CaRNeSS

3.2 (20120312.46)

Generated by Doxygen 1.7.6.1

Mon Mar 12 2012 14:17:40

Contents

Chapter 1

Catalytic Rections Network Stochastic Simulator - CaRNeSS 3.2 (20120312.46)

Author

Alessandro Filisetti

Version

3.2 (20120312.46)

Date

2012-03-12 sourceforge repository -- https://carness.svn.sourceforge.net/svnroot/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 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. 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
- QT4 library installed (you can download them from http://qt.nokia.-com/downloads, the download is a little bit large)
- GCC compiler, or similar, installed (if you need to compile the software on your machine)

2.3 IDEs

The QT package contains a very useful and powerful IDE called QT creator in which you can compile and develop your code. Alternatively on Mac Os sytems you can use xCode (http://developer.apple.com/xcode/). To create a valid xCode project it is sufficient, once you have installed the QT libraries, to open the terminal, go into the source code folder and type "qmake -spec macx-xcode QT_acs.pro".

ATTENTION!!! A version of the QT libraries specific for MaxOsX Lion (10.7) has not been yet released so it is not possible to create a valid xCode project, nonetheless you can always work on code by means of QT creator software paying attention to add the line "QMAKE_MAC_SDK = /Developer/SDKs/MacOSX10.-6.sdk" in the QT_acs.pro file

2.4 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. They are not handled from the software, if you like you can rearrange configuration file as you prefer, notice only that comments have to start with character *"#"*. Within the source code folder an example of the acsm2s.conf file is provided.

The following parameters are used both by the initializator and the simulator. - Nvertheless it is ALWAYS necessary having a complete configuration file even if the structures have been already created.

2.4.1 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 nSeconds 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 nReactions)
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)
timeStructuresSavingInterval	($>$ 0) All system structures (species, catalysis and reactions) are saved every *timeStructuresSavingInterval* seconds (simulation time)
fileTimesSaveInterval	($>$= 0) Times data are stored in file times.csv every *fileTimesSaveInterval* seconds

2.4.2 Environment

Parameters

lastFiringDiskSpeciesID	($>$ 0) The ID of the last firing disk species.
overallConcentration	($>$ 0) The overall initial concentration that will be divided between all the initial species according to the selected initial distribution.
ECConcentration	($>$ 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

2.4.3 Dynamic

Parameters

energy	(0 or 1) 0 no energy in the system, 1 energy constraints are applied
ratioSpeciesEnergizable	(%) The probability for a species to be potentially energized by the energy carriers
complexFormationSymmetry	(0 or 1) Complex Formation Symmetry • **1**: the catalyst can bound both substrates • **0 (default)**: catalyst binds only with the first substrate of the reaction
nonCatalyticMaxLength	(>= 0) Max length of non catalytic species
reactionProbability	(from 0 to 1) Probability for a species to catalyze a reaction
cleavageProbability	(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)
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)
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

2.5 Aknowledgments

- 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
- Roberto Serra, Marco Villani, Timoteo Carletti, Alex Graudenzi, 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):

- `acsm2s.conf` (described in the Input Parameters :: acsm2s.conf section)
- `_acsspecies.csv` - This file contains all the initial **species** with their proprieties
- `_acsreactions.csv` - This file contains all the initial **reactions** with their proprieties
- `_acscatalysis.csv` - This file contains all the **correspondances between species and reactions** with their proprieties

and, if the system is open

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

and. if the system is energy based

- `_acsnrgbooleanfunctions.csv` - This file contains all the possible boolean functions associated with the reactions

3.1 _acsspecies.csv

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

- Identification (ID)	- Sequence	- Concentration	- Diffusion enhancement	- Precipitation flag	- Dissociation Kinetic Constant	- Binding Point	- Already evaluated flag	- Species Age	- Number of reborns	- Catalyst ID	- Substrate 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

- *Identificator*: Reaction ID
- *Reaction type*: 0 Cleavage, 1 Condensation
- *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

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

- *Identificator*: Reaction ID
- *Catalyst ID*: species (as catalyst) ID
- *Reaction ID*: Reaction ID
- *Catalysis counter*: Catalysis counter
- *Identificator*: Reaction ID
- *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

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 (ms)	- Number of Species	- Number of Molecules	- Number of complex species	- Number of complexes	- Number of bricks	- Gillespie Computational Time	- Reaction Processes Computational Time	- Various Processes Computational Time	- New species formation computational ability

- *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

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

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.

5.1 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 6

Namespace Index

6.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 7

Class Index

7.1 Class Hierarchy

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

Chapter 8

Class Index

8.1 Class List

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

Chapter 9

File Index

9.1 File List

Here is a list of all files with brief descriptions:

Chapter 10

Namespace Documentation

10.1 filisettiLibrary Namespace Reference

Functions

- def PlotMatrix
- def PlotMatrixLOGY
- def PlotMatrixSingleSpeciesAmounts
- def PlotMatrixML
- def PlotMatrixErrorBar
- def PlotMatrix3D
- def PlotIMSHOWoverThreshold
- def PlotIMSHOW
- def zeroBeforeStrNum
- def writeOverallStatOnFile
- def writeOverallStatOnFileWhereISay

Variables

- int width = 8
- int height = 6

10.1.1 Function Documentation

10.1.1.1 def filisettiLibrary.PlotIMSHOW (*tmpFilename, tmpData, tmpThreshold, tmpDirSuffix*)

Definition at line 184 of file filisettiLibrary.py.

10.1.1.2 def filisettiLibrary.PlotIMSHOWoverThreshold (*tmpFilename, tmpData, tmpThreshold, tmpDirSuffix*)

Definition at line 159 of file filisettiLibrary.py.

10.1.1.3 def filisettiLibrary.PlotMatrix (*tmpFilename, tmpX, tmpY, tmpXlabel, tmpYlabel, tmpLegend, tmpDirSuffix = ''*)

```
PlotMatrix is a function for plotting one or more lines.
        - tmpFilename is the path + file name of the file to save
        - tmpX can be a matrix column or a vector
        - tmpY can be a matrix or a vector
        - tmpTitle, tmpXlabel and tmpYlabel are a strings
        - tmpLegend can be a set of strings or can be set to None value if you dont't need le
```

Definition at line 23 of file filisettiLibrary.py.

10.1.1.4 def filisettiLibrary.PlotMatrix3D (*tmpFilename, tmpX, tmpY, tmpZ, tmpXlabel, tmpYlabel, tmpZlabel*)

```
PlotMatrix is a function for plotting one or more lines.
        - tmpFilename is the path + file name of the file to save
        - tmpX can be a matrix column or a vector
        - tmpY can be a matrix or a vector
        - tmpTitle, tmpXlabel and tmpYlabel are a strings
        - tmpLegend can be a set of strings or can be set to None value if you dont't need le
```

Definition at line 133 of file filisettiLibrary.py.

10.1.1.5 def filisettiLibrary.PlotMatrixErrorBar (*tmpFilename, tmpX, tmpY, errX, errY, tmpXlabel, tmpYlabel, tmpLegend, tmpFolder = '__10_stastisticFiles'*)

```
PlotMatrix is a function for plotting one or more lines.
        - tmpFilename is the path + file name of the file to save
        - tmpX can be a matrix column or a vector
        - tmpY can be a matrix or a vector
        - tmpTitle, tmpXlabel and tmpYlabel are a strings
        - tmpLegend can be a set of strings or can be set to None value if you dont't need le
```

Definition at line 110 of file filisettiLibrary.py.

10.1.1.6 def filisettiLibrary.PlotMatrixLOGY (*tmpFilename, tmpX, tmpY, tmpXlabel, tmpYlabel, tmpLegend, tmpDirSuffix = ''*)

```
PlotMatrix is a function for plotting one or more lines.
        - tmpFilename is the path + file name of the file to save
        - tmpX can be a matrix column or a vector
        - tmpY can be a matrix or a vector
        - tmpTitle, tmpXlabel and tmpYlabel are a strings
        - tmpLegend can be a set of strings or can be set to None value if you dont't need le
```

Definition at line 46 of file filisettiLibrary.py.

10.1.1.7 def filisettiLibrary.PlotMatrixML (*tmpFilename, tmpX, tmpY, tmpXlabel, tmpYlabel, tmpLegend, tmpFolder = '__10_stastisticFiles_'*)

```
PlotMatrix is a function for plotting one or more lines.
        - tmpFilename is the path + file name of the file to save
        - tmpX can be a matrix column or a vector
        - tmpY can be a matrix or a vector
        - tmpTitle, tmpXlabel and tmpYlabel are a strings
        - tmpLegend can be a set of strings or can be set to None value if you dont't need legend
```

Definition at line 91 of file filisettiLibrary.py.

10.1.1.8 def filisettiLibrary.PlotMatrixSingleSpeciesAmounts (*tmpFilename, tmpX, tmpY, tmpA, tmpXlabel, tmpYlabel, tmpDirSuffix = ''*)

```
PlotMatrix is a function for plotting one or more lines.
        - tmpFilename is the path + file name of the file to save
        - tmpX can be a matrix column or a vector
        - tmpY can be a matrix or a vector
        - tmpTitle, tmpXlabel and tmpYlabel are a strings
        - tmpLegend can be a set of strings or can be set to None value if you dont't need legend
```

Definition at line 69 of file filisettiLibrary.py.

10.1.1.9 def filisettiLibrary.writeOverallStatOnFile (*tmpRgen, tmpGenID, tmpName, tmpStats, tmpFormat, tmpSims*)

Definition at line 222 of file filisettiLibrary.py.

10.1.1.10 def filisettiLibrary.writeOverallStatOnFileWhereISay (*tmpRgen, tmpGenID, tmpName, tmpStats, tmpFormat, tmpSims, tmpFolder*)

Definition at line 234 of file filisettiLibrary.py.

10.1.1.11 def filisettiLibrary.zeroBeforeStrNum (*tmpI, tmpL*)

Definition at line 212 of file filisettiLibrary.py.

10.1.2 Variable Documentation

10.1.2.1 int filisettiLibrary::height = 6

Definition at line 18 of file filisettiLibrary.py.

10.1.2.2 int filisettiLibrary::width = 8

Definition at line 17 of file filisettiLibrary.py.

10.2 generalStatistics Namespace Reference

Variables

- tuple initTime = time.time()
- list StrPath = sys.argv[1]
- tuple threshold = float(sys.argv[2])
- list singleGraphCreation = sys.argv[3]
- tuple speciesToObserve = np.array([[1,2,3,4]])
- tuple tmpDirs = sort(os.listdir(StrPath))
- int samples = 0
- int nFolders = 0
- string speciesFilesInThisSim = 'species_*'
- tuple speciesfileslist = sorted(glob.glob(speciesFilesInThisSim))
- string paramFile = "acsm2s.conf"
- tuple fid = open(paramFile, 'r')
- tuple strLine = line.split('=')
- tuple gens = int(strLine[1])
- tuple sims = int(strLine[1])
- tuple rcts = int(strLine[1])
- tuple nsec = int(strLine[1])
- tuple overallTimes = np.zeros((samples,nFolders))
- tuple timesFrames = range(0,samples)
- tuple overallLivSpe = np.zeros((samples,nFolders))
- tuple overallMols = np.zeros((samples,nFolders))
- tuple overallCpx = np.zeros((samples,nFolders))
- tuple overallCpxCopies = np.zeros((samples,nFolders))
- tuple overallDeath = np.zeros((samples,nFolders))
- tuple overallnewSpecies = np.zeros((samples,nFolders))
- tuple overallmaxAmount = np.zeros((samples,nFolders))
- tuple overallminAmount = np.zeros((samples,nFolders))
- tuple overallmeanAmount = np.zeros((samples,nFolders))
- tuple overallmedianAmount = np.zeros((samples,nFolders))
- tuple overallmaxL = np.zeros((samples,nFolders))
- tuple overallminL = np.zeros((samples,nFolders))
- tuple overallmeanL = np.zeros((samples,nFolders))
- tuple overallmedianL = np.zeros((samples,nFolders))
- tuple overallLoadedSpecies = np.zeros((samples,nFolders))
- string ndn = '_0_allStatResults_'
- tuple newdirAllResults = os.path.join(os.curdir, ndn)
- int dirCount = 0
- tuple rgens = fl.zeroBeforeStrNum(1, 1)
- tuple rsims = fl.zeroBeforeStrNum(1, 1)
- string ndnII = '__10_stastisticFiles_'
- tuple newdir = os.path.join(os.curdir, ndnII)
- string influxSpeciesFile = '_acsinflux.csv'

- tuple fidflux = open(influxSpeciesFile, 'r')
- list indexIn = []
- tuple timesfileslist = sorted(glob.glob('times_*'))
- int simulationID = 1
- tuple rrcts = fl.zeroBeforeStrNum(0, rcts)
- tuple rgen = fl.zeroBeforeStrNum(genID, gens)
- tuple species3DSIZEamountLAST = np.array([[int(0), int(0), float(0), int(0)]])
- int speciesFileID = 1
- tuple rsim = fl.zeroBeforeStrNum(simulationID, sims)
- tuple origDir = os.getcwd()
- tuple statDir = os.path.join(os.curdir, newdir)
- string outFnameStat = '_'
- tuple statfilesFlag = os.path.isfile(outFnameStat)
- float rctIDshow = 1.0
- float rctIDshowNoSave = 1.0
- int rctID = 1
- int previousTime = 0
- string filename = '01_gillespie_'
- list stat = []
- list species3Damount = []
- list species3DSIZEamount = []
- tuple concSpeciesHystory = np.zeros((samples,size(speciesToObserve)))
- tuple concSpeciesRelativeHystory = np.zeros((samples,size(speciesToObserve)))
- tuple totTimes = np.zeros((samples,size(speciesToObserve)))
- tuple strs = string.split(root, '_')
- tuple rctNumber = int(strs[len(strs)-1])
- tuple sngSpFileFid = open(sngSpFile, 'r')
- tuple speciesTable = sngSpFileFid.readlines()
- list sID = []
- list loadedSpecies = []
- tuple Kdeg = array(Kdeg)
- list sizes = []
- int seqID = 0
- realTheshold = threshold
- int totAmount = 0
- tuple eqLen = where(species3DSIZEamount[:,0] == len(seq))
- tuple eqLenLAST = where(species3DSIZEamountLAST[:,0] == len(seq))
- list stoPosition = speciesToObserve[0,:]
- tuple MSizes = int(max(sizes))
- tuple mSizes = int(min(sizes))
- tuple avSizes = float(mean(sizes))
- tuple meSizes = float(median(sizes))
- tuple overallControlledSpecies = float(sum(concSpeciesHystory[speciesFileID-1,:]))
- tuple overallControlledSpeciesOverTotConc = overallControlledSpecies/float(sum(amount[cutPnt==0]))

- tuple overallControlledSpeciesOverTotPlusCpxConc = overallControlled-Species/float(sum(amount[:]))
- tuple statControlled = np.vstack([statControlled,(overallControlledSpecies,overall-ControlledSpeciesOverTotConc,overallControlledSpeciesOverTotPlusCpx-Conc)])
- string finalStatDir = '__10_stastisticFiles_'
- species3DNoInflux = species3Damount
- species3DSIZEnoInflux = species3DSIZEamount
- string outFname3DSize = '_'
- string outFname3DSizeGrouped = '_'
- string outFname3DSizeGroupedLAST = '_'
- string outFname3DSizeGroupedLASTtot = '_'
- tuple saveFileStat = open(outFnameStat, 'w')
- tuple saveFile3Dsize = open(outFname3DSize, 'w')
- tuple saveFile3DsizeGR = open(outFname3DSizeGrouped, 'w')
- tuple saveFile3DsizeGRLAST = open(outFname3DSizeGroupedLAST, 'w')
- string outFnameStatToObserve = '_'
- string outFnameToObserve = '_'
- string outFnameRatioToObserve = '_'
- tuple saveFileStatKeepInEye = open(outFnameStatToObserve, 'w')
- tuple saveFileKeepInEye = open(outFnameToObserve, 'w')
- tuple saveFileRatioKeepInEye = open(outFnameRatioToObserve, 'w')
- string sngStatFile = '_'
- tuple sngStatFileFid = open(sngStatFile, 'r')
- int linesID = 0
- tuple Time = float(tmpTime)
- tuple LivSpe = int(tmpLivSpe)
- tuple Mols = float(tmpMols)
- tuple Death = int(tmpDeath)
- tuple NewS = int(tmpNewS)
- tuple MaxA = float(tmpMaxA)
- tuple MinA = float(tmpMinA)
- tuple MeanA = float(tmpMeanA)
- tuple MedianA = float(tmpMedianA)
- tuple Cpx = int(tmpCpx)
- tuple CpxCopies = float(tmpCpxCopies)
- tuple MaxL = int(tmpMaxL)
- tuple MinL = int(tmpMinL)
- tuple MeanL = float(tmpMeanL)
- tuple MedianL = float(tmpMedianL)
- tuple loadedConc = float(tmpLoadedConc)
- tuple statfileslastFID = open(outFname3DSizeGroupedLAST, 'r')
- tuple normTimes = np.mean(overallTimes,1)
- tuple normTimesStd = np.std(overallTimes,1)
- tuple normY = np.mean(overallLivSpe,1)
- tuple normYstd = np.std(overallLivSpe,1)
- tuple endTime = time.time()
- int minutes = 60

10.2.1 Variable Documentation

10.2.1.1 tuple generalStatistics::avSizes = float(mean(sizes))

Definition at line 450 of file generalStatistics.py.

10.2.1.2 tuple generalStatistics::concSpeciesHystory = np.zeros((samples,size(speciesToObserve)))

Definition at line 330 of file generalStatistics.py.

10.2.1.3 tuple generalStatistics::concSpeciesRelativeHystory = np.zeros((samples,size(speciesToObserve)))

Definition at line 331 of file generalStatistics.py.

10.2.1.4 tuple generalStatistics::Cpx = int(tmpCpx)

Definition at line 646 of file generalStatistics.py.

10.2.1.5 tuple generalStatistics::CpxCopies = float(tmpCpxCopies)

Definition at line 647 of file generalStatistics.py.

10.2.1.6 tuple generalStatistics::Death = int(tmpDeath)

Definition at line 640 of file generalStatistics.py.

10.2.1.7 int generalStatistics::dirCount = 0

Definition at line 213 of file generalStatistics.py.

10.2.1.8 tuple generalStatistics::endTime = time.time()

Definition at line 826 of file generalStatistics.py.

10.2.1.9 tuple generalStatistics::eqLen = where(species3DSIZEamount[:,0] == len(seq))

Definition at line 400 of file generalStatistics.py.

10.2.1.10 tuple generalStatistics::eqLenLAST = where(speciesSIZEamountLAST[-:,0] == len(seq))

Definition at line 418 of file generalStatistics.py.

10.2.1.11 tuple generalStatistics::fid = open(paramFile, 'r')

Definition at line 156 of file generalStatistics.py.

10.2.1.12 tuple generalStatistics::fidflux = open(influxSpeciesFile, 'r')

Definition at line 240 of file generalStatistics.py.

10.2.1.13 string generalStatistics::filename = '01_gillespie_'

Definition at line 295 of file generalStatistics.py.

10.2.1.14 string generalStatistics::finalStatDir = '__10_stastisticFiles_'

Definition at line 524 of file generalStatistics.py.

10.2.1.15 tuple generalStatistics::gens = int(strLine[1])

Definition at line 164 of file generalStatistics.py.

10.2.1.16 list generalStatistics::indexIn = [ ]

Definition at line 241 of file generalStatistics.py.

10.2.1.17 string generalStatistics::influxSpeciesFile = '_acsinflux.csv'

Definition at line 238 of file generalStatistics.py.

10.2.1.18 tuple generalStatistics::initTime = time.time()

Definition at line 102 of file generalStatistics.py.

10.2.1.19 tuple generalStatistics::Kdeg = array(Kdeg)

Definition at line 368 of file generalStatistics.py.

10.2.1.20 int generalStatistics::linesID = 0

Definition at line 633 of file generalStatistics.py.

10.2.1.21 tuple generalStatistics::LivSpe = int(tmpLivSpe)

Definition at line 638 of file generalStatistics.py.

10.2.1.22 tuple generalStatistics::loadedConc = float(tmpLoadedConc)

Definition at line 652 of file generalStatistics.py.

10.2.1.23 tuple generalStatistics::loadedSpecies = [ ]

Definition at line 356 of file generalStatistics.py.

10.2.1.24 tuple generalStatistics::MaxA = float(tmpMaxA)

Definition at line 642 of file generalStatistics.py.

10.2.1.25 tuple generalStatistics::MaxL = int(tmpMaxL)

Definition at line 648 of file generalStatistics.py.

10.2.1.26 tuple generalStatistics::MeanA = float(tmpMeanA)

Definition at line 644 of file generalStatistics.py.

10.2.1.27 tuple generalStatistics::MeanL = float(tmpMeanL)

Definition at line 650 of file generalStatistics.py.

10.2.1.28 tuple generalStatistics::MedianA = float(tmpMedianA)

Definition at line 645 of file generalStatistics.py.

10.2.1.29 tuple generalStatistics::MedianL = float(tmpMedianL)

Definition at line 651 of file generalStatistics.py.

10.2.1.30 tuple generalStatistics::meSizes = float(median(sizes))

Definition at line 451 of file generalStatistics.py.

10.2.1.31 tuple generalStatistics::MinA = float(tmpMinA)

Definition at line 643 of file generalStatistics.py.

10.2.1.32 tuple generalStatistics::MinL = int(tmpMinL)

Definition at line 649 of file generalStatistics.py.

10.2.1.33 int generalStatistics::minutes = 60

Definition at line 827 of file generalStatistics.py.

10.2.1.34 tuple generalStatistics::MoIs = float(tmpMoIs)

Definition at line 639 of file generalStatistics.py.

10.2.1.35 tuple generalStatistics::MSizes = int(max(sizes))

Definition at line 448 of file generalStatistics.py.

10.2.1.36 tuple generalStatistics::mSizes = int(min(sizes))

Definition at line 449 of file generalStatistics.py.

10.2.1.37 string generalStatistics::ndn = '_0_allStatResults_'

Definition at line 204 of file generalStatistics.py.

10.2.1.38 string generalStatistics::ndnII = '__10_stastisticFiles_'

Definition at line 229 of file generalStatistics.py.

10.2.1.39 tuple generalStatistics::newdir = os.path.join(os.curdir, ndnII)

Definition at line 230 of file generalStatistics.py.

10.2.1.40 tuple generalStatistics::newdirAllResults = os.path.join(os.curdir, ndn)

Definition at line 205 of file generalStatistics.py.

10.2.1.41 tuple generalStatistics::NewS = int(tmpNewS)

Definition at line 641 of file generalStatistics.py.

10.2.1.42 int generalStatistics::nFolders = 0

Definition at line 137 of file generalStatistics.py.

10.2.1.43 tuple generalStatistics::normTimes = np.mean(overallTimes,1)

Definition at line 721 of file generalStatistics.py.

10.2.1.44 tuple generalStatistics::normTimesStd = np.std(overallTimes,1)

Definition at line 722 of file generalStatistics.py.

10.2.1.45 tuple generalStatistics::normY = np.mean(overallLivSpe,1)

Definition at line 723 of file generalStatistics.py.

10.2.1.46 tuple generalStatistics::normYstd = np.std(overallLivSpe,1)

Definition at line 724 of file generalStatistics.py.

10.2.1.47 tuple generalStatistics::nsec = int(strLine[1])

Definition at line 170 of file generalStatistics.py.

10.2.1.48 tuple generalStatistics::origDir = os.getcwd()

Definition at line 274 of file generalStatistics.py.

10.2.1.49 string generalStatistics::outFname3DSize = '_'

Definition at line 564 of file generalStatistics.py.

10.2.1.50 string generalStatistics::outFname3DSizeGrouped = '_'

Definition at line 565 of file generalStatistics.py.

10.2.1.51 string generalStatistics::outFname3DSizeGroupedLAST = '_'

Definition at line 566 of file generalStatistics.py.

10.2.1.52 string generalStatistics::outFname3DSizeGroupedLASTtot = '_'

Definition at line 567 of file generalStatistics.py.

10.2.1.53 string generalStatistics::outFnameRatioToObserve = '_'

Definition at line 594 of file generalStatistics.py.

10.2.1.54 string generalStatistics::outFnameStat = '_'

Definition at line 276 of file generalStatistics.py.

10.2.1.55 string generalStatistics::outFnameStatToObserve = '_'

Definition at line 592 of file generalStatistics.py.

10.2.1.56 string generalStatistics::outFnameToObserve = '_'

Definition at line 593 of file generalStatistics.py.

10.2.1.57 tuple generalStatistics::overallControlledSpecies = float(sum(concSpeciesHystory[speciesFileID-1,:]))

Definition at line 468 of file generalStatistics.py.

10.2.1.58 tuple generalStatistics::overallControlledSpeciesOverTotConc = overallControlledSpecies/float(sum(amount[cutPnt==0]))

Definition at line 469 of file generalStatistics.py.

