazione(k, 2)=1%id del cleavage:1 cleavage 0 condensazione reazione(k, 3)=firing_disk(index_reaz, 1) sp=char(specie_def(:,:)) sp1=char(sp(index_reaz, 1:punto_di_taglio)) sp2=char(sp(index_reaz, punto_di_taglio+1:firing_disk(index_reaz, 2))) for i=1:length(specie_def(:, 1)) if strcmp(deblank(sp(i,:)), sp1)==1 reazione(k, 4)=i end if strcmp(deblank(sp(i,:)), sp2)==1 reazione(k, 5)=i end end for i=length(reazione(:, 1)):-1:1 if((sum(reazione(k, 2:5)==reazione(i, 2:5)))==4 &&i∼=k) reazione(k,:)=[ ] k=k-1 catalizzatore(kk, 3)=i trovato=1 break else%CONTROLLARE IL CONTROLLO!!!!!DOPO IL||%TOGLIERE IL CONTROLLO DA QUI E DA SOTTO!!!if(sum(reazione(k, 3:5)==reazione(i, 3:5))==3 &&reazione(k, 2)∼=reazione(i, 2))||((reazione(k, 3)==reazione(i, 3)&&reazione(k, 4)==reazione(i, 5)&&reazione(k, 5)==reazione(i, 4))) trovato=0 break else trovato=1 end end end end else%caso condensazione catalizzatore(kk, 4)=0 trovato=0 while trovato==0 reazione(k, 1)=k reazione(k, 2)=0%id della condensazione:1 cleavage 0 condensazione index_specie_1=ceil(rand ∗(numero_specie)) index_specie_2=ceil(rand ∗(numero_specie)) nuova_specie=[deblank(specie_def(index_specie_1,:)), deblank(specie_def(index_specie_2,:))] nuova_specie=char(nuova_specie) sp=char(specie_def(:,:)) found=0%controllo per vedere se trovo una specie che esiste gir i=1:length(specie_def(:, 1)) if strcmp(deblank(sp(i,:)), deblank(nuova_specie))==1 reazione(k, 3)=i found=1 end if strcmp(deblank(sp(i,:)), deblank(sp(index_specie_1,:)))==1 reazione(k, 4)=i end if strcmp(deblank(sp(i,:)), deblank(sp(index_specie_2,:)))==1 reazione(k, 5)=i end end if found==0%se ho trovato una specie che NON esiste h=h+1 reazione(k, 3)=length(specie_def(:, 1))+h specie_non_esistenti(h)={nuova_specie}for iii=h:-1:1 if strcmp(specie_non_esistenti(h), specie_non_esistenti(iii))==1 &&h∼=iii specie_non_esistenti(h)=[ ] h=h-1 reazione(k, 3)=length(specie_def(:, 1))+iii end end end%controllo che la reazione non esista gir i=length(reazione(:, 1)):-1:1 if(sum(reazione(k, 2:5)==reazione(i, 2:5)))==4 &&i∼=k reazione(k,:)=[ ] k=k-1 catalizzatore(kk, 3)=i trovato=1 break else if(sum(reazione(k, 3:5)==reazione(i, 3:5))==3 &&reazione(k, 2)∼=reazione(i, 2))||((reazione(k, 3)==reazione(i, 3)&&reazione(k, 4)==reazione(i, 5)&&reazione(k, 5)==reazione(i, 4))) trovato=0 break else trovato=1 end end end end end%controllo per evitare che lo stesso catalizzatore catalizzi la%stessa reazione trovato_uguale=1 for iiii=length(catalizzatore(:, 1)):-1:1 if sum(catalizzatore(kk, 2:3)==catalizzatore(iiii, 2:3))==2 &&kk∼=iiii catalizzatore(kk,:)=[ ] kk=kk-1 trovato_uguale=0 break end end if trovato_uguale==1 catalisi_reali=catalisi_reali-1 end end if exist('specie_non_esistenti')> specie_non_esistenti {1,1} =-9999

Definition at line 26 of file crea_catalizzatori.m.

13.38.2.15 end tot_cond = numero_specie^2

Definition at line 10 of file crea_catalizzatori.m.

13.38.2.16 tot_reaz =tot_cond+tot_cleav

Definition at line 12 of file crea_catalizzatori.m.

13.39 /Users/alessandrofilisetti/Documents/GIT/carness/_matlabinitializator/crea_concentrazioni_iniziali.m File Reference

Functions

- firing_disk (:, 3)
- if firing_disk (index, 3)
- else concentrazioni_iniziali (i)=0
- firing_disk_reale (k,:) =firing_disk(i,:)
- id Z ()
- valore (1:max(firing_disk_bck(:, 2)))=0
- valore (i)

- end end end STAMPA
  DISTRIBUZIONI DELLE
  CONCENTRAZIONI figure (123)%plot(valore

Variables

- function [concentrazioni_iniziali] = crea_concentrazioni_iniziali(alphabet,firing_disk,initialMaxLength,lunghezza-_max_fd,ratio_firing_disk, scelta_concentrazioni, overallConcentration, gamma)
- le specie fittizie create fino
  alla
  lunghezza_massima_per_calcolare_le_reazioni
  vengono poste uguali a in
  concentrazione tot_species = 0
- for i
- end lunghezza_totale = length(firing_disk(:,1))
- firing_disk_bck =firing_disk
- firing_disk_2 =firing_disk(1:tot_species,:)
- firing_disk =[]
- check di esistenza nel firing disk numero_molecole
- a seconda dei casi seleziono
  le specie esistenti del firing disk
- a seconda dei casi seleziono
  le specie esistenti del firing
  quelle non esistenti andranno
  semplicemente a concentrazione remaining_species = tot_species
- l_max =0
- species_to_delete = round(ratio_firing_disk∗remaining_species)
- la percentuale reazioni
  fra quelle che restano species_to_keep = remaining_species - species_to_delete + l_max
- while trovato
- end end end switch
  scelta_concentrazioni case
  distribuzione uniforme su
  tutte le specie esistenti del
  firing disk probabilita_uniforme = 1/species_to_keep
- end end case uniforme sulle
  lunghezza k =0
- end end vettore_ordinato_lunghezze = unique(firing_disk_reale(:,2))
- probabilita_per_lunghezza = 1/length(vettore_ordinato_lunghezze)
- end end normalizzazione per
  avere la concentrazione concentrazioni_iniziali =concentrazioni_iniziali/sum(concentrazioni_iniziali)∗overall-Concentration

13.39.1 Function Documentation

13.39.1.1 else concentrazioni_iniziali (i) [pure virtual]

13.39.1.2 end end end STAMPA DISTRIBUZIONI DELLE CONCENTRAZIONI figure (123)

13.39.1.3 firing_disk (: , 3)

13.39.1.4 if firing_disk (index , 3)

13.39.1.5 firing_disk_reale (k , :) =firing_disk(i,:)

13.39.1.6 valore (1: *maxfiring_disk_bck(:, 2)*) [pure virtual]

13.39.1.7 valore (i)

13.39.1.8 id Z ( ) [virtual]

13.39.2 Variable Documentation

13.39.2.1 sum(concentrazioni_iniziali) end concentrazioni_iniziali =concentrazioni_iniziali/sum(concentrazioni_iniziali)∗overallConcentration

Definition at line 129 of file crea_concentrazioni_iniziali.m.

13.39.2.2 a seconda dei casi seleziono le specie esistenti del firing disk

Definition at line 25 of file crea_concentrazioni_iniziali.m.

13.39.2.3 firing_disk =[ ]

Definition at line 17 of file crea_concentrazioni_iniziali.m.

13.39.2.4 firing_disk_2 =firing_disk(1:tot_species,:)

Definition at line 16 of file crea_concentrazioni_iniziali.m.

13.39.2.5 firing_disk_bck =firing_disk

Definition at line 14 of file crea_concentrazioni_iniziali.m.

13.39.2.6 function[concentrazioni_iniziali] = crea_concentrazioni_iniziali(alphabet,firing_disk,initialMaxLength,lunghezza_max_fd,ratio_firing_disk, scelta_concentrazioni, overallConcentration, gamma)

Definition at line 1 of file crea_concentrazioni_iniziali.m.

13.39.2.7 for i

Initial value:

```
= 1:initialMaxLength
    tot_species = tot_species+length(alphabet)^i
```

Definition at line 9 of file crea_concentrazioni_iniziali.m.

13.39.2.8 end end case favorire quelle corte con una scale free di esponente gamma k =0

Definition at line 73 of file crea_concentrazioni_iniziali.m.

13.39.2.9 l_max =0

Definition at line 31 of file crea_concentrazioni_iniziali.m.

13.39.2.10 end lunghezza_totale = length(firing_disk(:,1))

Definition at line 13 of file crea_concentrazioni_iniziali.m.

13.39.2.11 check di esistenza nel firing disk numero_molecole

Initial value:

```
= overallConcentration*volume
numero_specie = length(firing_disk(:,1))
```

Definition at line 22 of file crea_concentrazioni_iniziali.m.

13.39.2.12 probabilita_per_lunghezza = 1/length(vettore_ordinato_lunghezze)

Definition at line 82 of file crea_concentrazioni_iniziali.m.

13.39.2.13 end end end switch scelta_concentrazioni case distribuzione uniforme su tutte le specie esistenti del firing disk probabilita_uniforme = 1/species_to_keep

Definition at line 60 of file crea_concentrazioni_iniziali.m.

13.39.2.14 end remaining_species = tot_species

Definition at line 30 of file crea_concentrazioni_iniziali.m.

13.39.2.15 end end species_to_delete = round(ratio_firing_disk*remaining_species)

Definition at line 38 of file crea_concentrazioni_iniziali.m.

13.39.2.16 la percentuale reazioni fra quelle che restano species_to_keep = remaining_species - species_to_delete + l_max

Definition at line 39 of file crea_concentrazioni_iniziali.m.

13.39.2.17 le specie fittizie create fino alla lunghezza_massima_per_calcolare_le_reazioni vengono poste uguali a in concentrazione tot_species = 0

Definition at line 8 of file crea_concentrazioni_iniziali.m.

13.39.2.18 while trovato

Initial value:

```
==0
                index = l_max+ceil(rand*remaining_species)
```

Definition at line 44 of file crea_concentrazioni_iniziali.m.

13.39.2.19 end end vettore_ordinato_lunghezze = unique(firing_disk_reale(:,2))

Definition at line 81 of file crea_concentrazioni_iniziali.m.

13.40 /Users/alessandrofilisetti/Documents/GIT/carness/_matlabinitializator/crea_e_controlla_i_catalizzatori.m File Reference

Functions

- 1:length(influx(:, 1 matrice_adiacenza_sub_prod: (influx(:, 1 length)
- matrice_adiacenza_sub_prod (1:max(influx(:, 1)), 1:max(influx(:, 1)))=0
- matrice_adiacenza_sub_prod (reazione(i, 5), reazione(i, 3))
- else matrice_adiacenza_sub_prod (reazione(i, 3), reazione(i, 4))
- matrice_adiacenza_sub_prod (reazione(i, 3), reazione(i, 5))
- end end end if max (real(eig(matrice_adiacenza_sub_prod))) > 0 clear catalizzatore clear reazione clear specie_non_esistenti clear matrice_adiacenza_sub_prod check_ACS=0
- id check_ACS ()
- matrice_adiacenza_cat_prod (1:max(influx(:, 1)), 1:max(influx(:, 1)))=0
- else matrice_adiacenza_cat_prod (catalizzatore(i, 2), reazione(catalizzatore(i, 3), 4))
- matrice_adiacenza_cat_prod (catalizzatore(i, 2), reazione(catalizzatore(i, 3), 5))
- end end end if max (real(eig(matrice_adiacenza_cat_prod))) > 0 clear catalizzatore clear reazione clear specie_non_esistenti clear matrice_adiacenza_cat_prod check_ACS=0 = 1
- end end end size (matrice_adiacenza_sub_prod)%size(matrice_adiacenza_cat_prod) if max(real(eig(matrice_adiacenza_sub_prod))) > 0||max(real(eig(matrice_adiacenza_cat_prod))) > 0 clear catalizzatore clear reazione clear specie_non_esistenti clear matrice_adiacenza_cat_prod clear matrice_adiacenza_sub_prod check_ACS=0

Variables

- function [catalizzatore reazione specie_non_esistenti matrice_adiacenza_sub_prod matrice_adiacenza_cat_prod] = crea_e_controlla_i_catalizzatori (controllo_ACS_nel_ciclo, firing_disk, reactionProbability, decisione_catalizzatori, fino_a_che_lunghezza_i_polimeri_non_catalizzano, alphabet, cleavageProbability, specie_def, influx)
- controllo no ACS nell influx
  switch controllo_ACS_nel_ciclo
  case nessun controllo matrice_adiacenza_sub_prod =0
- matrice_adiacenza_cat_prod = 0
- check_ACS = 0
- for i
- inpudda = input(")
- counter_cicli = 0

13.40.1 Function Documentation

13.40.1.1 id check_ACS ( ) [virtual]

13.40.1.2 matrice_adiacenza_cat_prod (1: *maxinflux(:, 1),* 1: *maxinflux(:, 1)*) [pure virtual]

13.40.1.3 else matrice_adiacenza_cat_prod (catalizzatore(i, 2) , reazione(catalizzatore(i, 3), 4))

13.40.1.4 matrice_adiacenza_cat_prod (catalizzatore(i, 2) , reazione(catalizzatore(i, 3), 5))

13.40.1.5 matrice_adiacenza_sub_prod (1: *maxinflux(:, 1),* 1: *maxinflux(:, 1)*) [pure virtual]

13.40.1.6 matrice_adiacenza_sub_prod (reazione(i, 5) , reazione(i, 3))

13.40.1.7 else matrice_adiacenza_sub_prod (reazione(i, 3) , reazione(i, 4))

13.40.1.8 matrice_adiacenza_sub_prod (reazione(i, 3) , reazione(i, 5))

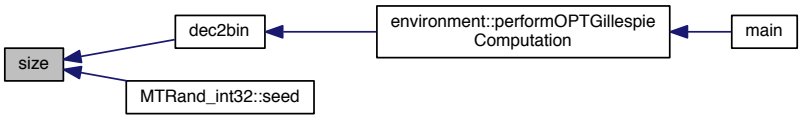
13.40.1.9 1:length(influx(:,1 matrice_adiacenza_sub_prod: ( ) [virtual]

13.40.1.10 end end end if max (real(eig(matrice_adiacenza_sub_prod))) [pure virtual]

13.40.1.11 end end end if max (real(eig(matrice_adiacenza_cat_prod))) = 1 [pure virtual]

13.40.1.12 end end end size (matrice_adiacenza_sub_prod) [pure virtual]

Here is the caller graph for this function:



13.40.2 Variable Documentation

13.40.2.1 while check_ACS = 0

Definition at line 18 of file crea_e_controlla_i_catalizzatori.m.

13.40.2.2 counter_cicli = 0

Definition at line 82 of file crea_e_controlla_i_catalizzatori.m.

