StabilityChecker

Miriam Leon, Chris Barnes

August 13, 2015

Contents

Introduction

`StabilityChecker` is a python package that can be utilised to identify regions of parameter space capable of producing a desired stability while handling uncertainty in biochemical rate constants and initial conditions. It can be used to design new systems of desired stability or study the stability of existing systems. `StabilityChecker` takes advantage of Sequential Monte Carlo methods, thus is able to harvest the power of parallelisation and speed up the algorithm. `StabilityChecker` is suitable for use if GPU access is available.

Overview

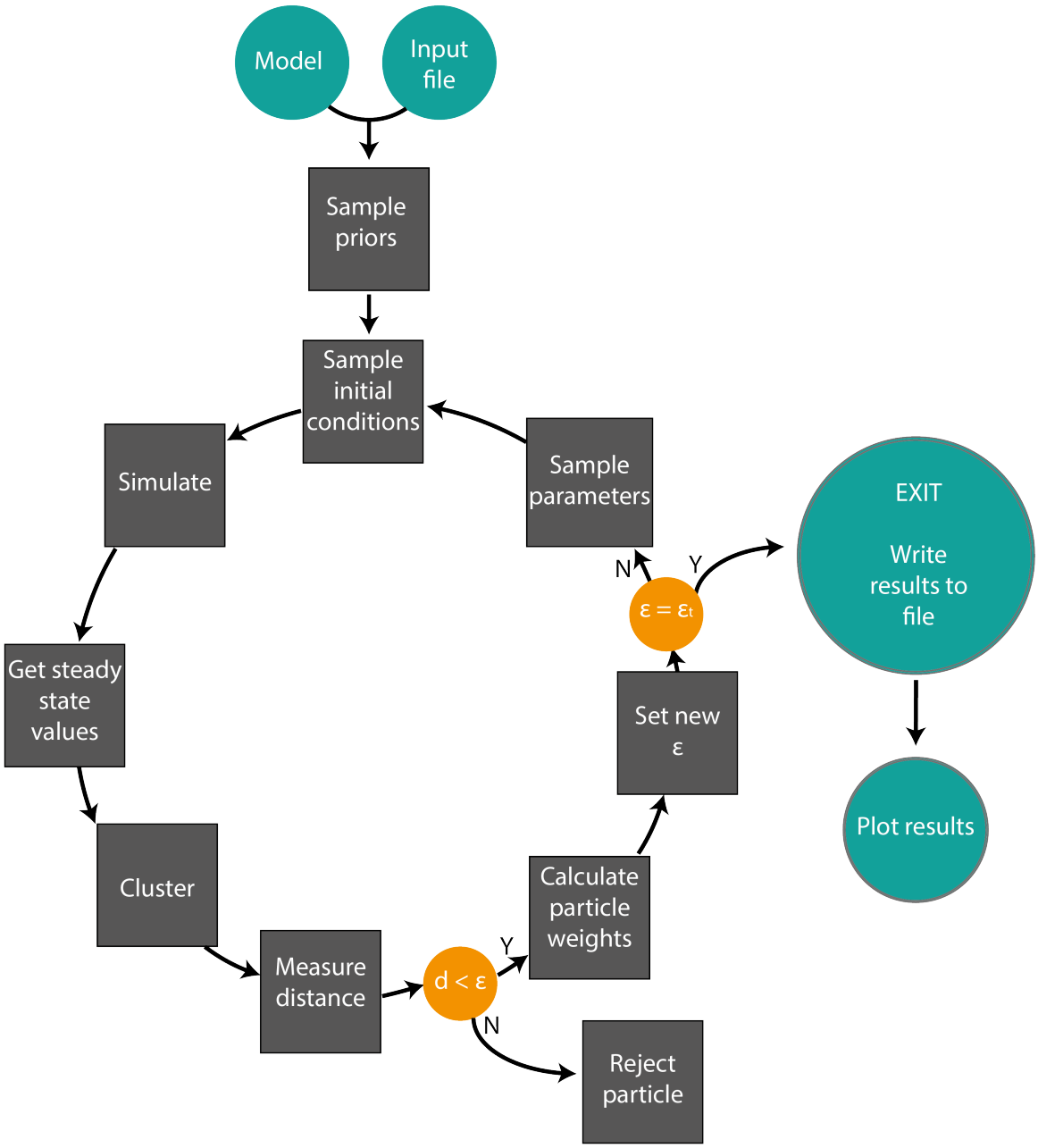

Figure 2.1: Flowchart of the algorithm used in StabilityChecker. ϵ stands for the threshold for the distance from the desired value. The user provides the model and input files as inputs and the algorithm outputs the posterior distribution, the parameter space capable of exhibiting the desired stability.

Installation

StabilityChecker has been developed to work on a Linux operating system, with GPU support.

The following dependencies are essential for the successful implementation of the package:

- numpy
- logging
- cuda-sim
- libsbml
- R
- ggplot2
- pycuda

```
$ cd StabilityChecker
$ python setup.py install
```

This will copy the module `StabilityChecker` into the `lib/pythonXX/site-packages` directory, where `XX` corresponds to the python version that was used. An `exe=<dir>` variable should also be added to the `run.sh` script pointing the script to the right package installation. Alternatively the path to the module can be added to the top of the `run.sh` file without the need for installation. The path to the cuda sim module must be added to the top of the `run.sh` file.

```
$ export PATH=<dir>:$PATH
```

Using the package

4.1 The working directory

The working directory must contain the following files. Each one is described in detail in the section following.

input_file.xml The user input file

run.sh The shell script that will initialise the scripts

plot_posterior.R The R code to plot the results

model file This can take two formats:

model.cu Cuda file of the model that is required to have this name

SBML model This can have any name, as long as this is provided in the `input_file.xml`. `Cuda-sim` will create the `model.cu` file using this SBML model.