10.2.1.59 tuple generalStatistics::overallControlledSpeciesOverTotPlusCpxConc = overallControlledSpecies/float(sum(amount[:]))

Definition at line 470 of file generalStatistics.py.

10.2.1.60 tuple generalStatistics::overallCpx = np.zeros((samples,nFolders))

Definition at line 188 of file generalStatistics.py.

10.2.1.61 tuple generalStatistics::overallCpxCopies = np.zeros((samples,nFolders))

Definition at line 189 of file generalStatistics.py.

10.2.1.62 tuple generalStatistics::overallDeath = np.zeros((samples,nFolders))

Definition at line 190 of file generalStatistics.py.

10.2.1.63 tuple generalStatistics::overallLivSpe = np.zeros((samples,nFolders))

Definition at line 186 of file generalStatistics.py.

10.2.1.64 tuple generalStatistics::overallLoadedSpecies = np.zeros((samples,nFolders))

Definition at line 200 of file generalStatistics.py.

10.2.1.65 tuple generalStatistics::overallmaxAmount = np.zeros((samples,nFolders))

Definition at line 192 of file generalStatistics.py.

10.2.1.66 tuple generalStatistics::overallmaxL = np.zeros((samples,nFolders))

Definition at line 196 of file generalStatistics.py.

10.2.1.67 tuple generalStatistics::overallmeanAmount = np.zeros((samples,nFolders))

Definition at line 194 of file generalStatistics.py.

10.2.1.68 tuple generalStatistics::overallmeanL = np.zeros((samples,nFolders))

Definition at line 198 of file generalStatistics.py.

10.2.1.69 tuple generalStatistics::overallmedianAmount = np.zeros((samples,nFolders))

Definition at line 195 of file generalStatistics.py.

10.2.1.70 tuple generalStatistics::overallmedianL = np.zeros((samples,nFolders))

Definition at line 199 of file generalStatistics.py.

10.2.1.71 tuple generalStatistics::overallminAmount = np.zeros((samples,nFolders))

Definition at line 193 of file generalStatistics.py.

10.2.1.72 tuple generalStatistics::overallminL = np.zeros((samples,nFolders))

Definition at line 197 of file generalStatistics.py.

10.2.1.73 tuple generalStatistics::overallMols = np.zeros((samples,nFolders))

Definition at line 187 of file generalStatistics.py.

10.2.1.74 tuple generalStatistics::overallnewSpecies = np.zeros((samples,nFolders))

Definition at line 191 of file generalStatistics.py.

10.2.1.75 tuple generalStatistics::overallTimes = np.zeros((samples,nFolders))

Definition at line 175 of file generalStatistics.py.

10.2.1.76 string generalStatistics::paramFile = "acsm2s.conf"

Definition at line 153 of file generalStatistics.py.

10.2.1.77 int generalStatistics::previousTime = 0

Definition at line 286 of file generalStatistics.py.

10.2.1.78 int generalStatistics::rctID = 1

Definition at line 285 of file generalStatistics.py.

10.2.1.79 float generalStatistics::rctIDshow = 1.0

Definition at line 283 of file generalStatistics.py.

10.2.1.80 float generalStatistics::rctIDshowNoSave = 1.0

Definition at line 284 of file generalStatistics.py.

10.2.1.81 tuple generalStatistics::rctNumber = int(strs[len(strs)-1])

Definition at line 338 of file generalStatistics.py.

10.2.1.82 tuple generalStatistics::rcts = int(strLine[1])

Definition at line 168 of file generalStatistics.py.

10.2.1.83 generalStatistics::realTheshold = threshold

Definition at line 377 of file generalStatistics.py.

10.2.1.84 tuple generalStatistics::rgen = fl.zeroBeforeStrNum(genID, gens)

Definition at line 265 of file generalStatistics.py.

10.2.1.85 tuple generalStatistics::rgens = fl.zeroBeforeStrNum(1, 1)

Definition at line 214 of file generalStatistics.py.

10.2.1.86 tuple generalStatistics::rrcts = fl.zeroBeforeStrNum(0, rcts)

Definition at line 263 of file generalStatistics.py.

10.2.1.87 tuple generalStatistics::rsim = fl.zeroBeforeStrNum(simulationID, sims)

Definition at line 272 of file generalStatistics.py.

10.2.1.88 tuple generalStatistics::rsims = fl.zeroBeforeStrNum(1, 1)

Definition at line 215 of file generalStatistics.py.

10.2.1.89 tuple generalStatistics::samples = 0

Definition at line 135 of file generalStatistics.py.

10.2.1.90 tuple generalStatistics::saveFile3Dsize = open(outFname3DSize, 'w')

Definition at line 569 of file generalStatistics.py.

10.2.1.91 tuple generalStatistics::saveFile3DsizeGR = open(outFname3DSizeGrouped, 'w')

Definition at line 570 of file generalStatistics.py.

10.2.1.92 tuple generalStatistics::saveFile3DsizeGRLAST = open(outFname3DSizeGroupedLAST, 'w')

Definition at line 571 of file generalStatistics.py.

10.2.1.93 tuple generalStatistics::saveFileKeepInEye = open(outFnameToObserve, 'w')

Definition at line 596 of file generalStatistics.py.

10.2.1.94 tuple generalStatistics::saveFileRatioKeepInEye = open(outFnameRatioToObserve, 'w')

Definition at line 597 of file generalStatistics.py.

10.2.1.95 tuple generalStatistics::saveFileStat = open(outFnameStat, 'w')

Definition at line 568 of file generalStatistics.py.

10.2.1.96 tuple generalStatistics::saveFileStatKeepInEye = open(outFnameStatToObserve, 'w')

Definition at line 595 of file generalStatistics.py.

10.2.1.97 int generalStatistics::seqID = 0

Definition at line 376 of file generalStatistics.py.

10.2.1.98 tuple generalStatistics::sID = [ ]

Definition at line 355 of file generalStatistics.py.

10.2.1.99 tuple generalStatistics::sims = int(strLine[1])

Definition at line 166 of file generalStatistics.py.

10.2.1.100 int generalStatistics::simulationID = 1

Definition at line 262 of file generalStatistics.py.

10.2.1.101 int generalStatistics::singleGraphCreation = sys.argv[3]

Definition at line 116 of file generalStatistics.py.

10.2.1.102 list generalStatistics::sizes = [ ]

Definition at line 375 of file generalStatistics.py.

10.2.1.103 tuple generalStatistics::sngSpFileFid = open(sngSpFile, 'r')

Definition at line 343 of file generalStatistics.py.

10.2.1.104 string generalStatistics::sngStatFile = '_'

Definition at line 624 of file generalStatistics.py.

10.2.1.105 tuple generalStatistics::sngStatFileFid = open(sngStatFile, 'r')

Definition at line 628 of file generalStatistics.py.

10.2.1.106 tuple generalStatistics::species3Damount = [ ]

Definition at line 325 of file generalStatistics.py.

10.2.1.107 list generalStatistics::species3DNoInflux = species3Damount

Definition at line 531 of file generalStatistics.py.

10.2.1.108 tuple generalStatistics::species3DSIZEamount = [ ]

Definition at line 326 of file generalStatistics.py.

10.2.1.109 tuple generalStatistics::species3DSlZEamountLAST = np.array([[int(0), int(0), float(0), int(0)]])

Definition at line 267 of file generalStatistics.py.

10.2.1.110 list generalStatistics::species3DSlZEnoInflux = species3DSlZEamount

Definition at line 532 of file generalStatistics.py.

10.2.1.111 int generalStatistics::speciesFileID = 1

Definition at line 271 of file generalStatistics.py.

10.2.1.112 string generalStatistics::speciesFilesInThisSim = 'species_*'

Definition at line 147 of file generalStatistics.py.

10.2.1.113 tuple generalStatistics::speciesfileslist = sorted(glob.glob(speciesFilesInThisSim))

Definition at line 148 of file generalStatistics.py.

10.2.1.114 tuple generalStatistics::speciesTable = sngSpFileFid.readlines()

Definition at line 344 of file generalStatistics.py.

10.2.1.115 tuple generalStatistics::speciesToObserve = np.array([[1,2,3,4]])

Definition at line 123 of file generalStatistics.py.

10.2.1.116 tuple generalStatistics::stat = [ ]

Definition at line 324 of file generalStatistics.py.

10.2.1.117 tuple generalStatistics::statControlled = np.vstack([statControlled,(overallControlledSpecies,overallControlled-SpeciesOverTotConc,overallControlledSpeciesOverTotPlusCpxConc)])

Definition at line 472 of file generalStatistics.py.

10.2.1.118 tuple generalStatistics::statDir = os.path.join(os.curdir, newdir)

Definition at line 275 of file generalStatistics.py.

10.2.1.119 generalStatistics::statfilesFlag = os.path.isfile(outFnameStat)

Definition at line 278 of file generalStatistics.py.

10.2.1.120 tuple generalStatistics::statfileslastFID = open(outFname3DSizeGroupedLAST, 'r')

Definition at line 678 of file generalStatistics.py.

10.2.1.121 list generalStatistics::stoPosition = speciesToObserve[0,:]

Definition at line 432 of file generalStatistics.py.

10.2.1.122 tuple generalStatistics::strLine = line.split('=')

Definition at line 162 of file generalStatistics.py.

10.2.1.123 list generalStatistics::StrPath = sys.argv[1]

Definition at line 110 of file generalStatistics.py.

10.2.1.124 tuple generalStatistics::strs = string.split(root, '_')

Definition at line 337 of file generalStatistics.py.

10.2.1.125 tuple generalStatistics::threshold = float(sys.argv[2])

Definition at line 111 of file generalStatistics.py.

10.2.1.126 tuple generalStatistics::Time = float(tmpTime)

Definition at line 637 of file generalStatistics.py.

10.2.1.127 tuple generalStatistics::timesfileslist = sorted(glob.glob('times_*'))

Definition at line 261 of file generalStatistics.py.

10.2.1.128 tuple generalStatistics::timesFrames = range(0,samples)

Definition at line 176 of file generalStatistics.py.

10.2.1.129 tuple generalStatistics::tmpDirs = sort(os.listdir(StrPath))

Definition at line 127 of file generalStatistics.py.

10.2.1.130 int generalStatistics::totAmount = 0

Definition at line 378 of file generalStatistics.py.

10.2.1.131 tuple generalStatistics::totTimes = np.zeros((samples,size(speciesToObserve)))

Definition at line 332 of file generalStatistics.py.

Chapter 11

Class Documentation

11.1 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)
- ~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
- void updateTotAmount ()
- void resetEventsCounter ()

11.1.1 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.

11.1.2 Constructor & Destructor Documentation

11.1.2.1 catalysis::catalysis ( )

11.1.2.2 catalysis::catalysis (acs_longInt *tmpCatId,* acs_longInt *tmpCat,* acs_longInt *tmpRctId,* acs_longInt *tmpAmount,* acs_double *tmpKass,* acs_double *tmpKdiss,* acs_double *tmpK_cpx*)

catalysis class constructor (FROM FILE)

Version

0.1

Date

2010-03-16

Definition at line 19 of file catalysis.cpp.

11.1.2.3 catalysis::~catalysis ( ) [inline]

Definition at line 34 of file catalysis.h.

11.1.3 Member Function Documentation

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

Definition at line 38 of file catalysis.h.

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

Definition at line 37 of file catalysis.h.

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

Definition at line 43 of file catalysis.h.

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

Definition at line 41 of file catalysis.h.

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

Definition at line 42 of file catalysis.h.

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

Definition at line 39 of file catalysis.h.

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

Definition at line 40 of file catalysis.h.

11.1.3.8 void catalysis::resetEventsCounter ( ) [inline]

Definition at line 48 of file catalysis.h.

11.1.3.9 void catalysis::updateTotAmount ( ) [inline]

Definition at line 47 of file catalysis.h.

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

- /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness_simulator/catalysis.h
- /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness_simulator/catalysis.cpp

11.2 commonFunctions Class Reference

This class contains all the common function of the system.

`#include <commonFunctions.h>`

11.2.1 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/Dropbox/ACS_SVN/carness/carness_simulator/commonFunctions.h

11.3 environment Class Reference

environment class

`#include <environment.h>`

Public Member Functions

- environment ()
- environment (QString tmpInitialPath)
- ~environment ()
- acs_int getNgen () const
- acs_int getNsim () const
- acs_double getActualTime () const
- acs_double getNseconds () const
- acs_int getNreactions () const
- acs_int 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
- bool getComplexFormationSymmetry () 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 getKdiss () const
- acs_double getKass () const
- acs_double getKcpx () const
- acs_double getKcpxDiss () const
- acs_double getKnrg () const
- acs_double getKirrad () 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 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 getOverallLoadedMolsCounter () const
- acs_int getTotNumberOfChargedMols ()
- void showGlobalParameter ()
- void printInitialCondition ()
- void printAllSpeciesIdAndSequence ()
- void printGillespieStructure ()
- void printNutrientsAndProbability ()
- void printAllChargeMols ()
- bool createInitialMoleculesPopulationFromFile (QString tmpSpeciesFilePath)
- bool createInitialReactionsLayerFromFile (QString tmpReactionsFilePath)
- bool createInitialCatalysisLayerFromFile (QString tmpCatalysisFilePath)
- bool createInfluxLayersFromFile (QString tmpInfluxFilePath)
- bool createNrgBooleanFunctionsFromFile (QString tmpInfluxFilePath)
- bool createInitialMoleculesPopulationFromSpecificFile (QString tmpSpeciesFilePath, acs_int tmpActGEN, acs_int tmpActSIM)
- bool createInitialReactionsLayerFromSpecificFile (QString tmpReactionsFilePath, acs_int tmpActGEN, acs_int tmpActSIM)
- bool createInitialCatalysisLayerFromSpecificFile (QString 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)
- 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 performGillespieComputation (MTRand &tmpRndDoubleGen, QTime &tmpTimeElapsed, acs_int tmpActGEN, acs_int tmpActSIM, acs_int tmpActSTEP, QString tmpStoringPath)

- bool performReaction (acs_longInt reaction_u, MTRand &tmp_RndDoubleGen, acs_int tmp_ActGEN, acs_int tmp_ActSIM, acs_int tmp_ActSTEP, QString tmp_StoringPath)
- bool newSpeciesEvaluation (string tmpNewSpecies, MTRand &tmp___RndDoubleGen)
- bool complexEvaluation (string tmpComplex, MTRand &tmp___RndDoubleGen, acs_int tmpCuttingPnt, acs_int tmpCatalyst_ID, acs_int tmpSubstrate_ID, bool tmpCpxType)
- acs_double computeSinglGilScore (acs_longInt tmpAmountI, acs_double tmpDifI, acs_int tmpSoII, acs_longInt tmpAmountII, acs_double tmpDifII, acs_int tmpSoIII, 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 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 incOverallLoadedMolsCounter ()
- void decOverallLoadedMolsCounter ()
- void resetCleavageCounter ()
- void resetEndoCleavageCounter ()
- void resetCondensationCounter ()
- void resetEndoCondensationCounter ()
- void resetOverallLoadedMolsCounter ()
- 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)
- 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, MTRand &tmp__RndDoubleGen)
- bool perform_endo_ComplexFormation (acs_longInt tmpCatalyst, acs_longInt tmpSubstrate, acs_int tmpNRGSide, MTRand &tmp__RndDoubleGen)
- bool performComplexDissociation (acs_longInt tmpComplex, acs_longInt tmpCatalyst, acs_longInt tmpSubstrate, 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)
- bool saveConfigurationFile (QString tmpStoringPath)
- bool saveInfluxStructure (QString tmpStoringPath)
- bool saveNrgBoolFncStructure (QString tmpStoringPath)
- QString zeroBeforeStringNumber (acs_int tmpTotN, acs_int tmpCurrentN)
- bool saveSpeciesStructure (acs_int tmpCurrentGen, acs_int tmpCurrentSim, acs_int tmpCurrentStep, QString tmpStoringPath)
- bool saveReactionsStructure (acs_int tmpCurrentGen, acs_int tmpCurrentSim, acs_int tmpCurrentStep, QString tmpStoringPath)

- bool saveCatalysisStructure (acs_int tmpCurrentGen, acs_int tmpCurrentSim, acs_int tmpCurrentStep, QString tmpStoringPath)
- bool saveTimes (acs_int tmpCurrentGen, acs_int tmpCurrentSim, acs_int tmpCurrentStep, QString tmpStoringPath)
- bool saveReactionsParameters (acs_int tmp__CurrentGen, acs_int tmp__CurrentSim, acs_int tmp__CurrentStep, QString tmp__StoringPath, acs_int tmpRctType, acs_longInt tmpCat, acs_longInt tmpMol_I, acs_longInt tmpMol_II, acs_longInt tmpMol_III)
- bool saveLivingSpeciesID (acs_int tmp__CurrentGen, acs_int tmp__CurrentSim, acs_int tmp__CurrentStep, QString tmp__StoringPath)
- bool saveLivingSpeciesAmount (acs_int tmp__CurrentGen, acs_int tmp__CurrentSim, QString tmp__StoringPath)
- bool saveLivingSpeciesConcentration (acs_int tmp__CurrentGen, acs_int tmp_-_CurrentSim, QString tmp__StoringPath)
- bool devStd ()
- bool entropy ()

11.3.1 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.

11.3.2 Constructor & Destructor Documentation

11.3.2.1 environment::environment ( )

Test environment costructor

Version

2.4

Date

2010-06-27

Parameters

tmpRnd-DoubleGen	randomGenerator reference

Definition at line 20 of file environment.cpp.

11.3.2.2 environment::environment (QString *tmpInitialPath*)

Environment Constructor

Version

1.0

Parameters

tmpInitial-Path	

Definition at line 73 of file environment.cpp.

11.3.2.3 environment::~environment ( ) [inline]

Definition at line 137 of file environment.h.

11.3.3 Member Function Documentation

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

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

Version

1.0

Date

2010-10-10

Parameters

acs_int	tmpSpeciesID Specie to charge

Definition at line 3533 of file environment.cpp.

11.3.3.2 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 2087 of file environment.cpp.

11.3.3.3 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 1557 of file environment.cpp.

11.3.3.4 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
tmpId-Reaction	reaction ID

Definition at line 1529 of file environment.cpp.

11.3.3.5 void environment::clearAllStructures ( )

Clear all structures after each simulation

Version

1.0

Definition at line 5449 of file environment.cpp.

11.3.3.6 bool environment::complexEvaluation (string *tmpComplex,* MTRand & *tmp___RndDoubleGen,* acs_int *tmpCuttingPnt,* acs_int *tmpCatalyst_ID,* acs_int *tmpSubstrate_ID,* bool *tmpCpxType*)

Evaluate new species

Version

1.1

Date

2010-06-04

Parameters

string	tmpNewSpecies New species sequence to evaluate
MTRand&	tmp___RndDoubleGen random number generator
tmpCutting-Pnt	Complex cutting point

tmpCatalyst_ID	catalyst ID
tmpSubstrate_ID	substrate ID
tmp_catalysisID	catalysis ID
tmpCpxType	ENDOERGONIC or ESOERGONIC

Definition at line 5112 of file environment.cpp.

11.3.3.7 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 2738 of file environment.cpp.

11.3.3.8 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 468 of file environment.cpp.

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

Create the initial amount of the molecules belonging to a particular species according to the species length

Version

1.0

Parameters

tmpAlphabetLength	Number of symbols in the alphabet
tmpSpeciesLength	Lenght of the species Create the initial concentration of the species according to the species length, the alphabet and the overall concentration

Version

1.0

Parameters

tmpAlphabetLength	Number of symbols in the alphabet
tmpSpeciesLength	Lenght of the species 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
tmpNew-Species-Length	Lenght of the species

Definition at line 1411 of file environment.cpp.

11.3.3.10 bool environment::createInfluxLayersFromFile (QString *tmpInfluxFilePath*)

Create influx layer from file

Version

1.0

Parameters

QString	tmpInfluxFilePath file path

Date

2010-05-18

Definition at line 1071 of file environment.cpp.

11.3.3.11 bool environment::createInitialCatalysisLayerFromFile (QString *tmpCatalysisFilePath*)

Initial catalysis layer creation from file

Version

1.0

Parameters

QString	tmpSpeciesFilePath file path

Definition at line 1237 of file environment.cpp.

11.3.3.12 bool environment::createInitialCatalysisLayerFromSpecificFile (QString *tmpCatalysisFilePath,* acs_int *tmpActGEN,* acs_int *tmpActSIM*)

Initial catalysis layer creation from SPECIFIC file

Version

1.0

Parameters

QString	tmpSpeciesFilePath file path

Definition at line 1278 of file environment.cpp.

11.3.3.13 bool environment::createInitialMoleculesPopulationFromFile (QString *tmpSpeciesFilePath*)

Initial molecule population creation from file

Version

1.0

Parameters

QString	tmpSpeciesFilePath file path

Definition at line 929 of file environment.cpp.

11.3.3.14 bool environment::createInitialMoleculesPopulationFromSpecificFile (QString *tmpSpeciesFilePath,* acs_int *tmpActGEN,* acs_int *tmpActSIM*)

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

Definition at line 1000 of file environment.cpp.

11.3.3.15 bool environment::createInitialReactionsLayerFromFile (QString *tmpReactionsFilePath*)

Initial reactions layer creation from file

Version

1.0

Parameters

QString	tmpSpeciesFilePath file path

Definition at line 1147 of file environment.cpp.

11.3.3.16 bool environment::createInitialReactionsLayerFromSpecificFile (QString *tmpReactionsFilePath,* acs_int *tmpActGEN,* acs_int *tmpActSIM*)

Initial reactions layer creation from SPECIFIC file

Version

1.0

Parameters

QString	tmpSpeciesFilePath file path

Definition at line 1186 of file environment.cpp.

11.3.3.17 bool environment::createNrgBooleanFunctionsFromFile (QString *tmpInfluxFilePath*)

load energy boolean function (in decimal format)

Version

1.0

Parameters

QString	tmpInfluxFilePath file path

Date

2011-04-13

Definition at line 1110 of file environment.cpp.

11.3.3.18 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 512 of file environment.cpp.

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

Definition at line 275 of file environment.h.

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

Definition at line 274 of file environment.h.

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

Definition at line 265 of file environment.h.

11.3.3.22 void environment::decNumberOfCpxMols ( ) [inline]

Definition at line 267 of file environment.h.

11.3.3.23 void environment::decNumberOfMols ( ) [inline]

Definition at line 263 of file environment.h.

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

Definition at line 272 of file environment.h.

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

Definition at line 270 of file environment.h.

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

Definition at line 261 of file environment.h.

11.3.3.27 void environment::decOverallLoadedMolsCounter ( ) [inline]

Definition at line 286 of file environment.h.

11.3.3.28 bool environment::devStd ( )

Definition at line 6313 of file environment.cpp.

11.3.3.29 bool environment::entropy ( )

Definition at line 6331 of file environment.cpp.

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

Definition at line 142 of file environment.h.

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

Definition at line 159 of file environment.h.

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

Definition at line 193 of file environment.h.

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

Definition at line 160 of file environment.h.

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

Definition at line 211 of file environment.h.

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

Definition at line 181 of file environment.h.

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

Definition at line 155 of file environment.h.

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

Definition at line 184 of file environment.h.

11.3.3.38 bool environment::getComplexFormationSymmetry ( ) const [inline]

Definition at line 152 of file environment.h.

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

Definition at line 182 of file environment.h.

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

Definition at line 213 of file environment.h.

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

Definition at line 183 of file environment.h.

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

Definition at line 147 of file environment.h.

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

Definition at line 203 of file environment.h.

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

Definition at line 190 of file environment.h.

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

Definition at line 212 of file environment.h.

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

Definition at line 214 of file environment.h.

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

Definition at line 157 of file environment.h.

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

Definition at line 149 of file environment.h.

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

Definition at line 170 of file environment.h.

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

Definition at line 168 of file environment.h.

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

Definition at line 169 of file environment.h.

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

Definition at line 191 of file environment.h.

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

Definition at line 175 of file environment.h.

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

Definition at line 176 of file environment.h.

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

Definition at line 177 of file environment.h.

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

Definition at line 174 of file environment.h.

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

Definition at line 179 of file environment.h.

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

Definition at line 178 of file environment.h.

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

Definition at line 150 of file environment.h.

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

Definition at line 146 of file environment.h.

11.3.3.61 acs_int environment::getMAXhours ( ) const [inline]

Definition at line 145 of file environment.h.

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

Definition at line 186 of file environment.h.

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

Definition at line 153 of file environment.h.

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

Definition at line 185 of file environment.h.

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

Definition at line 196 of file environment.h.

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

Definition at line 161 of file environment.h.

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

Definition at line 165 of file environment.h.

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

Definition at line 166 of file environment.h.

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

Definition at line 162 of file environment.h.

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

Definition at line 140 of file environment.h.

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

Definition at line 164 of file environment.h.

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

Definition at line 144 of file environment.h.