13.40.2.3 function[catalizzatore reazione specie_non_esistenti] = crea_e_controlla_i_catalizzatori (controllo_ACS_nel_ciclo, firing_disk, reactionProbability, decisione_catalizzatori, fino_a_che_lunghezza_i_polimeri_non_catalizzano, alphabet, cleavageProbability, specie_def, influx)

Definition at line 1 of file crea_e_controlla_i_catalizzatori.m.

13.40.2.4 for i

Initial value:

```
= 1:length(reazione(:,1))
                if sum(reazione(i,3)==influx(:,1))==1 && sum(
      reazione(i,4)==influx(:,1))==1 && sum(reazione(i,5)==influx(:,1))==1
                    if reazione(i,2)==0
                        matrice_adiacenza_sub_prod(
      reazione(i,4),reazione(i,3))=1
```

Definition at line 26 of file crea_e_controlla_i_catalizzatori.m.

13.40.2.5 inpudda = input(")

Definition at line 44 of file crea_e_controlla_i_catalizzatori.m.

13.40.2.6 matrice_adiacenza_cat_prod = 0

Definition at line 11 of file crea_e_controlla_i_catalizzatori.m.

13.40.2.7 matrice_adiacenza_sub_prod =0

Definition at line 10 of file crea_e_controlla_i_catalizzatori.m.

13.41 /Users/alessandrofilisetti/Documents/GIT/carness/_matlabinitializator/crea_firing_disk.m File Reference

Functions

- step (i)

Variables

- function [firing_disk]
- for i
- end k = 1
- end firing_disk =id_species

13.41.1 Function Documentation

13.41.1.1 step (i)

13.41.2 Variable Documentation

13.41.2.1 end firing_disk =id_species

Definition at line 24 of file crea_firing_disk.m.

13.41.2.2 function[firing_disk]

Initial value:

```
= crea_firing_disk(alphabet,
      massima_lunghezza_su_cui_calcolare_le_reazioni)
%function [firing_disk] = crea_firing_disk(alphabet,
      massima_lunghezza_su_cui_calcolare_le_reazioni)

tot_species = 0
```

Definition at line 3 of file crea_firing_disk.m.

13.41.2.3 for i

Initial value:

```
= 1:massima_lunghezza_su_cui_calcolare_le_reazioni
    tot_species = tot_species+length(alphabet)^i
```

Definition at line 8 of file crea_firing_disk.m.

13.41.2.4 k = 1

Definition at line 18 of file crea_firing_disk.m.

13.42 /Users/alessandrofilisetti/Documents/GIT/carness/_matlabinitializator/crea_influx.m File Reference

Functions

- influx (i, 2)
- influx (i,:)
- if sum (influx(i, 1)==influx(:, 1))
- influx (1, 1:2)=0
- if influx (1, 1)>0%%%for i

Variables

- function [influx]
- end for j
- end end tot_species = length(firing_disk(:,1))
- switch scelta_influx casuale case species_to_delete = round(ratio_influx*tot_species)
- species_to_keep = tot_species - species_to_delete
- for i
- while trovato == 0
- end end end influx =sort(influx)
- casuale tenendo i polimeri
  fino a lunghezza x e poi ne
  toglie l x case remaining_species = tot_species
- end proporzionale alla lunghezza

13.42.1 Function Documentation

13.42.1.1 influx (i , 2)

13.42.1.2 influx (i , :)

13.42.1.3 influx (1 , 1:2) [pure virtual]

13.42.1.4 if influx (1 , 1)

13.42.1.5 if sum (influx(i, 1) ==influx (: , 1))

Initial value:

```
==1
%                       trovato = 1
```

13.42.2 Variable Documentation

13.42.2.1 function[influx]

Initial value:

```
= crea_influx(concentrazioni_iniziali)
%function [influx] = crea_influx(concentrazioni_iniziali)


for i =1:length(concentrazioni_iniziali)

    influx(i,1)=i
```

Definition at line 3 of file crea_influx.m.

13.42.2.2 for i

Initial value:

```
= 1:species_to_keep
%             trovato = 0
```

Definition at line 56 of file crea_influx.m.

13.42.2.3 else clear influx influx =sort(influx)

Definition at line 65 of file crea_influx.m.

13.42.2.4 end for j

Initial value:

```
= length(concentrazioni_iniziali):-1:1
    if influx (j,2)==0
        influx(j,:)=[]
```

Definition at line 13 of file crea_influx.m.

13.42.2.5 end proporzionale alla lunghezza

Definition at line 108 of file crea_influx.m.

13.42.2.6 remaining_species = tot_species

Definition at line 71 of file crea_influx.m.

13.42.2.7 end end species_to_delete = round(ratio_influx∗tot_species)

Definition at line 53 of file crea_influx.m.

13.42.2.8 species_to_keep = tot_species - species_to_delete

Definition at line 54 of file crea_influx.m.

13.42.2.9 end end tot_species = length(firing_disk(:,1))

Definition at line 44 of file crea_influx.m.

13.42.2.10 while trovato == 0

Definition at line 58 of file crea_influx.m.

13.43 /Users/alessandrofilisetti/Documents/GIT/carness/_matlabinitializator/crea_influx-_semplice.m File Reference

Functions

- influx (j, 2)

Variables

- function [influx]
- for i
- end influx = zeros(sn,2)
- for j

13.43.1 Function Documentation

13.43.1.1 influx (j , 2)

13.43.2 Variable Documentation

13.43.2.1 function[influx]

Initial value:

```
= crea_influx_semplice(lMaxInflux,alphabet)

% total Number of species
sn = 0
```

Definition at line 3 of file crea_influx_semplice.m.

13.43.2.2 for i

Initial value:

```
= 1:lMaxInflux
   sn = sn + length(alphabet)^i
```

Definition at line 7 of file crea_influx_semplice.m.

13.43.2.3 end influx = zeros(sn,2)

Definition at line 11 of file crea_influx_semplice.m.

13.43.2.4 for j

Initial value:

```
= 1:sn
   influx(j,1)=j
```

Definition at line 12 of file crea_influx_semplice.m.

13.44 /Users/alessandrofilisetti/Documents/GIT/carness/_matlabinitializator/crea_tutte_-le_combinazioni_di_elementi.m File Reference

Variables

- prova m function [specie]
- numero_elementi = length(vettore_elementi)
- righe = numero_elementi^lunghezza_stringa
- for i

13.44.1 Variable Documentation

13.44.1.1 prova m function[specie]

Initial value:

```
=crea_tutte_le_combinazioni_di_elementi(vettore_elementi,K)
%function [specie]=crea_tutte_le_combinazioni_di_elementi(vettore_elementi)

%dato un vettore di elementi e una lunghezza massima K restituisce tutte le
%possibili combinazioni di elementi ordinati

lunghezza_stringa = K
```

Definition at line 4 of file crea_tutte_le_combinazioni_di_elementi.m.

13.44.1.2 for i

Initial value:

```
=1:lunghezza_stringa %colonne
   for j = 1:numero_elementi
       for z = (numero_elementi^(i-1))*(j-1)+1:numero_elementi^(
      i):righe
           specie(z:z+numero_elementi^(i-1)-1,i)=(vettore_elementi(
      j))
```

Definition at line 14 of file crea_tutte_le_combinazioni_di_elementi.m.

13.44.1.3 numero_elementi = length(vettore_elementi)

Definition at line 11 of file crea_tutte_le_combinazioni_di_elementi.m.

13.44.1.4 righe = numero_elementi^lunghezza_stringa

Definition at line 12 of file crea_tutte_le_combinazioni_di_elementi.m.

13.45 /Users/alessandrofilisetti/Documents/GIT/carness/_matlabinitializator/initial_-distribution.m File Reference

13.46 /Users/alessandrofilisetti/Documents/GIT/carness/_matlabinitializator/inizializzatore-_ACS.m File Reference

Functions

- mkdir (nome_cartella)
- mkdir ('res')
- cd (thisFolder)
- fid1,'nGEN=' fprintf ()
- perch sono input da file (i prossimi 4) count
- id i: (fid10,'%d\t', funzioni_booleane_in_dec(i, 1 fprintf)
- alphabet,
  massima_lunghezza_su_cui_calcolare_le_reazioni crea_firing_disk ()
- alphabet, firing_disk,
  initialMaxLength,
  lunghezza_max_fd,
  ratio_firing_disk,
  scelta_concentrazioni,
  overallConcentration,
  gamma_powerlaw_concentrazioni crea_concentrazioni_iniziali ()
- concentrazioni_iniziali crea_influx ()
- id k ()
- specie_def_2 (k,:)
- controllo_ACS_nel_ciclo,
  firing_disk,
  reactionProbability,
  decisione_catalizzatori,
  fino_a_che_lunghezza_i_polimeri_non_catalizzano,
  alphabet, cleavageProbability,
  specie_def, influx crea_e_controlla_i_catalizzatori ()
- id kk ()
- Punto di tagli del complesso (1--L-1) if concentrazioni_iniziali(i) $>$ 0||sum(i
- end else if reazione (i, 2)
- id specie_non_esistenti ()
- Coefficiente di degradazione (per complessi) count

Variables

- function [firing_disk concentrazioni_iniziali specie_def influx catalizzatore reazione specie_non_esistenti matrice_adiacenza_sub_prod matrice_adiacenza_cat_prod]
- fid1 = fopen('acsm2s.conf','w')
- fid2 = fopen('_acsspecies.csv','a')
- fid3 = fopen('_acsreactions.csv','a')
- fid4 = fopen('_acscatalysis.csv','a')
- fid5 = fopen('_acsinflux.csv','a')
- fid10 = fopen('_acsnrgbooleanfunctions.csv','a')
- count = fprintf(fid1,'%d\n',nGEN)
- st = fclose(fid1)
- vettore_rand =[0.5 1 2]
- counter id specie for i

- for j
- for z
- end end clear specie_temp end k = 0
- end specie_def =specie_def_2
- lunghezza_stringa = length(specie_def(1,:))
- for jjjj
- for xxxx
- end end if controllo
- end end if gg
- end end tmpStr = specie_def(i,:)
- Coefficiente di degradazione del complesso = fprintf(fid2,'%d\t',0)
- concentrazione molecole
  cariche if rand
  < ratioSpeciesEnergizable%count=fprintf(fid2,'%d\n',
  1);%specie energizzabile%else%count=fprintf(fid2,'%d\n',
  0);%specie NON energizzabile%end
  if influx_rate==0 if maxLOut >
  if i<=(2^(maxLOut+1)-2) count=fprintf(fid2,'%d\n',
  1);else count=fprintf(fid2,'%d\n',
  0);end else count=fprintf(fid2,'%d\n',
  0);end else count=fprintf(fid2,'%d\n',
  0);endend%----------%inizializzazione
  del file dell'influx"_influx.csv"%----------for
  i=1:length(influx(:, 1)) count=fprintf(fid5,'%d\t',
  influx(i, 1)-1);count=fprintf(fid5,'%d\n',
  influx(i, 2));endst=fclose(fid5);%----------%inizializzazione
  del file delle reazioni e dei
  catalizzatori%----------if
  catalizzatore(1,
  1)==-9999;else for i=1:length(catalizzatore(:,
  1)) count=fprintf(fid4,'%d\t',
  catalizzatore(i, 1)-1);count=fprintf(fid4,'%d\t',
  catalizzatore(i, 2)-1);count=fprintf(fid4,'%d\t',
  catalizzatore(i, 3)-1);count=fprintf(fid4,'%d\t',
  0);%quante volte if
  catalizzatore(i,
  4)==0 tmpKdiss=Kdiss/revRctRatio;count=fprintf(fid4,'%g\t',
  Kass);%kass count=fprintf(fid4,'%g\t',
  tmpKdiss);%kdiss count=fprintf(fid4,'%g\t',
  Kcpx);%k complex else tmpKass=Kass/revRctRatio;tmpKcpx=Kcpx/revRctRatio;count=fprintf(fid4,'%g\t',
  tmpKass);%kass count=fprintf(fid4,'%g\t',
  Kdiss);%kdiss count=fprintf(fid4,'%g\t',
  tmpKcpx);%k complex end
  index_rand=ceil(rand *2);count=fprintf(fid4,'%g\n',
  index_rand);%nuovo parametro
  end[righe_xx colonne_xx]=size(funzioni_booleane_in_dec);for
  i=1:length(reazione(:,
  1)) indexx=ceil(rand *righe_xx);funzione_giusta=funzioni_booleane_in_dec(indexx);count=fprintf(fid3,'%d\t',
  reazione(i, 1)-1);count=fprintf(fid3,'%d\t',
  reazione(i, 2));count=fprintf(fid3,'%d\t',
  reazione(i, 3)-1);count=fprintf(fid3,'%d\t',
  reazione(i, 4)-1);count=fprintf(fid3,'%d\t',
  reazione(i, 5)-1);count=fprintf(fid3,'%d\t',
  0);if energy==2 eso_endo=1;%perch?se
  non c'?l'energia sono tutte
  eso else if rand > energy if reazione (i, 2)
- else eso_endo =0
- temporal = specie_non_esistenti{i}

13.46.1 Function Documentation

13.46.1.1 cd (thisFolder)

13.46.1.2 Punto di tagli del complesso (1--L- *1*)

13.46.1.3 alphabet,firing_disk,initialMaxLength,lunghezza_max_fd,ratio_firing_disk, scelta_concentrazioni, overallConcentration, gamma_powerlaw_concentrazioni crea_concentrazioni_iniziali ( ) [virtual]

13.46.1.4 controllo_ACS_nel_ciclo, firing_disk, reactionProbability, decisione_catalizzatori, fino_a_che_lunghezza_i_polimeri_non_catalizzano, alphabet, cleavageProbability, specie_def, influx crea_e_controlla_i_catalizzatori ( ) [virtual]

13.46.1.5 alphabet,massima_lunghezza_su_cui_calcolare_le_reazioni crea_firing_disk ( ) [virtual]

13.46.1.6 concentrazioni_iniziali crea_influx ( ) [virtual]

13.46.1.7 Coefficiente di degradazione (per *complessi*)

13.46.1.8 perch sono input da file (i prossimi *4*)

13.46.1.9 fid2 d t fprintf ( ) [virtual]

13.46.1.10 id i: (fid10 , '%d\t') [virtual]

13.46.1.11 id k ( ) [virtual]

13.46.1.12 id kk ( ) [virtual]

13.46.1.13 mkdir (nome_cartella)

13.46.1.14 mkdir ('res')

13.46.1.15 end else if reazione (i , 2)

13.46.1.16 specie_def_2 (k , :)

13.46.1.17 id specie_non_esistenti ( ) [virtual]

Definition at line 366 of file inizializzatore_ACS.m.

13.46.2 Variable Documentation

13.46.2.1 Punto di tagli del complesso = fprintf(fid2,'%d\t',0)

Definition at line 252 of file inizializzatore_ACS.m.

13.46.2.2 end end if controllo

Initial value:

```
== 1
                count = fprintf(fid2,'%c',specie_def(i,
      jjjj))
```

Definition at line 223 of file inizializzatore_ACS.m.

