

STOCKS (STOCHastic Kinetic Simulator)

Version 1.01 (August 2001)

Exact stochastic simulation of Biochemical kinetics with Gillespie algorithm.

Andrzej M. Kierzek

Institute of Biochemistry and Biophysics,

Polish Academy of Sciences,

Pawińskiego 5a, 02-106 Warszawa

POLAND

andrzejk@ibb.waw.pl

1. INTRODUCTION

In order to run STOCKS you need to create 3 files: input, control and file list. Input file contains definition of your system. It specifies reaction formulas, stochastic rate constants and initial conditions. Control file contains parameters controlling the simulation: names of reactants to be monitored, number of Monte Carlo experiments, simulation time etc.. File list contains the paths to input, control, log and restart files. It also lists the path to directory in which output, log and restart files are written. After preparing these files you run simulation with the command:

```
stocks name_of_file_list_file
```

2. FILE FORMATS

All statements in all files described below are CASE SENSITIVE. The character # put as the first character in the line comments out the line.

2.1 Input file

Every reaction in the system must be specified in the input file. In order to specify reaction use following syntax:

```
reaction
  formula "A+B->C"
  rateC 10
end
```

In the reaction definition "reaction", "formula", "rateC" and "end" entries must be placed in separate lines. Reaction formula must be enclosed in quotes. The "->" string separates left and right side of the formula. Stoichiometric coefficients should be followed by "*" character (eg: "2*A->AA"). Names of the reactants should not exceed 20 characters and obviously must not contain "->" string or the quotes.

The label "rateC" followed by a real number specifies stochastic rate constant of the reaction. Stochastic rate constants should be expressed in reciprocal time units of your simulation. If you specify simulation time in seconds and want to have seconds on the "time" column in the output

files the unit of stochastic rate constant is 1/s . Before writing input file please have a look at point 3 (Reaction types and units) in this file, where you find details concerning units used in STOCKS software.

Initial conditions of the simulation are defined as in the following example:

```
initial_conditions
  A 100
  B 20
end
```

In this example A and B are names of the reactants that appear in reaction definitions followed by integer numbers of molecules. Amounts of reactants must be given as NUMBERS OF MOLECULES in the system rather than in any concentration unit (see chapter 3). Every reactant must be specified in separate line. The reactants for which initial numbers of molecules are not defined are assigned default values of 0.

There are three features of STOCKS dedicated to the simulation of biochemical systems: growing volume of reaction environment, replication reactions and random pools of reactants.

The following statement in the input file:

```
linear_volume_change
```

will make the volume of reaction environment double during `GenerationTime`. See chapter 3 for the explanation what effect will volume growth have on reaction rates.

Replication reaction is specified in the following way:

```
replication
  formula "A->2*A"
end
```

The purpose of replication reaction is to allow simple simulation of cell division. After every generation of the cell, specified by `GenerationTime` parameter (see control file specification) following operations are performed:

- 1) All replication reactions are executed once.
- 2) All the numbers of molecules in the system are divided by 2.

3) The volume of reaction environment is reset to initial value.

Now, imagine that you simulate the system in which A denote DNA element eg. promoter sequence and that initial number of A molecules is 1. You define replication reaction as in the example above. Then, after `GenerationTime` replication reaction increases the number of A elements to 2. Subsequently all the numbers of molecules in the system are divided by 2. You end up with the system that contains 1 molecule of A and half of all other molecules in the system. It is the simplest way in which cell division can be simulated and "attention" of the simulation program switched to one of the progeny cells. Please notice that these scheme allows you to execute any single, elementary reaction before the numbers of molecules are divided by 2. For instance if your DNA region is occupied by regulatory protein (let's name the complex AB) you would probably like to define replication reaction: `"AB->2*A+B"` This reaction would dissociate AB complex and replicate A element.

Metabolic pool of the reactant is defined by "pool" label:

```
pool
  A 35 3.5
  B 350 35
end
```

In this example substance A will be treated as a fluctuating pool with the mean number of molecules 35 and standard deviation 3.5 . B will be modeled as the random pool with the mean 350 and standard deviation 35. This means that numbers of A and B molecules would not depend on the reactions defined in your system. Before executing the step of Gillespie algorithm the numbers of A and B molecules will be generated as random numbers from Gaussian distribution with means and standard deviations equal to these specified in the "pool" entry. For the example of how random pools have been applied to the modeling of numbers of RNA polymerase molecules available for the given gene please read reference [1]. You will also find there justification of using Gaussian distribution.