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

Definition at line 143 of file environment.h.

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

Definition at line 141 of file environment.h.

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

Definition at line 163 of file environment.h.

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

Definition at line 206 of file environment.h.

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

Definition at line 207 of file environment.h.

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

Definition at line 208 of file environment.h.

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

Definition at line 205 of file environment.h.

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

Definition at line 204 of file environment.h.

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

Definition at line 151 of file environment.h.

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

Definition at line 215 of file environment.h.

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

Definition at line 195 of file environment.h.

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

Definition at line 171 of file environment.h.

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

Definition at line 158 of file environment.h.

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

Definition at line 154 of file environment.h.

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

Definition at line 202 of file environment.h.

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

Definition at line 192 of file environment.h.

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

Definition at line 156 of file environment.h.

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

Definition at line 187 of file environment.h.

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

Definition at line 148 of file environment.h.

11.3.3.92 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 2926 of file environment.cpp.

11.3.3.93 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 2905 of file environment.cpp.

11.3.3.94 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 2881 of file environment.cpp.

11.3.3.95 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 2947 of file environment.cpp.

11.3.3.96 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 2857 of file environment.cpp.

11.3.3.97 acs_int environment::getTotNumberOfChargedMols ( )

Get the total number of charged molecules

Definition at line 2967 of file environment.cpp.

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

Definition at line 194 of file environment.h.

11.3.3.99 void environment::incCleavageCounter ( ) [inline]

Definition at line 281 of file environment.h.

11.3.3.100 void environment::incCondensationCounter ( ) [inline]

Definition at line 283 of file environment.h.

11.3.3.101 void environment::incEndoCleavageCounter ( ) [inline]

Definition at line 282 of file environment.h.

11.3.3.102 void environment::incEndoCondensationCounter ( ) [inline]

Definition at line 284 of file environment.h.

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

Definition at line 276 of file environment.h.

11.3.3.104 void environment::incNumberOfCpx ( ) [inline]

Definition at line 264 of file environment.h.

11.3.3.105 void environment::incNumberOfCpxMols ( ) [inline]

Definition at line 266 of file environment.h.

11.3.3.106 void environment::incNumberOfMols ( ) [inline]

Definition at line 262 of file environment.h.

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

Definition at line 271 of file environment.h.

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

Definition at line 269 of file environment.h.

11.3.3.109 void environment::incNumberOfSpecies ( ) [inline]

Definition at line 260 of file environment.h.

11.3.3.110 void environment::incOverallLoadedMolsCounter ( ) [inline]

Definition at line 285 of file environment.h.

11.3.3.111 void environment::increaseAttempts ( ) [inline]

Definition at line 313 of file environment.h.

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

Definition at line 277 of file environment.h.

11.3.3.113 bool environment::newSpeciesEvaluation (string *tmpNewSpecies*, MTRand & *tmp___RndDoubleGen*)

Evaluate new species

Version

1.0

Parameters

string	tmpNewSpecies New species sequence to evaluate
MTRand&	tmp___RndDoubleGen random number generator

Definition at line 4878 of file environment.cpp.

11.3.3.114 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 2030 of file environment.cpp.

11.3.3.115 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 3033 of file environment.cpp.

11.3.3.116 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
tmpId-Reaction	Rections ID
tmpId-Catalysis	Catalysis ID
MTRand&	tmp__RndDoubleGen random number generator

Definition at line 4438 of file environment.cpp.

11.3.3.117 bool environment::perform_endo_ComplexFormation (acs_longInt *tmpCatalyst,* acs_longInt *tmpSubstrate,* 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 4643 of file environment.cpp.

11.3.3.118 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 4199 of file environment.cpp.

11.3.3.119 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 4335 of file environment.cpp.

11.3.3.120 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
MTRand&	tmp__RndDoubleGen random generator

Definition at line 4736 of file environment.cpp.

11.3.3.121 bool environment::performComplexFormation (acs_longInt *tmpCatalyst,* acs_longInt *tmpSubstrate,* 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 4559 of file environment.cpp.

11.3.3.122 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 4083 of file environment.cpp.

11.3.3.123 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

tmpTime-Interval	time elapsed since the last reaction
MTrand	&tmp_RndDoubleGen random number generator

Definition at line 3157 of file environment.cpp.

11.3.3.124 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 3832 of file environment.cpp.

11.3.3.125 bool environment::performGillespieComputation (MTRand & *tmpRndDoubleGen,* QTime & *tmpTimeElapsed,* acs_int *tmpActGEN,* acs_int *tmpActSIM,* acs_int *tmpActSTEP,* QString *tmpStoringPath*)

Perform all the gillespie algorithm procedure

Version

1.5

Date

2011.02.12

Parameters

tmpRnd-DoubleGen	random numbers generator
tmpTime-Elapsed	Computation time elapsed from the T0
tmpActGEN	actual generation
tmpActSIM	actual simulation
tmpActSTEP	actual step (reaction)
tmpStoring-Path	path where results are stored

Definition at line 2120 of file environment.cpp.

11.3.3.126 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 3752 of file environment.cpp.

11.3.3.127 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

tmpTime-Interval	time elapsed since the last reaction
MTrand	&tmp_RndDoubleGen random number generator

Definition at line 3589 of file environment.cpp.

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

Perform the reaction after the Gillespie computation

Version

1.0

Parameters

acs_longInt	reaction_u reaction ID in Gillespie structure
MTRand&	tmp_RndDoubleGen random generator

Definition at line 3884 of file environment.cpp.

11.3.3.129 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 3066 of file environment.cpp.

11.3.3.130 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 2763 of file environment.cpp.

11.3.3.131 void environment::printAllChargeMols ( )

Print all the energized molecules 1.0

Date

2010-11-14

Definition at line 5425 of file environment.cpp.

11.3.3.132 void environment::printAllSpeciesIdAndSequence ( )

Show all the species with their ID

Version

1.0

Definition at line 5379 of file environment.cpp.

11.3.3.133 void environment::printGillespieStructure ( )

Show all the Gillespie Structure

Version

1.0

Definition at line 5399 of file environment.cpp.

11.3.3.134 void environment::printInitialCondition ( )

Show all initial species in table format

Version

1.0

Definition at line 5322 of file environment.cpp.

11.3.3.135 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 3008 of file environment.cpp.

11.3.3.136 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 3485 of file environment.cpp.

11.3.3.137 void environment::resetCleavageCounter ( ) [inline]

Definition at line 288 of file environment.h.

11.3.3.138 void environment::resetConcentrationToInitialConditions ( )

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

Version

1.0

Definition at line 5515 of file environment.cpp.

11.3.3.139 void environment::resetCondensationCounter ( ) [inline]

Definition at line 290 of file environment.h.

11.3.3.140 void environment::resetEndoCleavageCounter ( ) [inline]

Definition at line 289 of file environment.h.

11.3.3.141 void environment::resetEndoCondensationCounter ( ) [inline]

Definition at line 291 of file environment.h.

11.3.3.142 void environment::resetOverallLoadedMolsCounter ( ) [inline]

Definition at line 292 of file environment.h.

11.3.3.143 void environment::resetReactionsCounter ( ) [inline]

Definition at line 294 of file environment.h.

11.3.3.144 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 1496 of file environment.cpp.

11.3.3.145 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

tmpNewSequence	New sequence to evaluate
MTRand&	tmp_RndDoubleGen random number generator
tmpNewSpeciesLength	Lenght of the species

Definition at line 1466 of file environment.cpp.

11.3.3.146 bool environment::saveCatalysisStructure (acs_int *tmpCurrentGen,* acs_int *tmpCurrentSim,* acs_int *tmpCurrentStep,* QString *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. The file is saved in the directory indicated as a second parameter in the run command

Version

1.0

Definition at line 5976 of file environment.cpp.

11.3.3.147 bool environment::saveConfigurationFile (QString *tmpStoringPath*)

Save a file with the configuration parameters

Version

1.0

Definition at line 5652 of file environment.cpp.

11.3.3.148 bool environment::saveInfluxStructure (QString *tmpStoringPath*)

Save influx structures in a file named

Parameters

bool	saveInfluxStructure(QString tmpStoringPath);

Version

1.0

Date

2010-04-04

Definition at line 5816 of file environment.cpp.

11.3.3.149 bool environment::saveLivingSpeciesAmount (acs_int *tmp__CurrentGen,* acs_int *tmp__CurrentSim,* QString *tmp__StoringPath*)

Save living species total AMOUNT in a file named livingAmount_[CurrentGen]_[current-Sim].csv. The file is saved in the directory indicated as a second parameter in the run

command

Version

1.0

Definition at line 6196 of file environment.cpp.

11.3.3.150 bool environment::saveLivingSpeciesConcentration (acs_int *tmp__CurrentGen,* acs_int *tmp__CurrentSim,* QString *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

Definition at line 6255 of file environment.cpp.

11.3.3.151 bool environment::saveLivingSpeciesID (acs_int *tmp__CurrentGen,* acs_int *tmp__CurrentSim,* acs_int *tmp__CurrentStep,* QString *tmp__StoringPath*)

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

Version

1.0

Definition at line 6144 of file environment.cpp.

11.3.3.152 bool environment::saveNrgBoolFncStructure (QString *tmpStoringPath*)

Save Energetic Boolean Function on a file named _acsnrgbooleanfunctions.csv

Version

1.0

Date

2011-04-15

Parameters

QString	tmpStoringPath Path of the saving folder

Definition at line 5847 of file environment.cpp.

11.3.3.153 bool environment::saveReactionsParameters (acs_int *tmp__CurrentGen,* acs_int *tmp__CurrentSim,* acs_int *tmp__CurrentStep,* QString *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

Definition at line 6097 of file environment.cpp.

11.3.3.154 bool environment::saveReactionsStructure (acs_int *tmpCurrentGen,* acs_int *tmpCurrentSim,* acs_int *tmpCurrentStep,* QString *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 5932 of file environment.cpp.

11.3.3.155 bool environment::saveSpeciesStructure (acs_int *tmpCurrentGen,* acs_int *tmpCurrentSim,* acs_int *tmpCurrentStep,* QString *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. The file is saved in the directory indicated as a second parameter in the run command

Version

1.0

Definition at line 5878 of file environment.cpp.

11.3.3.156 bool environment::saveTimes (acs_int *tmpCurrentGen,* acs_int *tmpCurrentSim,* acs_int *tmpCurrentStep,* QString *tmpStoringPath*)

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

Version

1.0

Definition at line 6021 of file environment.cpp.

11.3.3.157 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 906 of file environment.cpp.

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

Definition at line 310 of file environment.h.

11.3.3.159 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 3396 of file environment.cpp.

11.3.3.160 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 3437 of file environment.cpp.

11.3.3.161 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
tmpNew-Species-Length	Lenght of the species

Definition at line 1432 of file environment.cpp.

11.3.3.162 void environment::showGlobalParameter ( )

Shows all parameters uploaded from the configuration file

Version

1.0

Definition at line 5239 of file environment.cpp.

11.3.3.163 void environment::storeInitialStructures ( )

Store initial structures into storing variables

Version

1.0

Definition at line 5609 of file environment.cpp.

11.3.3.164 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 1596 of file environment.cpp.

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

Definition at line 279 of file environment.h.

11.3.3.166 void environment::updateSpeciesAges ( )

Update the species age

Version

1.0

Definition at line 5200 of file environment.cpp.

11.3.3.167 QString environment::zeroBeforeStringNumber (acs_int *tmpTotN,* acs_int *tmpCurrentN*)

This function creates a chain of zero as QString 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 5632 of file environment.cpp.

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

- /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness_simulator/environment.h
- /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness_simulator/environment.cpp

11.4 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

11.4.1 Detailed Description

Author

Alessandro Filisetti

Version

0.1

Date

2009-04-21

Definition at line 12 of file gillespie.h.

11.4.2 Constructor & Destructor Documentation

11.4.2.1 gillespie::gillespie ( )

11.4.2.2 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

tmpRnd-DoubleGen	randomGenerator reference

Definition at line 21 of file gillespie.cpp.

11.4.2.3 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.

11.4.2.4 gillespie::~gillespie ( ) [inline]

Definition at line 33 of file gillespie.h.

11.4.3 Member Function Documentation

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

Definition at line 44 of file gillespie.h.

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

Definition at line 36 of file gillespie.h.

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

Definition at line 37 of file gillespie.h.

11.4.3.4 acs_longInt gillespie::getMoII ( ) const [inline]

Definition at line 39 of file gillespie.h.

11.4.3.5 acs_longInt gillespie::getMoIII ( ) const [inline]

Definition at line 40 of file gillespie.h.

11.4.3.6 acs_longInt gillespie::getMoIIII ( ) const [inline]

Definition at line 41 of file gillespie.h.

11.4.3.7 acs_longInt gillespie::getMoIIV ( ) const [inline]

Definition at line 42 of file gillespie.h.

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

Definition at line 45 of file gillespie.h.

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

Definition at line 43 of file gillespie.h.

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

Definition at line 38 of file gillespie.h.

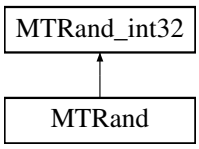
The documentation for this class was generated from the following files:

- /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness_simulator/gillespie.h
- /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness_simulator/gillespie.cpp

11.5 MTRand Class Reference

`#include <mtrand.h>`

Inheritance diagram for MTRand:



Public Member Functions

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

11.5.1 Detailed Description

Definition at line 97 of file mtrand.h.

11.5.2 Constructor & Destructor Documentation

11.5.2.1 MTRand::MTRand ( ) [inline]

Definition at line 99 of file mtrand.h.

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

Definition at line 100 of file mtrand.h.

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

Definition at line 101 of file mtrand.h.

11.5.2.4 MTRand::~MTRand ( ) [inline]

Definition at line 102 of file mtrand.h.

11.5.3 Member Function Documentation

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

Reimplemented from MTRand_int32.

Definition at line 103 of file mtrand.h.

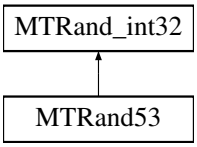
The documentation for this class was generated from the following file:

- /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness_simulator/mtrand.h

11.6 MTRand53 Class Reference

`#include <mtrand.h>`

Inheritance diagram for MTRand53:



Public Member Functions

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

11.6.1 Detailed Description

Definition at line 139 of file mtrand.h.

11.6.2 Constructor & Destructor Documentation

11.6.2.1 MTRand53::MTRand53 ( ) [inline]

Definition at line 141 of file mtrand.h.

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

Definition at line 142 of file mtrand.h.

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

Definition at line 143 of file mtrand.h.

11.6.2.4 MTRand53::~MTRand53 ( ) [inline]

Definition at line 144 of file mtrand.h.

11.6.3 Member Function Documentation

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

Reimplemented from MTRand_int32.

Definition at line 145 of file mtrand.h.

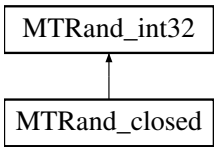
The documentation for this class was generated from the following file:

- /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness_simulator/mtrand.h

11.7 MTRand_closed Class Reference

`#include <mtrand.h>`

Inheritance 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() ()

11.7.1 Detailed Description

Definition at line 111 of file mtrand.h.

11.7.2 Constructor & Destructor Documentation

11.7.2.1 MTRand_closed::MTRand_closed ( ) [inline]

Definition at line 113 of file mtrand.h.

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

Definition at line 114 of file mtrand.h.

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

Definition at line 115 of file mtrand.h.

11.7.2.4 MTRand_closed::~MTRand_closed ( ) [inline]

Definition at line 116 of file mtrand.h.

11.7.3 Member Function Documentation

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

Reimplemented from MTRand_int32.

Definition at line 117 of file mtrand.h.

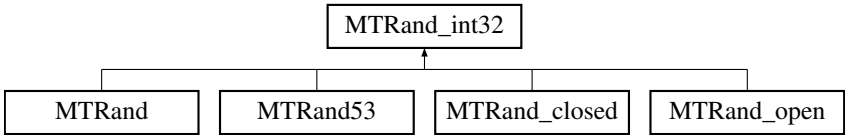
The documentation for this class was generated from the following file:

- /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness_simulator/mtrand.h

11.8 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 ()

11.8.1 Detailed Description

Definition at line 48 of file mtrand.h.

11.8.2 Constructor & Destructor Documentation

11.8.2.1 MTRand_int32::MTRand_int32 ( ) [inline]

Definition at line 51 of file mtrand.h.

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

Definition at line 53 of file mtrand.h.

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

Definition at line 55 of file mtrand.h.

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

Definition at line 62 of file mtrand.h.

11.8.3 Member Function Documentation

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

Reimplemented in MTRand53, MTRand_open, MTRand_closed, and MTRand.

Definition at line 60 of file mtrand.h.

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

Definition at line 85 of file mtrand.h.

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

Definition at line 23 of file mtrand.cpp.

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

Definition at line 35 of file mtrand.cpp.

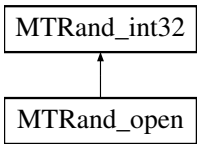
The documentation for this class was generated from the following files:

- /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness_simulator/mtrand.h
- /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness_simulator/mtrand.cpp

11.9 MTRand_open Class Reference

`#include <mtrand.h>`

Inheritance 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() ()

11.9.1 Detailed Description

Definition at line 125 of file mtrand.h.

11.9.2 Constructor & Destructor Documentation

11.9.2.1 MTRand_open::MTRand_open ( ) [inline]

Definition at line 127 of file mtrand.h.

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

Definition at line 128 of file mtrand.h.

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

Definition at line 129 of file mtrand.h.

11.9.2.4 MTRand_open::~MTRand_open ( ) [inline]

Definition at line 130 of file mtrand.h.

11.9.3 Member Function Documentation

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

Reimplemented from MTRand_int32.

Definition at line 131 of file mtrand.h.

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

- /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness_simulator/mtrand.h

11.10 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)

  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
- void updateTotEvents ()
- void resetEventsCounter ()

11.10.1 Detailed Description

Definition at line 17 of file reactions.h.

11.10.2 Constructor & Destructor Documentation

11.10.2.1 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*)

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.

11.10.2.2 reactions::~reactions ( ) [inline]

Definition at line 32 of file reactions.h.

11.10.3 Member Function Documentation

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

Definition at line 41 of file reactions.h.

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

Definition at line 40 of file reactions.h.

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

Definition at line 35 of file reactions.h.

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

Definition at line 37 of file reactions.h.

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

Definition at line 38 of file reactions.h.

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

Definition at line 39 of file reactions.h.

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

Definition at line 36 of file reactions.h.

11.10.3.8 void reactions::resetEventsCounter ( ) [inline]

Definition at line 45 of file reactions.h.

11.10.3.9 void reactions::updateTotEvents ( ) [inline]

Definition at line 44 of file reactions.h.

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

- /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness_simulator/reactions.h
- /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness_simulator/reactions.cpp

11.11 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
- 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)

11.11.1 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.

11.11.2 Constructor & Destructor Documentation

11.11.2.1 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.

11.11.2.2 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
tmpComplexDegEnh	complex dissociation kinetic constant
tmpComplexCuttingPoint	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 51 of file species.cpp.

11.11.2.3 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)
tmpConcentration	species initial concentration
tmpDiffusionEnh	Diffusion enhancement degree
tmpSoluble	1 if the species is soluble, 0 otherwise
tmpComplexDegEnh	complex dissociation kinetic constant
tmpComplexCuttingPoint	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 96 of file species.cpp.

11.11.2.4 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 134 of file species.cpp.

11.11.2.5 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
tmp-Sequence	species sequence (e.g. ABABAABABA)
tmpAmount	species initial amount of molecules
tmp-Reactions_-constant	????

Definition at line 175 of file species.cpp.

11.11.2.6 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
tmp-Sequence	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 222 of file species.cpp.

11.11.2.7 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 267 of file species.cpp.

11.11.2.8 species::~species ( ) [inline]

Definition at line 78 of file species.h.

11.11.3 Member Function Documentation

11.11.3.1 bool species::chargeMol ( )

to charge molecules

Definition at line 342 of file species.cpp.

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

Definition at line 123 of file species.h.

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

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

Definition at line 305 of file species.cpp.

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

Definition at line 89 of file species.h.

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

Definition at line 84 of file species.h.

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

Definition at line 96 of file species.h.

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

Definition at line 86 of file species.h.

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

Definition at line 94 of file species.h.

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

Definition at line 93 of file species.h.

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

Definition at line 87 of file species.h.

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

Definition at line 100 of file species.h.

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

Definition at line 91 of file species.h.

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

Definition at line 99 of file species.h.

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

Definition at line 95 of file species.h.

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

Definition at line 101 of file species.h.

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

Definition at line 81 of file species.h.

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

Definition at line 98 of file species.h.

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

return the concentration of the loaded molecules

Definition at line 369 of file species.cpp.

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

Definition at line 85 of file species.h.

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

Definition at line 90 of file species.h.

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

Definition at line 82 of file species.h.

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

Definition at line 83 of file species.h.

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

Definition at line 92 of file species.h.

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

Definition at line 97 of file species.h.

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

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

Definition at line 294 of file species.cpp.

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

Definition at line 124 of file species.h.

11.11.3.27 void species::rebornsIncrement ( ) [inline]

Definition at line 121 of file species.h.

11.11.3.28 void species::resetAge ( ) [inline]

Definition at line 126 of file species.h.

11.11.3.29 void species::resetReborns ( ) [inline]

Definition at line 127 of file species.h.

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

Definition at line 128 of file species.h.

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

Definition at line 107 of file species.h.

11.11.3.32 bool species::setChargeMols (acs_int *tmpMolsToCharge*)

to charge a specific number of molecules

Definition at line 329 of file species.cpp.

11.11.3.33 void species::setConcentration (acs_double *tmpConc,* acs_double *tmpVolume*) [inline]

Definition at line 108 of file species.h.

11.11.3.34 void species::setDiffusion (acs_double *tmpDiff*) [inline]

Definition at line 117 of file species.h.

11.11.3.35 void species::setEvaluated ( ) [inline]

Definition at line 116 of file species.h.

11.11.3.36 void species::setKphospho (acs_double *tmpKphospho*) [inline]

Definition at line 119 of file species.h.

11.11.3.37 void species::setNewAge (acs_double *tmpLastTimeInterval*) [inline]

Definition at line 120 of file species.h.

11.11.3.38 void species::setSolubility (acs_int *tmpSol*) [inline]

Definition at line 118 of file species.h.

11.11.3.39 bool species::setSpecificChargeMols (acs_int *tmpMolsToCharge*)

to charge a specific number of molecules

Definition at line 316 of file species.cpp.

11.11.3.40 void species::specificIncrement (acs_int *tmpIncrement,* acs_double *tmpVolume*) [inline]

Definition at line 106 of file species.h.

11.11.3.41 bool species::unchargeMol ( )

to uncharge molecules

Definition at line 357 of file species.cpp.

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

- /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness_simulator/species.h
- /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness_simulator/species.cpp

Chapter 12

File Documentation

12.1 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness-_simulator/_analysis/allTimesAnalysis.m File Reference

Functions

- params do not prompt figures (0)

Variables

- function [out]
- params figureVisible = prompt figures (1)
- params deltaT = Delta T
- params totT = total time of the simulation
- OUT currentDir = cd()

12.1.1 Function Documentation

12.1.1.1 params do not prompt figures (0)

12.1.2 Variable Documentation

12.1.2.1 OUT currentDir = cd()

Definition at line 18 of file allTimesAnalysis.m.

12.1.2.2 params deltaT = Delta T

Definition at line 13 of file allTimesAnalysis.m.

12.1.2.3 params figureVisible = prompt figures (1)

Definition at line 12 of file allTimesAnalysis.m.

12.1.2.4 function[out]

Initial value:

```
 allTimesAnalysis(params)
% function [out] = allTimesAnalysis(params)
%
% 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
%    params.tmpPath = simulations path
```

Definition at line 1 of file allTimesAnalysis.m.

12.1.2.5 params totT = total time of the simulation

Definition at line 14 of file allTimesAnalysis.m.