4.2 The input file

The `input.xml` file contains all the necessary information that is not contained in the model itself. This file must follow the specific format showed in the 'Examples' folder. No whitespaces are to be included between the tags.

4.2.1 Required arguments

epsilons The acceptable distance from the desired value.

- **epsilon_t** Total variance in the data
  - **start** The initial cut-off value for the distance from the desired values.
  - **end** The final cut-off value for the distance from the desired values.
- **epsilon_vcl** Within-cluster variance
  - **start** The initial cut-off value for the distance from the desired values.
  - **end** The final cut-off value for the distance from the desired values.
- **eps_cl** Number of clusters
  - **start** The initial cut-off value for the distance from the desired values.
  - **end** The final cut-off value for the distance from the desired values.

Desired final values The desired values the algorithm will converge to.

- **number_of_clusters** The number of clusters in the data,describing its stability.
- **total_variance** The total variance in the data.
- **cluster_variance** The within-cluster variance in the data.

particles The number of accepted parameter sets required to proceed to the next iteration.

number_to_sample Number of parameter sets to sample from for iteration. This needs to be greater or equal to the particles.

initial_conditions_samples Number of initial condition sets to sample for each parameter set at each iteration.

alpha A value that describes the step size for the incremental reduction of the epsilon at each iteration. This should be set to a small value (a = 0.1) to speed up the algorithm.

times The time points for the simulation are given as a whitespace delimited list.

species_numb_to_fit Which two species numbers to be fit by the algorithm. The should be whitespace delimited.

stoch_determ The type of simulation to be used. The two options are stochastic or deterministic.

model_file Whether the model provided is in SBML or cuda format. The two options are sbml and cuda.

sbml_name The name of the SBML file that will be provided. Note that if a cuda file is provided, that must be named model.cu.

parameters The parameters included in the model.