13.46.2.3 k_fosforilazione velocit con cui l atp count = fprintf(fid1,'%d\n',nGEN)

Definition at line 35 of file inizializzatore_ACS.m.

13.46.2.4 else eso_endo =0

Definition at line 348 of file inizializzatore_ACS.m.

13.46.2.5 fid1 = fopen('acsm2s.conf','w')

Definition at line 19 of file inizializzatore_ACS.m.

13.46.2.6 fid10 = fopen('_acsnrgbooleanfunctions.csv','a')

Definition at line 24 of file inizializzatore_ACS.m.

13.46.2.7 fid2 = fopen('_acsspecies.csv','a')

Definition at line 20 of file inizializzatore_ACS.m.

13.46.2.8 fid3 = fopen('_acsreactions.csv','a')

Definition at line 21 of file inizializzatore_ACS.m.

13.46.2.9 fid4 = fopen('_acscatalysis.csv','a')

Definition at line 22 of file inizializzatore_ACS.m.

13.46.2.10 fid5 = fopen('_acsinflux.csv','a')

Definition at line 23 of file inizializzatore_ACS.m.

13.46.2.11 function[firing_disk concentrazioni_iniziali specie_def influx catalizzatore reazione specie_non_esistenti matrice_adiacenza_sub_prod matrice_adiacenza_cat_prod]

Initial value:

```
= inizializzatore_ACS(nGEN, nSIM, nSeconds,
      nReactions, initialMaxLength,
      massima_lunghezza_su_cui_calcolare_le_reazioni,
      overallConcentration, alphabet,
      complexFormationSymmetry,
      fino_a_che_lunghezza_i_polimeri_non_catalizzano,
      reactionProbability, cleavageProbability,
      diffusion_contribute, solubility_threshold,
      influx_rate, reverseReactions,K_nrg,
      moleculeDecay_KineticConstant, ratio_firing_disk,
      lunghezza_max_fd, scelta_concentrazioni,
      gamma_powerlaw_concentrazioni,
      decisione_catalizzatori, lastFiringDiskSpeciesID,
      ECConcentration, volume, energy, controllo_ACS_nel_ciclo, K_nrg_decay, nome_cartella,
      funzioni_booleane_in_dec,ratioSpeciesEnergizable,
      Kass,Kdiss,Kcpx,K_cpx, randomSeed, debugLevel,
      timeStructuresSavingInterval,maxLOut,
      simFolder,lMaxInflux,fileTimesSaveInterval,nHours,nAttempts,
      revRctRatio,newSpeciesProbMinThreshold,volumeGrowth,
      stochDivision)
%function [firing_disk concentrazioni_iniziali specie_def influx catalizzatore reazione
      specie_non_esistenti matrice_adiacenza_sub_prod matrice_adiacenza_cat_prod] = inizializzatore_ACS(
```

```
      nGEN, nSIM, nSeconds, nReactions, initialMaxLength,
      massima_lunghezza_su_cui_calcolare_le_reazioni,
      overallConcentration, alphabet,
      complexFormationSymmetry,
      fino_a_che_lunghezza_i_polimeri_non_catalizzano,
      reactionProbability, cleavageProbability,
      diffusion_contribute, solubility_threshold,
      influx_rate, reverseReactions,K_nrg,
      moleculeDecay_KineticConstant, ratio_firing_disk,
      lunghezza_max_fd, scelta_concentrazioni,
      gamma_powerlaw_concentrazioni,
      decisione_catalizzatori, lastFiringDiskSpeciesID,
      ECConcentration, volume, energy, controllo_ACS_nel_ciclo, K_nrg_decay, nome_cartella,
      funzioni_booleane_in_dec,ratioSpeciesEnergizable,
      Kass,Kdiss,Kcpx,K_cpx, randomSeed, debugLevel,
      timeStructuresSavingInterval,maxLOut,
      simFolder,lMaxInflux,fileTimesSaveInterval,nHours,nAttempts,
      revRctRatio,newSpeciesProbMinThreshold,volumeGrowth,
      stochDivision)


 rand

%------------------------------------------------------------------------
%apertura file
%------------------------------------------------------------------------
thisFolder = pwd
```

Definition at line 1 of file inizializzatore_ACS.m.

13.46.2.12 end end if gg

Initial value:

```
= input ('')
% %              specie_def(i,jjjj)
% %              isspace(specie_def(i,jjjj))
% %              specie_def(i,jjjj)==('')
%              if sum(alphabetspecie_def(i,jjjj)) || specie_def(
      i,jjjj)==(' ')
%              else
%                  count = fprintf(fid2,'%c',specie_def(
      i,jjjj))
```

Definition at line 232 of file inizializzatore_ACS.m.

13.46.2.13 for i

Initial value:

```
= massima_lunghezza_su_cui_calcolare_le_reazioni:-1:1
    specie_temp=crea_tutte_le_combinazioni_di_elementi(alphabet,i)
```

Definition at line 174 of file inizializzatore_ACS.m.

13.46.2.14 for j

Initial value:

```
= length(specie_temp(:,1)):-1:1
        k = k+1
```

Definition at line 177 of file inizializzatore_ACS.m.

13.46.2.15 for jjjj

Initial value:

```
=1:length(specie_def(1,:))
            controllo = 0
```

Definition at line 216 of file inizializzatore_ACS.m.

13.46.2.16 end end clear specie_temp end k = 0

Definition at line 188 of file inizializzatore_ACS.m.

13.46.2.17 lunghezza_stringa = length(specie_def(1,:))

Definition at line 215 of file inizializzatore_ACS.m.

13.46.2.18 concentrazione molecole cariche if rand< ratioSpeciesEnergizable% count = fprintf(fid2,'%d\n',1); % specie energizzabile% else% count = fprintf(fid2,'%d\n',0); % specie NON energizzabile% end if influx_rate == 0 if maxLOut > if i<= (2^(maxLOut+1)-2) count = fprintf(fid2,'%d\n',1); else count = fprintf(fid2,'%d\n',0); end else count = fprintf(fid2,'%d\n',0); end else count = fprintf(fid2,'%d\n',0); endend%----------%inizializzazione del file dell'influx "_influx.csv"%----------for i = 1:length(influx(:,1)) count = fprintf(fid5,'%d \t', influx(i,1)-1); count = fprintf(fid5,'%d \n', influx(i,2));endst = fclose(fid5);%----------%inizializzazione del file delle reazioni e dei catalizzatori%----------if catalizzatore(1,1) == -9999;else for i=1:length(catalizzatore(:,1)) count = fprintf(fid4,'%d\t',catalizzatore(i,1)-1); count = fprintf(fid4,'%d\t',catalizzatore(i,2)-1); count = fprintf(fid4,'%d\t',catalizzatore(i,3)-1); count = fprintf(fid4,'%d\t',0); %quante volte if catalizzatore(i,4)==0 tmpKdiss = Kdiss / revRctRatio; count = fprintf(fid4,'%g\t',Kass); %kass count = fprintf(fid4,'%g\t',tmpKdiss); %kdiss count = fprintf(fid4,'%g\t',Kcpx); %k complex else tmpKass = Kass / revRctRatio; tmpKcpx = Kcpx / revRctRatio; count = fprintf(fid4,'%g\t',tmpKass); %kass count = fprintf(fid4,'%g\t',Kdiss); %kdiss count = fprintf(fid4,'%g\t',tmpKcpx); %k complex end index_rand = ceil(rand*2); count = fprintf(fid4,'%g\n',index_rand); %nuovo parametro end [righe_xx colonne_xx]=size(funzioni_booleane_in_dec); for i=1:length(reazione(:,1)) indexx = ceil(rand*righe_xx); funzione_giusta = funzioni_booleane_in_dec(indexx); count = fprintf(fid3,'%d\t',reazione(i,1)-1); count = fprintf(fid3,'%d\t',reazione(i,2)); count = fprintf(fid3,'%d\t',reazione(i,3)-1); count = fprintf(fid3,'%d\t',reazione(i,4)-1); count = fprintf(fid3,'%d\t',reazione(i,5)-1); count = fprintf(fid3,'%d\t',0); if energy == 2 eso_endo = 1; %perch? se non c'? l'energia sono tutte eso else if rand > energy if reazione(i, 2)

Initial value:

```
==1
                                    eso_endo =1
```

Definition at line 345 of file inizializzatore_ACS.m.

13.46.2.19 end specie_def =specie_def_2

Definition at line 193 of file inizializzatore_ACS.m.

13.46.2.20 inserisco il numero decimale relativo alla funzione booleana della reazione end end st = fclose(fid1)

Definition at line 126 of file inizializzatore_ACS.m.

13.46.2.21 temporal = specie_non_esistenti{i}

Definition at line 376 of file inizializzatore_ACS.m.

13.46.2.22 tmpStr = specie_def(i,:)

Definition at line 243 of file inizializzatore_ACS.m.

13.46.2.23 vettore_rand =[0.5 1 2]

Definition at line 127 of file inizializzatore_ACS.m.

13.46.2.24 for xxxx

Initial value:

```
= 1:length(alphabet)
                if sum(alphabet(xxxx)==specie_def(i,
      jjjj))>0
                    controllo = 1
```

Definition at line 218 of file inizializzatore_ACS.m.

13.46.2.25 for z

Initial value:

```
= 1:length(specie_temp(1,:))
           specie_def(k,z)=specie_temp(j,z)
```

Definition at line 179 of file inizializzatore_ACS.m.

13.47 /Users/alessandrofilisetti/Documents/GIT/carness/_matlabinitializator/lancia_acs.m File Reference

Functions

- Keq (k, j)

Variables

- lancia_acs m clear all close
  all parametri kdiss = 0
- kcomplex = 1e6
- kcond = 6e8
- k_complex = 1e-4
- AB = 0
- C = 2e-7
- CA = 0
- k = 0
- for A
- for B

13.47.1 Function Documentation

13.47.1.1 Keq (k , j)

13.47.2 Variable Documentation

13.47.2.1 for A

Initial value:

```
= 10e-7:10e-7:10e-6

    k=k+1
    j=0
```

Definition at line 17 of file lancia_acs.m.

13.47.2.2 AB = 0

Definition at line 12 of file lancia_acs.m.

13.47.2.3 for B

Initial value:

```
= 10e-7:10e-7:10e-6

        j=j+1


        [t y] = ACS_reverse_reaction_con_input(A,B,AB,C,CA,kdiss,
      kcomplex,kcond,k_complex)
```

Definition at line 21 of file lancia_acs.m.

13.47.2.4 C = 2e-7

Definition at line 13 of file lancia_acs.m.

13.47.2.5 CA = 0

Definition at line 14 of file lancia_acs.m.

13.47.2.6 k = 0

Definition at line 16 of file lancia_acs.m.

13.47.2.7 k_complex = 1e-4

Definition at line 10 of file lancia_acs.m.

13.47.2.8 kcomplex = 1e6

Definition at line 8 of file lancia_acs.m.

13.47.2.9 kcond = 6e8

Definition at line 9 of file lancia_acs.m.

13.47.2.10 lancia_acs m clear all close all parametri kdiss = 0

Definition at line 7 of file lancia_acs.m.

13.48 /Users/alessandrofilisetti/Documents/GIT/carness/_matlabinitializator/lancia_-inizializzatore_acs.m File Reference

Functions

- id nGEN ()
- experiment all condensation are eso (and cleavage endo)) energy
- id ratio_firing_disk ()

Variables

- nSIM =1
- nSeconds =400
- nReactions =200000000
- initialPopulationNumber =0
- initialMaxLength =3
- massima_lunghezza_su_cui_calcolare_le_reazioni = 3
- experiment __pad1__
- alphabet = ['AB']
- lastFiringDiskSpeciesID = 0
- for i
- overallConcentration =1e-4
- volume =1e-15
- experiment __pad2__
- energyTarget =0
- complexFormationSymmetry =0
- **NEW ** fino_a_che_lunghezza_i_polimeri_non_catalizzano = 2
- reactionProbability =0.004
- experiment __pad3__
- reverseReactions =0
- costanti CINETICHE K_eq =1000
- parte da rivedere e correggere Kass_o_Kdiss = 0
- se cleavage kdiss = 100
- se cleavage se a condensazione kass =100
- rapporto_Kfor_Kback = 0
- e g Kass
- K_cpxDiss =0
- coefficiente di fosforilazione
- K_nrg = 0
- K_irrad = 0
- altri parametri moleculeDecay_KineticConstant =0.02
- diffusion_contribute =0
- solubility_threshold =0

- experimentveri influx =2
- influx_rate =1e-6
- percentuale di specie da cancellare rispetto a tutte quelle che restano dopo aver conservato i polimeri fino a lunghezzamaxfd lunghezza_max_fd = 1
- lunghezza dei monomeri polimeri da conservare scelta_concentrazioni =1
- parametro switch
- parametro uniforme sulle lunghezze
- parametro uniforme sulle powerlaw con esponente gamma gamma_powerlaw_concentrazioni = 2.1
- esponente della powerlaw in caso parametri per la distribuzione dei catalizzatori
- richiamo la funzione inizializzatore_ACS [fd concentrazioni specie_def influx catalizzatore reazione specie_non_esistenti matrice_adiacenza_sub_prod matrice_adiacenza_cat_prod] = inizializzatore_ACS(nGEN, nSIM, nSeconds, nReactions, initialPopulationNumber, initialMaxLength, massima_lunghezza_su_cui_calcolare_le_reazioni, overallConcentration, alphabet, complexFormationSymmetry, fino_a_che_lunghezza_i_polimeri_non_catalizzano, reactionProbability, cleavageProbability, diffusion_contribute, solubility_threshold, influx, influx_rate, reverseReactions, Kass_o_Kdiss,rapporto_Kfor_Kback,rapporto_Kcpx_K_ass,K_cpxDiss, K_nrg, moleculeDecay_KineticConstant, ratio_firing_disk, lunghezza_max_fd, scelta_concentrazioni, gamma_powerlaw_concentrazioni,decisione_catalizzatori, lastFiringDiskSpeciesID, EC-Concentration, volume, energy, energyTarget, controllo_ACS_nel_ciclo, K_irrad)

13.48.1 Function Documentation

13.48.1.1 experiment all condensation are eso (and cleavage *endo*)

13.48.1.2 id nGEN ( ) [virtual]

13.48.1.3 id ratio_firing_disk ( ) [virtual]

13.48.2 Variable Documentation

13.48.2.1 experiment __pad1__

Definition at line 27 of file lancia_inizializzatore_acs.m.

13.48.2.2 experiment __pad2__

Definition at line 45 of file lancia_inizializzatore_acs.m.

13.48.2.3 experiment __pad3__

Definition at line 57 of file lancia_inizializzatore_acs.m.

13.48.2.4 alphabet = ['AB']

Definition at line 29 of file lancia_inizializzatore_acs.m.

13.48.2.5 esponente della powerlaw in caso parametri per la distribuzione dei catalizzatori

Definition at line 89 of file lancia_inizializzatore_acs.m.

13.48.2.6 complexFormationSymmetry =0

Definition at line 51 of file lancia_inizializzatore_acs.m.