12.2 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness-_simulator/_analysis/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)
- Species sequence overallConcMatrix (j, indice)
- overallConcMatrix (end,:)
- if exist ('0_concentrations_ALL.csv','file') delete('0_concentrations_ALL.csv')
- fclose (fidC)
- fclose (fid)
- :,(and(evaluated >0, bindpnt==0 overallConcMatrix ()
- grid on set (gca,'fontsize', 15,'fontname','times')
- xlabel ('Time', 'Interpreter', 'latex', 'fontsize', 15)
- ylabel ('Concentration', 'Interpreter', 'latex', 'fontsize', 15)

- id gcf ()
- saveas (figure1, '0_conc.fig')
- compute corrcoef of species file (NO complexes NO notEvaluated) cct
- '0_concentrations_EVAL.csv' delete ()
- if exist ('0_concentrations_EVALNOCPX.csv','file') delete('0_concentrations_EVALNOCPX.csv')
- cd ('./../..')
- end end cd (currentDir)

Variables

- function [out]
- images are shown This function performs statistics on each single simulation
- images are shown This function performs statistics on each single on each single species file currentDir = cd()
- if nargin< 1params.path=currentDir();params.deltaT=10;params.totT=1000;params.-showFig=0;endcd(params.path);disp('start analysis...')%READ ALL THE DIRECTORY CONTAINING SIMULATIONSsimDirs=dir('*sim *');%File Containing all Timestimes=0:params.deltaT:params.totT;%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'));speciesFiles=dir('*species_ *');%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 = size(overallConcMatrix)
- 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
- fixed Concentration indice = indice+1
- else per ora ho memorizzato solo nome specie e concentrazione
- end fidC = fopen('0_concentrations_ALL.csv','a')
- for a
- pause
- for k
- overallConcMatrixEval = overallConcMatrix(:,evaluated>0)
- if params showFig
- clear overallConcMatrix
- clear overallConcMatrixEvalNoComplex
- clear specie
- clear bindpnt
- clear evaluated
- out = 0

12.2.1 Function Documentation

12.2.1.1 Species Age (in *seconds*) = fscanf(fid,'%d',1)

12.2.1.2 Dissociation Kinetic Constant bindpnt (indice)

12.2.1.3 cd ('./../..')

12.2.1.4 end end cd (currentDir)

12.2.1.5 Species sequence concentrazione (indice)

12.2.1.6 '0_concentrations_EVAL.csv' delete ( ) [virtual]

12.2.1.7 Binding point evaluated (indice)

12.2.1.8 if exist ('0_concentrations_ALL.csv' , 'file')

12.2.1.9 if exist ('0_concentrations_EVALNOCPX.csv' , 'file')

12.2.1.10 fclose (fidC)

12.2.1.11 fclose (fid)

12.2.1.12 compute corrcoef of species file (NO complexes NO *notEvaluated*)

12.2.1.13 fid s fscanf ( ) [virtual]

12.2.1.14 id gcf ( ) [virtual]

12.2.1.15 ID ho raggiunto la fine del file if isempty (itmp)

12.2.1.16 Species sequence overallConcMatrix (j , indice)

12.2.1.17 overallConcMatrix (end , :)

12.2.1.18 :,(and(evaluated>0,bindpnt==0 overallConcMatrix ( ) [virtual]

12.2.1.19 saveas (figure1 , '0_conc.fig')

12.2.1.20 grid on set (gca , 'fontsize' , 15 , 'fontname' , 'times')

12.2.1.21 xlabel ('Time' , 'Interpreter' , 'latex' , 'fontsize' , 15)

12.2.1.22 ylabel ('Concentration' , 'Interpreter' , 'latex' , 'fontsize' , 15)

12.2.2 Variable Documentation

12.2.2.1 ID __pad0__

Definition at line 59 of file concAnalysis.m.

12.2.2.2 for a

Initial value:

```
1:indice-1
                    fprintf(fidC,'%s \t',specie(a).nome)
```

Definition at line 111 of file concAnalysis.m.

12.2.2.3 clear bindpnt

Definition at line 200 of file concAnalysis.m.

12.2.2.4 clear concentrazione

Definition at line 84 of file concAnalysis.m.

12.2.2.5 Phosphorilation Kinetic constant

Definition at line 80 of file concAnalysis.m.

12.2.2.6 images are shown This function performs statistics on each single on each single species file currentDir = cd()

Definition at line 13 of file concAnalysis.m.

12.2.2.7 end end = size(overallConcMatrix)

Definition at line 63 of file concAnalysis.m.

12.2.2.8 clear evaluated

Definition at line 201 of file concAnalysis.m.

12.2.2.9 Phosphorilation Kinetic f

Definition at line 80 of file concAnalysis.m.

12.2.2.10 end fidC = fopen('0_concentrations_ALL.csv','a')

Definition at line 110 of file concAnalysis.m.

12.2.2.11 Phosphorilation Kinetic Charged Molecules Concentration ftmp = fscanf(fid,'%f',1)

Definition at line 72 of file concAnalysis.m.

12.2.2.12 function[out]

Initial value:

```
 concAnalysis(params)
% function [out] = concAnalysis(params)
%
% INPUT
% params.path = path of the simulations root
% params.deltaT = delta T between one file and the next one
% params.totT = total seconds of the simulation
% params.showFig = 0 images are not shown
```

Definition at line 1 of file concAnalysis.m.

12.2.2.13 fixed Concentration indice = indice+1

Definition at line 82 of file concAnalysis.m.

12.2.2.14 Catalyst ID itmp = fscanf(fid,'%d',1)

Definition at line 70 of file concAnalysis.m.

12.2.2.15 end end case favorire quelle corte con una scale free di esponente gamma k

Initial value:

```
1 : ss
            fprintf(fidC,' \n')
```

Definition at line 126 of file concAnalysis.m.

12.2.2.16 if nargin< 1params.path=currentDir();params.deltaT=10;params.totT=1000;params.-showFig=0;endcd(params.path);disp('start analysis...')%READ ALL THE DIRECTORY CONTAINING SIMULATIONSsimDirs=dir('∗sim ∗');%File Containing all Timestimes=0:params.deltaT:params.totT;%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'));species-Files=dir('∗species_∗');%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 >

Initial value:

```
1

                itmp=fscanf(fid,'%d',1)
```

Definition at line 55 of file concAnalysis.m.

12.2.2.17 if nFile

Initial value:

```
= 1
                    % per ora ho memorizzato solo nome specie e concentrazione
```

Definition at line 65 of file concAnalysis.m.

12.2.2.18 end end out = 0

Definition at line 210 of file concAnalysis.m.

12.2.2.19 clear overallConcMatrix

Definition at line 195 of file concAnalysis.m.

12.2.2.20 clear overallConcMatrixEval = overallConcMatrix(:,evaluated>0)

Definition at line 134 of file concAnalysis.m.

12.2.2.21 clear overallConcMatrixEvalNoComplex

Definition at line 197 of file concAnalysis.m.

12.2.2.22 pause

Definition at line 116 of file concAnalysis.m.

12.2.2.23 params showFig

Initial value:

```
= 0
            figure('Visible','off')
        else
            figure(1)
        end
        plot(times', overallConcMatrixEvalNoComplex, '-')
```

Definition at line 139 of file concAnalysis.m.

12.2.2.24 images are shown This function performs statistics on each single simulation

Definition at line 1 of file concAnalysis.m.

12.2.2.25 clear specie

Definition at line 198 of file concAnalysis.m.

12.3 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness-_simulator/_analysis/filisettiLibrary.py File Reference

Namespaces

- namespace filisettiLibrary

Functions

- def filisettiLibrary.PlotMatrix
- def filisettiLibrary.PlotMatrixLOGY
- def filisettiLibrary.PlotMatrixSingleSpeciesAmounts
- def filisettiLibrary.PlotMatrixML
- def filisettiLibrary.PlotMatrixErrorBar
- def filisettiLibrary.PlotMatrix3D
- def filisettiLibrary.PlotIMSHOWoverThreshold
- def filisettiLibrary.PlotIMSHOW
- def filisettiLibrary.zeroBeforeStrNum
- def filisettiLibrary.writeOverallStatOnFile
- def filisettiLibrary.writeOverallStatOnFileWhereISay

Variables

- int filisettiLibrary.width = 8
- int filisettiLibrary.height = 6

12.4 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness-_simulator/_analysis/garbageSearch.m File Reference

Functions

- fclose (fid)
- 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)
- printing results cd (currentDir)
- fprintf (fid,'NET%d\n\n', x)
- fprintf (fid,'%f\n\n', garbageMatrix(i, 2))

Variables

- currentDir = cd()
- simDirs = dir('sim_*')
- fid = fopen('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 = size(reactionsMatrix) 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

12.4.1 Function Documentation

12.4.1.1 end catalysisMatrix (continua , 1:4)

12.4.1.2 printing results cd (currentDir)

12.4.1.3 cleavagesMatrix (cleavages , 1:6)

12.4.1.4 condensationsMatrix (condensations , 1:6)

12.4.1.5 fclose (fid)

12.4.1.6 fprintf (fid , 'NET%d\n\n' , x)

12.4.1.7 fprintf (fid , '%f\n\n' , garbageMatrix(i, 2))

12.4.1.8 garbageMatrix (gRows , 1:2)

12.4.1.9 break end reactionsMatrix (continua , 1:6)

12.4.1.10 speciesMatrix (continua , 2)

12.4.1.11 if conc&& sValues (5)

12.4.2 Variable Documentation

12.4.2.1 break

Definition at line 78 of file garbageSearch.m.

12.4.2.2 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 116 of file garbageSearch.m.

12.4.2.3 catalysisFile = cFiles(length(cFiles)).name

Definition at line 69 of file garbageSearch.m.

12.4.2.4 catalysisMatrix = 0

Definition at line 72 of file garbageSearch.m.

12.4.2.5 creating catalysis matrix cFiles = dir('catalysis∗')

Definition at line 68 of file garbageSearch.m.

12.4.2.6 if check1

Initial value:

```
= 0 && check2 == 0 && check3 == 0
            gRows = gRows+1
```

Definition at line 93 of file garbageSearch.m.

12.4.2.7 check2 = ismember(speciesMatrix(i,1),condensationsMatrix(:,4:5))

Definition at line 91 of file garbageSearch.m.

12.4.2.8 check3 = ismember(speciesMatrix(i,1),catalysisMatrix(:,2))

Definition at line 92 of file garbageSearch.m.

12.4.2.9 cleavages = 0

Definition at line 52 of file garbageSearch.m.

12.4.2.10 cleavagesMatrix = size(reactionsMatrix) 0

Definition at line 54 of file garbageSearch.m.

12.4.2.11 conc = fscanf(fid,'%f',1)

Definition at line 22 of file garbageSearch.m.

12.4.2.12 else condensations = 0

Definition at line 51 of file garbageSearch.m.

12.4.2.13 condensationsMatrix = 0

Definition at line 55 of file garbageSearch.m.

12.4.2.14 continua = 1

Definition at line 12 of file garbageSearch.m.

12.4.2.15 currentDir = cd()

Definition at line 1 of file garbageSearch.m.

12.4.2.16 while continua cValues = fscanf(fid,'%f',7)

Definition at line 74 of file garbageSearch.m.

12.4.2.17 else[r c] = size(garbageMatrix)

Definition at line 107 of file garbageSearch.m.

12.4.2.18 fid = fopen('garbage.txt','w')

Definition at line 3 of file garbageSearch.m.

12.4.2.19 identifying garbage[rsm csm] = size(speciesMatrix)

Definition at line 86 of file garbageSearch.m.

12.4.2.20 if garbageMatrix = 0

Definition at line 88 of file garbageSearch.m.

12.4.2.21 gRows = 0

Definition at line 87 of file garbageSearch.m.

12.4.2.22 end

****
S T A R T **********for i

Initial value:

```
1:rrm
       if reactionsMatrix(i,2) == 1
           cleavages = cleavages+1
```

Definition at line 56 of file garbageSearch.m.

12.4.2.23 else index = fscanf(fid,'%d',1)

Definition at line 15 of file garbageSearch.m.

12.4.2.24 break end name = fscanf(fid,'%s',1)

Definition at line 21 of file garbageSearch.m.

12.4.2.25 clear r

Definition at line 115 of file garbageSearch.m.

12.4.2.26 reactionsFile = rFiles(length(rFiles)-1).name

Definition at line 34 of file garbageSearch.m.

12.4.2.27 reactionsMatrix = 0

Definition at line 37 of file garbageSearch.m.

12.4.2.28 creating reactions matrix rFiles = dir('reactions*')

Definition at line 33 of file garbageSearch.m.

12.4.2.29 while continua rValues = fscanf(fid,'%d',7)

Definition at line 39 of file garbageSearch.m.

12.4.2.30 creating species matrix sFiles = dir('species*')

Definition at line 9 of file garbageSearch.m.

12.4.2.31 simDirs = dir('sim_*')

Definition at line 2 of file garbageSearch.m.

12.4.2.32 speciesFile = sFiles(length(sFiles)).name

Definition at line 10 of file garbageSearch.m.

12.4.2.33 speciesMatrix = 0

Definition at line 13 of file garbageSearch.m.

12.4.2.34 if stop = isempty(index)

Definition at line 16 of file garbageSearch.m.

12.4.2.35 sValues = fscanf(fid,'%f',12)

Definition at line 23 of file garbageSearch.m.

12.4.2.36 for x

Initial value:

```
1:length(simDirs)
   cd(strcat(simDirs(x).name,'/res'))
```

Definition at line 5 of file garbageSearch.m.

12.5 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness-_simulator/_analysis/generalConcentrationOverThreshold.m File Reference

Functions

- cd (params.tmpPath)
- disp ('start analysis...')%CREATE HEADER ROW IN THE FINAL OUTCOMES FILES outFilenName
- 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 '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,:)
- saveGraphSUBToFile (folderSub, reaction, rtime, confparams.nReactions, graph-SUB, 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 ∗analysisTime-IntervalNoSave),(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
- params params Otherwise if the reaction is already present its parameters are updated if sum (and((graph(:, 1)==cat), graph(:, 2)==mol_I))
- graph (position,:)
- :, 2 mol_I ()
- graphSUB (position,:)
- params params Otherwise if the reaction is already present its parameters are updated if sum (and((graph(:, 1)==cat), graph(:, 2)==mol_II))
- :, 2 mol_II ()
- end fclose (fid)
- :, 2 k ()
- grid on set (gca,'fontsize', 15,'fontname','times')
- xlabel ('Time', 'Interpreter', 'latex', 'fontsize', 15)

- ylabel ('Gillespie Mean', 'Interpreter', 'latex', 'fontsize', 15)
- id gcf ()
- saveas (figure1, '1_gilleMean.fig')
- ylabel ('Gillespie SD', 'Interpreter', 'latex', 'fontsize', 15)
- saveas (figure2, '2_gilleSD.fig')
- ylabel ('Entropy', 'Interpreter', 'latex', 'fontsize', 15)
- saveas (figure3, '3_entropy.fig')
- ylabel ('New species Probability', 'Interpreter', 'latex', 'fontsize', 15)
- saveas (figure4, '4_nsp.fig')
- ylabel ('Flux Molecules Dynamics', 'Interpreter', 'latex', 'fontsize', 15)
- saveas (figure5, '5_fluxEconomy.fig')
- 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_by-SCC))
- 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', autocataly-sis))

- 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%s\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)
- disp (fprintf('\n ANALYSIS of the SIMULATION%s IS FINISHED\n', simDirs(IDF).name))
- cd (currentDir)
- tmpl, tmpL strZero ()
- while ischar (tline)%tline
- if isequal (tline(1:param-1),'nSeconds')%confparams.nSeconds = str2num(tline(param+1-:length(tline)))
- elseif isequal (tline(1:param-1),'reactionProbability')%confparams.reactionProbability = str2num(tline(param+1:length(tline)))
- elseif isequal (tline(1:param-1),'energy')%confparams.energy = str2num(tline(param+1-:length(tline)))
- elseif isequal (tline(1:param-1),'nReactions')%confparams.nReactions = str2num(tline(param+1:length(tline)))
- elseif isequal (tline(1:param-1),'influx_rate')%confparams.influx_rate = str2num(tline(param+1-:length(tline)))
- elseif isequal (tline(1:param-1),'maxLOut')%confparams.maxLOut = str2num(tline(param+1-:length(tline)))
- elseif isequal (tline(1:param-1),'ECConcentration')%confparams.ECConcentration = str2num(tline(param+1:length(tline)))
- end end fclose (fidConf)
- 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
- elseif isequal (tline(1:param-1),'nGEN') confparams.nGEN

- elseif isequal (tline(1:param-1),'nSIM') confparams.nSIM
- elseif isequal (tline(1:param-1),'timeStructuresSavingInterval') confparams.timeStructuresSavingInterval
- elseif isequal (tline(1:param-1),'volume') confparams.volume

Variables

- function out
- params threshold = 0
- params decayTime = 100
- params tmpResFold = 'res'
- params distinctiveSubStr = 'K_cpx5_rete_n_'
- params tmpIDsim = '5'
- params tmpRctFileToLoad = ''
- params tmpRctSUBFileToLoad = ''
- params showFig = 0
- end Set current date and current directory currentDate = date()
- currentDir = cd()
- end fidFINAL = fopen(outFilenName,'w')
- READ ALL THE DIRECTORY CONTAINING SIMULATIONS search = strcat('*',params.distinctiveSubStr,'*')
- 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=confparams.nSeconds./(confparams.nSeconds/params.decayTime);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
- LOAD FIRST SPECIES FILE [tmpID, tmpSeq, tmpConc, tmpDiff, tmpSol, tmpCpxDiss, tmpCpxCut, tmpEval, tmpAge, tmpReb, tmpCatID, tmpSubID, tmpKpho, tmpLoadConc, tmpConcFixed] = textread(speciesFiles(length(speciesFiles)).name,'%d %s %f %f %f %f %d %d %f %d %d %d %f %f %d','headerlines',0)
- Compute species over threshold IDsOverThreshold = tmpID(and(tmpConc>params.threshold,tmpCpxCut==0))
- concVec = tmpConc
- reaction parameters files rcsFiles = dir('*reactions_parameters*')
- filextPre
- 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
- 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)
- end if nrg
- printTemporalMessage = 1
- If the time is righe save iGraph structures to file if(rtime > rctIDshow ∗analysis-TimeInterval) disp(sprintf('|-%s|Reaction%d-tim saveGraphToFile)(folderCat, reaction, rtime, confparams.nReactions, graph, filextPre)
- 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
- 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 = graphconncomp(catSparse) histc(scc,1:max(scc))
- realSccs = self_loops(graph) 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
- nZeros = length(num2str(tmpL)) - length(num2str(tmpl))
- if nZeros for p
- tline = fgets(fidConf)
- param = findstr(tline,'=')
- 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)

12.5.1 Function Documentation

12.5.1.1 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))

12.5.1.2 end cd (params. *tmpPath*)

12.5.1.3 Go into the results folder cd (strcat(simDirs(IDF).name,'/res'))

12.5.1.4 cd (currentDir)

12.5.1.5 cd ('./res')

12.5.1.6 cd (strcat('../../', tmpDir))

12.5.1.7 disp ('start analysis...')

12.5.1.8 For each folder the necessary computations are performed disp ('**********')

12.5.1.9 disp ('∗NEW SIMULATION ANALYSIS PROCESS')

12.5.1.10 end If there are species over theshold network analysis is performed disp ('|-Graph Creation')

12.5.1.11 disp (strcat('|-File ', rcsFiles(rfileID).name, 'processing...'))

12.5.1.12 disp (msg)

12.5.1.13 NET ANALYSIS disp ('|-∗∗∗∗∗∗∗NETWORK ANALYSIS ∗∗∗∗∗∗∗')

12.5.1.14 disp ('|-Strongly connected components analysis... ')

12.5.1.15 disp (sprintf('<> SCC n.%d', c))

12.5.1.16 end end disp (sprintf('|-Number of ACS:%d', realSccs))

12.5.1.17 disp (sprintf('|-Number of ACS(length 1):%d', self))

12.5.1.18 disp (sprintf('|-Species over threshold:%d', length(IDsOverThreshold)))

12.5.1.19 disp (fprintf('\t Within Acs%d->%d#%d-[%d]%d%6.4f', incomingNodes(innode), IDsOverThreshold(idot), weightToDistribute,...IDsOverThreshold(idot), IDsOverThreshold(idot), concVec(idot)))

12.5.1.20 disp (fprintf('\t From Acs%d->%d#%d-[%d]%d%6.4f', incomingNodes(innode), IDsOverThreshold(idot), weightToDistribute,...IDsOverThreshold(idot), IDsOverThreshold(idot), concVec(idot)))

12.5.1.21 end end disp (fprintf('\t\t<> Number of Structural Autocatalytic set of molecules:%d', realSccs))

12.5.1.22 end disp (fprintf('\t\t<>----------')))

12.5.1.23 disp (fprintf('\t\t<> Species over threshold produced by a CHAIN:%d', prod_chain))

12.5.1.24 if realSccs disp (fprintf('\t\t<> Species over threshold produced INTO an ACS:%d', prod_inSCC))

12.5.1.25 disp (fprintf('\t\t<> Species over threshold produced INTO an ACS(weigthed):%d', prod_inSCC_weight))

12.5.1.26 disp (fprintf('\t\t<> Species over threshold produced BY an ACS:%d', prod_bySCC))

12.5.1.27 disp (fprintf('\t\t<> Species over threshold produced BY an ACS(weigthed):%d', prod_bySCC_weight))

12.5.1.28 disp (fprintf('\t\t<> Species over threshold produced by an overlap:%d', prod_overlap))

12.5.1.29 disp (fprintf('\t\t<> Species over threshold produced by an overlap(weigthed):%d', prod_overlap_weight))

12.5.1.30 disp (fprintf('\t\t<> Species over threshold produced by itself:%d', autocatalysis))

12.5.1.31 disp (fprintf('\t\t<> Species over threshold produced by itself(weigthed):%d', self_loop_weight))

12.5.1.32 disp (fprintf('\t\t<> Concentration in ACSs:%6.4f', conc_inSCC))

12.5.1.33 disp (fprintf('\t\t<> Concentration in ACSs leaves:%6.4f', conc_bySCC))

12.5.1.34 disp (fprintf('\t\t<> Concentration in chains:%6.4f', conc_chain))

12.5.1.35 disp (fprintf('\t\t<> Concentration of autocatalyst:%6.4f', conc_selfCat))

12.5.1.36 disp (fprintf('\t\t<> Number of endo condensations:%6.4f', endo_condensation_counter))

12.5.1.37 disp (fprintf('\t\t<> Number of condensations:%6.4f', condensation_counter))

12.5.1.38 disp (fprintf('\t\t<> Number of endo cleavages:%6.4f', endo_cleavage_counter))

12.5.1.39 disp (fprintf('\t\t<> Number of cleavages:%6.4f', cleavage_counter))

12.5.1.40 disp (fprintf('\n ANALYSIS of the SIMULATION%s IS FINISHED\n', simDirs(IDF).name))

12.5.1.41 if exist (outFilenName , 'file')

12.5.1.42 if exist (outFname , 'file')

12.5.1.43 fclose (fidFINAL)

12.5.1.44 end fclose (fid)

12.5.1.45 end end fclose (fidConf)

12.5.1.46 end fclose (fid1)

12.5.1.47 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\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')

12.5.1.48 fprintf (fidFINAL , '\n%s\t%6.4f\t%6.4f\t%6.-4f\t%d\t%s\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)

12.5.1.49 id gcf ( ) [virtual]

12.5.1.50 end gillTimeSeries (rlineID , :)

12.5.1.51 graph (: , 5)

12.5.1.52 graph (position , :)

12.5.1.53 graphSUB (: , 5)

12.5.1.54 graphSUB (position , :)

12.5.1.55 end end if (cc =0)

12.5.1.56 while ischar (rlineb)

12.5.1.57 while ischar (tline)

12.5.1.58 if isequal (tline(1:param-1) , 'nSeconds') =
str2num(tline(param+1:length(tline)))

12.5.1.59 elseif isequal (tline(1:param-1) , 'reactionProbability') =
str2num(tline(param+1:length(tline)))

12.5.1.60 elseif isequal (tline(1:param-1) , 'energy') =
str2num(tline(param+1:length(tline)))

12.5.1.61 elseif isequal (tline(1:param-1) , 'nReactions') =
str2num(tline(param+1:length(tline)))

12.5.1.62 elseif isequal (tline(1:param-1) , 'influx_rate') =
str2num(tline(param+1:length(tline)))

12.5.1.63 elseif isequal (tline(1:param-1) , 'maxLOut') =
str2num(tline(param+1:length(tline)))

12.5.1.64 elseif isequal (tline(1:param-1) , 'ECConcentration') =
str2num(tline(param+1:length(tline)))

12.5.1.65 elseif isequal (tline(1:param-1) , 'nGEN')

12.5.1.66 elseif isequal (tline(1:param-1) , 'nSIM')

12.5.1.67 elseif isequal (tline(1:param-1) , 'timeStructuresSavingInterval')

12.5.1.68 elseif isequal (tline(1:param-1) , 'volume')

12.5.1.69 :,2 k ( ) [virtual]

12.5.1.70 :,2 mol_I ( ) [virtual]

12.5.1.71 :,2 mol_II ( ) [virtual]

12.5.1.72 while rline (end)

12.5.1.73 saveas (figure1 , '1_gilleMean.fig')

12.5.1.74 saveas (figure2 , '2_gilleSD.fig')

12.5.1.75 saveas (figure3 , '3_entropy.fig')

12.5.1.76 saveas (figure4 , '4_nsp.fig')

12.5.1.77 saveas (figure5 , '5_fluxEconomy.fig')

12.5.1.78 saveGraphSUBToFile (folderSub , reaction , rtime , confparams. *nReactions,* graphSUB , filextPre)

12.5.1.79 end function saveGraphSUBToFile (tmpDir , tmpRct , tmprTime , tmpRcts , tmpGraph , tmpFilextPre)

12.5.1.80 end function saveGraphToFile (tmpDir , tmpRct , tmprTime , tmpRcts , tmpGraph , tmpFilextPre)

12.5.1.81 grid on set (gca , 'fontsize' , 15 , 'fontname' , 'times')

12.5.1.82 tmpl, tmpL strZero ( ) [virtual]

12.5.1.83 params params Substrate (If different from *1*)

12.5.1.84 end if sum (influx *=* =mol_l)

Initial value:

```
1
                        fluxEconomy = fluxEconomy - 1
```

12.5.1.85 params params Otherwise if the reaction is already present its parameters are updated if sum (and((graph(:, 1)==cat), graph(:, 2)==mol_l))

12.5.1.86 params params Otherwise if the reaction is already present its parameters are updated if sum (and((graph(:, 1)==cat), graph(:, 2)==mol_ll))

12.5.1.87 if sum (find(scc==c) *=* =IDsOverThreshold (idot))

12.5.1.88 end end end end if self if sum (selfID *=* =idot)

12.5.1.89 :,4 timeInterval ( ) [virtual]

12.5.1.90 xlabel ('Time' , 'Interpreter' , 'latex' , 'fontsize' , 15)

12.5.1.91 ylabel ('Gillespie Mean' , 'Interpreter' , 'latex' , 'fontsize' , 15)

12.5.1.92 ylabel ('Gillespie SD' , 'Interpreter' , 'latex' , 'fontsize' , 15)

12.5.1.93 ylabel ('Entropy' , 'Interpreter' , 'latex' , 'fontsize' , 15)

12.5.1.94 ylabel ('New species Probability' , 'Interpreter' , 'latex' , 'fontsize' , 15)

12.5.1.95 ylabel ('Flux Molecules Dynamics' , 'Interpreter' , 'latex' , 'fontsize' , 15)

12.5.1.96 if ∼isdir (strcat('../../', folderCat))

12.5.1.97 end if ∼isdir (strcat('../../', folderSub))

12.5.1.98 if ∼isempty (IDsOverThreshold)

12.5.1.99 if ∼isempty (incomingNodes)

12.5.2 Variable Documentation

12.5.2.1 else Otherwise it has been produced by an ACS

Definition at line 494 of file generalConcentrationOverThreshold.m.