- **item** One item corresponds to one parameter. The number and order of the items must strictly correspond to those of the parameters in the model provided.
  - **name** The name of the parameter. Note that the parameters are read by order and not name from the model. The name provided here is used in the plotting module.
  - **distribution** From what distribution to sample values from. The two options are constant and uniform.
  - **start** The lower limit of the distribution to sample from.

end The upper limit of the distribution to sample from. Note that if the distribution is constant this value is not read.

initial_conditions The species included in the model.

item One item corresponds to one species. The number and order of the items must strictly correspond to those of the species in the model provided.

name The name of the species. Note that the species are read by order and not name from the model.

distribution From what distribution to sample values from. The two options are constant and uniform. Strictly two species must be set to uniform and the rest constant.

start The lower limit of the distribution to sample from.

end The upper limit of the distribution to sample from. Note that if the distribution is constant this value is not read.

4.2.2 Output

The outputs from running `StabilityChecker` are saved in the `results_txt_files` folder. This folder contains two files and one folder per population. The two files are the following:

- `Parameter_values_final` Each line contains the values of the parameters in the order set in the `input_file.xml` file and each line corresponds to an accepted particle in the final accepted population.
- `Parameter_weights_final` Contains the weights of each particle (parameter set) in the final accepted population.

Each population folder contains the following files:

- `data_PopulationN.txt` Each line contains the values of the parameters in the order set in the `input_file.xml` file and each line corresponds to an accepted particle.
- `data_WeightsN.txt` Contains the weights of each particle (parameter set).
- One `set_resultXX` per parameter set. Each line contains the steady state value of each species, in order specified in the `input_file.xml` file, species_numb_to_fit. Each line corresponds to one initial condition set.

The plot of the posterior is saved in the `posterior.pdf` file in the working directory. The plot is interpreted as follows:

- The marginal distributions of each parameter are found on the diagonal
- The pairwise joint distributions are found along the sides. The location of each pairwise joint distribution is determined by which pair of parameters are compared.

4.2.3 Running StabilityChecker

To run `StabilityChecker`, and once the `input_file.xml` file is completed, the `run.sh` file must be executed. The progress of the algorithm can be followed in the `my_abc_scan.log` file. Once StabilityChecker is finished the posterior can be seen in the `posterior.pdf` file.
The phase plots of the populations can be visualised by plotting the contents of the `set_resultXX` file for each parameter set.

Examples

5.1 The simple genetic toggle switch using ODEs

5.1.1 Background

In this example, the simple toggle switch model developed by Gardner *et al.* (2000) was used. This model consists of two mutually repressing transcription factors and is defined by the following ODEs:

$$\frac{du}{dt} = \frac{a_1}{1+v^\beta} - u$$
$$\frac{dv}{dt} = \frac{a_2}{1+u^\gamma} - v$$

Where a_1 and a_2 are the effective rates of synthesis of repressors 1 and 2 respectively. u is the concentration of repressor 1 and v the concentration of repressor 2. β is the cooperativity of repression of promoter 1 and γ of repressor 2. This model is capable of bistability, and `StabilityChecker` will be used to find the parameter space in which this behaviour is found.

5.1.2 Setting up

Model

This example does not use an SBML model but instead provides `StabilityChecker` with the cuda file directly. This is the `model.cu` file.

Input file

The input file is set up as shown in the examples folder. As this model only contains two species, u and v, these are selected for the fit as well as initial condition scan.

run file

The pythonpath must be set to point to the directory in which cuda sim was installed.

```
$export PATH=<dir>:$PATH
```

In addition, the `exe=<dir>` variable must be set to the directory in which `StabilityChecker` is installed.

5.1.3 Running StabilityChecker

The working directory must contain the following:

- The `model.cu` file
- The completed `input.xml` file
- The customised `run.sh` file
- The `plot_posterior.R` file

The `run.sh` file is then executed:

```
./run.sh
```

The progress of the algorithm can be followed in the `my_abc_scan.log` file.

5.1.4 Results

The posterior is found in the `posterior.pdf` file. It is the one shown in figure 5.1. The plots on the diagonal represent the marginal distributions for the values of each parameter that were found in the final population, thus the ones that can produce bistable behaviour. The pairwise joint distributions are found on the side plots.