13.48.2.7 diffusion_contribute =0

Definition at line 75 of file lancia_inizializzatore_acs.m.

13.48.2.8 energyTarget =0

Definition at line 49 of file lancia_inizializzatore_acs.m.

13.48.2.9 ∗∗ NEW∗∗ fino_a_che_lunghezza_i_polimeri_non_catalizzano = 2

Definition at line 54 of file lancia_inizializzatore_acs.m.

13.48.2.10 coefficiente di fosforilazione

Definition at line 68 of file lancia_inizializzatore_acs.m.

13.48.2.11 parametro uniforme sulle powerlaw con esponente gamma gamma_powerlaw_concentrazioni = 2.1

Definition at line 85 of file lancia_inizializzatore_acs.m.

13.48.2.12 for i

Initial value:

```
= 1:massima_lunghezza_su_cui_calcolare_le_reazioni
    lastFiringDiskSpeciesID = lastFiringDiskSpeciesID +
      length(alphabet)^i
```

Definition at line 33 of file lancia_inizializzatore_acs.m.

13.48.2.13 controllo che non ci siano cicli nell influx =2

Definition at line 77 of file lancia_inizializzatore_acs.m.

13.48.2.14 influx_rate =1e-6

Definition at line 78 of file lancia_inizializzatore_acs.m.

13.48.2.15 initialMaxLength =3

Definition at line 23 of file lancia_inizializzatore_acs.m.

13.48.2.16 initialPopulationNumber =0

Definition at line 21 of file lancia_inizializzatore_acs.m.

13.48.2.17 richiamo la funzione inizializzatore_ACS[fd concentrazioni specie_def influx catalizzatore reazione specie_non_esistenti matrice_adiacenza_sub_prod matrice_adiacenza_cat_prod] = inizializzatore_ACS(nGEN, nSIM, nSeconds, nReactions, initialPopulationNumber, initialMaxLength, massima_lunghezza_su_cui_calcolare_le_reazioni, overallConcentration, alphabet, complexFormationSymmetry, fino_a_che_lunghezza_i_polimeri_non_catalizzano, reactionProbability, cleavageProbability, diffusion_contribute, solubility_threshold, influx, influx_rate, reverseReactions, Kass_o_Kdiss,rapporto_Kfor_Kback,rapporto_Kcpx_K_ass,K_cpxDiss, K_nrg, moleculeDecay_KineticConstant, ratio_firing_disk, lunghezza_max_fd, scelta_concentrazioni, gamma_powerlaw_concentrazioni,decisione_catalizzatori, lastFiringDiskSpeciesID, ECConcentration, volume, energy, energyTarget, controllo_ACS_nel_ciclo, K_irrad)

Definition at line 100 of file lancia_inizializzatore_acs.m.

13.48.2.18 K_cpxDiss =0

Definition at line 66 of file lancia_inizializzatore_acs.m.

13.48.2.19 costanti CINETICHE K_eq =1000

Definition at line 61 of file lancia_inizializzatore_acs.m.

13.48.2.20 K_irrad = 0

Definition at line 71 of file lancia_inizializzatore_acs.m.

13.48.2.21 K_nrg = 0

Definition at line 69 of file lancia_inizializzatore_acs.m.

13.48.2.22 se cleavage se a condensazione kass =100

Definition at line 63 of file lancia_inizializzatore_acs.m.

13.48.2.23 e g Kass

Initial value:

```
= 100 --> Kdiss = 100/100 = 1
rapporto_Kcpx_K_ass = 0
```

Definition at line 64 of file lancia_inizializzatore_acs.m.

13.48.2.24 parte da rivedere e correggere Kass_o_Kdiss = 0

Definition at line 63 of file lancia_inizializzatore_acs.m.

13.48.2.25 se cleavage kdiss = 100

Definition at line 63 of file lancia_inizializzatore_acs.m.

13.48.2.26 end lastFiringDiskSpeciesID = 0

Definition at line 32 of file lancia_inizializzatore_acs.m.

13.48.2.27 percentuale di specie da cancellare rispetto a tutte quelle che restano dopo aver conservato i polimeri fino a lunghezzamaxfd lunghezza_max_fd = 1

Definition at line 83 of file lancia_inizializzatore_acs.m.

13.48.2.28 parametro uniforme sulle lunghezze

Definition at line 84 of file lancia_inizializzatore_acs.m.

13.48.2.29 massima_lunghezza_su_cui_calcolare_le_reazioni = 3

Definition at line 24 of file lancia_inizializzatore_acs.m.

13.48.2.30 altri parametri moleculeDecay_KineticConstant =0.02

Definition at line 74 of file lancia_inizializzatore_acs.m.

13.48.2.31 nReactions =200000000

Definition at line 20 of file lancia_inizializzatore_acs.m.

13.48.2.32 nSeconds =400

Definition at line 19 of file lancia_inizializzatore_acs.m.

13.48.2.33 nSIM =1

Definition at line 18 of file lancia_inizializzatore_acs.m.

13.48.2.34 overallConcentration =1e-4

Definition at line 38 of file lancia_inizializzatore_acs.m.

13.48.2.35 rapporto_Kfor_Kback = 0

Definition at line 64 of file lancia_inizializzatore_acs.m.

13.48.2.36 reactionProbability =0.004

Definition at line 56 of file lancia_inizializzatore_acs.m.

13.48.2.37 reverseReactions =0

Definition at line 58 of file lancia_inizializzatore_acs.m.

13.48.2.38 lunghezza dei monomeri polimeri da conservare scelta_concentrazioni =1

Definition at line 84 of file lancia_inizializzatore_acs.m.

13.48.2.39 solubility_threshold =0

Definition at line 76 of file lancia_inizializzatore_acs.m.

13.48.2.40 parametro switch

Definition at line 84 of file lancia_inizializzatore_acs.m.

13.48.2.41 volume =1e-15

Definition at line 40 of file lancia_inizializzatore_acs.m.

13.49 /Users/alessandrofilisetti/Documents/GIT/carness/_matlabinitializator/start.m File Reference

Functions

- Numero di simulazioni (diversi semi random) nSeconds
- Numero massimo di reazioni (secondo parametro di stop oltre al numero di secondi) nHours=0
- Numero massimo di ore per la simulazione (=0 no vincolo) nAttempts=0
- Numero di volte in cui sistema
  ritenta la stessa rete (=0 no vincolo) initialMaxLength
- Number of different networks
  for every network will be
  performed nSim different simulation (differnt random seeds) simFolder.name
- if exist (simFolder.name, 'dir')
- introduzione delle FUNZIONI
  BOOLEANE nell che verranno
  successivamente assegnate con
  probabilit uniforme alle
  reazioni la funzione divisa a
  la prima met corrisponde alla condensazione (8 bit)
- converto in decimale for per
  default uniforme funzioni_booleane_in_dec (i, 2)

Variables

- lancia_serie_di_inizializzatore
  m clear all close all
  **********Inserimento
  dei PARAMETRI VARIABILI sui
  quali fare lo SCREENING

- lancia_serie_di_inizializzatore
  m clear all close all
  **********Inserimento
  dei PARAMETRI VARIABILI sui
  quali fare lo sottoforma di
  matrici o vettori reactionProbability = [0.000516529
- nome_prob = [0.125
- lMaxInflux = [2]
- parametro_screening = lMaxInflux
- nome_folder = [2]
- **********
  dei PARAMETRI FISSI
- **********
  dei PARAMETRI quelli che
  restano cio costanti in tutti
  gli esperimenti della serie nSIM =1
- Numero di secondi randomSeed =0
- lasciare a debugLevel =0
- livello di dettaglio messaggi
  durante simulazione
- livello di dettaglio messaggi
  durante lasciare a
- livello di dettaglio messaggi
  durante lasciare per debug
  software timeStructuresSavingInterval =nSeconds/100
- definisce il tempo in cui
  vengono salvati i file durante
  la simulazione fileTimesSaveInterval =0
- Definisce il tempo in cui
  vengono salvati i dati sui
  file times
- Definisce il tempo in cui
  vengono salvati i dati sui
  file reaction_parameter e i
  vari living nReactions =200000000
- Lunghezza massima delle specie
  da creare massima_lunghezza_su_cui_calcolare_le_reazioni = 6
- Lunghezza massima fino alla
  quale creare le reazioni maxLOut = 2
- se =0 non viene considerato
- altrimenti Quando influx_rate
  indica la lunghezza massima
  delle molecole che possono
  uscire dal contenitore
- altrimenti Quando influx_rate
  indica la lunghezza massima
  delle molecole che possono
  uscire dal quando influx_rate
- alphabet = ['AB']
- overallConcentration =0.0333
- volume =1e-18
- energy =0
- energia considerata
- energia non considerata complexFormationSymmetry =0
- fino_a_che_lunghezza_i_polimeri_non_catalizzano = 2
- cleavageProbability =0.5

- reverseReactions =0
- Kass = 50
- Kdiss = 25
- Kcpx = 50
- K_cpx = 1
- K_nrg = 0
- K_nrg_decay =0
- coefficiente di decadimento
  delle molecole o dei carrier
  dalla propria componente
  energetica revRctRatio = 10
- ratioSpeciesEnergizable = 0
- percentuale di specie presenti
  nel sistema che possono essere
  energizzate per ogni specie
  create c una certa probabilit
  di essere energizzabile o meno moleculeDecay_KineticConstant =0.02
- diffusion_contribute =0
- solubility_threshold =0
- lunghezza massima delle
  molecole presenti nell influx ratio_firing_disk =0
- percentuale di specie da
  cancellare rispetto a tutte
  quelle che restano dopo aver
  conservato i polimeri fino a
  lunghezzamaxfd lunghezza_max_fd = 7
- lunghezza dei monomeri
  polimeri da conservare scelta_concentrazioni =1
- parametro switch
- parametro uniforme sulle lunghezze
- parametro uniforme sulle
  powerlaw con esponente gamma gamma_powerlaw_concentrazioni = 2.1
- esponente della powerlaw in caso decisione_catalizzatori =1
- parametri per la distribuzione dei catalizzatori
- controllo che non ci siano
  cicli nell influx
- NUOVO PARAMETRO volumeGrowth = 0
- NUOVO PARAMETRO stochDivision = 0
- NUOVO PARAMETRO simFolder nets = 10
- Number of different networks ensambles
- Nome della cartella dove verr
  salvata la simulazione
  simFolder path = 'SIMS'
- Percorso dove verr creata la
  cartella simFolder dove
  verranno salvati tutti i file **********
  nGEN =10
- Numero di generazioni
- Numero di al momento significa
  che alla fine di ogni
  generazione da ogni file di
  fine sim partono altre Nsim
- Numero di al momento significa
  che alla fine di ogni
  generazione da ogni file di
  fine sim partono altre
  lasciare ad !lastFiringDiskSpeciesID = 0

- calcolata in automatico for i
- end lastFiringDiskSpeciesID = lastFiringDiskSpeciesID -1
- thisFolder = pwd
- end fid20 = fopen('lanciatore.sh','w')
- introduzione delle FUNZIONI BOOLEANE nell energia
- introduzione delle FUNZIONI BOOLEANE nell che verranno successivamente assegnate con probabilit uniforme alle reazioni la funzione divisa a met
- introduzione delle FUNZIONI BOOLEANE nell che verranno successivamente assegnate con probabilit uniforme alle reazioni la funzione divisa a la prima met corrisponde alla la seconda parte al cleavage(4 bit).%esempio funzioni_booleane_in_dec = bi2de(funzioni_booleane,'left-msb')
- parte2_nome_cartella = num2str(nome_folder(i))
- parte3_nome_cartella = ('_rete_n_')

13.49.1 Function Documentation

13.49.1.1 introduzione delle FUNZIONI BOOLEANE nell che verranno successivamente assegnate con probabilit uniforme alle reazioni la funzione divisa a la prima met corrisponde alla condensazione (8 *bit*)

13.49.1.2 if exist (simFolder. *name,* 'dir')

13.49.1.3 converto in decimale for per default uniforme funzioni_booleane_in_dec (i , 2)

13.49.1.4 Numero massimo di reazioni (secondo parametro di stop oltre al numero di *secondi*) [pure virtual]

13.49.1.5 Numero di volte in cui sistema ritenta la stessa rete ( )

13.49.1.6 Number of different networks for every network will be performed nSim different simulation (differnt random *seeds*)

13.49.1.7 Numero massimo di ore per la simulazione ( ) [pure virtual]

13.49.1.8 Numero di simulazioni (diversi semi *random*)

13.49.2 Variable Documentation

13.49.2.1 Numero di al momento significa che alla fine di ogni generazione da ogni file di fine sim partono altre lasciare ad !lastFiringDiskSpeciesID = 0

Definition at line 79 of file start.m.

13.49.2.2 livello di dettaglio messaggi durante lasciare a

Definition at line 26 of file start.m.

13.49.2.3 alphabet = ['AB']

Definition at line 36 of file start.m.

13.49.2.4 parametri per la distribuzione dei catalizzatori

Definition at line 63 of file start.m.

13.49.2.5 cleavageProbability =0.5

Definition at line 42 of file start.m.

13.49.2.6 energia non considerata complexFormationSymmetry =0

Definition at line 40 of file start.m.

13.49.2.7 energia considerata

Definition at line 39 of file start.m.

13.49.2.8 altrimenti Quando influx_rate indica la lunghezza massima delle molecole che possono uscire dal contenitore

Definition at line 34 of file start.m.

13.49.2.9 lasciare a debugLevel =0

Definition at line 26 of file start.m.

13.49.2.10 esponente della powerlaw in caso decisione_catalizzatori =1

Definition at line 62 of file start.m.

13.49.2.11 diffusion_contribute =0

Definition at line 54 of file start.m.

13.49.2.12 introduzione delle FUNZIONI BOOLEANE nell energia

Definition at line 111 of file start.m.

13.49.2.13 energy =0

Definition at line 39 of file start.m.

13.49.2.14 Number of different networks ensambles

Definition at line 72 of file start.m.

13.49.2.15 end fid20 = fopen('lanciatore.sh','w')

Definition at line 92 of file start.m.

13.49.2.16 definisce il tempo in cui vengono salvati i file durante la simulazione fileTimesSaveInterval =0

Definition at line 28 of file start.m.

13.49.2.17 fino_a_che_lunghezza_i_polimeri_non_catalizzano = 2

Definition at line 41 of file start.m.

13.49.2.18 ********** ********** Inserimento dei PARAMETRI FISSI

Definition at line 20 of file start.m.

13.49.2.19 converto in decimale funzioni_booleane_in_dec = bi2de(funzioni_booleane,'left-msb')

Definition at line 119 of file start.m.

13.49.2.20 parametro uniforme sulle powerlaw con esponente gamma gamma_powerlaw_concentrazioni = 2.1

Definition at line 61 of file start.m.

13.49.2.21 Numero di generazioni

Definition at line 78 of file start.m.

13.49.2.22 end********** ********** ***** S T A R T***** ********** for i

Initial value:

```
= 1:massima_lunghezza_su_cui_calcolare_le_reazioni
    lastFiringDiskSpeciesID = lastFiringDiskSpeciesID +
      length(alphabet)^i
```

Definition at line 80 of file start.m.