12.5.2.2 alreadyAdded_ACS

Initial value:

```
= 0 % Add concentration to conc of ACSs
                                                    conc_inSCC = conc_inSCC +
      concVec(idot)
```

Definition at line 478 of file generalConcentrationOverThreshold.m.

12.5.2.3 if alreadyAdded_chain = 0

Definition at line 460 of file generalConcentrationOverThreshold.m.

12.5.2.4 else Otherwise it has been produced by an so it is a first layer leaf if alreadyAdded_leaves = 0

Definition at line 459 of file generalConcentrationOverThreshold.m.

12.5.2.5 autocatalysis = 0

Definition at line 428 of file generalConcentrationOverThreshold.m.

12.5.2.6 if realSccs If there are ACS for c

Initial value:

```
1 : length(numScc)
                if numScc(c) > 1
                    sccID = sccID + 1
```

Definition at line 445 of file generalConcentrationOverThreshold.m.

12.5.2.7 cat = rline(4)

Definition at line 157 of file generalConcentrationOverThreshold.m.

12.5.2.8 catSparse = sparse(from,to,true,max(max(from,to)),max(max(from,to)))

Definition at line 440 of file generalConcentrationOverThreshold.m.

12.5.2.9 params params end end end else if cc = rline(3)

Definition at line 156 of file generalConcentrationOverThreshold.m.

12.5.2.10 else cleavage_counter = 0

Definition at line 128 of file generalConcentrationOverThreshold.m.

12.5.2.11 while cntrl

Definition at line 151 of file generalConcentrationOverThreshold.m.

12.5.2.12 conc_bySCC = 0

Definition at line 436 of file generalConcentrationOverThreshold.m.

12.5.2.13 conc_chain = 0

Definition at line 435 of file generalConcentrationOverThreshold.m.

12.5.2.14 conc_inSCC = 0

Definition at line 434 of file generalConcentrationOverThreshold.m.

12.5.2.15 conc_selfCat = 0

Definition at line 437 of file generalConcentrationOverThreshold.m.

12.5.2.16 concVec = tmpConc

Definition at line 113 of file generalConcentrationOverThreshold.m.

12.5.2.17 condensation_counter = 0

Definition at line 126 of file generalConcentrationOverThreshold.m.

12.5.2.18 end function confparams = readParameters()

Definition at line 68 of file generalConcentrationOverThreshold.m.

12.5.2.19 end Set current date and current directory currentDate = date()

Definition at line 24 of file generalConcentrationOverThreshold.m.

12.5.2.20 currentDir = cd()

Definition at line 25 of file generalConcentrationOverThreshold.m.

12.5.2.21 params params decayTime = 100

Definition at line 14 of file generalConcentrationOverThreshold.m.

12.5.2.22 params distinctiveSubStr = 'K_cpx5_rete_n_'

Definition at line 16 of file generalConcentrationOverThreshold.m.

12.5.2.23 ecc = confparams.ECConcentration

Definition at line 567 of file generalConcentrationOverThreshold.m.

12.5.2.24 endo_cleavage_counter = 0

Definition at line 127 of file generalConcentrationOverThreshold.m.

12.5.2.25 else endo_condensation_counter = endo_condensation_counter + 1

Definition at line 219 of file generalConcentrationOverThreshold.m.

12.5.2.26 fid = fopen(rcsFiles(rfileID).name,'r')

Definition at line 132 of file generalConcentrationOverThreshold.m.

12.5.2.27 end fid1 = fopen(outFname,'a')

Definition at line 649 of file generalConcentrationOverThreshold.m.

12.5.2.28 Come back to the original folder fidFINAL = fopen(outFilenName,'w')

Definition at line 36 of file generalConcentrationOverThreshold.m.

12.5.2.29 LOAD FIRST SPECIES FILE[tmpID, tmpSeq, tmpConc, tmpDiff, tmpSol, tmpCpxDiss, tmpCpxCut, tmpEval, tmpAge, tmpReb, tmpCatID, tmpSubID, tmpKpho, tmpLoadConc, tmpConcFixed] = textread(speciesFiles(length(speciesFiles)).name,'%d %s %f %f %f %f %d %d %f %d %d %d %f %f %d','headerlines',0)

Definition at line 108 of file generalConcentrationOverThreshold.m.

12.5.2.30 fileSpeciesID = 1

Definition at line 104 of file generalConcentrationOverThreshold.m.

12.5.2.31 filextPre

Initial value:

```
strcat('_',zeroBeforeStrNum(IDsimFOLDER, length(simDirs)),int2str(IDsimFOLDER)
    )


      for rfileID=1:length(rcsFiles)
          % read species file

          %Initialize reaction type counter
          endo_condensation_counter = 0
```

Definition at line 118 of file generalConcentrationOverThreshold.m.

12.5.2.32 FOR EACH REACTION fluxEconomy = 0

Definition at line 145 of file generalConcentrationOverThreshold.m.

12.5.2.33 params params end end end end fluxEconomyArray = [ ]

Definition at line 146 of file generalConcentrationOverThreshold.m.

12.5.2.34 if folderCat = strcat('__0_iGraph_CAT_', int2str(params.decayTime))

Definition at line 92 of file generalConcentrationOverThreshold.m.

12.5.2.35 end if folderSub = strcat('__0_iGraph_SUB_', int2str(params.decayTime))

Definition at line 93 of file generalConcentrationOverThreshold.m.

12.5.2.36 Computing actual number od strongly connected components from = graph(:,1)+1

Definition at line 420 of file generalConcentrationOverThreshold.m.

12.5.2.37 end function[N, ids]

Initial value:

```
self_loops(graph)
   %Compute self-loop
   N = sum(graph(:,1)==graph(:,2))
```

Definition at line 675 of file generalConcentrationOverThreshold.m.

12.5.2.38 gillEntropy = rline(12)

Definition at line 165 of file generalConcentrationOverThreshold.m.

12.5.2.39 gillMean = rline(10)

Definition at line 163 of file generalConcentrationOverThreshold.m.

12.5.2.40 gillSD = rline(11)

Definition at line 164 of file generalConcentrationOverThreshold.m.

12.5.2.41 clear gillTimeSeries

Definition at line 576 of file generalConcentrationOverThreshold.m.

12.5.2.42 else graph = graph(graph(:,5)>0,:)

Definition at line 206 of file generalConcentrationOverThreshold.m.

12.5.2.43 else graphSUB = graphSUB(graphSUB(:,5)>0,:)

Definition at line 212 of file generalConcentrationOverThreshold.m.

12.5.2.44 for IDF

Initial value:

```
1:length(simDirs)

    if isdir(strcat(simDirs(IDF).name))

        disp(sprintf('|- Processing dir %s',simDirs(IDF).name))
```

Definition at line 58 of file generalConcentrationOverThreshold.m.

12.5.2.45 idOt = length(IDsOverThreshold)

Definition at line 568 of file generalConcentrationOverThreshold.m.

12.5.2.46 ids = graph(graph(:,1)==graph(:,2),1)

Definition at line 678 of file generalConcentrationOverThreshold.m.

12.5.2.47 IDsimFOLDER = 1

Definition at line 56 of file generalConcentrationOverThreshold.m.

12.5.2.48 Compute species over threshold IDsOverThreshold = tmpID(and(tmpConc>params.threshold,tmpCpxCut==0))

Definition at line 112 of file generalConcentrationOverThreshold.m.

12.5.2.49 incomingNodes = graph(graph(:,2)==IDsOverThreshold(idot),1)

Definition at line 462 of file generalConcentrationOverThreshold.m.

12.5.2.50 controllo che non ci siano cicli nell influx = loadInflux()'

Definition at line 85 of file generalConcentrationOverThreshold.m.

12.5.2.51 for innode

Initial value:

```
1 : length(incomingNodes)
                            alreadyAdded_ACS = 0
```

Definition at line 468 of file generalConcentrationOverThreshold.m.

12.5.2.52 if inSCCFlag = = 1 % If the node is in an ACS

Definition at line 488 of file generalConcentrationOverThreshold.m.

12.5.2.53 clear j

Initial value:

```
size(tmpGraph) 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 651 of file generalConcentrationOverThreshold.m.

12.5.2.54 loadedMols = rline(9)

Definition at line 162 of file generalConcentrationOverThreshold.m.

12.5.2.55 loadedMolsConc = rline(8)

Definition at line 161 of file generalConcentrationOverThreshold.m.

12.5.2.56 mol_I = rline(5)

Definition at line 158 of file generalConcentrationOverThreshold.m.

12.5.2.57 params params end if mol_II = rline(6)

Definition at line 159 of file generalConcentrationOverThreshold.m.

12.5.2.58 mol_III = rline(7)

Definition at line 160 of file generalConcentrationOverThreshold.m.

12.5.2.59 newSpeciesProb = rline(13)

Definition at line 166 of file generalConcentrationOverThreshold.m.

12.5.2.60 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 471 of file generalConcentrationOverThreshold.m.

12.5.2.61 end if nrg

Initial value:

```
= 1
                    nrgTimeSeries(rlineID,:) = [rtime,loadedMolsConc,loadedMols
      ]
```

Definition at line 169 of file generalConcentrationOverThreshold.m.

12.5.2.62 clear nrgTimeSeries

Definition at line 575 of file generalConcentrationOverThreshold.m.

12.5.2.63 numScc = graphconncomp(catSparse) histc(scc,1:max(scc))

Definition at line 442 of file generalConcentrationOverThreshold.m.

12.5.2.64 nZeros = length(num2str(tmpL)) - length(num2str(tmpl))

Definition at line 597 of file generalConcentrationOverThreshold.m.

12.5.2.65 end end out

Initial value:

```
"_" char
% 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.

12.5.2.66 outFname = strcat('_iGraph_CAT',tmpFilextPre,'_',tmpStrZeros,num2str(tmpRct),'_',num2str(tmprTime),'.csv')

Definition at line 645 of file generalConcentrationOverThreshold.m.

12.5.2.67 if nZeros for p

Initial value:

```
1:nZeros
            strZero = strcat(strZero,'0')
```

Definition at line 599 of file generalConcentrationOverThreshold.m.

12.5.2.68 param = findstr(tline,'=')

Definition at line 612 of file generalConcentrationOverThreshold.m.

12.5.2.69 previousTime = 0

Definition at line 135 of file generalConcentrationOverThreshold.m.

12.5.2.70 printTemporalMessage = 1

Definition at line 175 of file generalConcentrationOverThreshold.m.

12.5.2.71 end prod_bySCC = 0

Definition at line 425 of file generalConcentrationOverThreshold.m.

12.5.2.72 prod_bySCC_weight = 0

Definition at line 431 of file generalConcentrationOverThreshold.m.

12.5.2.73 prod_chain = 0

Definition at line 424 of file generalConcentrationOverThreshold.m.

12.5.2.74 prod_chain_weight = 0

Definition at line 430 of file generalConcentrationOverThreshold.m.

12.5.2.75 if then it has been produced within an ACS prod_inSCC = 0

Definition at line 423 of file generalConcentrationOverThreshold.m.

12.5.2.76 prod_inSCC_weight = 0

Definition at line 429 of file generalConcentrationOverThreshold.m.

12.5.2.77 **end** Compute the overlap between the different counter **prod_overlap** = 0

Definition at line 426 of file generalConcentrationOverThreshold.m.

12.5.2.78 **prod_overlap_weight** = 0

Definition at line 432 of file generalConcentrationOverThreshold.m.

12.5.2.79 **reaction** parameters files **rcsFiles** = **dir**('∗reactions_parameters∗')

Definition at line 116 of file generalConcentrationOverThreshold.m.

12.5.2.80 **rct** = **confparams.reactionProbability**

Definition at line 566 of file generalConcentrationOverThreshold.m.

12.5.2.81 **rctID** = 1

Definition at line 138 of file generalConcentrationOverThreshold.m.

12.5.2.82 **rctIDshow** = 1

Definition at line 136 of file generalConcentrationOverThreshold.m.

12.5.2.83 **rctIDshowNoSave** = 1

Definition at line 137 of file generalConcentrationOverThreshold.m.

12.5.2.84 Craete different variables **reaction** = **rline**(1)

Definition at line 154 of file generalConcentrationOverThreshold.m.

12.5.2.85 **realSccs** = self_loops(**graph**) 0

Definition at line 444 of file generalConcentrationOverThreshold.m.

12.5.2.86 **end** **rline** = fgetl(**fid**)

Definition at line 140 of file generalConcentrationOverThreshold.m.

12.5.2.87 rlineb = rline

Definition at line 141 of file generalConcentrationOverThreshold.m.

12.5.2.88 Update rlineID rlineID = 1

Definition at line 142 of file generalConcentrationOverThreshold.m.

12.5.2.89 rtime = rline(2)

Definition at line 155 of file generalConcentrationOverThreshold.m.

12.5.2.90 If the time is righe save iGraph structures to file if (rtime > rctIDshow ∗ analysisTimeInterval) disp(sprintf(' |- %s | Reaction %d - tim saveGraphToFile)(folderCat, reaction, rtime, confparams.nReactions, graph, filextPre)

Definition at line 179 of file generalConcentrationOverThreshold.m.

12.5.2.91 sccID = 0

Definition at line 427 of file generalConcentrationOverThreshold.m.

12.5.2.92 READ ALL THE DIRECTORY CONTAINING SIMULATIONS search = strcat('∗',params.distinctiveSubStr,'∗')

Definition at line 45 of file generalConcentrationOverThreshold.m.

12.5.2.93 self_loop_weight = 0

Definition at line 433 of file generalConcentrationOverThreshold.m.

12.5.2.94 if params showFig = 0

Definition at line 20 of file generalConcentrationOverThreshold.m.

12.5.2.95 simDirs = dir(search)

Definition at line 46 of file generalConcentrationOverThreshold.m.

12.5.2.96 **end end if the species concentration but the species is not produced by other species**

Definition at line 529 of file generalConcentrationOverThreshold.m.

12.5.2.97 **Select Times and reactions files speciesFiles = dir('species_1*')**

Definition at line 71 of file generalConcentrationOverThreshold.m.

12.5.2.98 **tempProd_chain_weight = 0**

Definition at line 464 of file generalConcentrationOverThreshold.m.

12.5.2.99 **params threshold = 0**

Definition at line 13 of file generalConcentrationOverThreshold.m.

12.5.2.100 **: timeInterval = rtime - previousTime**

Definition at line 193 of file generalConcentrationOverThreshold.m.

12.5.2.101 **File Containing all Times times = 0:params.deltaT:params.totT**

Definition at line 49 of file generalConcentrationOverThreshold.m.

12.5.2.102 **while ischar(tline tline = fgets(fidConf)**

Definition at line 609 of file generalConcentrationOverThreshold.m.

12.5.2.103 **params tmpIDsim = '5'**

Definition at line 17 of file generalConcentrationOverThreshold.m.

12.5.2.104 **tmpProd_chain = 0**

Definition at line 461 of file generalConcentrationOverThreshold.m.

12.5.2.105 **params tmpRctFileToLoad = ''**

Definition at line 18 of file generalConcentrationOverThreshold.m.

12.5.2.106 params tmpRctSUBFileToLoad = ''

Definition at line 19 of file generalConcentrationOverThreshold.m.

12.5.2.107 params tmpResFold = 'res'

Definition at line 15 of file generalConcentrationOverThreshold.m.

12.5.2.108 tmpStrZeros = zeroBeforeStrNum(tmpRct, tmpRcts)

Definition at line 644 of file generalConcentrationOverThreshold.m.

12.5.2.109 to = graph(:,2)+1

Definition at line 421 of file generalConcentrationOverThreshold.m.

12.5.2.110 wasteSpecies = 0

Definition at line 438 of file generalConcentrationOverThreshold.m.

12.5.2.111 end end If the species concentration but the species is not produced by other so it is waste if wasteSpeciesFLAG = 0

Definition at line 465 of file generalConcentrationOverThreshold.m.

12.5.2.112 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 485 of file generalConcentrationOverThreshold.m.

12.6 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness-_simulator/_analysis/generalStatistics.py File Reference

Namespaces

- namespace generalStatistics

Variables

- tuple generalStatistics.initTime = time.time()
- list generalStatistics.StrPath = sys.argv[1]

- tuple generalStatistics.threshold = float(sys.argv[2])
- list generalStatistics.singleGraphCreation = sys.argv[3]
- tuple generalStatistics.speciesToObserve = np.array([[1,2,3,4]])
- tuple generalStatistics.tmpDirs = sort(os.listdir(StrPath))
- int generalStatistics.samples = 0
- int generalStatistics.nFolders = 0
- string generalStatistics.speciesFilesInThisSim = 'species_*'
- tuple generalStatistics.speciesfileslist = sorted(glob.glob(speciesFilesInThisSim))
- string generalStatistics.paramFile = "acsm2s.conf"
- tuple generalStatistics.fid = open(paramFile, 'r')
- tuple generalStatistics.strLine = line.split('=')
- tuple generalStatistics.gens = int(strLine[1])
- tuple generalStatistics.sims = int(strLine[1])
- tuple generalStatistics.rcts = int(strLine[1])
- tuple generalStatistics.nsec = int(strLine[1])
- tuple generalStatistics.overallTimes = np.zeros((samples,nFolders))
- tuple generalStatistics.timesFrames = range(0,samples)
- tuple generalStatistics.overallLivSpe = np.zeros((samples,nFolders))
- tuple generalStatistics.overallMols = np.zeros((samples,nFolders))
- tuple generalStatistics.overallCpx = np.zeros((samples,nFolders))
- tuple generalStatistics.overallCpxCopies = np.zeros((samples,nFolders))
- tuple generalStatistics.overallDeath = np.zeros((samples,nFolders))
- tuple generalStatistics.overallnewSpecies = np.zeros((samples,nFolders))
- tuple generalStatistics.overallmaxAmount = np.zeros((samples,nFolders))
- tuple generalStatistics.overallminAmount = np.zeros((samples,nFolders))
- tuple generalStatistics.overallmeanAmount = np.zeros((samples,nFolders))
- tuple generalStatistics.overallmedianAmount = np.zeros((samples,nFolders))
- tuple generalStatistics.overallmaxL = np.zeros((samples,nFolders))
- tuple generalStatistics.overallminL = np.zeros((samples,nFolders))
- tuple generalStatistics.overallmeanL = np.zeros((samples,nFolders))
- tuple generalStatistics.overallmedianL = np.zeros((samples,nFolders))
- tuple generalStatistics.overallLoadedSpecies = np.zeros((samples,nFolders))
- string generalStatistics.ndn = '_0_allStatResults_'
- tuple generalStatistics.newdirAllResults = os.path.join(os.curdir, ndn)
- int generalStatistics.dirCount = 0
- tuple generalStatistics.rgens = fl.zeroBeforeStrNum(1, 1)
- tuple generalStatistics.rsims = fl.zeroBeforeStrNum(1, 1)
- string generalStatistics.ndnII = '__10_stastisticFiles_'
- tuple generalStatistics.newdir = os.path.join(os.curdir, ndnII)
- string generalStatistics.influxSpeciesFile = '_acsinflux.csv'
- tuple generalStatistics.fidflux = open(influxSpeciesFile, 'r')
- list generalStatistics.indexIn = []
- tuple generalStatistics.timesfileslist = sorted(glob.glob('times_*'))
- int generalStatistics.simulationID = 1
- tuple generalStatistics.rrcts = fl.zeroBeforeStrNum(0, rcts)
- tuple generalStatistics.rgen = fl.zeroBeforeStrNum(genID, gens)

- tuple generalStatistics.species3DSIZEamountLAST = np.array([[int(0), int(0), float(0), int(0)]])
- int generalStatistics.speciesFileID = 1
- tuple generalStatistics.rsim = fl.zeroBeforeStrNum(simulationID, sims)
- tuple generalStatistics.origDir = os.getcwd()
- tuple generalStatistics.statDir = os.path.join(os.curdir, newdir)
- string generalStatistics.outFnameStat = '_'
- tuple generalStatistics.statfilesFlag = os.path.isfile(outFnameStat)
- float generalStatistics.rctIDshow = 1.0
- float generalStatistics.rctIDshowNoSave = 1.0
- int generalStatistics.rctID = 1
- int generalStatistics.previousTime = 0
- string generalStatistics.filename = '01_gillespie_'
- list generalStatistics.stat = []
- list generalStatistics.species3Damount = []
- list generalStatistics.species3DSIZEamount = []
- tuple generalStatistics.concSpeciesHystory = np.zeros((samples,size(speciesToObserve)))
- tuple generalStatistics.concSpeciesRelativeHystory = np.zeros((samples,size(speciesToObserve)))
- tuple generalStatistics.totTimes = np.zeros((samples,size(speciesToObserve)))
- tuple generalStatistics.strs = string.split(root, '_')
- tuple generalStatistics.rctNumber = int(strs[len(strs)-1])
- tuple generalStatistics.sngSpFileFid = open(sngSpFile, 'r')
- tuple generalStatistics.speciesTable = sngSpFileFid.readlines()
- list generalStatistics.sID = []
- list generalStatistics.loadedSpecies = []
- tuple generalStatistics.Kdeg = array(Kdeg)
- list generalStatistics.sizes = []
- int generalStatistics.seqID = 0
- generalStatistics.realTheshold = threshold
- int generalStatistics.totAmount = 0
- tuple generalStatistics.eqLen = where(species3DSIZEamount[:,0] == len(seq))
- tuple generalStatistics.eqLenLAST = where(species3DSIZEamountLAST[:,0] == len(seq))
- list generalStatistics.stoPosition = speciesToObserve[0,:]
- tuple generalStatistics.MSizes = int(max(sizes))
- tuple generalStatistics.mSizes = int(min(sizes))
- tuple generalStatistics.avSizes = float(mean(sizes))
- tuple generalStatistics.meSizes = float(median(sizes))
- tuple generalStatistics.overallControlledSpecies = float(sum(concSpeciesHystory[speciesFileID-1,:]))
- tuple generalStatistics.overallControlledSpeciesOverTotConc = overallControlledSpecies/float(sum(amount[cutPnt==0]))
- tuple generalStatistics.overallControlledSpeciesOverTotPlusCpxConc = overallControlledSpecies/float(sum(amount[:]))

