User's Guide

for BNG2Smoldyn 2013

Weiren Cui

CAS-MPG Partner Institute for Computational Biology, Shanghai, China
Contact: cuiweiren@picb.ac.cn

Overview

BNG2Smoldyn is a software that combines BioNetGen[1] and Smoldyn[2] together in order to do the spatial and temporal simulation with rule-based modeling. BioNetGen defines a biochemical reaction system with executable rules and generates the corresponding reaction network. Smoldyn is a first principle spatial and temporal tool for simulating biochemical reaction systems. BNG2Smoldyn provides a user interface for rule-based spatio-temporal modeling and simulation.

1. Software requirement

The package of BNG2Smoldyn requires the version of BNGL-2.2.2-stable (http:// bionetgen.googlecode.com/ files/BioNetGen-2.2.2-stable.zip) and Smoldyn-2.28 (http:// www.smoldyn.org/ download.html). We recommend users to obtain the most current versions of both.

1.1 Installation of BNGL

In the version of BNGL-2.2.2-stable, the simulation engine was pre-compiled for all the operating systems (Windows, Linux and Mac OS). The Perl package is required to run the BNGL and therefore must be installed.

1.2 Installation of Smoldyn

Several extra packages are needed for the full function of Smoldyn. Detailed guidance for installing Smoldyn is in the "Smoldyn_doc2.pdf" under the home folder of Smoldyn.

For Windows users, files under the folder "dll" should be copied to the directory "C:\Windows\System32". (for the users of Windows 64 bits, the directory should be "C:\Windows\System64").

For Linux users (e.g., Ubuntu), C++ compiler 'g++', Python head file 'python-dev', OpenGL glut library 'freeglut3-dev' and libtiff library 'libtiff4-dev' are required. At the command line, use "sudo apt-get install g++" to install g++ (similar for other packages). After that, enter the 'cmake' folder and typing "cmake ..", then "make".

For Mac OS users, an Intel-based machine running Mac OS 10.6 or later is required for the latest version of Smoldyn. After the corresponding package is downloaded