13.49.2.23 controllo che non ci siano cicli nell influx

Definition at line 68 of file start.m.

13.49.2.24 influx_rate

Initial value:

```
= 0 indica fino a quale lunghezza le molecole non variano in quantita' (simulazione membrana
       permeabile)
ECConcentration=0
```

Definition at line 34 of file start.m.

13.49.2.25 K_cpx = 1

Definition at line 48 of file start.m.

13.49.2.26 K_nrg = 0

Definition at line 49 of file start.m.

13.49.2.27 K_nrg_decay =0

Definition at line 50 of file start.m.

13.49.2.28 Kass = 50

Definition at line 45 of file start.m.

13.49.2.29 Kcpx = 50

Definition at line 47 of file start.m.

13.49.2.30 Kdiss = 25

Definition at line 46 of file start.m.

13.49.2.31 end lastFiringDiskSpeciesID = lastFiringDiskSpeciesID -1

Definition at line 83 of file start.m.

13.49.2.32 se il sistema chiuso lMaxInflux = [2]

Definition at line 13 of file start.m.

13.49.2.33 percentuale di specie da cancellare rispetto a tutte quelle che restano dopo aver conservato i polimeri fino a lunghezzamaxfd lunghezza_max_fd = 7

Definition at line 59 of file start.m.

13.49.2.34 parametro uniforme sulle lunghezze

Definition at line 60 of file start.m.

13.49.2.35 Lunghezza massima delle specie da creare massima_lunghezza_su_cui_calcolare_le_reazioni = 6

Definition at line 33 of file start.m.

13.49.2.36 Lunghezza massima fino alla quale creare le reazioni maxLOut = 2

Definition at line 34 of file start.m.

13.49.2.37 introduzione delle FUNZIONI BOOLEANE nell che verranno successivamente assegnate con probabilit uniforme alle reazioni la funzione divisa a met

Definition at line 111 of file start.m.

13.49.2.38 percentuale di specie presenti nel sistema che possono essere energizzate per ogni specie create c una certa probabilit di essere energizzabile o meno moleculeDecay_KineticConstant =0.02

Definition at line 53 of file start.m.

13.49.2.39 NUOVO PARAMETRO simFolder nets = 10

Definition at line 72 of file start.m.

13.49.2.40 Percorso dove verr creata la cartella simFolder dove verranno salvati tutti i file********** nGEN =10

Definition at line 78 of file start.m.

13.49.2.41 nome_folder = [2]

Definition at line 15 of file start.m.

13.49.2.42 nome_prob = [0.125

Definition at line 12 of file start.m.

13.49.2.43 Definisce il tempo in cui vengono salvati i dati sui file reaction_parameter e i vari living nReactions =200000000

Definition at line 29 of file start.m.

13.49.2.44 ********** ********** Inserimento dei PARAMETRI quelli che restano cio costanti in tutti gli esperimenti della serie nSIM =1

Definition at line 23 of file start.m.

13.49.2.45 Numero di al momento significa che alla fine di ogni generazione da ogni file di fine sim partono altre Nsim

Definition at line 78 of file start.m.

13.49.2.46 overallConcentration =0.0333

Definition at line 37 of file start.m.

13.49.2.47 parametro_screening = IMaxInflux

Definition at line 14 of file start.m.

13.49.2.48 parte2_nome_cartella = num2str(nome_folder(i))

Definition at line 142 of file start.m.

13.49.2.49 parte3_nome_cartella = ('_rete_n_')

Definition at line 143 of file start.m.

13.49.2.50 concAnalysis(params) clear all close all params path = 'SIMS'

Definition at line 74 of file start.m.

13.49.2.51 Numero di secondi randomSeed =0

Definition at line 25 of file start.m.

13.49.2.52 lunghezza massima delle molecole presenti nell influx ratio_firing_disk =0

Definition at line 58 of file start.m.

13.49.2.53 ratioSpeciesEnergizable = 0

Definition at line 52 of file start.m.

13.49.2.54 reactionProbability = [0.000516529

Definition at line 11 of file start.m.

13.49.2.55 reverseReactions =0

Definition at line 43 of file start.m.

13.49.2.56 coefficiente di decadimento delle molecole o dei carrier dalla propria componente energetica revRctRatio = 10

Definition at line 51 of file start.m.

13.49.2.57 lunghezza dei monomeri polimeri da conservare scelta_concentrazioni =1

Definition at line 60 of file start.m.

13.49.2.58 lancia_serie_di_inizializzatore m clear all close all********** Inserimento dei PARAMETRI VARIABILI sui quali fare lo SCREENING

Definition at line 8 of file start.m.

13.49.2.59 se =0 non viene considerato

Definition at line 34 of file start.m.

13.49.2.60 livello di dettaglio messaggi durante simulazione

Definition at line 26 of file start.m.

13.49.2.61 solubility_threshold =0

Definition at line 55 of file start.m.

13.49.2.62 NUOVO PARAMETRO stochDivision = 0

Definition at line 70 of file start.m.

13.49.2.63 parametro switch

Definition at line 60 of file start.m.

13.49.2.64 thisFolder = pwd

Definition at line 86 of file start.m.

13.49.2.65 Definisce il tempo in cui vengono salvati i dati sui file times

Definition at line 28 of file start.m.

13.49.2.66 livello di dettaglio messaggi durante lasciare per debug software timeStructuresSavingInterval =nSeconds/100

Definition at line 27 of file start.m.

13.49.2.67 volume =1e-18

Definition at line 38 of file start.m.

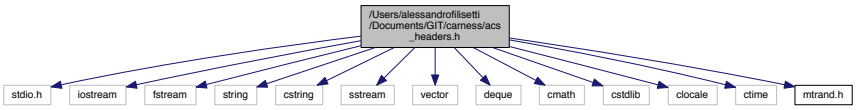
13.49.2.68 NUOVO PARAMETRO volumeGrowth = 0

Definition at line 69 of file start.m.

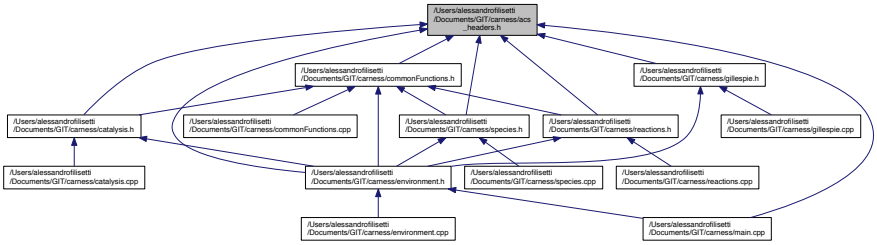
13.50 /Users/alessandrofilisetti/Documents/GIT/carness/acs_headers.h File Reference

```
#include <stdio.h>
#include <iostream>
#include <fstream>
#include <string>
#include <cstring>
#include <sstream>
#include <vector>
#include <deque>
#include <cmath>
#include <cstdlib>
#include <clocale>
#include <ctime>
#include "mtrand.h"
```

Include dependency graph for acs_headers.h:



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



Macros

- #define __SOFTVERSION__ "4.8b20131209.62"
- #define MINIMAL_PROMPT -1
- #define RUNNING_VERSION 0
- #define SMALL_DEBUG 1
- #define MEDIUM_DEBUG 2
- #define HIGH_DEBUG 3
- #define FINDERRORDURINGRUNTIME -10
- #define COMPLEXSTUFF -20
- #define GILLESPIESTUFF -30
- #define RANDOMRANGE random()
- #define PROPORTIONALMOLECULEAMOUNT 1
- #define UNIFORMMOLECULEAMOUNT 2
- #define INVPROPORTIONALMOLECULEAMOUNT 3
- #define CONDENSATION 0
- #define CLEAVAGE 1
- #define COMPLEXFORMATION 2
- #define COMPLEXDEGRADATION 3
- #define SPECIESDECAY 4
- #define PHOSPHORILATION 5
- #define ENDO_CLEAVAGE 6
- #define ENDO_CONDENSATION 7
- #define ENDO_COMPLEXFORMATION 8
- #define ENERGYEFFLUX 9
- #define SPONTANEOUS_CONDENSATION 10
- #define SPONTANEOUS_CLEAVAGE 11
- #define SOLUBLE 1
- #define PRECIPITATED 0
- #define ESOERGONIC 1
- #define ENDOERGONIC 0
- #define CLEAVAGEBASED 1

- #define CONDENSATIONBASED 0
- #define ENERGYBASED 1
- #define ENERGYFREE 0
- #define TRUENRG '1'
- #define FALSENRG '0'
- #define ENERGIZABLE 1
- #define NOTENERGIZABLE 0
- #define SUBSTRATELOAD 0
- #define CATALYSTLOAD 1
- #define BOTHLOAD 2
- #define COMPLEXLOAD 3
- #define NOTHINGLOAD 4
- #define NEWREACTIONS 1
- #define UPGRADEREACTIONS 0
- #define NOSPONTANEOUS 0
- #define NEP 2.71828182845904523536028874
- #define AVO 6.02214179e+23
- #define MINIMALRCTTIMEMULTI 100

Typedefs

- typedef long double acs_double
- typedef unsigned long int acs_longInt
- typedef unsigned int acs_int

13.50.1 Macro Definition Documentation

13.50.1.1 #define __SOFTVERSION__ "4.8b20131209.62"

Definition at line 42 of file acs_headers.h.

13.50.1.2 #define AVO 6.02214179e+23

Definition at line 108 of file acs_headers.h.

13.50.1.3 #define BOTHLOAD 2

Definition at line 95 of file acs_headers.h.

13.50.1.4 #define CATALYSTLOAD 1

Definition at line 94 of file acs_headers.h.

13.50.1.5 #define CLEAVAGE 1

Definition at line 65 of file acs_headers.h.

13.50.1.6 #define CLEAVAGEBASED 1

Definition at line 84 of file acs_headers.h.

13.50.1.7 #define COMPLEXDEGRADATION 3

Definition at line 67 of file acs_headers.h.

13.50.1.8 #define COMPLEXFORMATION 2

Definition at line 66 of file acs_headers.h.

13.50.1.9 #define COMPLEXLOAD 3

Definition at line 96 of file acs_headers.h.

13.50.1.10 #define COMPLEXSTUFF -20

Definition at line 52 of file acs_headers.h.

13.50.1.11 #define CONDENSATION 0

Definition at line 64 of file acs_headers.h.

13.50.1.12 #define CONDENSATIONBASED 0

Definition at line 85 of file acs_headers.h.

13.50.1.13 #define ENDO_CLEAVAGE 6

Definition at line 70 of file acs_headers.h.

13.50.1.14 #define ENDO_COMPLEXFORMATION 8

Definition at line 72 of file acs_headers.h.

13.50.1.15 #define ENDO_CONDENSATION 7

Definition at line 71 of file acs_headers.h.

13.50.1.16 #define ENDOERGONIC 0

Definition at line 83 of file acs_headers.h.

13.50.1.17 #define ENERGIZABLE 1

Definition at line 90 of file acs_headers.h.

13.50.1.18 #define ENERGYBASED 1

Definition at line 86 of file acs_headers.h.

13.50.1.19 #define ENERGYEFFLUX 9

Definition at line 73 of file acs_headers.h.

13.50.1.20 #define ENERGYFREE 0

Definition at line 87 of file acs_headers.h.

13.50.1.21 #define ESOERGONIC 1

Definition at line 82 of file acs_headers.h.

13.50.1.22 #define FALSENRG '0'

Definition at line 89 of file acs_headers.h.

13.50.1.23 #define FINDERRORDURINGRUNTIME -10

Definition at line 51 of file acs_headers.h.

13.50.1.24 #define GILLESPIESTUFF -30

Definition at line 53 of file acs_headers.h.

13.50.1.25 #define HIGH_DEBUG 3

Definition at line 50 of file acs_headers.h.

13.50.1.26 #define INVPROPORTIONALMOLECULEAMOUNT 3

Definition at line 61 of file acs_headers.h.

13.50.1.27 #define MEDIUM_DEBUG 2

Definition at line 49 of file acs_headers.h.

13.50.1.28 #define MINIMAL_PROMPT -1

Definition at line 46 of file acs_headers.h.

13.50.1.29 #define MINIMALRCTTIMEMULTI 100

Definition at line 111 of file acs_headers.h.

13.50.1.30 #define NEP 2.71828182845904523536028874

Definition at line 107 of file acs_headers.h.

13.50.1.31 #define NEWREACTIONS 1

Definition at line 100 of file acs_headers.h.

13.50.1.32 #define NOSPONTANEOUS 0

Definition at line 104 of file acs_headers.h.

13.50.1.33 #define NOTENERGIZABLE 0

Definition at line 91 of file acs_headers.h.

13.50.1.34 #define NOTHINGLOAD 4

Definition at line 97 of file acs_headers.h.

13.50.1.35 #define PHOSPHORILATION 5

Definition at line 69 of file acs_headers.h.

13.50.1.36 #define PRECIPITATED 0

Definition at line 79 of file acs_headers.h.

13.50.1.37 #define PROPORTIONALMOLECULEAMOUNT 1

Definition at line 59 of file acs_headers.h.

13.50.1.38 #define RANDOMRANGE random()

Definition at line 56 of file acs_headers.h.

13.50.1.39 #define RUNNING_VERSION 0

Definition at line 47 of file acs_headers.h.

13.50.1.40 #define SMALL_DEBUG 1

Definition at line 48 of file acs_headers.h.

13.50.1.41 #define SOLUBLE 1

Definition at line 78 of file acs_headers.h.

13.50.1.42 #define SPECIESDECAY 4

Definition at line 68 of file acs_headers.h.

13.50.1.43 #define SPONTANEOUS_CLEAVAGE 11

Definition at line 75 of file acs_headers.h.

13.50.1.44 #define SPONTANEOUS_CONDENSATION 10

Definition at line 74 of file acs_headers.h.

13.50.1.45 #define SUBSTRATELOAD 0

Definition at line 93 of file acs_headers.h.

13.50.1.46 #define TRUENRG '1'

Definition at line 88 of file acs_headers.h.

13.50.1.47 #define UNIFORMMOLECULEAMOUNT 2

Definition at line 60 of file acs_headers.h.

13.50.1.48 #define UPGRADEREACTIONS 0

Definition at line 101 of file acs_headers.h.

13.50.2 Typedef Documentation

13.50.2.1 typedef long double acs_double

Definition at line 34 of file acs_headers.h.

13.50.2.2 typedef unsigned int acs_int

Definition at line 36 of file acs_headers.h.

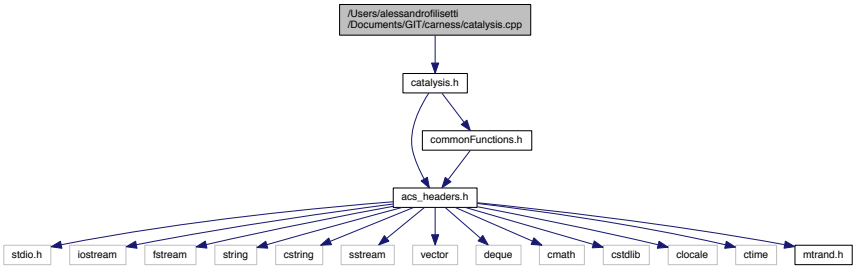
13.50.2.3 typedef unsigned long int acs_longInt

Definition at line 35 of file acs_headers.h.