- tuple generalStatistics.statControlled = np.vstack([statControlled,(overallControlledSpecies,overallControlledSpeciesOverTotConc,overallControlledSpeciesOverTotPlusCpxConc)])
- string generalStatistics.finalStatDir = '__10_stastisticFiles_'
- generalStatistics.species3DNoInflux = species3Damount
- generalStatistics.species3DSIZEnoInflux = species3DSIZEamount
- string generalStatistics.outFname3DSize = '_'
- string generalStatistics.outFname3DSizeGrouped = '_'
- string generalStatistics.outFname3DSizeGroupedLAST = '_'
- string generalStatistics.outFname3DSizeGroupedLASTtot = '_'
- tuple generalStatistics.saveFileStat = open(outFnameStat, 'w')
- tuple generalStatistics.saveFile3Dsize = open(outFname3DSize, 'w')
- tuple generalStatistics.saveFile3DsizeGR = open(outFname3DSizeGrouped, 'w')
- tuple generalStatistics.saveFile3DsizeGRLAST = open(outFname3DSizeGroupedLAST, 'w')
- string generalStatistics.outFnameStatToObserve = '_'
- string generalStatistics.outFnameToObserve = '_'
- string generalStatistics.outFnameRatioToObserve = '_'
- tuple generalStatistics.saveFileStatKeepInEye = open(outFnameStatToObserve, 'w')
- tuple generalStatistics.saveFileKeepInEye = open(outFnameToObserve, 'w')
- tuple generalStatistics.saveFileRatioKeepInEye = open(outFnameRatioToObserve, 'w')
- string generalStatistics.sngStatFile = '_'
- tuple generalStatistics.sngStatFileFid = open(sngStatFile, 'r')
- int generalStatistics.linesID = 0
- tuple generalStatistics.Time = float(tmpTime)
- tuple generalStatistics.LivSpe = int(tmpLivSpe)
- tuple generalStatistics.Mols = float(tmpMols)
- tuple generalStatistics.Death = int(tmpDeath)
- tuple generalStatistics.NewS = int(tmpNewS)
- tuple generalStatistics.MaxA = float(tmpMaxA)
- tuple generalStatistics.MinA = float(tmpMinA)
- tuple generalStatistics.MeanA = float(tmpMeanA)
- tuple generalStatistics.MedianA = float(tmpMedianA)
- tuple generalStatistics.Cpx = int(tmpCpx)
- tuple generalStatistics.CpxCopies = float(tmpCpxCopies)
- tuple generalStatistics.MaxL = int(tmpMaxL)
- tuple generalStatistics.MinL = int(tmpMinL)
- tuple generalStatistics.MeanL = float(tmpMeanL)
- tuple generalStatistics.MedianL = float(tmpMedianL)
- tuple generalStatistics.loadedConc = float(tmpLoadedConc)
- tuple generalStatistics.statfileslastFID = open(outFname3DSizeGroupedLAST, 'r')
- tuple generalStatistics.normTimes = np.mean(overallTimes,1)
- tuple generalStatistics.normTimesStd = np.std(overallTimes,1)
- tuple generalStatistics.normY = np.mean(overallLivSpe,1)

- tuple generalStatistics.normYstd = np.std(overallLivSpe,1)
- tuple generalStatistics.endTime = time.time()
- int generalStatistics.minutes = 60

12.7 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness-_simulator/_analysis/KillSpam.m File Reference

Functions

- fclose (fid)
- 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)
- end cd (currentDir)
- 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))

Variables

- currentDir = cd()
- simDirs = dir('sim_∗')
- fid = fopen('results_TS1.txt','w')
- for x
- blocked = 0:5
- 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 = size(reactionsMatrix) 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 = size(catalysisMatrix) 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

12.7.1 Function Documentation

12.7.1.1 end catalysisMatrix (continua , 1:4)

12.7.1.2 end cd (currentDir)

12.7.1.3 cleavagesMatrix (cleavages , 1:6)

12.7.1.4 condensationsMatrix (condensations , 1:6)

12.7.1.5 fclose (fid)

12.7.1.6 fprintf (fid , 'NET%d\n\n' , x)

12.7.1.7 fprintf (fid , 'KILLERS\n\n')

12.7.1.8 end fprintf (fid , '\n\n')

12.7.1.9 fprintf (fid , 'Catalysts\n\n')

12.7.1.10 fprintf (fid , '%d\t' , killerCatalysts(i, 2))

12.7.1.11 fprintf (fid , '%d\n\n' , killerCatalysts(i, 3))

12.7.1.12 end fprintf (fid , 'KILLERS-SPAMMERS\n\n')

12.7.1.13 fprintf (fid , '%d\t' , KSCatalysts(KSCRows, 2))

12.7.1.14 fprintf (fid , '%d\n\n' , KSCatalysts(KSCRows, 3))

12.7.1.15 end fprintf (fid , 'SPAMMERS\n\n')

12.7.1.16 fprintf (fid , '%d\t' , spammerCatalysts(SCRows, 2))

12.7.1.17 fprintf (fid , '%d\n\n' , spammerCatalysts(SCRows, 3))

12.7.1.18 if ((outliersMatrix(i, 2)==1)&&(check2==1 &&check3==1))

12.7.1.19 if ((outliersMatrix(i, 2)==1)&&(check2==0 &&check3==0))

12.7.1.20 killerCatalysts (KCRows , 1)

12.7.1.21 killerCatalysts (KCRows , 2)

12.7.1.22 killerCatalysts (KCRows , 3)

12.7.1.23 killersMatrix (killRows , 1:6)

12.7.1.24 KSCatalysts (KSCRows , 1)

12.7.1.25 KSCatalysts (KSCRows , 2)

12.7.1.26 KSCatalysts (KSCRows , 3)

12.7.1.27 KSMatrix (KSRows , 1:6)

12.7.1.28 KSMatrix (i , :)

12.7.1.29 id KSRows ( ) [virtual]

12.7.1.30 outliersMatrix (outRows , 1:6)

12.7.1.31 if outliersMatrix (i , 2)

12.7.1.32 break end reactionsMatrix (continua , 1:6)

12.7.1.33 spammerCatalysts (KSCRows , 1)

12.7.1.34 spammerCatalysts (KSCRows , 2)

12.7.1.35 spammerCatalysts (KSCRows , 3)

12.7.1.36 SpammersMatrix (SpamRows , 1:6)

12.7.1.37 if SpamRows SpammersMatrix (: , 7)

12.7.1.38 SpammersMatrix (i , :)

12.7.1.39 else SpammersMatrix (i , 7)

12.7.2 Variable Documentation

12.7.2.1 blocked = 0:5

Definition at line 7 of file KillSpam.m.

12.7.2.2 break

Definition at line 54 of file KillSpam.m.

12.7.2.3 clear c

Definition at line 251 of file KillSpam.m.

12.7.2.4 catalysisFile = cFiles(length(cFiles)).name

Definition at line 45 of file KillSpam.m.

12.7.2.5 catalysisMatrix = 0

Definition at line 48 of file KillSpam.m.

12.7.2.6 creating catalysis matrix cFiles = dir('catalysis*')

Definition at line 44 of file KillSpam.m.

12.7.2.7 check2 = ismember(outliersMatrix(i,4),blocked)

Definition at line 86 of file KillSpam.m.

12.7.2.8 check3 = ismember(outliersMatrix(i,5),blocked)

Definition at line 87 of file KillSpam.m.

12.7.2.9 check4 = xor(check2,check3)

Definition at line 119 of file KillSpam.m.

12.7.2.10 if check5

Initial value:

```
= 1 || check6 == 1 || check7 == 1
                    KSRows = KSRows-1
```

Definition at line 132 of file KillSpam.m.

12.7.2.11 check6 = ismember(KSMatrix(i,5),cleavagesMatrix(:,3))

Definition at line 130 of file KillSpam.m.

12.7.2.12 check7 = ismember(KSMatrix(i,5),catalysisMatrix(:,2))

Definition at line 131 of file KillSpam.m.

12.7.2.13 if checkKM = isempty(killersMatrix)

Definition at line 237 of file KillSpam.m.

12.7.2.14 if checkKSM = isempty(KSMatrix)

Definition at line 148 of file KillSpam.m.

12.7.2.15 if checkSM = isempty(SpammersMatrix)

Definition at line 214 of file KillSpam.m.

12.7.2.16 cleavages = 0

Definition at line 28 of file KillSpam.m.

12.7.2.17 cleavagesMatrix = size(reactionsMatrix) 0

Definition at line 30 of file KillSpam.m.

12.7.2.18 else condensations = 0

Definition at line 27 of file KillSpam.m.

12.7.2.19 condensationsMatrix = 0

Definition at line 31 of file KillSpam.m.

12.7.2.20 continua = 1

Definition at line 12 of file KillSpam.m.

12.7.2.21 currentDir = cd()

Definition at line 1 of file KillSpam.m.

12.7.2.22 while continua cValues = fscanf(fid,'%f',7)

Definition at line 50 of file KillSpam.m.

12.7.2.23 else[r c] = size(killersMatrix)

Definition at line 241 of file KillSpam.m.

12.7.2.24 fid = fopen('results_TS1.txt','w')

Definition at line 3 of file KillSpam.m.

12.7.2.25 end clear i

Initial value:

```
1:rrm
        if reactionsMatrix(i,2) == 1
            cleavages = cleavages+1
```

Definition at line 32 of file KillSpam.m.

12.7.2.26 clear j

Initial value:

```
1:rcm
                if killerReaction == catalysisMatrix(j,3) && catalysisMatrix(j,
     4) > 0
                    KCRows = KCRows+1
```

Definition at line 100 of file KillSpam.m.

12.7.2.27 KCRows = 0

Definition at line 97 of file KillSpam.m.

12.7.2.28 killerCatalysts = 0

Definition at line 96 of file KillSpam.m.

12.7.2.29 killersMatrix = 0

Definition at line 83 of file KillSpam.m.

12.7.2.30 if killRows = 0

Definition at line 82 of file KillSpam.m.

12.7.2.31 KSCatalysts = 0

Definition at line 151 of file KillSpam.m.

12.7.2.32 KSCRows = 0

Definition at line 152 of file KillSpam.m.

12.7.2.33 KSMatrix = 0

Definition at line 115 of file KillSpam.m.

12.7.2.34 outliersMatrix = 0

Definition at line 71 of file KillSpam.m.

12.7.2.35 outRows = 0

Definition at line 70 of file KillSpam.m.

12.7.2.36 if possible

Initial value:

```
= 1
            outRows = outRows+1
```

Definition at line 74 of file KillSpam.m.

12.7.2.37 clear r

Definition at line 250 of file KillSpam.m.

12.7.2.38 identifying outliers reactionsCounters = size(catalysisMatrix) reactionsMatrix(:,6)

Definition at line 63 of file KillSpam.m.

12.7.2.39 reactionsFile = rFiles(length(rFiles)-1).name

Definition at line 10 of file KillSpam.m.

12.7.2.40 reactionsMatrix = 0

Definition at line 13 of file KillSpam.m.

12.7.2.41 creating reactions matrix rFiles = dir('reactions*')

Definition at line 9 of file KillSpam.m.

12.7.2.42 while continua rValues = fscanf(fid,'%d',7)

Definition at line 15 of file KillSpam.m.

12.7.2.43 SCRows = 0

Definition at line 218 of file KillSpam.m.

12.7.2.44 simDirs = dir('sim_*')

Definition at line 2 of file KillSpam.m.

12.7.2.45 spammersCatalysts = 0

Definition at line 217 of file KillSpam.m.

12.7.2.46 SpammersMatrix = 0

Definition at line 170 of file KillSpam.m.

12.7.2.47 end identifying spammers SpamRows = 0

Definition at line 169 of file KillSpam.m.

12.7.2.48 if stop = isempty(rValues)

Definition at line 16 of file KillSpam.m.

12.7.2.49 for x

Initial value:

```
1:length(simDirs)
   cd(strcat(simDirs(x).name,'/res'))
```

Definition at line 5 of file KillSpam.m.

12.8 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness-_simulator/_analysis/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 = size(reactionsMatrix) 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

12.8.1 Function Documentation

12.8.1.1 break end catalysisMatrix (continua , 1:7)

12.8.1.2 cleavagesMatrix (i , 1:6)

12.8.1.3 condensationsMatrix (i , 1:6)

12.8.1.4 end fclose (fid)

12.8.1.5 if ((outliersMatrix(i, 2)==0)&&(check2==1 &&check3==1))

12.8.1.6 if ((outliersMatrix(i, 2)==0)&&(check2==0 &&check3==0))

12.8.1.7 killersMatrix (killRows , 1:6)

12.8.1.8 KSMatrix (KSRows , 1:6)

12.8.1.9 KSMatrix (i , :)

12.8.1.10 id KSRows ( ) [virtual]

12.8.1.11 outliersMatrix (outRows , 1:6)

12.8.1.12 if outliersMatrix (i , 2)

12.8.1.13 break end reactionsMatrix (continua , 1:6)

12.8.1.14 SpammersMatrix (SpamRows , 1:6)

12.8.1.15 if SpamRows SpammersMatrix (: , 7)

12.8.1.16 SpammersMatrix (i , :)

12.8.1.17 else SpammersMatrix (i , 7)

12.8.2 Variable Documentation

12.8.2.1 blocked = 0:5

Definition at line 1 of file KSSearch.m.

12.8.2.2 catalysisFile = cFiles(length(cFiles)).name

Definition at line 40 of file KSSearch.m.

12.8.2.3 catalysisMatrix = 0

Definition at line 43 of file KSSearch.m.

12.8.2.4 creating catalysis matrix cFiles = dir('catalysis*')

Definition at line 39 of file KSSearch.m.

12.8.2.5 check2 = ismember(outliersMatrix(i,4),blocked)

Definition at line 82 of file KSSearch.m.

12.8.2.6 check3 = ismember(outliersMatrix(i,5),blocked)

Definition at line 83 of file KSSearch.m.

12.8.2.7 check4 = xor(check2,check3)

Definition at line 97 of file KSSearch.m.

12.8.2.8 if check5

Initial value:

```
= 1 || check6 == 1 || check7 == 1
                KSRows = KSRows-1
```

Definition at line 110 of file KSSearch.m.

12.8.2.9 check6 = ismember(KSMatrix(i,5),cleavagesMatrix(:,3))

Definition at line 108 of file KSSearch.m.

12.8.2.10 check7 = ismember(KSMatrix(i,5),catalysisMatrix(:,2))

Definition at line 109 of file KSSearch.m.

12.8.2.11 cleavages = 0

Definition at line 23 of file KSSearch.m.

12.8.2.12 cleavagesMatrix = size(reactionsMatrix) 0

Definition at line 25 of file KSSearch.m.

12.8.2.13 else condensations = 0

Definition at line 22 of file KSSearch.m.

12.8.2.14 condensationsMatrix = 0

Definition at line 26 of file KSSearch.m.

12.8.2.15 continua = 1

Definition at line 7 of file KSSearch.m.

12.8.2.16 while continua cValues = fscanf(fid,'%f',7)

Definition at line 45 of file KSSearch.m.

12.8.2.17 fid = fopen(reactionsFile)

Definition at line 6 of file KSSearch.m.

12.8.2.18 fpc = ceil(rrm/20∗19)

Definition at line 58 of file KSSearch.m.

12.8.2.19 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.

12.8.2.20 killersMatrix = 0

Definition at line 79 of file KSSearch.m.

12.8.2.21 identifying killers killRows = 0

Definition at line 78 of file KSSearch.m.

12.8.2.22 KSMatrix = 0

Definition at line 93 of file KSSearch.m.

12.8.2.23 outliersCounter = reactionsCounters(fpc:rrm)

Definition at line 59 of file KSSearch.m.

12.8.2.24 outliersMatrix = 0

Definition at line 67 of file KSSearch.m.

12.8.2.25 outRows = 0

Definition at line 66 of file KSSearch.m.

12.8.2.26 if possible

Initial value:

```
= 1
        outRows = outRows+1
```

Definition at line 70 of file KSSearch.m.

12.8.2.27 identifying outliers reactionsCounters = sort(reactionsMatrix(:,6))'

Definition at line 57 of file KSSearch.m.

12.8.2.28 reactionsFile = rFiles(length(rFiles)-1).name

Definition at line 5 of file KSSearch.m.

12.8.2.29 reactionsMatrix = 0

Definition at line 8 of file KSSearch.m.

12.8.2.30 creating reactions matrix rFiles = dir('reactions*')

Definition at line 4 of file KSSearch.m.

12.8.2.31 while continua rValues = fscanf(fid,'%d',7)

Definition at line 10 of file KSSearch.m.

12.8.2.32 SpammersMatrix = 0

Definition at line 129 of file KSSearch.m.

12.8.2.33 identifying spammers SpamRows = 0

Definition at line 128 of file KSSearch.m.

12.8.2.34 if stop = isempty(rValues)

Definition at line 11 of file KSSearch.m.

12.9 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness_simulator/_analysis/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)

12.9.1 Function Documentation

12.9.1.1 cd (currentDir)

12.9.1.2 fclose (fid)

12.9.1.3 fprintf (fid , 'NET%d\n\n' , x)

12.9.1.4 fprintf (fid , 'KILLERS\n\n')

12.9.1.5 end fprintf (fid , '\n\n')

12.9.1.6 fprintf (fid , 'KILLERS-SPAMMERS\n\n')

12.9.1.7 fprintf (fid , 'SPAMMERS\n\n')

12.9.2 Variable Documentation

12.9.2.1 clear c

Definition at line 27 of file KSSearchLauncher.m.

12.9.2.2 if checkKM = isempty(killersMatrix)

Definition at line 12 of file KSSearchLauncher.m.

12.9.2.3 if checkKSM = isempty(KSMatrix)

Definition at line 29 of file KSSearchLauncher.m.

12.9.2.4 if checkSM = isempty(SpammersMatrix)

Definition at line 46 of file KSSearchLauncher.m.

12.9.2.5 currentDir = cd()

Definition at line 1 of file KSSearchLauncher.m.

12.9.2.6 else[r c] = size(killersMatrix)

Definition at line 16 of file KSSearchLauncher.m.

12.9.2.7 fid = fopen('results.txt','w')

Definition at line 3 of file KSSearchLauncher.m.

12.9.2.8 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.

12.9.2.9 clear j

Definition at line 25 of file KSSearchLauncher.m.

12.9.2.10 KSSearch

Definition at line 7 of file KSSearchLauncher.m.

12.9.2.11 clear r

Definition at line 26 of file KSSearchLauncher.m.

12.9.2.12 simDirs = dir('sim_*')

Definition at line 2 of file KSSearchLauncher.m.

12.9.2.13 for x

Initial value:

```
1:length(simDirs)
   cd(strcat(simDirs(x).name,'/res'))
```

Definition at line 5 of file KSSearchLauncher.m.

12.10 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness-_simulator/_analysis/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.time-StructuresSavingInterval
- elseif isequal (tline(1:param-1),'reactionProbability') confparams.reaction-Probability
- 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,'=')

12.10.1 Function Documentation

12.10.1.1 end end fclose (fidConf)

12.10.1.2 while ischar (tline)

12.10.1.3 if isequal (tline(1:param-1) , 'nSeconds')

12.10.1.4 elseif isequal (tline(1:param-1) , 'nGEN')

12.10.1.5 elseif isequal (tline(1:param-1) , 'nSIM')

12.10.1.6 elseif isequal (tline(1:param-1) , 'nReactions')

12.10.1.7 elseif isequal (tline(1:param-1) , 'timeStructuresSavingInterval')

12.10.1.8 elseif isequal (tline(1:param-1) , 'reactionProbability')

12.10.1.9 elseif isequal (tline(1:param-1) , 'energy')

12.10.1.10 elseif isequal (tline(1:param-1) , 'ECConcentration')

12.10.1.11 elseif isequal (tline(1:param-1) , 'influx_rate')

12.10.1.12 elseif isequal (tline(1:param-1) , 'maxLOut')

12.10.1.13 elseif isequal (tline(1:param-1) , 'volume')

12.10.2 Variable Documentation

12.10.2.1 function confparams

Initial value:

```
readParameters()

    fidConf=fopen('acsm2s.conf','r')
```

Definition at line 1 of file readParameters.m.

12.10.2.2 param = findstr(tline,'=')

Definition at line 9 of file readParameters.m.

12.10.2.3 tline = fgets(fidConf)

Definition at line 5 of file readParameters.m.

12.11 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness-_simulator/_analysis/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
- params threshold = 0
- params decayTime = 10
- params tmpResFold = 'res'
- params distinctiveSubStr = 'sim'
- params tmpIDsim = '5'
- params tmpRctFileToLoad = ''
- params tmpRctSUBFileToLoad = ''

12.11.1 Function Documentation

12.11.1.1 concAnalysis (params)

12.11.1.2 function stats ( )

12.11.2 Variable Documentation

12.11.2.1 params decayTime = 10

Definition at line 84 of file stats.m.

12.11.2.2 params deltaT = 10

Definition at line 7 of file stats.m.

12.11.2.3 params distinctiveSubStr = 'sim'

Definition at line 86 of file stats.m.

12.11.2.4 params figureVisible

Initial value:

```
 timesAnalysis(tmpPath)
%
% clear all
% close all
% tmpPath = '~/Documents/lavoro/panini/reactionIN3' timesAnalysis(tmpPath)
%
% clear all
% close all
% tmpPath = '~/Documents/lavoro/panini/reactionIN4' timesAnalysis(tmpPath)

% clear all
% close all
% params.tmpPath = '~/Documents/lavoro/panini/reactionIN2' 0
```

Definition at line 57 of file stats.m.

12.11.2.5 params showFig = 0

Definition at line 9 of file stats.m.

12.11.2.6 params threshold = 0

Definition at line 83 of file stats.m.

12.11.2.7 params tmpIDsim = '5'

Definition at line 87 of file stats.m.

12.11.2.8 params tmpRctFileToLoad = ''

Definition at line 88 of file stats.m.

12.11.2.9 params tmpRctSUBFileToLoad = ''

Definition at line 89 of file stats.m.

12.11.2.10 params tmpResFold = 'res'

Definition at line 85 of file stats.m.

12.11.2.11 params totT = 1000

Definition at line 8 of file stats.m.

12.12 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness-_simulator/_analysis/timesAnalysis.m File Reference

Variables

- function [out]

12.12.1 Variable Documentation

12.12.1.1 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.m.

12.13 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness-_simulator/_analysis/timesAnalysis_PANINI.m File Reference

Variables

- function [out]

12.13.1 Variable Documentation

12.13.1.1 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.

12.14 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness-_simulator/_matlabinitializator/crea_catalizzatori.m File - Reference

Functions

- controllo while tot_reaz ∗reactionProbability ∗cleavageProbability tot_cleav fprintf ('la combinazione di reactionProbability e cleavageProbability\n agliata, non ci sono cleavage possibili sufficienti\n') reactionProbability
- 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
- cleavageProbability = input('introduci la nuova cleavageProbability \n')
- end catalisi_reali = round(tot_reaz∗reactionProbability∗numero_specie)

- reazione = -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
- 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,:)= specie_non_esistenti [];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 con-densazione catalizzatore(kk, 4)=0;trovato=0;while trovato==0 reazione(k, 1)=k;reazione(k, 2)=0;%id della condensazione:1 cleavage 0 conden-sazione index_specie_1=ceil(rand ∗(numero_specie));index_specie_2=ceil(rand ∗(numero_specie));nuova_specie=[deblank(specie_def(index_specie_1,:)), de-blank(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'

12.14.1 Function Documentation

12.14.1.1 end catalizzatore (kk , 1)

12.14.1.2 id catalizzatore catalizzatore (kk , 2)

12.14.1.3 id specie catalizzatore (kk , 3)

12.14.1.4 controllo while tot_reaz∗ reactionProbability∗ cleavageProbability tot_cleav fprintf ('la combinazione di reactionProbability e cleavageProbability\n [non ci sono cleavage possibili sufficienti\n'])

12.14.1.5 distribuendo le catalisi a caso viene una distribuzione uniforme switch (decisione_catalizzatori) [pure virtual]

12.14.2 Variable Documentation

12.14.2.1 if catalisi_reali = round(tot_reaz∗reactionProbability∗numero_specie)

Definition at line 21 of file crea_catalizzatori.m.

12.14.2.2 cleavageProbability = input('introduci la nuova cleavageProbability \n')

Definition at line 18 of file crea_catalizzatori.m.

12.14.2.3 if fino_a_che_lunghezza_i_polimeri_non_catalizzano = = 0 %i monomeri non catalizzano 0

Definition at line 54 of file crea_catalizzatori.m.

12.14.2.4 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.

12.14.2.5 else h = 0

Definition at line 29 of file crea_catalizzatori.m.

12.14.2.6 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.

12.14.2.7 else index = ceil(rand∗(numero_specie-numero_specie_da_-togliere))+numero_specie_da_togliere

Definition at line 49 of file crea_catalizzatori.m.

12.14.2.8 indice catalizzatore while catalisi_reali k = k+1

Definition at line 40 of file crea_catalizzatori.m.

12.14.2.9 kk = 0

Definition at line 36 of file crea_catalizzatori.m.

12.14.2.10 numero_specie = length(firing_disk(:,1))

Definition at line 5 of file crea_catalizzatori.m.

12.14.2.11 numero_specie_da_togliere = 0

Definition at line 43 of file crea_catalizzatori.m.

12.14.2.12 reazione = -9999

Definition at line 25 of file crea_catalizzatori.m.