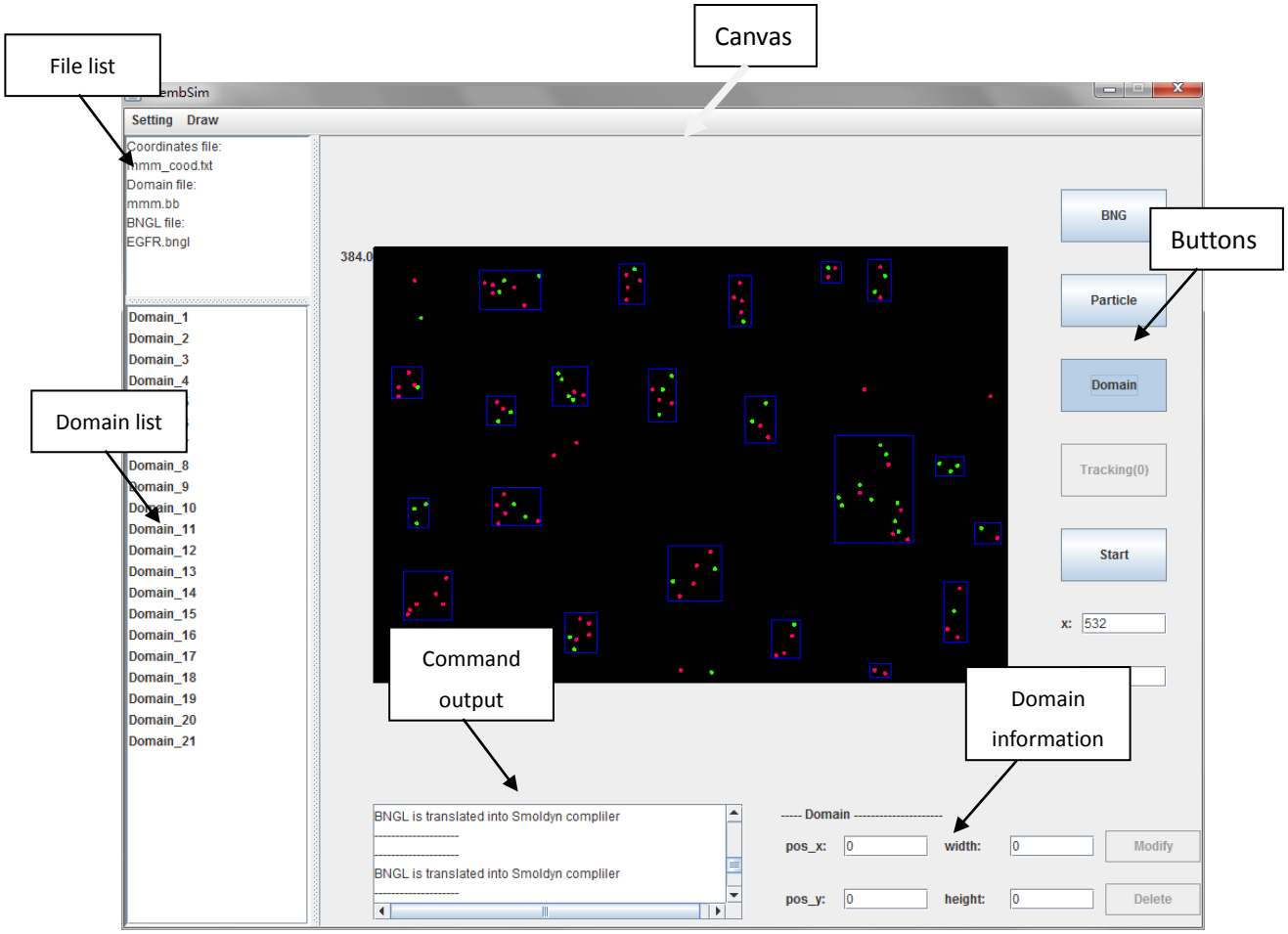
and extracted, type "./install.sh" or "sudo ./install.sh" (for the administrator) to install Smoldyn. The executable file 'smoldyn' is under the folder of "./bin/". To install old versions of Smoldyn, please refer to the 'README' file in the package for guidance.

Smoldyn requires the package of OpenGL for 3D visualization. However, a user can opt out without this feature. Open the file 'CMakeList.txt', go to line 136 and change "option(OPTION_USE_OPENGL "Build with OpenGL support" ON)" into "option(OPTION_USE_OPENGL "Build with OpenGL support" OFF)". Then enter the folder 'cmake' and type "cmake .." and then "make".

1.3 Installation of BNG2Smoldyn

The latest package of BNG2Smoldyn can be downloaded from our wiki page: http://www.picb.ac.cn/stabwiki/images/8/8c/B2s-1.42.zip. BNG2Smoldyn requires Java, Perl and JRE-1.6.

2. Usage



BNG2Smoldyn has a graphical user interface to define particles, domains and simulation properties. Once the BNG2Smoldyn is correctly installed, the program can be executed directly by double clicks on the 'b2s.jar' package, or using command

"java -jar b2s.jar" under the package folder. The main window will be shown as the figure below after the program started.

Canvas: the membrane area in 2D space, where the particles distribute. One can add particles and draw domains directly onto this canvas.

Buttons: five control buttons, 'BNG', 'Particles', 'Domain', 'Tracking' and 'Start', to define (in an top-down order) different parameters required for a simulation. Clicking a button opens a pop-up dialog for parameter input. The text fields (x:, and y:) below these buttons display the current mouse coordinate on the canvas.

File list: listing the file names of a BNGL file, a file of initial particle coordinates and a file of domains, which are imported from pop-up dialogs related to the corresponding control buttons.

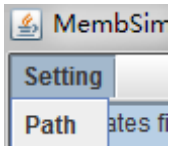
Domain list: list of existing domains on the canvas, named as "Domain_#", '#' is the index of that domain. Each domain in the list can be selected on the canvas and the selected domain is colored orange while the other domains are blue.