13.51 /Users/alessandrofilisetti/Documents/GIT/carness/catalysis.cpp File Reference

```
#include "catalysis.h"
```

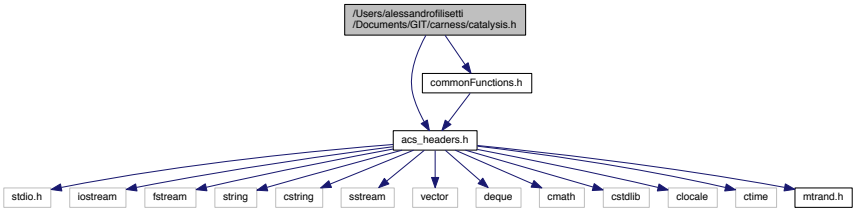
Include dependency graph for catalysis.cpp:



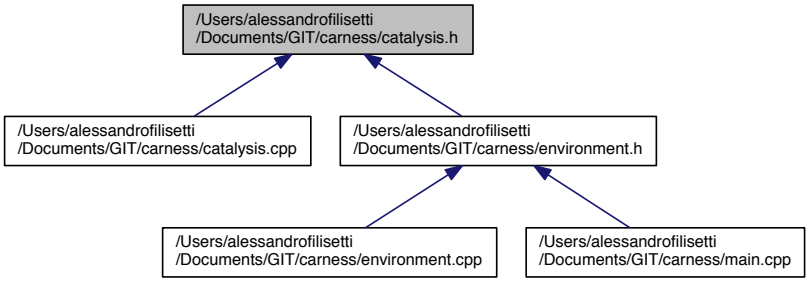
13.52 /Users/alessandrofilisetti/Documents/GIT/carness/catalysis.h File Reference

```
#include "acs_headers.h"
#include "commonFunctions.h"
```

Include dependency graph for catalysis.h:



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



Classes

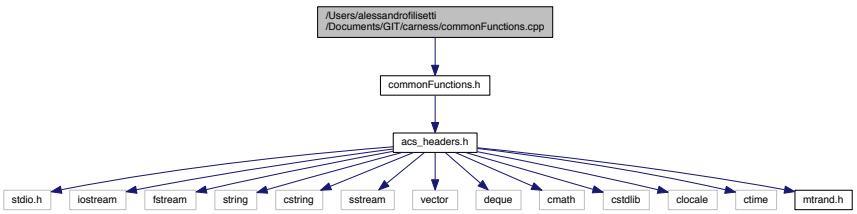
- class catalysis

  CATALYSIS class.

13.53 /Users/alessandrofilisetti/Documents/GIT/carness/commonFunctions.cpp File Reference

```
#include "commonFunctions.h"
```

Include dependency graph for commonFunctions.cpp:



Functions

- acs_longInt returnSelectionIdFromAWeightProbVector (acs_double *tmpArray, MTRand &tmpRandomGenerator, acs_int tmpRow)

  This funtion returns a random position in a probability weight array of N elements.

- acs_longInt returnSelectionIdFromAWeightProbVector (vector< acs_double > &tmpVector, acs_double tmpMaxValue, MTRand &tmpRandomGenerator, acs_int tmpRow)
- acs_longInt returnSelectionIdFromAWeightProbVectorAlreadyNormalized (vector< acs_double > &tmpVector, MTRand &tmpRandomGenerator)
- acs_longInt returnUniformSelection_LONG_IdFromVector (vector< acs_longInt > &tmpVector, MTRand &tmpRandomGenerator)
- acs_longInt getIntRandom (acs_longInt tmpFromNum, acs_longInt tmpToNum, MTRand &tmpRandomGenerator)
- acs_double getDoubleRandom (acs_double tmpFromNum, acs_double tmpToNum, MTRand &tmpRandomGenerator)
- acs_longInt random_poisson (acs_double tmpLambda, MTRand &tmpRandomGenerator)
- acs_longInt random_binomial (acs_longInt n, acs_double tmpP, MTRand &tmpRandomGenerator)
- acs_double acsround (acs_double tmpX)
- string dec2bin (acs_int tmpInt)
- bool ExitWithError (string strFunctionName, string strError)
- vector< string > split (string str, const char *delim)
- bool fromStrToBool (string const &string)

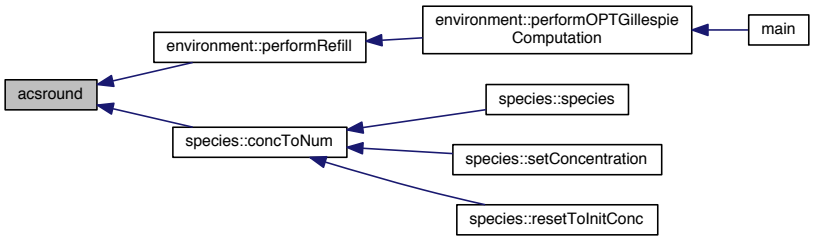
13.53.1 Function Documentation

13.53.1.1 acs_double acsround (acs_double *tmpX*)

Function to round double numbers in integers

Definition at line 229 of file commonFunctions.cpp.

Here is the caller graph for this function:

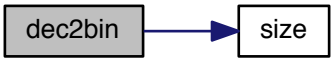


13.53.1.2 string dec2bin (acs_int *tmpInt*)

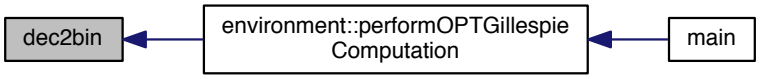
Function to convert a decimal number in a binary string composed of 12 bit such Example -> input: 10, binary 1010 –> 000000001010

Definition at line 241 of file commonFunctions.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



13.53.1.3 bool ExitWithError (string *strFunctionName,* string *strError*)

Function to close the program after having en error

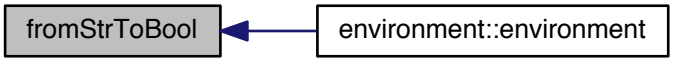
Definition at line 269 of file commonFunctions.cpp.

13.53.1.4 bool fromStrToBool (string const & *string*)

Function to convert from string to boolean

Definition at line 302 of file commonFunctions.cpp.

Here is the caller graph for this function:



13.53.1.5 acs_double getDoubleRandom (acs_double *tmpFromNum,* acs_double *tmpToNum,* MTRand & *tmpRandomGenerator*)

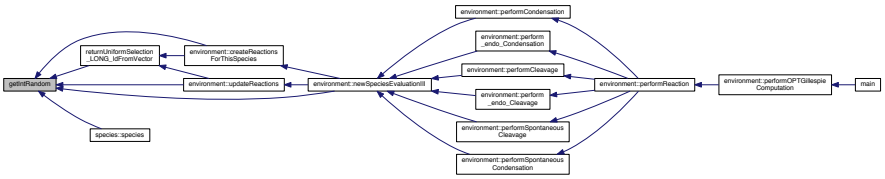
Definition at line 151 of file commonFunctions.cpp.

Here is the caller graph for this function:

13.53.1.6 acs_longInt getIntRandom (acs_longInt *tmpFromNum,* acs_longInt *tmpToNum,* MTRand & *tmpRandomGenerator*)

Definition at line 131 of file commonFunctions.cpp.

Here is the caller graph for this function:

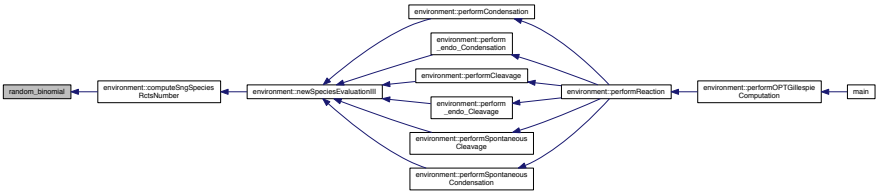


13.53.1.7 acs_longInt random_binomial (acs_longInt *n,* acs_double *tmpP,* MTRand & *tmpRandomGenerator*)

Function to return a number from a binomial distribution

Definition at line 203 of file commonFunctions.cpp.

Here is the caller graph for this function:



13.53.1.8 acs_longInt random_poisson (acs_double *tmpLambda,* MTRand & *tmpRandomGenerator*)

Function to return a number from a poisson random distribution

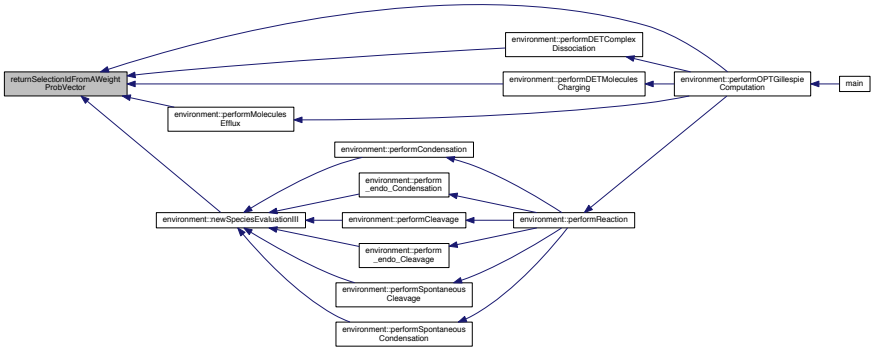
Definition at line 183 of file commonFunctions.cpp.

13.53.1.9 acs_longInt returnSelectionIdFromAWeightProbVector (acs_double * *tmpArray,* MTRand & *tmpRandomGenerator,* acs_int *tmpRow*)

This funtion returns a random position in a probability weight array of N elements.

Definition at line 15 of file commonFunctions.cpp.

Here is the caller graph for this function:



13.53.1.10 acs_longInt returnSelectionIdFromAWeightProbVector (vector< acs_double > & *tmpVector,* acs_double *tmpMaxValue,* MTRand & *tmpRandomGenerator,* acs_int *tmpRow*)

Return position of a randomly selected element from a vector containing cumulative values for each element

Version

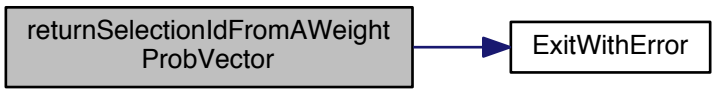
1.0

Parameters

vector<acs_double>&	tmpVector
acs_double	MAX VALUE contained within the QList (being a cumulative list this is the last value)
MTRand&	tmpRandomGenerator

Definition at line 40 of file commonFunctions.cpp.

Here is the call graph for this function:



13.53.1.11 acs_longInt returnSelectionIdFromAWeightProbVectorAlreadyNormalized (vector< acs_double > & *tmpVector,* MTRand & *tmpRandomGenerator*)

Return position of a randomly selected element from a normalized vector containing cumulative values for each element

Version

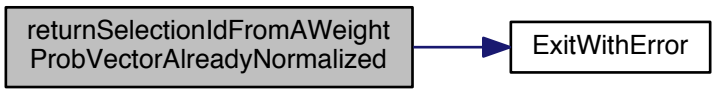
1.0

Parameters

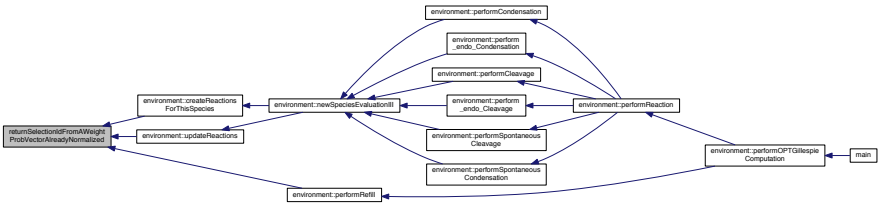
vector<acs_double>&	tmpQList
MTRand&	tmpRandomGenerator

Definition at line 72 of file commonFunctions.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



13.53.1.12 acs_longInt returnUniformSelection_LONG_IdFromVector (vector< acs_longInt > & *tmpVector,* MTRand & *tmpRandomGenerator*)

Return position of a LONG INT randomly selected element from a normalized vector containing cumulative values for each element

Version

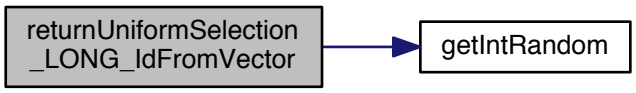
1.0

Parameters

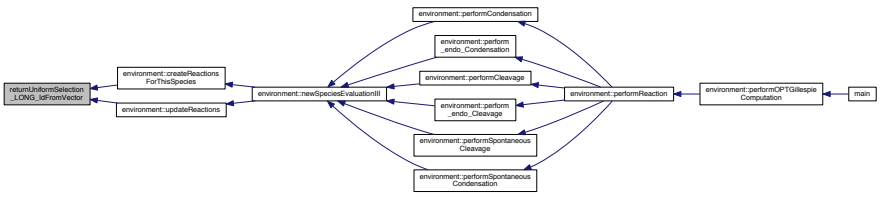
vector<acs_-double>&	tmpVector
MTRand&	tmpRandomGenerator

Definition at line 124 of file commonFunctions.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

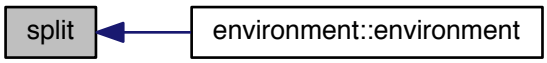


13.53.1.13 vector<string> split (string *str*, const char * *delim*)

Function to split a string and save tokens in a vector

Definition at line 279 of file commonFunctions.cpp.