12.14.2.13 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,:)= specie_non_esistenti[]; 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'

Definition at line 244 of file crea_catalizzatori.m.

12.14.2.14 end tot_cond = numero_specie^2

Definition at line 10 of file crea_catalizzatori.m.

12.14.2.15 tot_reaz = tot_cond+tot_cleav

Definition at line 12 of file crea_catalizzatori.m.

12.15 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness-_simulator/_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)∗overallConcentration

12.15.1 Function Documentation

12.15.1.1 else concentrazioni_iniziali (i) [pure virtual]

12.15.1.2 end end end STAMPA DISTRIBUZIONI DELLE CONCENTRAZIONI figure (123)

12.15.1.3 firing_disk (: , 3)

12.15.1.4 if firing_disk (index , 3)

12.15.1.5 firing_disk_reale (k , :) = firing_disk(i,:)

12.15.1.6 valore (1: *maxfiring_disk_bck(:, 2)*) [pure virtual]

12.15.1.7 valore (i)

12.15.1.8 id Z ( ) [virtual]

12.15.2 Variable Documentation

12.15.2.1 sum(concentrazioni_iniziali) end concentrazioni_iniziali = concentrazioni_iniziali/sum(concentrazioni_iniziali)∗overall-Concentration

Definition at line 129 of file crea_concentrazioni_iniziali.m.

12.15.2.2 a seconda dei casi seleziono le specie esistenti del firing disk

Definition at line 25 of file crea_concentrazioni_iniziali.m.

12.15.2.3 firing_disk = []

Definition at line 17 of file crea_concentrazioni_iniziali.m.

12.15.2.4 firing_disk_2 = firing_disk(1:tot_species,:)

Definition at line 16 of file crea_concentrazioni_iniziali.m.

12.15.2.5 firing_disk_bck = firing_disk

Definition at line 14 of file crea_concentrazioni_iniziali.m.

12.15.2.6 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.

12.15.2.7 for i

Initial value:

```
1:initialMaxLength
   tot_species = tot_species+length(alphabet)^i
```

Definition at line 9 of file crea_concentrazioni_iniziali.m.

12.15.2.8 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.

12.15.2.9 l_max = 0

Definition at line 31 of file crea_concentrazioni_iniziali.m.

12.15.2.10 end lunghezza_totale = length(firing_disk(:,1))

Definition at line 13 of file crea_concentrazioni_iniziali.m.

12.15.2.11 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.

12.15.2.12 probabilita_per_lunghezza = 1/length(vettore_ordinato_lunghezze)

Definition at line 82 of file crea_concentrazioni_iniziali.m.

12.15.2.13 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.

12.15.2.14 end remaining_species = tot_species

Definition at line 30 of file crea_concentrazioni_iniziali.m.

12.15.2.15 end end species_to_delete = round(ratio_firing_disk∗remaining_species)

Definition at line 38 of file crea_concentrazioni_iniziali.m.

12.15.2.16 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.

12.15.2.17 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.

12.15.2.18 while trovato

Initial value:

```
=0
                index = l_max+ceil(rand*remaining_species)
```

Definition at line 44 of file crea_concentrazioni_iniziali.m.

12.15.2.19 end end vettore_ordinato_lunghezze = unique(firing_disk_reale(:,2))

Definition at line 81 of file crea_concentrazioni_iniziali.m.

12.16 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness-_simulator/_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
- 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

12.16.1 Function Documentation

12.16.1.1 id check_ACS ( ) [virtual]

12.16.1.2 matrice_adiacenza_cat_prod (1: *maxinflux(:, 1),* 1: *maxinflux(:, 1)*) [pure virtual]

12.16.1.3 **else matrice_adiacenza_cat_prod (catalizzatore(i, 2) , reazione(catalizzatore(i, 3), 4))**

12.16.1.4 **matrice_adiacenza_cat_prod (catalizzatore(i, 2) , reazione(catalizzatore(i, 3), 5))**

12.16.1.5 **matrice_adiacenza_sub_prod (1: *maxinflux(:, 1),* 1: *maxinflux(:, 1)*)** [pure virtual]

12.16.1.6 **matrice_adiacenza_sub_prod (reazione(i, 5) , reazione(i, 3))**

12.16.1.7 **else matrice_adiacenza_sub_prod (reazione(i, 3) , reazione(i, 4))**

12.16.1.8 **matrice_adiacenza_sub_prod (reazione(i, 3) , reazione(i, 5))**

12.16.1.9 **1:length(influx(:,1 matrice_adiacenza_sub_prod: ( )** [virtual]

12.16.1.10 **end end end if max (real(eig(matrice_adiacenza_sub_prod)))** [pure virtual]

12.16.1.11 **end end end if max (real(eig(matrice_adiacenza_cat_prod)))** [pure virtual]

12.16.1.12 **end end end size (matrice_adiacenza_sub_prod)** [pure virtual]

12.16.2 Variable Documentation

12.16.2.1 **while check_ACS = 0**

Definition at line 19 of file crea_e_controlla_i_catalizzatori.m.

12.16.2.2 **counter_cicli = 0**

Definition at line 83 of file crea_e_controlla_i_catalizzatori.m.

12.16.2.3 **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.

12.16.2.4 **end end end for i**

Initial value:

```
crea_catalizzatori (firing_disk, reactionProbability, decisione_catalizzatori,
      fino_a_che_lunghezza_i_polimeri_non_catalizzano, alphabet, cleavageProbability,
       specie_def) 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 27 of file crea_e_controlla_i_catalizzatori.m.

12.16.2.5 inpudda = input('')

Definition at line 45 of file crea_e_controlla_i_catalizzatori.m.

12.16.2.6 matrice_adiacenza_cat_prod = 0

Definition at line 12 of file crea_e_controlla_i_catalizzatori.m.

12.16.2.7 matrice_adiacenza_sub_prod = 0

Definition at line 11 of file crea_e_controlla_i_catalizzatori.m.

12.17 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness_simulator/_matlabinitializator/crea_firing_disk.m File Reference

Functions

- step (i)

Variables

- function [firing_disk]
- for i
- end k = 1
- end firing_disk = id_species

12.17.1 Function Documentation

12.17.1.1 step (i)

12.17.2 Variable Documentation

12.17.2.1 end firing_disk = id_species

Definition at line 24 of file crea_firing_disk.m.

12.17.2.2 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.

12.17.2.3 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.

12.17.2.4 k = 1

Definition at line 18 of file crea_firing_disk.m.

12.18 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness-_simulator/_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

12.18.1 Function Documentation

12.18.1.1 influx (i , 2)

12.18.1.2 influx (i , :)

12.18.1.3 influx (1 , 1:2) [pure virtual]

12.18.1.4 if influx (1 , 1)

12.18.1.5 if sum (influx(i, 1) ==influx (: , 1))

Initial value:

```
=1
%                          trovato = 1
```

12.18.2 Variable Documentation

12.18.2.1 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.

12.18.2.2 for i

Initial value:

```
 1:species_to_keep
%             trovato = 0
```

Definition at line 56 of file crea_influx.m.

12.18.2.3 else clear influx influx = sort(influx)

Definition at line 65 of file crea_influx.m.

12.18.2.4 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.

12.18.2.5 end proporzionale alla lunghezza

Definition at line 108 of file crea_influx.m.

12.18.2.6 remaining_species = tot_species

Definition at line 71 of file crea_influx.m.

12.18.2.7 end end species_to_delete = round(ratio_influx∗tot_species)

Definition at line 53 of file crea_influx.m.

12.18.2.8 species_to_keep = tot_species - species_to_delete

Definition at line 54 of file crea_influx.m.

12.18.2.9 end end tot_species = length(firing_disk(:,1))

Definition at line 44 of file crea_influx.m.

12.18.2.10 while trovato = = 0

Definition at line 58 of file crea_influx.m.

12.19 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness_simulator/_matlabinitializator/crea_influx_semplice.m File Reference

Functions

- influx (j, 2)

Variables

- function [influx]
- for i
- end influx = zeros(sn,2)
- for j

12.19.1 Function Documentation

12.19.1.1 influx (j , 2)

12.19.2 Variable Documentation

12.19.2.1 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.

12.19.2.2 for i

Initial value:

```
 1:lMaxInflux
   sn = sn + length(alphabet)^i
```

Definition at line 7 of file crea_influx_semplice.m.

12.19.2.3 end influx = zeros(sn,2)

Definition at line 11 of file crea_influx_semplice.m.

12.19.2.4 for j

Initial value:

```
1:sn
  influx(j,1)=j
```

Definition at line 12 of file crea_influx_semplice.m.

12.20 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness-_simulator/_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

12.20.1 Variable Documentation

12.20.1.1 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.

12.20.1.2 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.

12.20.1.3 numero_elementi = length(vettore_elementi)

Definition at line 11 of file crea_tutte_le_combinazioni_di_elementi.m.

12.20.1.4 righe = numero_elementi^lunghezza_stringa

Definition at line 12 of file crea_tutte_le_combinazioni_di_elementi.m.

12.21 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness_simulator/_matlabinitializator/initial_distribution.m File Reference

12.22 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness_simulator/_matlabinitializator/inizializzatore_ACS.m File Reference

Functions

- cd (simFolder.path) cd(simFolder.name) mkdir(nome_cartella) = fopen('lanciatore.sh','w')
- cd (nome_cartella) fid1
- mkdir ('res')
- cd (thisFolder)
- fid1,'nGEN=' fprintf ()
- perchno 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]
- 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)
- counter id specie for i
- for j
- for z
- end end clear specie_temp end k = 0
- end specie_def = specie_def_2
- 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 lMaxInflux > if i <=(2^(lMaxInflux+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\n', 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\n', tmpKcpx);%k complex end end reazione [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;%perchse non c'energia sono tutte eso else if rand > energy if(i, 2)
- else eso_endo = 0
- temporal = specie_non_esistenti{i}

12.22.1 Function Documentation

12.22.1.1 **end cd (simFolder. *path*) = fopen('lanciatore.sh','w')**

12.22.1.2 **cd (nome_cartella)**

12.22.1.3 **cd (thisFolder)**

12.22.1.4 **Punto di tagli del complesso (1--L- *1*)**

12.22.1.5 **alphabet,firing_disk,initialMaxLength,lunghezza_max_fd,ratio-_firing_disk, scelta_concentrazioni, overallConcentration, gamma_powerlaw_concentrazioni crea_concentrazioni_iniziali ( )** [virtual]

12.22.1.6 **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]

12.22.1.7 **alphabet,massima_lunghezza_su_cui_calcolare_le_reazioni crea_firing_disk ( )** [virtual]

12.22.1.8 **concentrazioni_iniziali crea_influx ( )** [virtual]

12.22.1.9 **Coefficiente di degradazione (per *complessi*)**

12.22.1.10 **perchno input da file (i prossimi *4*)**

12.22.1.11 **fid2 d t fprintf ( )** [virtual]

12.22.1.12 **id i: (fid10 , '%d\t')** [virtual]

12.22.1.13 **id k ( )** [virtual]

12.22.1.14 **id kk ( )** [virtual]

12.22.1.15 **mkdir ('res')**

12.22.1.16 **end else if reazione (i , 2)**

12.22.1.17 **specie_def_2 (k , :)**

12.22.1.18 **id specie_non_esistenti ( )** [virtual]

Definition at line 322 of file inizializzatore_ACS.m.

12.22.2 Variable Documentation

12.22.2.1 Punto di tagli del complesso = fprintf(fid2,'%d\t',0)

Definition at line 211 of file inizializzatore_ACS.m.

12.22.2.2 k_fosforilazione velocitn cui l atp count = fprintf(fid1,'%d\n',nGEN)

Definition at line 35 of file inizializzatore_ACS.m.

12.22.2.3 else eso_endo = 0

Definition at line 304 of file inizializzatore_ACS.m.

12.22.2.4 fid10 = fopen('_acsnrgbooleanfunctions.csv','a')

Definition at line 24 of file inizializzatore_ACS.m.

12.22.2.5 fid2 = fopen('_acsspecies.csv','a')

Definition at line 20 of file inizializzatore_ACS.m.

12.22.2.6 fid3 = fopen('_acsreactions.csv','a')

Definition at line 21 of file inizializzatore_ACS.m.

12.22.2.7 fid4 = fopen('_acscatalysis.csv','a')

Definition at line 22 of file inizializzatore_ACS.m.

12.22.2.8 fid5 = fopen('_acsinflux.csv','a')

Definition at line 23 of file inizializzatore_ACS.m.

12.22.2.9 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,ratioSpeciesEnergiza
      ,Kass,Kdiss,Kcpx,K_cpx, randomSeed, debugLevel, timeStructuresSavingInterval,
      maxLOut,simFolder,lMaxInflux,fileTimesSaveInterval,nHours,nAttempts,revRctRatio
      )
%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, initialMaxLeng
      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,ratioSpeciesEnergiza
      ,Kass,Kdiss,Kcpx,K_cpx,onlyEnvironmentCreation, randomSeed, debugLevel,
      timeStructuresSavingInterval,maxLOut)

 rand('state',sum(100*clock))
 rand

%--------------------------------------------------------------------------
%apertura file
%--------------------------------------------------------------------------
thisFolder = pwd
```

Definition at line 1 of file inizializzatore_ACS.m.

12.22.2.10 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 166 of file inizializzatore_ACS.m.

12.22.2.11 for j

Initial value:

```
 length(specie_temp(:,1)):-1:1
        k = k+1
```

Definition at line 169 of file inizializzatore_ACS.m.

12.22.2.12 end end clear specie_temp end k = 0

Definition at line 180 of file inizializzatore_ACS.m.

12.22.2.13 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 IMaxInflux > if i<= (2^(IMaxInflux+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\n',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\n',tmpKcpx); %k complex end end reazione[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; %perchon c'energia sono tutte eso else if rand > energy if(i, 2)

Initial value:

```
=1
                                    eso_endo =1
```

Definition at line 301 of file inizializzatore_ACS.m.

12.22.2.14 end specie_def = specie_def_2

Definition at line 185 of file inizializzatore_ACS.m.

12.22.2.15 inserisco il numero decimale relativo alla funzione booleana della reazione end end st = fclose(fid1)

Definition at line 119 of file inizializzatore_ACS.m.

12.22.2.16 temporal = specie_non_esistenti{i}

Definition at line 332 of file inizializzatore_ACS.m.

12.22.2.17 tmpStr = specie_def(i,:)

Definition at line 203 of file inizializzatore_ACS.m.

12.22.2.18 for z

Initial value:

```
1:length(specie_temp(1,:))
          specie_def(k,z)=specie_temp(j,z)
```

Definition at line 171 of file inizializzatore_ACS.m.

12.23 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness_simulator/_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

12.23.1 Function Documentation

12.23.1.1 Keq (k , j)

12.23.2 Variable Documentation

12.23.2.1 for A

Initial value:

```
10e-7:10e-7:10e-6

   k=k+1
   j=0
```

Definition at line 17 of file lancia_acs.m.

12.23.2.2 AB = 0

Definition at line 12 of file lancia_acs.m.

12.23.2.3 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.

12.23.2.4 C = 2e-7

Definition at line 13 of file lancia_acs.m.

12.23.2.5 CA = 0

Definition at line 14 of file lancia_acs.m.

12.23.2.6 k = 0

Definition at line 16 of file lancia_acs.m.

12.23.2.7 k_complex = 1e-4

Definition at line 10 of file lancia_acs.m.

12.23.2.8 kcomplex = 1e6

Definition at line 8 of file lancia_acs.m.

12.23.2.9 kcond = 6e8

Definition at line 9 of file lancia_acs.m.

12.23.2.10 lancia_acs m clear all close all parametri kdiss = 0

Definition at line 7 of file lancia_acs.m.

12.24 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness_simulator/_matlabinitializator/lancia_inizializzatore_acs.m - File Reference

Functions

- lancia_inizializzatore_acs m script che lancia l inizializzatore per il simulatore ACSM2M e in cui sono contenuti tutti i parametri dell inizializzazione comandi di sistema clear all close all seme random rand ('state', sum(100 ∗clock)) rand%----------%PARAMETRI%----------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, ECConcentration, volume, energy, energyTarget, controllo_ACS_nel_ciclo, K_irrad)

12.24.1 Function Documentation

12.24.1.1 experiment all condensation are eso (and cleavage *endo*)

12.24.1.2 lancia_inizializzatore_acs m script che lancia l inizializzatore per il simulatore ACSM2M e in cui sono contenuti tutti i parametri dell inizializzazione comandi di sistema clear all close all seme random rand ('state' , sum(100 ∗clock))

12.24.1.3 id ratio_firing_disk ( ) [virtual]

12.24.2 Variable Documentation

12.24.2.1 experiment __pad1__

Definition at line 27 of file lancia_inizializzatore_acs.m.

12.24.2.2 experiment __pad2__

Definition at line 45 of file lancia_inizializzatore_acs.m.

12.24.2.3 experiment __pad3__

Definition at line 57 of file lancia_inizializzatore_acs.m.

12.24.2.4 alphabet = ['AB']

Definition at line 29 of file lancia_inizializzatore_acs.m.

12.24.2.5 esponente della powerlaw in caso parametri per la distribuzione dei catalizzatori

Definition at line 89 of file lancia_inizializzatore_acs.m.

12.24.2.6 complexFormationSymmetry = 0

Definition at line 51 of file lancia_inizializzatore_acs.m.

12.24.2.7 diffusion_contribute = 0

Definition at line 75 of file lancia_inizializzatore_acs.m.

12.24.2.8 energyTarget = 0

Definition at line 49 of file lancia_inizializzatore_acs.m.

12.24.2.9 ∗∗ NEW∗∗ fino_a_che_lunghezza_i_polimeri_non_catalizzano = 2

Definition at line 54 of file lancia_inizializzatore_acs.m.

12.24.2.10 coefficiente di fosforilazione

Definition at line 68 of file lancia_inizializzatore_acs.m.

12.24.2.11 parametro uniforme sulle powerlaw con esponente gamma gamma_powerlaw_concentrazioni = 2.1

Definition at line 85 of file lancia_inizializzatore_acs.m.

12.24.2.12 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.

12.24.2.13 controllo che non ci siano cicli nell influx = 2

Definition at line 77 of file lancia_inizializzatore_acs.m.

12.24.2.14 influx_rate = 1e-6

Definition at line 78 of file lancia_inizializzatore_acs.m.

12.24.2.15 initialMaxLength = 3

Definition at line 23 of file lancia_inizializzatore_acs.m.

12.24.2.16 initialPopulationNumber = 0

Definition at line 21 of file lancia_inizializzatore_acs.m.

12.24.2.17 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.

12.24.2.18 K_cpxDiss = 0

Definition at line 66 of file lancia_inizializzatore_acs.m.

12.24.2.19 costanti CINETICHE K_eq = 1000

Definition at line 61 of file lancia_inizializzatore_acs.m.

12.24.2.20 K_irrad = 0

Definition at line 71 of file lancia_inizializzatore_acs.m.

12.24.2.21 K_nrg = 0

Definition at line 69 of file lancia_inizializzatore_acs.m.

12.24.2.22 se cleavage se a condensazione kass = 100

Definition at line 63 of file lancia_inizializzatore_acs.m.

12.24.2.23 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.

12.24.2.24 parte da rivedere e correggere Kass_o_Kdiss = 0

Definition at line 63 of file lancia_inizializzatore_acs.m.

12.24.2.25 se cleavage kdiss = 100

Definition at line 63 of file lancia_inizializzatore_acs.m.

12.24.2.26 end lastFiringDiskSpeciesID = 0

Definition at line 32 of file lancia_inizializzatore_acs.m.

12.24.2.27 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.

12.24.2.28 parametro uniforme sulle lunghezze

Definition at line 84 of file lancia_inizializzatore_acs.m.

12.24.2.29 massima_lunghezza_su_cui_calcolare_le_reazioni = 3

Definition at line 24 of file lancia_inizializzatore_acs.m.

12.24.2.30 altri parametri moleculeDecay_KineticConstant = 0.02

Definition at line 74 of file lancia_inizializzatore_acs.m.

12.24.2.31 nReactions = 200000000

Definition at line 20 of file lancia_inizializzatore_acs.m.

12.24.2.32 nSeconds = 400

Definition at line 19 of file lancia_inizializzatore_acs.m.

12.24.2.33 nSIM = 1

Definition at line 18 of file lancia_inizializzatore_acs.m.

12.24.2.34 overallConcentration = 1e-4

Definition at line 38 of file lancia_inizializzatore_acs.m.

12.24.2.35 rapporto_Kfor_Kback = 0

Definition at line 64 of file lancia_inizializzatore_acs.m.

12.24.2.36 reactionProbability = 0.004

Definition at line 56 of file lancia_inizializzatore_acs.m.

12.24.2.37 reverseReactions = 0

Definition at line 58 of file lancia_inizializzatore_acs.m.

12.24.2.38 lunghezza dei monomeri polimeri da conservare scelta_concentrazioni = 1

Definition at line 84 of file lancia_inizializzatore_acs.m.

12.24.2.39 solubility_threshold = 0

Definition at line 76 of file lancia_inizializzatore_acs.m.

12.24.2.40 parametro switch

Definition at line 84 of file lancia_inizializzatore_acs.m.

12.24.2.41 volume = 1e-15

Definition at line 40 of file lancia_inizializzatore_acs.m.

12.25 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness_simulator/_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
- cd (simFolder.path) if exist(simFolder.name
- converto in decimale for per default iforme funzioni_booleane_in_dec (i, 2)
- if z< 10 parte_a=num2str(0);parte_b=num2str(z);parte4_nome_cartella=strcat(parte_a, parte_b);else parte4_nome_cartella=num2str(z);end nome_cartella=strcat(parte1_nome_cartella, parte2_nome_cartella, parte3_nome_cartella, parte4_nome_cartella);%scrivo il lanciatore.sh riga_lanciatore=strcat('nice./acsm2s./', nome_cartella,'/./', nome_cartella,'/res/./', nome_cartella,'/ > num2str (z)
- clear riga_lanciatore lancio il vero e proprio inizializzatore inizializzatore_ACS (nGEN, nSIM, nSeconds, nReactions, initialMaxLength, virtual_fd_max_length, overallConcentration, alphabet, complexFormationSymmetry, fino_a_che_lunghezza_i_polimeri_non_catalizzano, reactionProbability(i), 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)

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
- nome_folder = [3]
- ********** Inserimento dei PARAMETRI FISSI
- ********** Inserimento dei PARAMETRI quelli che restano ciostanti 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 = 10
- 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 virtual_fd_max_length = 4
- Lunghezza massima fino alla quale creare le reazioni maxLOut = 3
- 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 = 1e-6
- K_nrg = 0
- K_nrg_decay = 0
- coefficiente di decadimento delle molecole o dei carrier dalla propria componente energetica revRctRatio = 1000000
- ratioSpeciesEnergizable = 0
- percentuale di specie presenti nel sistema che possono essere energizzate per ogni specie create c a certa probabilit essere energizzabile o meno moleculeDecay_KineticConstant = 0.02
- diffusion_contribute = 0
- solubility_threshold = 0
- se sistema iuso lMaxInflux = 3
- 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 = 2
- 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
- Number of different networks ensambles
- Nome della cartella dove verr salvata la simulazione simFolder path = '~/Documents'

- Percorso dove verr creata la cartella simFolder dove verranno salvati tutti i file ********** nGEN = 1
- 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
- dir
- cd(thisFolder)%introduzione delle FUNZIONI BOOLEANE nell'energia funzioni_booleane_in_dec = bi2de(funzioni_booleane,'left-msb')
- parte2_nome_cartella = num2str(nome_folder(i))
- parte3_nome_cartella = ('_rete_n_')
- if z< 10 parte_a=num2str(0);parte_b=num2str(z);parte4_nome_cartella=strcat(parte_a, parte_b);else parte4_nome_cartella=num2str(z);end nome_cartella=strcat(parte1_nome_cartella, parte2_nome_cartella, parte3_nome_cartella, parte4_nome_cartella);%scrivo il lanciatore.sh riga_lanciatore=strcat('nice./acsm2s./', nome_cartella,'/./', nome_cartella,'/res/./', nome_cartella,'/ > sims
- if z< 10 parte_a=num2str(0);parte_b=num2str(z);parte4_nome_cartella=strcat(parte_a, parte_b);else parte4_nome_cartella=num2str(z);end nome_cartella=strcat(parte1_nome_cartella, parte2_nome_cartella, parte3_nome_cartella, parte4_nome_cartella);%scrivo il lanciatore.sh riga_lanciatore=strcat('nice./acsm2s./', nome_cartella,'/./', nome_cartella,'/res/./', nome_cartella,'/ > _
- if z< 10 parte_a=num2str(0);parte_b=num2str(z);parte4_nome_cartella=strcat(parte_a, parte_b);else parte4_nome_cartella=num2str(z);end nome_cartella=strcat(parte1_nome_cartella, parte2_nome_cartella, parte3_nome_cartella, parte4_nome_cartella);%scrivo il lanciatore.sh riga_lanciatore=strcat('nice./acsm2s./', nome_cartella,'/./', nome_cartella,'/res/./', nome_cartella,'/ > log
- riga_lanciatore_2 = strcat('echo "probabilita_reazione/',nome_cartella,'" ;',riga_lanciatore)
- count = fprintf(fid20,'%s\n ',riga_lanciatore_2)
- end end st = fclose(fid20)

12.25.1 Function Documentation

12.25.1.1 cd (simFolder. *path*)

12.25.1.2 converto in decimale for per default iforme funzioni_booleane_in_dec (i , 2)

12.25.1.3 clear riga_lanciatore lancio il vero e proprio inizializzatore **inizializzatore_ACS** (**nGEN** , **nSIM** , **nSeconds** , **nReactions** , **initialMaxLength** , **virtual_fd_max_length** , **overallConcentration** , **alphabet** , **complexFormationSymmetry** , **fino_a_che_lunghezza_i_polimeri-_non_catalizzano** , **reactionProbability(i)** , **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** , **IMaxInflux** , **fileTimesSaveInterval** , **nHours** , **nAttempts** , **revRctRatio**)

12.25.1.4 **if** z<10 parte_a=**num2str**(0); parte_b=**num2str**(z); parte4_nome_cartella = strcat(parte_a,parte_b); **else** parte4_nome_cartella = **num2str**(z); **end** nome_cartella= strcat(parte1_nome_cartella,**parte2_nome_cartella**,**parte3_-nome_cartella**,parte4_nome_cartella); %scrivo il lanciatore.sh riga_lanciatore = strcat(' nice ./acsm2s ./',nome_cartella,'/ ./',nome_cartella,'/res/ ./', nome_cartella,'/ > **num2str** (**z**)

12.25.1.5 Numero massimo di **reazioni** (secondo parametro di **stop** oltre al numero di *secondi*) [pure virtual]

12.25.1.6 Numero di volte in cui sistema ritenta la stessa **rete** ( )

12.25.1.7 Number of different networks for every network will be performed nSim different **simulation** (differnt random *seeds*)

12.25.1.8 Numero massimo di ore per la **simulazione** ( ) [pure virtual]

12.25.1.9 Numero di **simulazioni** (diversi semi *random*)

12.25.2 Variable Documentation

12.25.2.1 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 70 of file start.m.