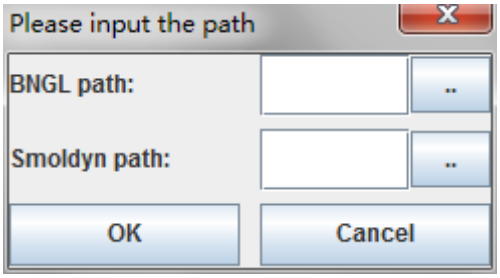
Domain information: showing the dimension of the selected domain rectangle, including the x-y coordinate of the top-left corner, the width and the height of the domain. These parameters can be modified via the "Modify" button. The selected domain can be deleted using the “Delete”.

Command output: showing the command-line outputs from both BioNetGen and Smoldyn. These outputs include the reaction network size by BioNetGen and particle binding radius by Smoldyn and are informative for users,.

2.1 Setting the path for BNGL and Smoldyn

For the first time use of BNG2Smoldyn, it is necessary to tell BNG2Smoldyn where the BNGL and Smoldyn installed. Set this parameter at the menu "Setting" - "Path" as illustrated as follows.





The path can be directly inputted or selected from browsing a folder selection by

clicking on the button "..". For Linux users, the compiled Smoldyn file is under the folder of 'cmake'.

2.2 BioNetGen file

We use a model of dimerization-induced EGFR phosphorylation as an example to demonstrate the usage of the package. The system is defined in the file 'EGFR.bngl' as shown below. Users can refer to the BNG user guide for grammar details.

```
begin model

begin parameters
    NA          6.02e23     # Avogadro number (molecues/mol)
    f           1           # Fraction of the cell to simulate
    Vo          f*1.0e-10   # Extracellular volume=1/cell_density (L)
    EGFR_tot    116
    EGF_tot     50
    kp0         0.01
    km0         0.99
    kp1         9.0e7/(NA*Vo) # ligand-monomer binding
    km1         0.0           # ligand-monomer dissociation
    kp2         0.00542           # 0:2 dimer on
    km2         1.24              # 0:2 dimer off
    kp3         0.00291           # 1:2 dimer on
    km3         0.738             # 1:2 dimer off
    kp4         0.00845           # 2:2 dimer on
    km4         0.273             # 2:2 dimer off
    kp5         0.0733          #dimer phosphorylation
    km5         0.13            #dimer dephosphorylation
    kp6         0.0733
    km6         0.13
    km7         0.13
end parameters

begin molecule types
    EGFR(ecto~t~e,L,Y1068~U~P)
    EGF(R)
end molecule types

begin seed species
    EGFR(ecto~t,L,Y1068~U)   EGFR_tot
    EGF(R!1).EGFR(ecto~e,L!1,Y1068~U)   EGF_tot
end seed species

begin reaction rules
```

```
    # conformation change of EGFR
    EGFR(ecto~t,Y1068~U) -> EGFR(ecto~e,Y1068~U)  kp0
    EGFR(ecto~e,L,Y1068~U) -> EGFR(ecto~t,L,Y1068~U) km0

    # dimerazation of EGFR
   EGFR(ecto~e,L) + EGFR(ecto~e,L) <-> EGFR(ecto~e!1,L).EGFR(ecto~e!1,L)
kp2, km2
    EGF(R!1).EGFR(ecto~e,L!1) + EGFR(ecto~e,L) <->
EGF(R!1).EGFR(ecto~e!2,L!1).EGFR(ecto~e!2,L) kp3, km3
    EGF(R!1).EGFR(ecto~e,L!1) + EGF(R!2).EGFR(ecto~e,L!2) <->
EGF(R!1).EGFR(ecto~e!3,L!1).EGF(R!2).EGFR(ecto~e!3,L!2) kp4, km4

    # Phosphorylation and dephosphorylation
    EGFR(ecto~e!1,Y1068~U).EGFR(ecto~e!1,Y1068~U) <->
EGFR(ecto~e!1,Y1068~P).EGFR(ecto~e!1,Y1068~U) kp5, km5
    EGFR(ecto~e!1,Y1068~P).EGFR(ecto~e!1,Y1068~U) ->
EGFR(ecto~e!1,Y1068~P).EGFR(ecto~e!1,Y1068~P) kp6
    EGFR(ecto~e!1,Y1068~P).EGFR(ecto~e!1,Y1068~P) ->
EGFR(ecto~e!1,Y1068~U).EGFR(ecto~e!1,Y1068~U) km6
    EGFR(ecto~e,Y1068~P) -> EGFR(ecto~e,Y1068~U) km7
end reaction rules

begin observables
    Molecules  RR  EGFR(ecto~e!1,Y1068~U).EGFR(ecto~e!1,Y1068~U)
    Molecules  RRP EGFR(ecto~e!1,Y1068~U).EGFR(ecto~e!1,Y1068~P)
    Molecules  RPRP    EGFR(ecto~e!1,Y1068~P).EGFR(ecto~e!1,Y1068~P)
    Molecules  EGFR_total EGFR()
end observables

end model

#actions
generate_network({overwrite=>1})
```