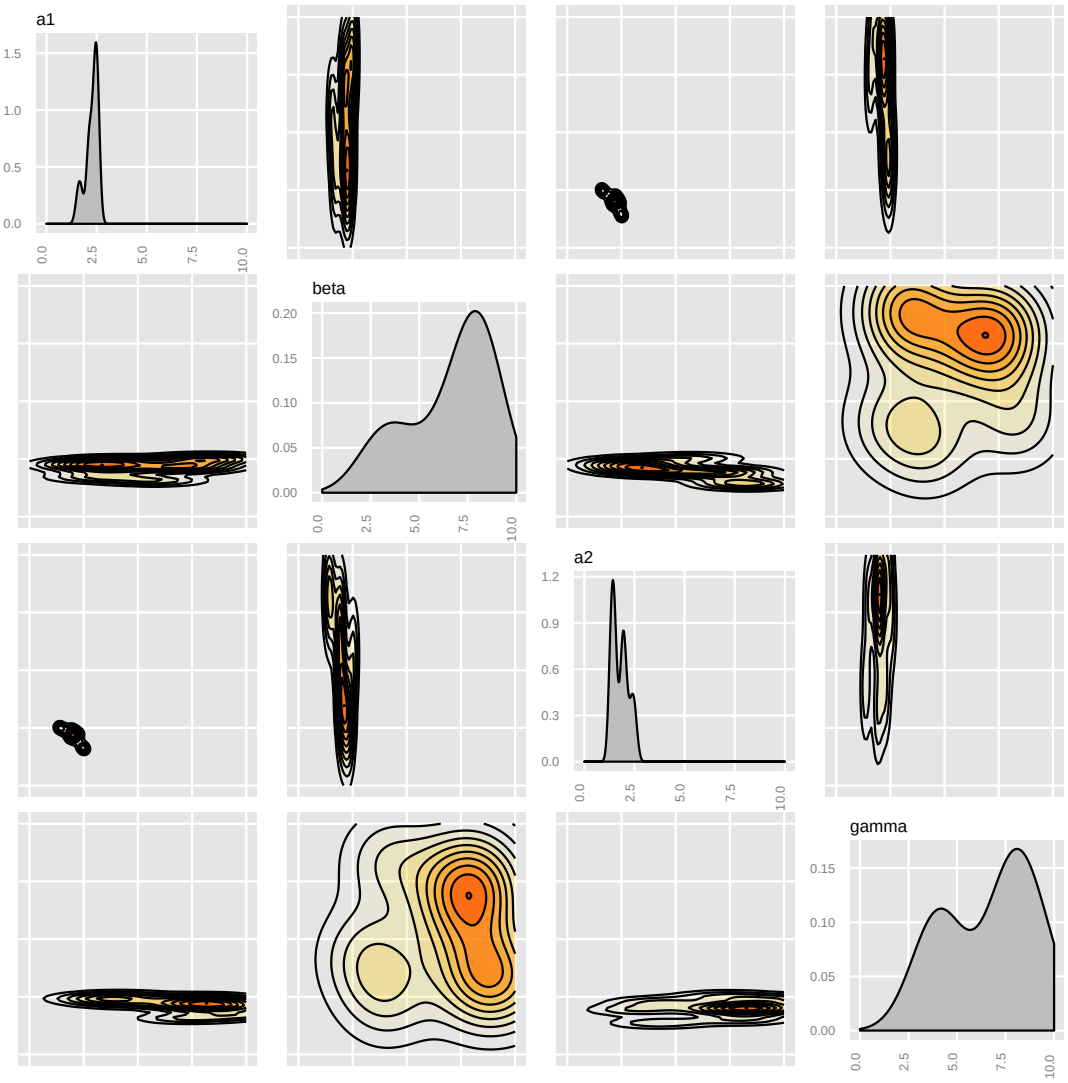

Figure 5.1: The posterior distribution of the Gardner toggle switch produced by StabilityChecker. The marginal distributions of each parameter are found on the diagonal and pairwise joint distributions along the sides.

Bibliography

Gardner, Timothy S, Cantor, Charles R, & Collins, James J. 2000. Construction of a genetic toggle switch in Escherichia coli. *Nature*.