If you want to make the number of reactant molecules constant for the substances A use the following statement:

```
constant A
```

As the result the number of A molecules will not be changed in the simulation and remain equal to the value specified in `initial_conditions` entry.

If you want to start counting the time of the simulation from specified time rather than from 0 you can write:

```
initial_time 20.626
```

in order to start the simulation from 20.626 time units. This option is useful when you want to restart the simulation that crashed (see 2.5).

2.2 Control file

The format of the control file will be explained using following example:

```
Monitor A
Monitor B

GenerationTime 2100
NumberOfExperiments 100
NumberOfGenerations 10
MonitorTime 10
Seed 62433
Saveperiod 10
```

If you want to save the number of molecules as the function of time for the given reactant it should be specified by the `Monitor` statement. In the example above the changes in the number of molecules of the substances A and B will be saved. For every substance specified by `Monitor` separate file will be created. I will refer to this file as trajectory file. Trajectory file will contain the number of molecules saved after every preset time interval. The length of the interval is specified by the `MonitorTime` statement. Therefore, in the example above the numbers of A and B molecules will be recorded every 10 time units.

The timescale of the single simulation is determined by `GenerationTime` and `NumberOfGenerations` parameters. `GenerationTime` specifies the length of the single cell generation. After every `GenerationTime` replication process is simulated in the way described in the point 1.1 and simulation continues. In the example above 10 generations are simulated and generation time is 2100 time units. Thus, the timescale of the single simulation will be 21000 time units. If you want to run the simulation without simulating cell division process `GenerationTime` becomes the time of the simulation. You can also use `TimeOfExperiment` statement which sets the value of the same variable as `GenerationTime` statement.

The single simulation specified above needs to be repeated many times in order to collect data necessary for the computations of statistical parameters describing time evolution of the system (mean number of molecules vs time, standard deviation of this number etc). The number of independent runs is specified by `NumberOfExperiments` parameter. After every run numbers of molecules are reset to initial conditions, time is reset to 0 and next simulation is run. In our example 100 independent simulations will be performed.

Trajectory files are written to disk at the end of generation or at the end of single simulation if cell divisions are not simulated. In order to make restarting simulation possible, in the case of computer's crash, restart file is written. The frequency of writing restart file is specified by `Saveperiod` parameter. `Saveperiod` is expressed in "monitor times". Therefore, in the example above restart file will be saved after every 10 times the numbers of molecules are saved (100 time units).

2.3 File list file.

The structure of file list file is simple and will be explained using following example:

```
Input directory/job.inp
Output directory/
Control directory/job.ctrl
Restart directory/job.rst
Log directory/job.log
```

Input, Control, Restart and Log statements specify the names of input, control, log and restart files respectively. Output statement specifies the name of the directory (this directory must be created before simulation) in which trajectory files will be saved. The files paths should be absolute (starting from /) or be specified relatively to the directory from which the software is run (current working directory at the moment of executing `stocks` command).

2.4 Trajectory files

For every substance to be monitored the program writes trajectory file. The name of the substance is used as the filename. The file contains two columns. First column contain the time and the second one the number of molecules recorded at the given time. All the trajectories (independent runs of the simulation) are written to the single trajectory file. They are separated by empty line followed by the comment of the kind:

```
# TRAJECTORY 20
```

The line above is the header of 20th trajectory. This format is well suited for the examination with GNUPLOT software (the program I strongly recommend for plotting trajectories) which understands “#” as a comment and blank line as separator of data series. Thus gnuplot command:

```
plot 'trajectory_file_name' with lines
```

will nicely plot the data from trajectory file. The format of trajectory file is also recognized by utility programs `mtrj` and `trjadd` described in details in point 4 of this file.

After running first examples please note that times in first column are not exactly equal to multiplications of `MonitorTime`. This is due to the fact that Gillespie algorithm does not have constant timestep of the simulation. During the simulation only the times at which elementary reactions occur are generated. Thus the time in the first column is the first **exact time** of some reaction in the system which exceeds the multiplication of monitor time. If for some analysis you need to have equal time intervals on time axis the `mt_rj` utility may be of help.