12.25.2.2 if z< 10 parte_a=num2str(0) parte_b=num2str(z) parte4_nome_-cartella=strcat(parte_a, parte_b) else parte4_nome_cartella=num2str(z) end nome_cartella=strcat(parte1_nome_cartella, parte2_nome_cartella, parte3_nome_cartella, parte4_nome_cartella)%scrivo il lanciatore.sh riga_lanciatore=strcat('nice./acsm2s./', nome_cartella,'/./', nome_cartella,'/res/./', nome_cartella,'/ > _

Definition at line 142 of file start.m.

12.25.2.3 livello di dettaglio messaggi durante lasciare a

Definition at line 25 of file start.m.

12.25.2.4 alphabet = ['AB']

Definition at line 35 of file start.m.

12.25.2.5 parametri per la distribuzione dei catalizzatori

Definition at line 61 of file start.m.

12.25.2.6 cleavageProbability = 0.5

Definition at line 41 of file start.m.

12.25.2.7 energia non considerata complexFormationSymmetry = 0

Definition at line 39 of file start.m.

12.25.2.8 energia considerata

Definition at line 38 of file start.m.

12.25.2.9 altrimenti Quando influx_rate indica la lunghezza massima delle molecole che possono uscire dal contenitore

Definition at line 33 of file start.m.

12.25.2.10 count = fprintf(fid20,'%s\n ',riga_lanciatore_2)

Definition at line 144 of file start.m.

12.25.2.11 lasciare a debugLevel = 0

Definition at line 25 of file start.m.

12.25.2.12 esponente della powerlaw in caso decisione_catalizzatori = 1

Definition at line 60 of file start.m.

12.25.2.13 diffusion_contribute = 0

Definition at line 52 of file start.m.

12.25.2.14 dir

Initial value:

```
= 0
    mkdir(simFolder.name)
```

Definition at line 79 of file start.m.

12.25.2.15 energy = 0

Definition at line 38 of file start.m.

12.25.2.16 Number of different networks ensambles

Definition at line 63 of file start.m.

12.25.2.17 definisce il tempo in cui vengono salvati i file durante la simulazione fileTimesSaveInterval = 10

Definition at line 27 of file start.m.

12.25.2.18 fino_a_che_lunghezza_i_polimeri_non_catalizzano = 2

Definition at line 40 of file start.m.

12.25.2.19 ********** ********** Inserimento dei PARAMETRI FISSI

Definition at line 19 of file start.m.

12.25.2.20 cd (thisFolder) %introduzione delle FUNZIONI BOOLEANE nell'energia funzioni_booleane_in_dec = bi2de(funzioni_booleane,'left-msb')

Definition at line 110 of file start.m.

12.25.2.21 parametro uniforme sulle powerlaw con esponente gamma gamma_powerlaw_concentrazioni = 2.1

Definition at line 59 of file start.m.

12.25.2.22 Numero di generazioni

Definition at line 69 of file start.m.

12.25.2.23 end********** ***** S T A R T***** ********** for i

Initial value:

```
 1:virtual_fd_max_length
    lastFiringDiskSpeciesID = lastFiringDiskSpeciesID + length(alphabet)^i
```

Definition at line 71 of file start.m.

12.25.2.24 controllo che non ci siano cicli nell influx

Definition at line 63 of file start.m.

12.25.2.25 influx_rate

Initial value:

```
 0 indica fino a quale lunghezza le molecole non variano in quantita'
       (simulazione membrana permeabile)
ECConcentration=0
```

Definition at line 33 of file start.m.

12.25.2.26 K_cpx = 1e-6

Definition at line 46 of file start.m.

12.25.2.27 K_nrg = 0

Definition at line 47 of file start.m.

12.25.2.28 K_nrg_decay = 0

Definition at line 48 of file start.m.

12.25.2.29 Kass = 50

Definition at line 43 of file start.m.

12.25.2.30 Kcpx = 50

Definition at line 45 of file start.m.

12.25.2.31 Kdiss = 25

Definition at line 44 of file start.m.

12.25.2.32 end lastFiringDiskSpeciesID = lastFiringDiskSpeciesID -1

Definition at line 74 of file start.m.

12.25.2.33 se sistema iuso lMaxInflux = 3

Definition at line 55 of file start.m.

12.25.2.34 if z<10 parte_a=num2str(0); parte_b=num2str(z); parte4_nome_cartella = strcat(parte_a,parte_b); else parte4_nome_cartella = num2str(z); end nome_cartella= strcat(parte1_nome_cartella,parte2_nome_cartella,parte3_-nome_cartella,parte4_nome_cartella); %scrivo il lanciatore.sh riga_lanciatore = strcat(' nice ./acsm2s ./',nome_cartella,'/ ./',nome_cartella,'/res/ ./', nome_cartella,'/ > log

Definition at line 142 of file start.m.

12.25.2.35 percentuale di specie da cancellare rispetto a tutte quelle che restano dopo aver conservato i polimeri fino a lunghezzamaxfd lunghezza_max_fd = 2

Definition at line 57 of file start.m.

12.25.2.36 parametro uniforme sulle lunghezze

Definition at line 58 of file start.m.

12.25.2.37 Lunghezza massima fino alla quale creare le reazioni maxLOut = 3

Definition at line 33 of file start.m.

12.25.2.38 percentuale di specie presenti nel sistema che possono essere energizzate per ogni specie create c a certa probabilit essere energizzabile o meno moleculeDecay_KineticConstant = 0.02

Definition at line 51 of file start.m.

12.25.2.39 Percorso dove verr creata la cartella simFolder dove verranno salvati tutti i file********** nGEN = 1

Definition at line 69 of file start.m.

12.25.2.40 nome_folder = [3]

Definition at line 14 of file start.m.

12.25.2.41 nome_prob = [0.125

Definition at line 12 of file start.m.

12.25.2.42 Definisce il tempo in cui vengono salvati i dati sui file reaction_parameter e i vari living nReactions = 200000000

Definition at line 28 of file start.m.

12.25.2.43 ********** Inserimento dei PARAMETRI quelli che restano ciostanti in tutti gli esperimenti della serie nSIM = 1

Definition at line 22 of file start.m.

12.25.2.44 Numero di al momento significa che alla fine di ogni generazione da ogni file di fine sim partono altre Nsim

Definition at line 69 of file start.m.

12.25.2.45 overallConcentration = 0.0333

Definition at line 36 of file start.m.

12.25.2.46 **if z< 10 parte_a=num2str(0) parte_b=num2str(z) parte4_nome_-cartella=strcat(parte_a, parte_b) else parte4_nome_cartella=num2str(z) end nome_cartella=strcat(parte1_nome_cartella, parte2_nome_cartella, parte3_nome_cartella, parte4_nome_cartella)%scrivo il lanciatore.sh riga_lanciatore=strcat('nice./acsm2s./', nome_cartella,'/./', nome_cartella,'/res/./', nome_cartella,'/ > parte2_nome_cartella = num2str(nome_folder(i))**

Definition at line 130 of file start.m.

12.25.2.47 **parte3_nome_cartella = ('_rete_n_')**

Definition at line 131 of file start.m.

12.25.2.48 **concAnalysis(params) clear all close all params path = '~/Documents'**

Definition at line 65 of file start.m.

12.25.2.49 **Numero di secondi randomSeed = 0**

Definition at line 24 of file start.m.

12.25.2.50 **lunghezza massima delle molecole presenti nell influx ratio_firing_disk = 0**

Definition at line 56 of file start.m.

12.25.2.51 **ratioSpeciesEnergizable = 0**

Definition at line 50 of file start.m.

12.25.2.52 **reactionProbability = [0.000516529**

Definition at line 11 of file start.m.

12.25.2.53 **reverseReactions = 0**

Definition at line 42 of file start.m.

12.25.2.54 **coefficiente di decadimento delle molecole o dei carrier dalla propria componente energetica revRctRatio = 1000000**

Definition at line 49 of file start.m.

12.25.2.55 **riga_lanciatore_2 = strcat('echo "probabilita_reazione/',nome_cartella,'" ;',riga_lanciatore)**

Definition at line 143 of file start.m.

12.25.2.56 **lunghezza dei monomeri polimeri da conservare scelta_concentrazioni = 1**

Definition at line 58 of file start.m.

12.25.2.57 **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.

12.25.2.58 **se = 0 non viene considerato**

Definition at line 33 of file start.m.

12.25.2.59 **if z<10 parte_a=num2str(0); parte_b=num2str(z); parte4_nome_cartella = strcat(parte_a,parte_b); else parte4_nome_cartella = num2str(z); end nome_cartella= strcat(parte1_nome_cartella,parte2_nome_cartella,parte3_-nome_cartella,parte4_nome_cartella); %scrivo il lanciatore.sh riga_lanciatore = strcat(' nice ./acsm2s ./',nome_cartella,'/ ./',nome_cartella,'/res/ ./', nome_cartella,'/ > sims**

Definition at line 142 of file start.m.

12.25.2.60 **livello di dettaglio messaggi durante simulazione**

Definition at line 25 of file start.m.

12.25.2.61 **solubility_threshold = 0**

Definition at line 53 of file start.m.

12.25.2.62 **end end st = fclose(fid20)**

Definition at line 153 of file start.m.

12.25.2.63 **parametro switch**

Definition at line 58 of file start.m.

12.25.2.64 thisFolder = pwd

Definition at line 77 of file start.m.

12.25.2.65 Definisce il tempo in cui vengono salvati i dati sui file times

Definition at line 27 of file start.m.

12.25.2.66 livello di dettaglio messaggi durante lasciare per debug software timeStructuresSavingInterval = nSeconds/100

Definition at line 26 of file start.m.

12.25.2.67 Lunghezza massima delle specie da creare virtual_fd_max_length = 4

Definition at line 32 of file start.m.

12.25.2.68 volume = 1e-18

Definition at line 37 of file start.m.

12.26 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness-_simulator/acs_headers.h File Reference

```
#include <QtCore/QCoreApplication>     #include <QString-
List>  #include <QTextStream>  #include <QFile>  #include
<QList>  #include <QTime>  #include <iostream>  #include
<string>  #include <vector>  #include <deque>  #include
<cmath> #include <cstdlib> #include "mtrand.h" #include
<time.h>
```

Defines

- #define MINIMAL_PROMPT -1
- #define RUNNING_VERSION 0
- #define SMALL_DEBUG 1
- #define MEDIUM_DEBUG 2
- #define HIGH_DEBUG 3
- #define FINDERRORDURINGRUNTIME -10
- #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 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 NEP 2.71828182845904523536028747
- #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

12.26.1 Define Documentation

12.26.1.1 #define AVO 6.02214179e+23

Definition at line 100 of file acs_headers.h.

12.26.1.2 #define BOTHLOAD 2

Definition at line 90 of file acs_headers.h.

12.26.1.3 #define CATALYSTLOAD 1

Definition at line 89 of file acs_headers.h.

12.26.1.4 #define CLEAVAGE 1

Definition at line 62 of file acs_headers.h.

12.26.1.5 #define CLEAVAGEBASED 1

Definition at line 79 of file acs_headers.h.

12.26.1.6 #define COMPLEXDEGRADATION 3

Definition at line 64 of file acs_headers.h.

12.26.1.7 #define COMPLEXFORMATION 2

Definition at line 63 of file acs_headers.h.

12.26.1.8 #define COMPLEXLOAD 3

Definition at line 91 of file acs_headers.h.

12.26.1.9 #define CONDENSATION 0

Definition at line 61 of file acs_headers.h.

12.26.1.10 #define CONDENSATIONBASED 0

Definition at line 80 of file acs_headers.h.

12.26.1.11 #define ENDO_CLEAVAGE 6

Definition at line 67 of file acs_headers.h.

12.26.1.12 #define ENDO_COMPLEXFORMATION 8

Definition at line 69 of file acs_headers.h.

12.26.1.13 #define ENDO_CONDENSATION 7

Definition at line 68 of file acs_headers.h.

12.26.1.14 #define ENDOERGONIC 0

Definition at line 78 of file acs_headers.h.

12.26.1.15 #define ENERGIZABLE 1

Definition at line 85 of file acs_headers.h.

12.26.1.16 #define ENERGYBASED 1

Definition at line 81 of file acs_headers.h.

12.26.1.17 #define ENERGYEFFLUX 9

Definition at line 70 of file acs_headers.h.

12.26.1.18 #define ENERGYFREE 0

Definition at line 82 of file acs_headers.h.

12.26.1.19 #define ESOERGONIC 1

Definition at line 77 of file acs_headers.h.

12.26.1.20 #define FALSENRG '0'

Definition at line 84 of file acs_headers.h.

12.26.1.21 #define FINDERRORDURINGRUNTIME -10

Definition at line 50 of file acs_headers.h.

12.26.1.22 #define HIGH_DEBUG 3

Definition at line 49 of file acs_headers.h.

12.26.1.23 #define INVPROPORTIONALMOLECULEAMOUNT 3

Definition at line 58 of file acs_headers.h.

12.26.1.24 #define MEDIUM_DEBUG 2

Definition at line 48 of file acs_headers.h.

12.26.1.25 #define MINIMAL_PROMPT -1

Definition at line 45 of file acs_headers.h.

12.26.1.26 #define MINIMALRCTTIMEMULTI 100

Definition at line 103 of file acs_headers.h.

12.26.1.27 #define NEP 2.71828182845904523536028874

Definition at line 99 of file acs_headers.h.

12.26.1.28 #define NEWREACTIONS 1

Definition at line 95 of file acs_headers.h.

12.26.1.29 #define NOTENERGIZABLE 0

Definition at line 86 of file acs_headers.h.

12.26.1.30 #define NOTHINGLOAD 4

Definition at line 92 of file acs_headers.h.

12.26.1.31 #define PHOSPHORILATION 5

Definition at line 66 of file acs_headers.h.

12.26.1.32 #define PRECIPITATED 0

Definition at line 74 of file acs_headers.h.

12.26.1.33 #define PROPORTIONALMOLECULEAMOUNT 1

Definition at line 56 of file acs_headers.h.

12.26.1.34 #define RANDOMRANGE random()

Definition at line 53 of file acs_headers.h.

12.26.1.35 #define RUNNING_VERSION 0

Definition at line 46 of file acs_headers.h.

12.26.1.36 #define SMALL_DEBUG 1

Definition at line 47 of file acs_headers.h.

12.26.1.37 #define SOLUBLE 1

Definition at line 73 of file acs_headers.h.

12.26.1.38 #define SPECIESDECAY 4

Definition at line 65 of file acs_headers.h.

12.26.1.39 #define SUBSTRATELOAD 0

Definition at line 88 of file acs_headers.h.

12.26.1.40 #define TRUENRG '1'

Definition at line 83 of file acs_headers.h.

12.26.1.41 #define UNIFORMMOLECULEAMOUNT 2

Definition at line 57 of file acs_headers.h.

12.26.1.42 #define UPGRADEREACTIONS 0

Definition at line 96 of file acs_headers.h.

12.26.2 Typedef Documentation

12.26.2.1 typedef long double acs_double

Definition at line 35 of file acs_headers.h.

12.26.2.2 typedef unsigned int acs_int

Definition at line 37 of file acs_headers.h.

12.26.2.3 typedef unsigned long int acs_longInt

Definition at line 36 of file acs_headers.h.

12.27 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness_simulator/catalysis.cpp File Reference

```
#include "catalysis.h"
```

12.28 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness_simulator/catalysis.h File Reference

```
#include "acs_headers.h" #include "commonFunctions.h"
```

Classes

- class catalysis

  CATALYSIS class.

12.29 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness_simulator/commonFunctions.cpp File Reference

```
#include "commonFunctions.h"
```

Functions

- int returnSelectionIdFromAWeightProbVector (acs_double *tmpArray, MTRand &tmpRandomGenerator)

  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_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 round (acs_double tmpX)
- string dec2bin (acs_int tmpInt)
- bool ExitWithError (string strFunctionName, string strError)

12.29.1 Function Documentation

12.29.1.1 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.

12.29.1.2 bool ExitWithError (string *strFunctionName,* string *strError*)

Function to close the program after having en error

Definition at line 269 of file commonFunctions.cpp.

12.29.1.3 acs_double getDoubleRandom (acs_double *tmpFromNum,* acs_double *tmpToNum,* MTRand & *tmpRandomGenerator*)

Definition at line 151 of file commonFunctions.cpp.

12.29.1.4 acs_longInt getIntRandom (acs_longInt *tmpFromNum,* acs_longInt *tmpToNum,* MTRand & *tmpRandomGenerator*)

Definition at line 131 of file commonFunctions.cpp.

12.29.1.5 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.

12.29.1.6 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.

12.29.1.7 int returnSelectionIdFromAWeightProbVector (acs_double * *tmpArray,* MTRand & *tmpRandomGenerator*)

This funtion returns a random position in a probability weight array of N elements.

Definition at line 15 of file commonFunctions.cpp.

12.29.1.8 acs_longInt returnSelectionIdFromAWeightProbVector (vector< acs_double > & *tmpVector,* acs_double *tmpMaxValue,* MTRand & *tmpRandomGenerator*)

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.

12.29.1.9 acs_longInt returnSelectionIdFromAWeightProbVectorAlready-Normalized (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.

12.29.1.10 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.

12.29.1.11 acs_double round (acs_double *tmpX*)

Function to round double numbers in integers

Definition at line 229 of file commonFunctions.cpp.

12.30 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness-_simulator/commonFunctions.h File Reference

```
#include "acs_headers.h"
```

Functions

- int returnSelectionIdFromAWeightProbVector (acs_double *tmpArray, MTRand &tmpRandomGenerator)

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_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 round (acs_double tmpX)
- string dec2bin (acs_int tmpInt)
- bool ExitWithError (string strFunctionName, string strError)

12.30.1 Function Documentation

12.30.1.1 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.

12.30.1.2 bool ExitWithError (string *strFunctionName,* string *strError*)

Function to close the program after having en error

Definition at line 269 of file commonFunctions.cpp.

12.30.1.3 acs_double getDoubleRandom (acs_double *tmpFromNum,* acs_double *tmpToNum,* MTRand & *tmpRandomGenerator*)

Definition at line 151 of file commonFunctions.cpp.

12.30.1.4 acs_longInt getIntRandom (acs_longInt *tmpFromNum,* acs_longInt *tmpToNum,* MTRand & *tmpRandomGenerator*)

Definition at line 131 of file commonFunctions.cpp.

12.30.1.5 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.

12.30.1.6 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.

12.30.1.7 int returnSelectionIdFromAWeightProbVector (acs_double * *tmpArray,* MTRand & *tmpRandomGenerator*)

This funtion returns a random position in a probability weight array of N elements.

Definition at line 15 of file commonFunctions.cpp.

12.30.1.8 acs_longInt returnSelectionIdFromAWeightProbVector (vector< acs_double > & *tmpVector,* acs_double *tmpMaxValue,* MTRand & *tmpRandomGenerator*)

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.

12.30.1.9 acs_longInt returnSelectionIdFromAWeightProbVectorAlready-Normalized (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.

12.30.1.10 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.

12.30.1.11 acs_double round (acs_double *tmpX*)

Function to round double numbers in integers

Definition at line 229 of file commonFunctions.cpp.

12.31 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness-_simulator/environment.cpp File Reference

```
#include "environment.h"
```

12.32 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness-_simulator/environment.h File Reference

```
#include "acs_headers.h"   #include "species.h"   #include
```

```
"reactions.h" #include "commonFunctions.h" #include "catalysis.-
h" #include "gillespie.h"
```

Classes

- class environment

  environment class

12.33 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness-_simulator/gillespie.cpp File Reference

```
#include "gillespie.h"
```

12.34 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness-_simulator/gillespie.h File Reference

```
#include "acs_headers.h"
```

Classes

- class gillespie

12.35 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness-_simulator/main.cpp File Reference

```
#include "acs_headers.h" #include "environment.h"
```

Functions

- void saveToFile (QString tmpSavingPath, environment *tmpEnvironment, acs_int tmpGen, acs_int tmpSim, acs_int tmpStep)
- void saveTimesToFile (QString tmpSavingPath, environment *tmpEnvironment, acs_int tmpGen, acs_int tmpSim, acs_int tmpStep)
- void saveInitialConditionsToFile (QString tmpSavingPath, environment *tmpEnvironment, acs_int tmpGen, acs_int tmpSim, acs_int tmpStep)
- int main (int argc, char *argv[])

12.35.1 Function Documentation

12.35.1.1 int main (int *argc,* char * *argv[]*)

double random number generator

Definition at line 324 of file main.cpp.

12.35.1.2 void saveInitialConditionsToFile (QString *tmpSavingPath,* environment * *tmpEnvironment,* acs_int *tmpGen,* acs_int *tmpSim,* acs_int *tmpStep*)

Save to file all the INITIAL structures

Version

1.0

Parameters

QString	tmpSavingPath Saving files path
environment	*tmpEnvironment environment instance reference
tmpSim	Current simulation
acs_int	Current step

Definition at line 628 of file main.cpp.

12.35.1.3 void saveTimesToFile (QString *tmpSavingPath,* environment * *tmpEnvironment,* acs_int *tmpGen,* acs_int *tmpSim,* acs_int *tmpStep*)

Save TIMES to file

Version

1.0

Parameters

QString	tmpSavingPath Saving files path
environment	*tmpEnvironment environment instance reference
tmpSim	Current simulation
acs_int	Current step

Definition at line 612 of file main.cpp.

12.35.1.4 void saveToFile (QString *tmpSavingPath,* environment * *tmpEnvironment,* acs_int *tmpGen,* acs_int *tmpSim,* acs_int *tmpStep*)

Save to file structures at step tmpStep

Version

1.0

Parameters

QString	tmpSavingPath Saving files path
environment	∗tmpEnvironment environment instance reference
tmpSim	Current simulation
acs_int	Current step

Definition at line 592 of file main.cpp.

12.36 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness-_simulator/mtrand.cpp File Reference

```
#include "mtrand.h"
```

12.37 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness-_simulator/mtrand.h File Reference

Classes

- class MTRand_int32
- class MTRand
- class MTRand_closed
- class MTRand_open
- class MTRand53

12.38 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness-_simulator/reactions.cpp File Reference

```
#include "reactions.h"
```

12.39 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness-_simulator/reactions.h File Reference

```
#include "acs_headers.h" #include "commonFunctions.h"
```

Classes

- class reactions

12.40 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness_simulator/species.cpp File Reference

```
#include "species.h"
```

12.41 /Users/alessandrofilisetti/Dropbox/ACS_SVN/carness/carness_simulator/species.h File Reference

```
#include "acs_headers.h" #include "commonFunctions.h"
```

Classes

- class species

  This class contains declarations of the species class.