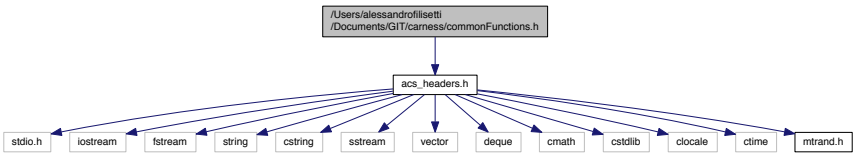
Here is the caller graph for this function:



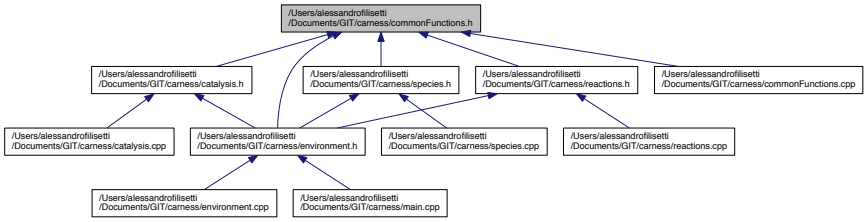
13.54 /Users/alessandrofilisetti/Documents/GIT/carness/commonFunctions.h File Reference

```
#include "acs_headers.h"
```

Include dependency graph for commonFunctions.h:



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



Functions

- acs_longInt returnSelectionIdFromAWeightProbVector (acs_double *tmpArray, MTRand &tmpRandomGenerator, acs_int tmpRow=0)

  This funtion returns a random position in a probability weight array of N elements.
- acs_longInt returnSelectionIdFromAWeightProbVector (vector< acs_double > &tmpVector, acs_double tmpMaxValue, MTRand &tmpRandomGenerator, acs_int tmpRow=0)
- acs_longInt returnSelectionIdFromAWeightProbVectorAlreadyNormalized (vector< acs_double > &tmpVector, MTRand &tmpRandomGenerator)

- acs_longInt returnUniformSelection_LONG_IdFromVector (vector< acs_longInt > &tmpVector, MTRand &tmpRandomGenerator)
- acs_longInt getIntRandom (acs_longInt tmpFromNum, acs_longInt tmpToNum, MTRand &tmpRandomGenerator)
- acs_double getDoubleRandom (acs_double tmpFromNum, acs_double tmpToNum, MTRand &tmpRandomGenerator)
- acs_longInt random_poisson (acs_double tmpLambda, MTRand &tmpRandomGenerator)
- acs_longInt random_binomial (acs_longInt n, acs_double tmpP, MTRand &tmpRandomGenerator)
- acs_double acsround (acs_double tmpX)
- string dec2bin (acs_int tmpInt)
- vector< string > split (string str, const char *delim)
- bool ExitWithError (string strFunctionName, string strError)
- bool fromStrToBool (string const &string)

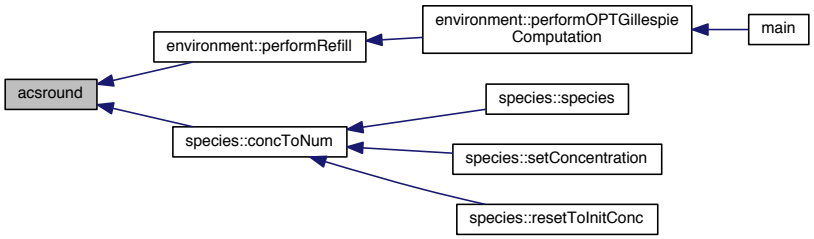
13.54.1 Function Documentation

13.54.1.1 acs_double acsround (acs_double *tmpX*)

Function to round double numbers in integers

Definition at line 229 of file commonFunctions.cpp.

Here is the caller graph for this function:

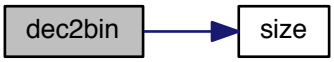


13.54.1.2 string dec2bin (acs_int *tmpInt*)

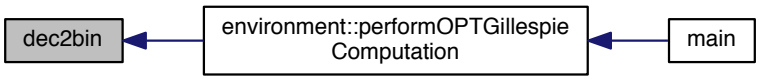
Function to convert a decimal number in a binary string composed of 12 bit such Example -> input: 10, binary 1010 –> 000000001010

Definition at line 241 of file commonFunctions.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



13.54.1.3 bool ExitWithError (string *strFunctionName,* string *strError*)

Function to close the program after having en error

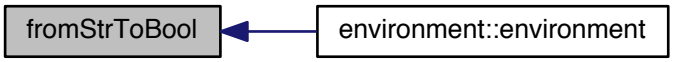
Definition at line 269 of file commonFunctions.cpp.

13.54.1.4 bool fromStrToBool (string const & *string*)

Function to convert from string to boolean

Definition at line 302 of file commonFunctions.cpp.

Here is the caller graph for this function:



13.54.1.5 acs_double getDoubleRandom (acs_double *tmpFromNum,* acs_double *tmpToNum,* MTRand & *tmpRandomGenerator*)

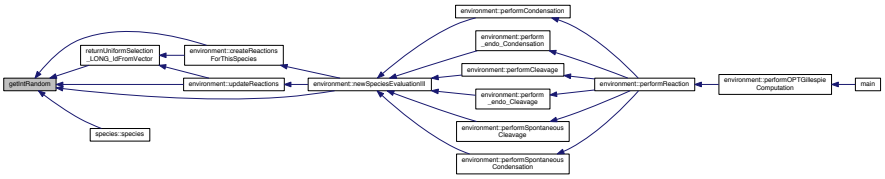
Definition at line 151 of file commonFunctions.cpp.

Here is the caller graph for this function:

13.54.1.6 acs_longInt getIntRandom (acs_longInt *tmpFromNum,* acs_longInt *tmpToNum,* MTRand & *tmpRandomGenerator*)

Definition at line 131 of file commonFunctions.cpp.

Here is the caller graph for this function:

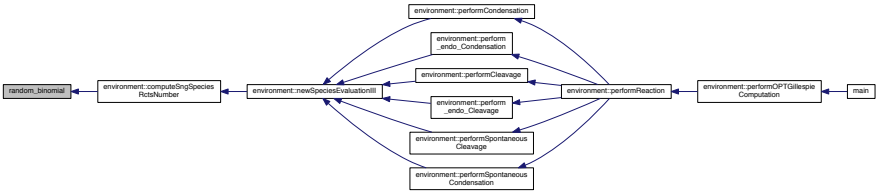


13.54.1.7 acs_longInt random_binomial (acs_longInt *n,* acs_double *tmpP,* MTRand & *tmpRandomGenerator*)

Function to return a number from a binomial distribution

Definition at line 203 of file commonFunctions.cpp.

Here is the caller graph for this function:



13.54.1.8 acs_longInt random_poisson (acs_double *tmpLambda,* MTRand & *tmpRandomGenerator*)

Function to return a number from a poisson random distribution

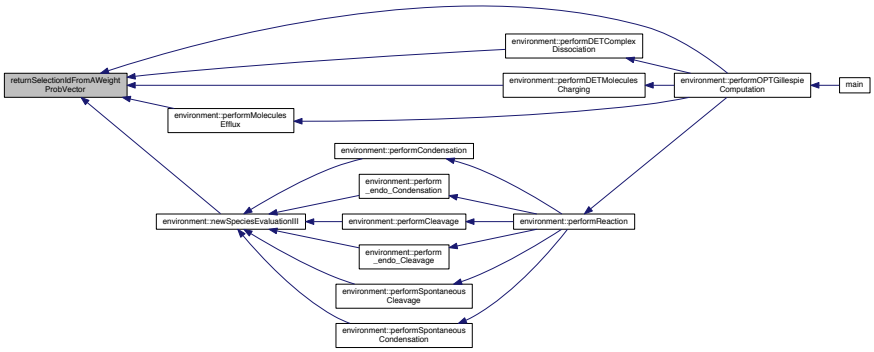
Definition at line 183 of file commonFunctions.cpp.

13.54.1.9 acs_longInt returnSelectionIdFromAWeightProbVector (acs_double * *tmpArray,* MTRand & *tmpRandomGenerator,* acs_int *tmpRow* = 0)

This funtion returns a random position in a probability weight array of N elements.

Definition at line 15 of file commonFunctions.cpp.

Here is the caller graph for this function:



13.54.1.10 acs_longInt returnSelectionIdFromAWeightProbVector (vector< acs_double > & *tmpVector,* acs_double *tmpMaxValue,* MTRand & *tmpRandomGenerator,* acs_int *tmpRow*)

Return position of a randomly selected element from a vector containing cumulative values for each element

Version

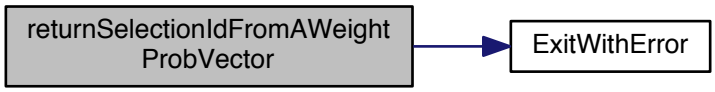
1.0

Parameters

vector<acs_-double>&	tmpVector
acs_double	MAX VALUE contained within the QList (being a cumulative list this is the last value)
MTRand&	tmpRandomGenerator

Definition at line 40 of file commonFunctions.cpp.

Here is the call graph for this function:



13.54.1.11 acs_longInt returnSelectionIdFromAWeightProbVectorAlreadyNormalized (vector< acs_double > & *tmpVector,* MTRand & *tmpRandomGenerator*)

Return position of a randomly selected element from a normalized vector containing cumulative values for each element

Version

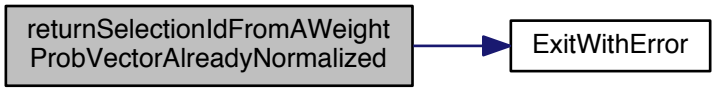
1.0

Parameters

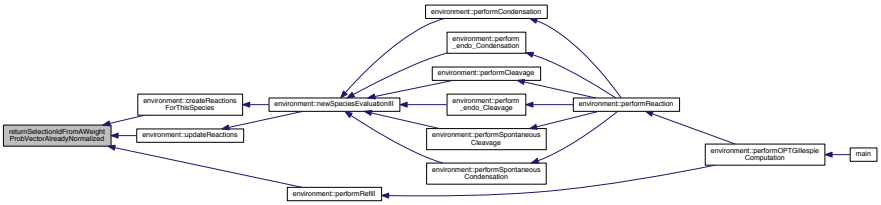
vector<acs_-double>&	tmpQList
MTRand&	tmpRandomGenerator

Definition at line 72 of file commonFunctions.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



13.54.1.12 acs_longInt returnUniformSelection_LONG_IdFromVector (vector< acs_longInt > & *tmpVector,* MTRand & *tmpRandomGenerator*)

Return position of a LONG INT randomly selected element from a normalized vector containing cumulative values for each element

Version

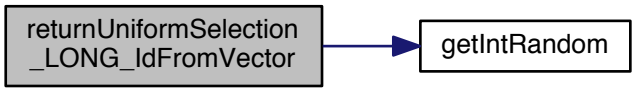
1.0

Parameters

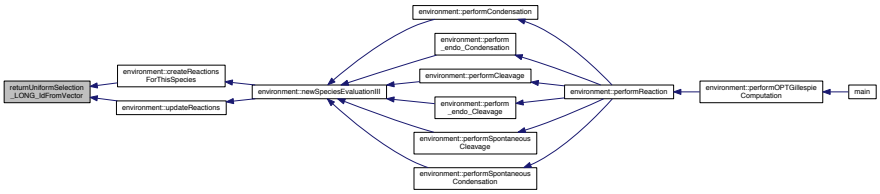
vector<acs_-double>&	tmpVector
MTRand&	tmpRandomGenerator

Definition at line 124 of file commonFunctions.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

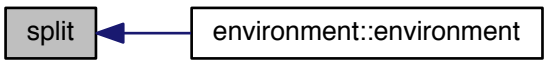


13.54.1.13 vector<string> split (string *str,* const char * *delim*)

Function to split a string and save tokens in a vector

Definition at line 279 of file commonFunctions.cpp.

Here is the caller graph for this function:



13.55 /Users/alessandrofilisetti/Documents/GIT/carness/Debug/catalysis.d File Reference

13.56 /Users/alessandrofilisetti/Documents/GIT/carness/Debug/commonFunctions.d File Reference

13.57 /Users/alessandrofilisetti/Documents/GIT/carness/Debug/environment.d File Reference

13.58 /Users/alessandrofilisetti/Documents/GIT/carness/Debug/gillespie.d File Reference

13.59 /Users/alessandrofilisetti/Documents/GIT/carness/Debug/main.d File Reference

13.60 /Users/alessandrofilisetti/Documents/GIT/carness/Debug/mtrand.d File Reference

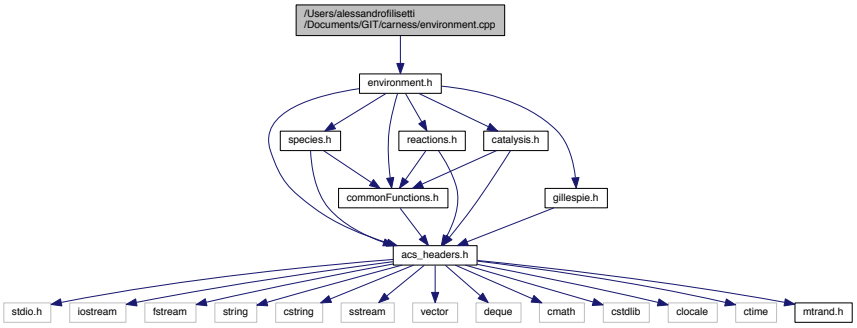
13.61 /Users/alessandrofilisetti/Documents/GIT/carness/Debug/reactions.d File Reference

13.62 /Users/alessandrofilisetti/Documents/GIT/carness/Debug/species.d File Reference

13.63 /Users/alessandrofilisetti/Documents/GIT/carness/environment.cpp File Reference

```
#include "environment.h"
```

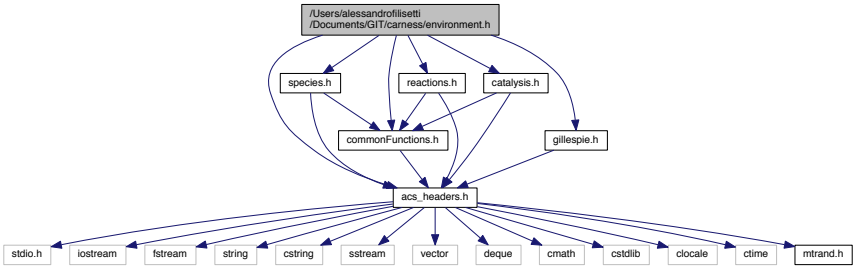
Include dependency graph for environment.cpp:



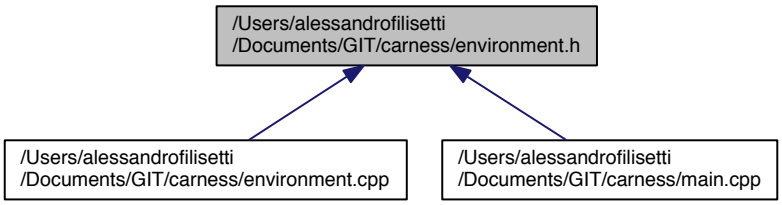
13.64 /Users/alessandrofilisetti/Documents/GIT/carness/environment.h File Reference

```
#include "acs_headers.h"
#include "species.h"
#include "reactions.h"
#include "commonFunctions.h"
#include "catalysis.h"
#include "gillespie.h"
```

Include dependency graph for environment.h:



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



Classes

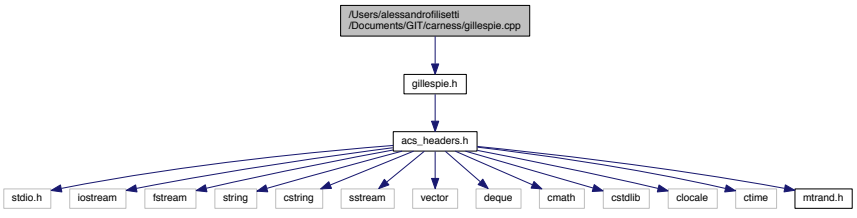
- class environment

  environment class

13.65 /Users/alessandrofilisetti/Documents/GIT/carness/gillespie.cpp File Reference

```
#include "gillespie.h"
```