2.5 Restart file

Restart file is formatted in such a way that it can be added to input file in order to specify initial conditions corresponding to the state of the system at the moment of saving the file. Thus the file contains `initial_conditions` and `initial_time` entries in the format of input file. As the `initial_conditions` the numbers of molecules of all the reactants are given. Thus in order to restart the simulation you need to delete `initial_conditions` section from the input file and include restart file in this place. The trajectories written from restarted simulation will have the times starting from the time at which previous simulation was interrupted as specified by `initial_time` line in restart file.

2.6 Log file

As the program is expected to be run in background all the messages it generates are written to log file rather than to console. Thus if for some reason your simulation did not start see if the log file has been created. You may find there an error message pointing to the line with mistyped reaction formula or other syntax error. After every Monte Carlo repetition is finished the message is written to the log file. Another message is written if the system reached the state at which all substrate molecules have been consumed so no reaction can happen and the trajectory is finished before preset simulation time.

3. REACTION TYPES AND UNITS

3.1 Reaction types

Gillespie algorithm deals with elementary reactions i.e. reactions representing actual reactive collisions among the molecules. For that reason you cannot put into the input file complex reaction mechanism e.g. you cannot say that reaction k is Michaelis-Menten process with defined K_M and turnover values. The system needs to be modeled using three types of elementary reactions:

- 1) **First order reaction.** This reaction type includes both isomerisation reactions (eg: $A \rightarrow B$) when single substrate is converted into single product molecules and degradations (eg: $A \rightarrow B + 3C$) when the single substrate molecule is split into multiple products.
- 2) **Second order reaction.** This type is assigned to the reaction involving encounter of two molecules of **different** reactants (eg: $A + B \rightarrow C$, $A + B \rightarrow 2C + D$ etc ...).
- 3) **Homodimer formation.** This type is assigned to reactions involving encounter of two molecules of **the same** reactant (eg: $2A \rightarrow B$, $2A \rightarrow B + C$ etc ..). The reason why this kind of reaction cannot be treated in the same way as second order reactions is that it has different number of possible reactant combinations. If you have in your system N_A molecules of A and N_B molecules of B the number of possible distinct reactive encounters is $N_A N_B$ whereas the number of possible distinct encounters among A molecules is $N_A(N_A-1)/2$.

The third and higher order reactions are not supported by STOCKS as they can be reasonably estimated by the combination of second order reactions. Actually, the collision of three molecules at **exactly** the same time is not possible so the third order reaction is always the combination of two second order ones.

3.2 The unit of reactant amount

The amount of reactant should be given as the number of molecules. Therefore any concentration should be multiplied by the volume of reaction environment and expressed as the **INTEGER** number of molecules. In the trajectory and restart files the numbers of molecules are also given.

3.2 Time units

You can use any time unit in the simulation provided that all stochastic rate constants are expressed in reciprocal of this unit. The time column in the trajectory files is then expressed in the time unit you have chosen. Let us assume that we want to have a second as the time unit. Then, all stochastic rate constants should be expressed in 1/s and the simulation, generation and monitor times in the control file should be given in seconds. As a result the time column in trajectory files and initial time in restart file will be output in seconds.

3.3 Relation between stochastic and experimental rate constants

The experimentally measured rate constants that should form a basis of any simulation are usually given in 1/s and $M^{-1} s^{-1}$ units. There is fortunately a straightforward way to convert them into stochastic rate constants provided that the volume V of reaction environment is known. The experimental rate constant should be expressed as the number of events occurring in the volume V in the time unit of the simulation. Therefore, in the case of first order reaction the stochastic rate constant equals experimental rate constant as the rate of first order reaction does not depend on the volume. In the case of second order reaction the experimental rate constant should be divided by volume and converted from moles to the number of events. Let

us calculate stochastic rate constant for the experimental value $1 \text{ M}^{-1} \text{ s}^{-1}$ in the volume of 10^{-15} L:

$$C = 1 (\text{mol/L})^{-1} \text{ s}^{-1} / 10^{-15} \text{ L} = 10^{15} \text{ mol}^{-1} \text{ s}^{-1}$$

$$C = 10^{15} \text{ mol}^{-1} \text{ s}^{-1} / (6.02 \cdot 10^{23} \text{ mol}^{-1}) = 1.7 \cdot 10^{-9} \text{ 1/s}$$

In the case of homodimer formation the experimental value should be divided by V and multiplied by 2 ($C = 2 \cdot \text{determ_rate}/\text{volume}$) due to the difference in the number of distinct reactive encounters. Therefore if the reaction $A+A \rightarrow B$ has the experimental rate of $1 \text{ M}^{-1} \text{ s}^{-1}$ then the stochastic rate constant in the volume of 10^{-15} L is $3.4 \cdot 10^{-9} \text{ 1/s}$.