The above BioNetGen file contains six sections.

Parameters. This section defines the model parameters, including initial conditions and reaction rate constants. Specifically, we note that the second order rate constant for particle association is in the unit of per (number per cell) per second in BioNetGen, as in this software, the actual simulation is done by Smoldyn, so this unit should be converted into volume per second, which is the standard unit in Smoldyn.

Molecule type. This section defines molecular entities in the model. These entities include binding sites, interaction domains and phosphor-sites, etc. For example, the

EGFR molecule is define by the following syntax

```
EGFR(ecto~t~e,L,Y1068~U~P)
```

The molecule is specified with three intra-molecular sites (or subdomains). Component 'ecto', indicating the ectodomain of the receptor, with two states denoted as 't' or 'e', corresponding to the “tethered” or “extended” conformations. The symbol '~' is a delimiter used to separate the state enumeration of the site. Component 'L' specifies the ligand binding site. Component 'Y1068' specifies the tyrosine residue at position 1068, with two states, 'U', unphosphorylated state, or 'P', phosphorylated state.

Seed species. This section defines the initial species for the BNG to generate a reaction network.

Reaction rules. The section specifies all reaction rules. Rules for a reversible reaction can be written in a single line. In the EGFR example, three sets of rules are defined.

1. Conformation change of EGFR monomer: T <-> E, T is the “tethered” monomer, while E is the “extended” monomer.
2. Dimerization and dissociation with two “extended” monomers: E + E <-> E.E, E.E is dimer.
3. Phosphorylation and dephosphorylation of EGFR dimer: E.E <-> E(p).E(p). E(p). E(p) can be dimer with either one or two phosphor group.

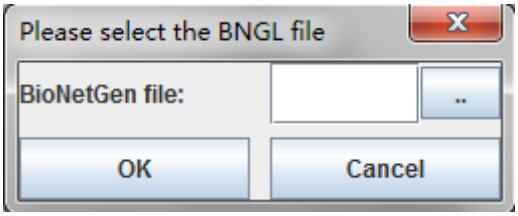
These rules implicate 39 reactions.

Observables: This section defines the patterns of chemical species serving as simulation outputs.

Actions: This section defines the actions. In this example, only 'generate_network' is needed for later Smoldyn simulation.

2.3 Read in the BioNetGen file

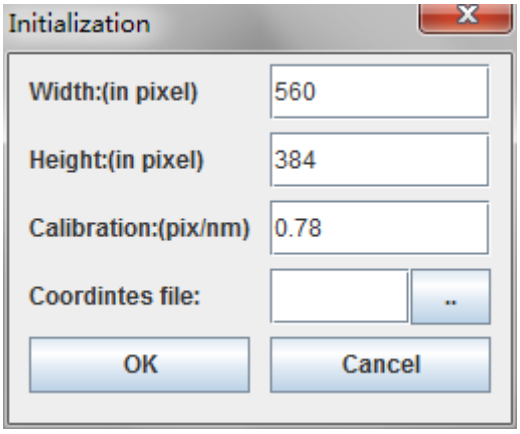
Users can load the BNG file by clicking the "BNG" button. A pop-up dialog will show for importing the BNG file.