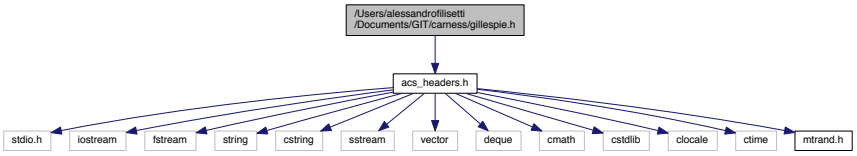
Include dependency graph for gillespie.cpp:



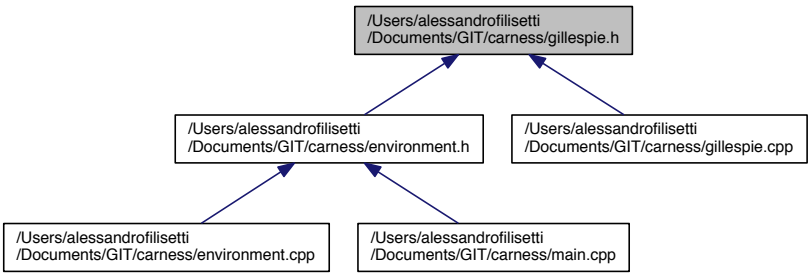
13.66 /Users/alessandrofilisetti/Documents/GIT/carness/gillespie.h File Reference

```
#include "acs_headers.h"
```

Include dependency graph for gillespie.h:



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



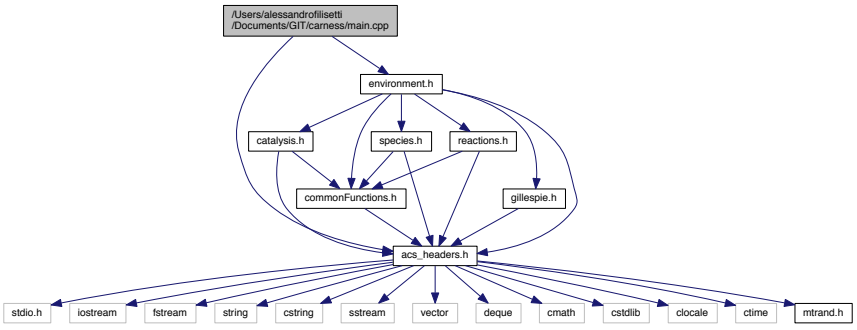
Classes

- class gillespie

13.67 /Users/alessandrofilisetti/Documents/GIT/carness/main.cpp File Reference

```
#include "acs_headers.h"
#include "environment.h"
```

Include dependency graph for main.cpp:



Functions

- void saveToFile (string tmpSavingPath, environment *tmpEnvironment, acs_int tmpGen, acs_int tmpSim, acs_int tmpStep)
- void saveTimesToFile (string tmpSavingPath, environment *tmpEnvironment, acs_int tmpGen, acs_int tmpSim, acs_int tmpStep)
- void saveInitialConditionsToFile (string tmpSavingPath, environment *tmpEnvironment, acs_int tmpGen, acs_int tmpSim, acs_int tmpStep)
- int main (int argc, char *argv[])

13.67.1 Function Documentation

13.67.1.1 int main (int *argc,* char * *argv[]*)

double random number generator

Definition at line 354 of file main.cpp.

13.67.1.2 void saveInitialConditionsToFile (string *tmpSavingPath,* environment * *tmpEnvironment,* acs_int *tmpGen,* acs_int *tmpSim,* acs_int *tmpStep*)

Save to file all the INITIAL structures

Version

2.0

Parameters

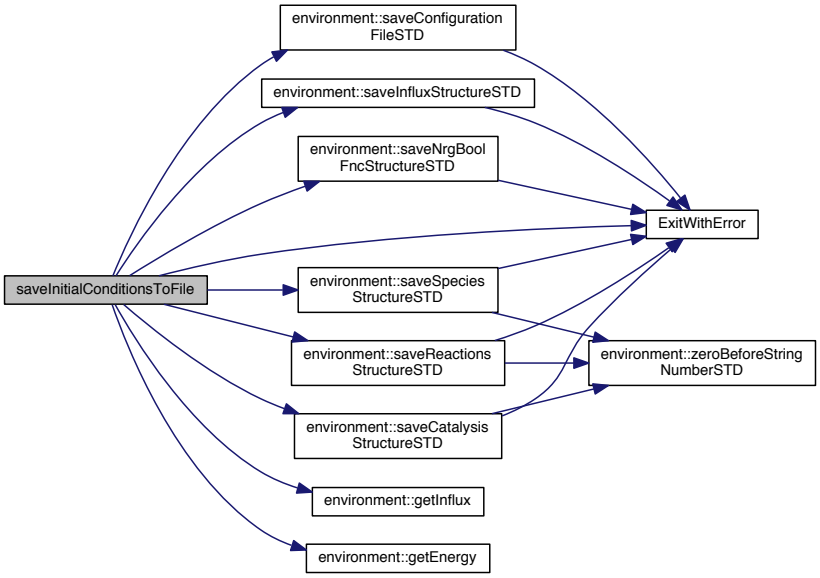
string	tmpSavingPath Saving files path
environment	*tmpEnvironment environment instance reference
tmpSim	Current simulation
acs_int	Current step

Date

2013/07/03

Definition at line 669 of file main.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

13.67.1.3 void saveTimesToFile (string *tmpSavingPath,* environment * *tmpEnvironment,* acs_int *tmpGen,* acs_int *tmpSim,* acs_int *tmpStep*)

Save TIMES to file

Version

2.0

Parameters

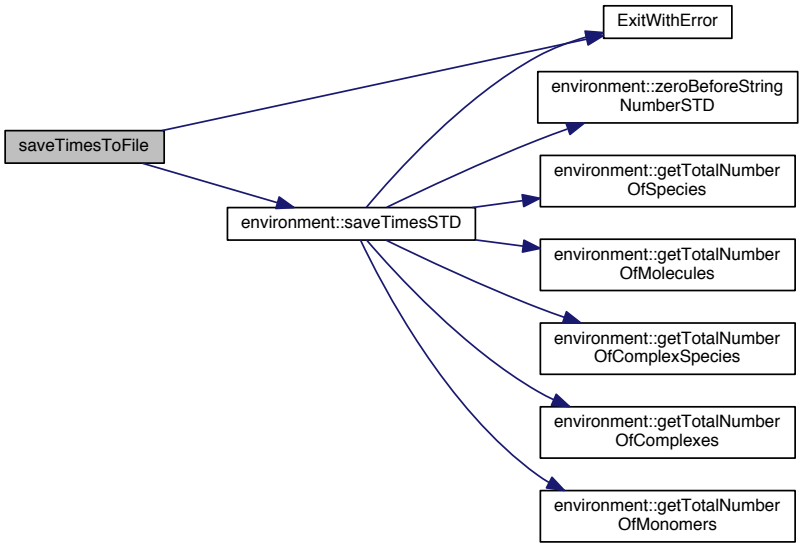
string	tmpSavingPath Saving files path
environment	∗tmpEnvironment environment instance reference
tmpSim	Current simulation
acs_int	Current step

Date

2013/07/03

Definition at line 653 of file main.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

13.67.1.4 void saveToFile (string *tmpSavingPath,* environment ∗ *tmpEnvironment,* acs_int *tmpGen,* acs_int *tmpSim,* acs_int *tmpStep*)

Save to file structures at step tmpStep

Version

2.0

Parameters

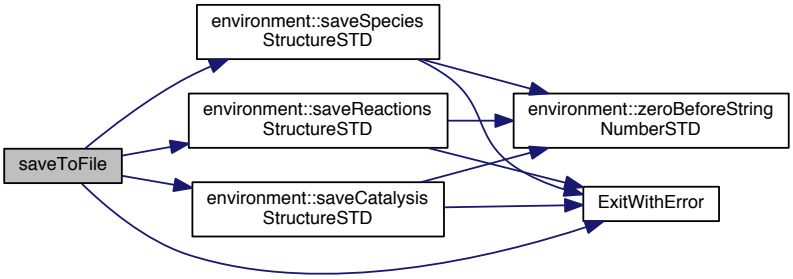
string	tmpSavingPath Saving files path
environment	∗tmpEnvironment environment instance reference
tmpSim	Current simulation
acs_int	Current step

Date

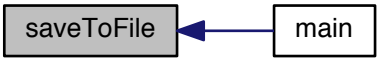
2013/07/13

Definition at line 633 of file main.cpp.

Here is the call graph for this function:



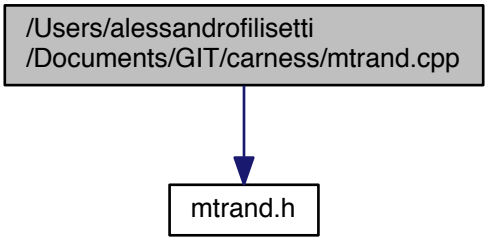
Here is the caller graph for this function:



13.68 /Users/alessandrofilisetti/Documents/GIT/carness/mtrand.cpp File Reference

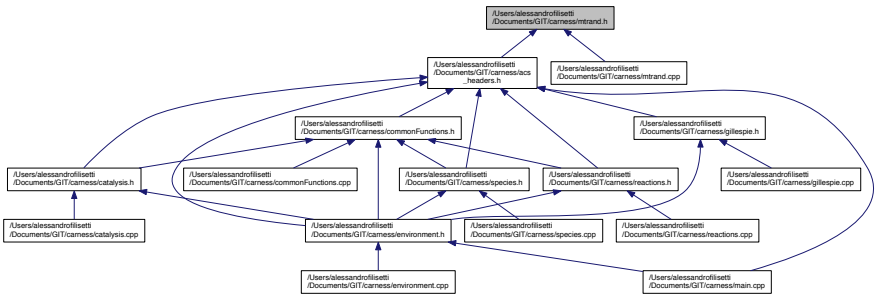
```
#include "mtrand.h"
```

Include dependency graph for mtrand.cpp:



13.69 /Users/alessandrofilisetti/Documents/GIT/carness/mtrand.h File Reference

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



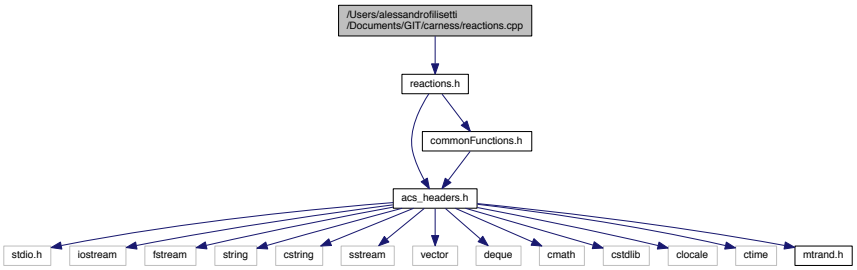
Classes

- class MTRand_int32
- class MTRand
- class MTRand_closed
- class MTRand_open
- class MTRand53

13.70 /Users/alessandrofilisetti/Documents/GIT/carness/reactions.cpp File Reference

```
#include "reactions.h"
```

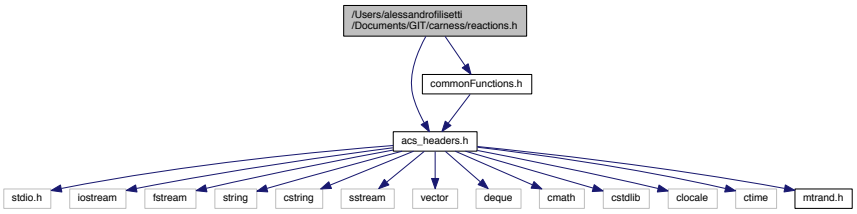
Include dependency graph for reactions.cpp:



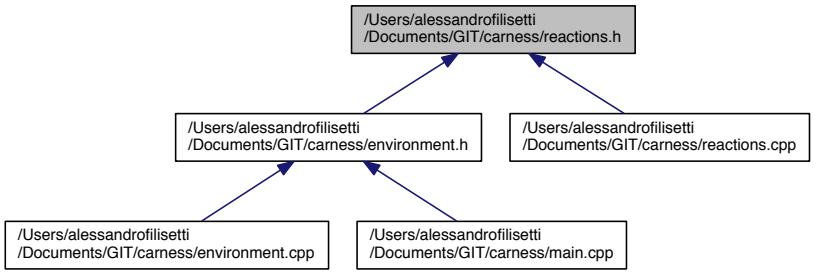
13.71 /Users/alessandrofilisetti/Documents/GIT/carness/reactions.h File Reference

```
#include "acs_headers.h"
#include "commonFunctions.h"
```

Include dependency graph for reactions.h:



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



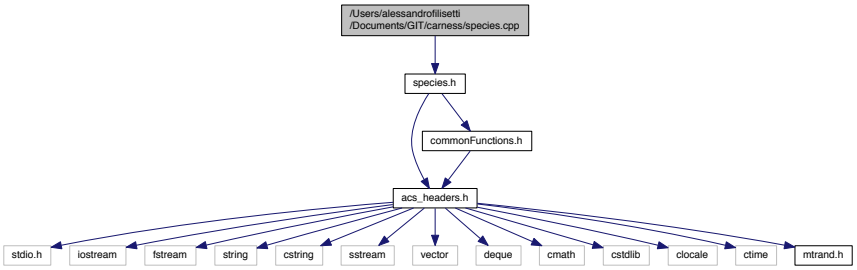
Classes

- class reactions

13.72 /Users/alessandrofilisetti/Documents/GIT/carness/species.cpp File Reference

```
#include "species.h"
```

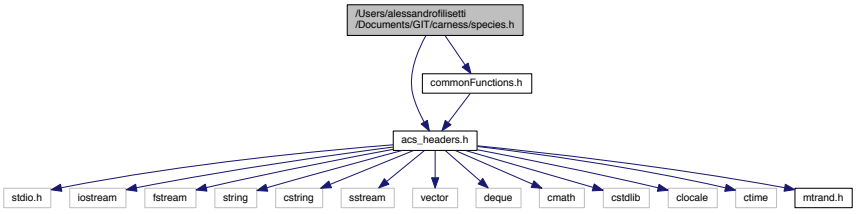
Include dependency graph for species.cpp:



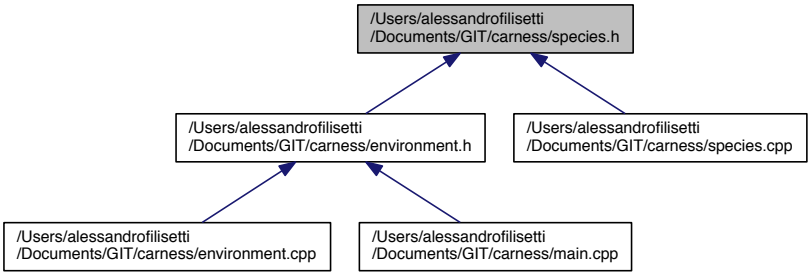
13.73 /Users/alessandrofilisetti/Documents/GIT/carness/species.h File Reference

```
#include "acs_headers.h"
#include "commonFunctions.h"
```

Include dependency graph for species.h:



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



Classes

- class species

  This class contains declarations of the species class.