The current release of STOCKS does not contain any utility for conversion of units although I plan to write it in the near future. It is also safer to do computations carefully by hand as very small numbers of molecules can easily be computed wrongly due to round-off errors if the concentrations are expressed in “nanomol” and the volume is in the range of “femtoliters”.

The original paper of Gillespie [2] very clearly explains the relation between experimental and stochastic rate constants.

3.4 Simulations in changing volumes.

If `linear_volume_change` statement is present in the input file the volume is linearly doubled during the simulation in the following way. The initial volume of the system is assumed to be 1. Before every step of Gillespie algorithm the volume is calculated using the following formula:

$$V = (1+t/T)$$

where t is the time of the simulation and T generation time. Then, the stochastic rate constants of both types of second order reactions are divided by V. The rates of first order reactions remain unchanged.

The practical consequence of the above is that the values of stochastic rate constants present in the system should be calculated for the **initial volume** of reaction environment. As the volume will grow during simulation the rates will be scaled appropriately.

4. UTILITY PROGRAMS

There are four utility programs in the current distributions of STOCKS. The PERL script phase2D.pl in phase2d example shows how they can be combined in order to perform analysis of simulation results in parameter-scanning application.

4.1 *mtrj*

mtrj is the program for calculating mean trajectory and its +/- n standard deviations “envelopes”. The program reads following command line parameters:

- i **name** specifies the name of the trajectory file to be analyzed.
- bin **n** specifies the length of time interval in which data are averaged.
- nstd **n** the program will calculate +/- n standard deviations envelopes.
- vc the program will output variation coefficient instead of mean trajectory.

mtrj reads all the trajectories from trajectory file. Using this data it computes mean number of molecules present in the system in each time interval specified by **-bin** option. The standard deviation of this number is also computed. Program outputs the trajectory file containing three trajectories. The first one contains mean number of molecules the second the mean + n standard deviations and the third the mean – n standard deviations. In this trajectory file the times will be equal to the multiplications of the time interval in which averaging has been done. It is reasonable to set this time interval equal to the monitor time of the simulations.

Alternatively, the program outputs the variation coefficient as a function of time i.e. the ratio of standard deviation to the mean in every time interval.

Results are printed to standard output so they need to be redirected to the file.

4.2 *trjadd*

The program adds or subtracts trajectories which are results of the same simulation. It simply adds the numbers of molecules corresponding to the same exact times in two trajectory files. If the files are coming from different simulations the times will not match and the program generates error message. The program is run in the following way:

```
trjadd file1 file2 [-sub]
```

where `file1` and `file2` are names of trajectory files and `-sub` option makes the program subtract trajectory `file2` from the trajectory `file1`.

4.3 *linefit*

The program fits the line to the specified part of trajectory. The trajectory file must be either specified with `-i` option or fed to standard input. Options `-bgn t1` and `-end t2` specify the part of the trajectory to which the line should be fitted (`t1` and `t2` are in time units). Program outputs three numbers to standard output. The numbers are slope, intercept and correlation coefficient respectively.

4.4 *mean*

The program computes mean number of molecules observed in the specified part of trajectory. The input and options are the same as in the case of `linefit`. Two numbers printed to standard output are mean and standard deviation respectively.

5. REFERENCES

1. Kierzek AM, Zaim J, Zielenkiewicz P. The Effect of Transcription and Translation Initiation Frequencies on the Stochastic Fluctuations in Prokaryotic Gene Expression. *J. Biol. Chem* 276(11) 8165-8172 (2001).
2. Gillespie DT. Exact Stochastic Simulation of Coupled Chemical Reactions. *J. Phys. Chem* 81(25) 2340-2361 (1977).