README

These files are open-source for computing cell signaling pathways and are made up of C/C++.

Compiling

In Linux and macOS, the *exec.sh* can compile the *main.cpp* executables, as well as *Boolean.cpp*, *functions.cpp*, and *initial\_data.cpp* with

chmod +x *exec.sh*

./*exec.sh*

According to the user computer system, *-std=c++11*in *exec.sh* can be changeable.

If the user wants to calculate variation by a specific parameter,

use *cat filename.txt | xargs -L 1 ./a.out* in *exec.sh.* (Some examples are written in *exec.sh.*)

Parameters

* *Np*: the number of proteins in the signaling pathway.
* *Nw*: the number of weights in the signaling pathway.
* *rb*: the parameter of the Waller-Kraft operator.
* *p*: the number of inhibitors.
* *taneps*: the parameter of the hyper tangent function.
* *dt*: time shift.
* *iter*: calculation time.
* cut: a specific interval of simulation time.
* *simul\_time*: simulation time.
* *cvinterval*: coefficient variation period on simulation time.
* *spinterval*: smooth spline period on calculation time.

The upstream (), downstream (), and stochastic rate () imply the control parameters of the protein’s activation, and they are indicated in the main script (*main.cpp*).

Boolean operator

When the user calculates the Boolean value of each protein, use the *get\_min\_max* function in *Boolean.cpp*. Our open source considers a maximum of five weights for each protein, and they have to be inserted sequentially. For the upstream value, the user can use the *get\_input* function and the protein and its boolean value has to be inserted crosswise. For example, if one protein is activated by two proteins, the upstream value is calculated:

*input = get\_input(2, protein1, boolean\_value1, protein2, boolean\_value2)*

Finally, using the Boolean value and input, the activation of protein can be computed by *first\_Boolean\_gate*, *T*, and *protein\_calc*functions.

Calculation Functions

Single simulation: Each protein's amplitude, smooth spline, and trajectory are calculated via a single simulation (*simul\_time*=1).

* Amplitude is estimated as the difference between the maximum and minimum of protein’s activation from a specific start point to calculation time without stochastic effects (*amplitude* in *functions.cpp*).
* Smooth spline is estimated at a constant period (*splinterval*).
* A trajectory can be imaged by the output file.

Multiple simulations: Each protein’s root mean square error (RMSE), coefficient variation (CV), and bliss index are calculated via multiple simulations (*simul\_time*=1000).

* The RMSE (*RMSE* in *functions.cpp*) is to find suitable parameters set based on a specific parameter.
* CV (*data\_sort*, *interCV*, *CV* in *functions.cpp*) is estimated as the mean (*MEAN* in *functions.cpp*) and standard deviation (*VAR* in *functions.cpp*) of multiple simulations to compare relative variation at a constant period (*cvlinterval*).
* Since the bliss index is calculated via inhibitors, the user has to choose the number of inhibitors (*p*). A protein is set as zero at a specific simulation time and estimates the mean and standard deviation for before and after inhibition while multiple simulations. The reduction rate (*reduction* in *functions.cpp*) is calculated using the estimated mean and standard deviation and makes a file for each inhibitor. The user has to be careful about the order of inhibitors. This open-source considers the order from single to combination inhibitor. After making a file of inhibition, the bliss index can be calculated using the file (*bliss* in *functions.cpp*).

Main code

Protein has to be set individually because of different weights, upstream, and downstream.

* The proteins including a negative feedback loop indicate the *phi* (=-1)index in *main.cpp*.
* The *inhibi\_point* in *main.cpp* can be changeable by the user (It is set as half of the calculation time in open-source).
* The *num* implies the order of inhibition for each inhibitor.
* Each inhibition has to be compiled individually to make a file.
* When the user wants to do multiple stimulations, set 1 to the receptor protein, in contrast, set zero to the receptor protein for single stimulation.
* *Non-negative feedback* in *main.cpp* represents the case without a negative feedback loop.