The basic input files are BioSimWare-compatible; this list provides the additional files requested by cupSODA, cupSODA*L, coagSODA, and cuTauLeaping. For each input file the content and formatting are described. In all tables, N represents the number of parallel threads; M is the number of parameters (i.e., reactions); S is the number of chemical species; T is the number of time instants (i.e., the number of points in the dynamics); D is the number of experiments in the target time series; E is the number of repetitions of each experiment. All files are tab-separated, except where specified. All indices are 0-based, that is, the first index is always 0. Please avoid empty lines in input files. Please, try to avoid the scientific notation.

cupSODA / coagSODA / cupSODA*L / LASSIE

File name	*Content*	*Format*
c_matrix	Matrix of kinetic parameters[1]	N rows, M columns
cs_vector	Vector of chemical species to be sampled	$K \leq S$ rows
MX_0	Matrix of initial quantities[2]	N rows, S columns
t_vector	Vector of sampling times	T rows
M_feed	Vector of the feed species (values are zero if the species is not in feed; equal to the fixed amount if the species is in feed)	1 row, S columns
atol_vector	Vector of absolute error tolerances	S rows
rtol	Relative tolerance (scalar value)	1 value
volume	Volume[3]	1 value
modelkind	Type of input model: concentration-based (“deterministic”) or molecules-based (“stochastic”)	{stochastic, deterministic}
max_steps	Maximum number of steps during integration (i.e., LSODA’s MXSTEP value)	1 value

The following parameters are required only in the case of **fitness evaluation** (experimental):

ts_rep	Number of experiments repetitions	1 value (E)
ts_numtgt	Number of target chemical species in the time series	1 value (K')

[1] Can be specified both in terms of deterministic or stochastic parameters (please see modelkind file).
[2] Can be specified both in terms of concentrations or molecular amounts (please see modelkind file).
[3] Required in the case of molecules amounts based models

ts_matrix	Matrix of target time series. First column is the sample time instant.	T rows, $E * D * K' + 1$ columns
tts_vector	Vector of threads assignment to different experimental conditions (e.g., for a single population of 128 individuals, the file contains a single number: 128. For three populations of 32 individuals each, it contains 32 64 96)	D rows

cuTauleaping

Nome file	*Contenuto*	*Formato*
c_matrix	Matrix of stochastic parameters	N rows, M columns
cs_vector	Vector of chemical species to be sampled	$K \leq S$ rows
MX_0	Matrix of initial amounts (molecules)	N rows, S columns
t_vector	Vector of sampling times	T rows
MX_feed	Matrix of feed species (indices of chemical species whose amount never changes)[4]	N rows, $K \leq S$ columns
eps	Error control (NOT SUPPORTED)	1 value
ML_vector	List of feed species	$K \leq S$ rows

The following parameters are required only in the case of fitness evaluation (experimental):

ts_rep	Number of repetitions of the experiments	1 value (E)
ts_numtgt	Number of target chemical species in the time series	1 value (K')
ts_matrix	Matrix of target time series. First column is the sample time instant.	T rows, $E * D * K' + 1$ columns
tts_vector	Vector of threads assignment to different experimental conditions	D rows
p_rep	Number of repetitions used to assess the average dynamics (G)	1 value (G)

[4]No longer supported

Command line arguments for cupSODA, cupSODA*L and coagSODA

```
./cupSODA inputfolder BLOCKS outputfolder outputprefix GPU FITNESS
MEMORY_CONFIG DUMP
```

- `inputfolder` is the directory which contains the input model, encoded as described above;
- `BLOCKS` is the number of CUDA blocks to be used during the simulation; it is supposed to be a number perfectly dividing the number of parallel threads (e.g.., different parameterizations), that is, the number of rows in the c_matrix file. Its value must be compatible with CUDA requirements[5];
- `outputfolder` is the folder containing the outputs of simulations;
- `outputprefix` is the prefix of output files;
- `GPU` is the ID of the target GPU used for simulations;
- `FITNESS` is a numeric value which indicates the type of fitness value to be calculated:
    - 0: fitness calculation disabled;
    - 1: fitness based on point-2-point distances;
    - 2: fitness based on point-2-point distances redirected on stdout;
    - 3: not supported;
    - 4: fitness for robust oscillation evaluations.
- `MEMORY_CONFIG` is a numeric value indicating the types of low-latency memories to be used to speed up simulations:
    - 0: just global memory;
    - 1: use shared memory;
    - 2: usa shared and constant memory.
    - Negative values: same semantics as above + adaptive max steps
- `DUMP` if it is different from 0, it activates debug dumps
    - 0: no dump
    - 1: normal debug
    - 2: extended debug dump

[5]The number of blocks B is supposed to be equal to the number of simulations T divided by the number of desired threads-per-block P, i.e., B=T/P. P must not exceed 1024 (CUDA limitations) and, in any case, it should remain below a threshold due to the availability of shared memory. Since shared memory is used to store the state of the system during simulations, assuming S chemical species, we have P<floor(SM/((S+1)*8) where SM is the amount of shared memory available on each streaming multiprocessor.

Command line arguments for cuTauLeaping

```
./cuTauLeaping input_folder THREADS BLOCKS GPU OFFSET output_folder
output_prefix FITNESS FORCE_SSA VERBOSE
```

- `FITNESS` is a numeric value which indicates the type of fitness to be calculated
    - 0: no fitness;
    - 1: classic point-2-point fitness;
    - 2: classic point-2-point fitness redirected to stdout.
- `FORCE_SSA` forces the use of Gillespie's direct method
- `VERBOSE` enables the verbose mode