In the EGFR example, we select the './example/EGFR.bngl'. After loading the BNG file, the program will execute the BNG file and generate the reaction network. The output for the BioNetGen results will appear on the 'command output', such as the number of iterations for generating the network.

2.4 Initialization of particles

After the reaction network is generated, the button of "Particle" will become enabled. The pop-up dialog of initialization of particles (coordinates) will be shown after clicking the “Particle” button.



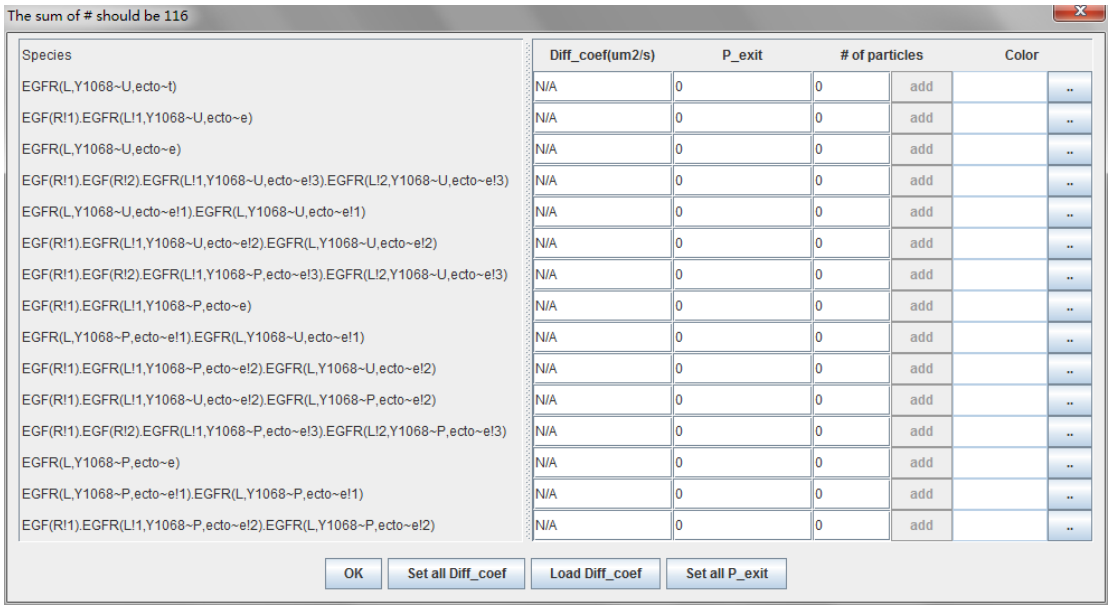
Width: dimension of the canvas along the x axis (in the unit of pixel).

Height: dimension of the canvas along the y axis (in the unit of pixel).

Calibration: the correspondence between a pixel and its real physical length.

Coordinates file: Input for the path of the coordinates file, which is in the format of three columns of 'index', 'x' and 'y'. As an alternative, one can also manually input particles onto the main canvas.

In the EGFR example, we input the coordinates file as "./example/mmm_cood.txt". After initialization, a pop-up dialog with parameters of all species appears as shown below.



Species: list of all the species in the model, expressed in the BioNetGen syntax.

Diff_coef: diffusion coefficients for species. A diffusion coefficient must be specified to each species, either manually into the text field or by importing from a file.

P_exit: exit probability (within the interval [0,1]) of the domains for different species.

# of particles: assign the initial copy number for each species. The total number of initial species should be consistent with the total number implicated in the coordinate file. The "add" button is disabled for the first time of the display of this dialog. When it's enabled, by clicking the corresponding buttons, one can add the corresponding species by drawing a rectangle and inputting the number of particles into the rectangle.

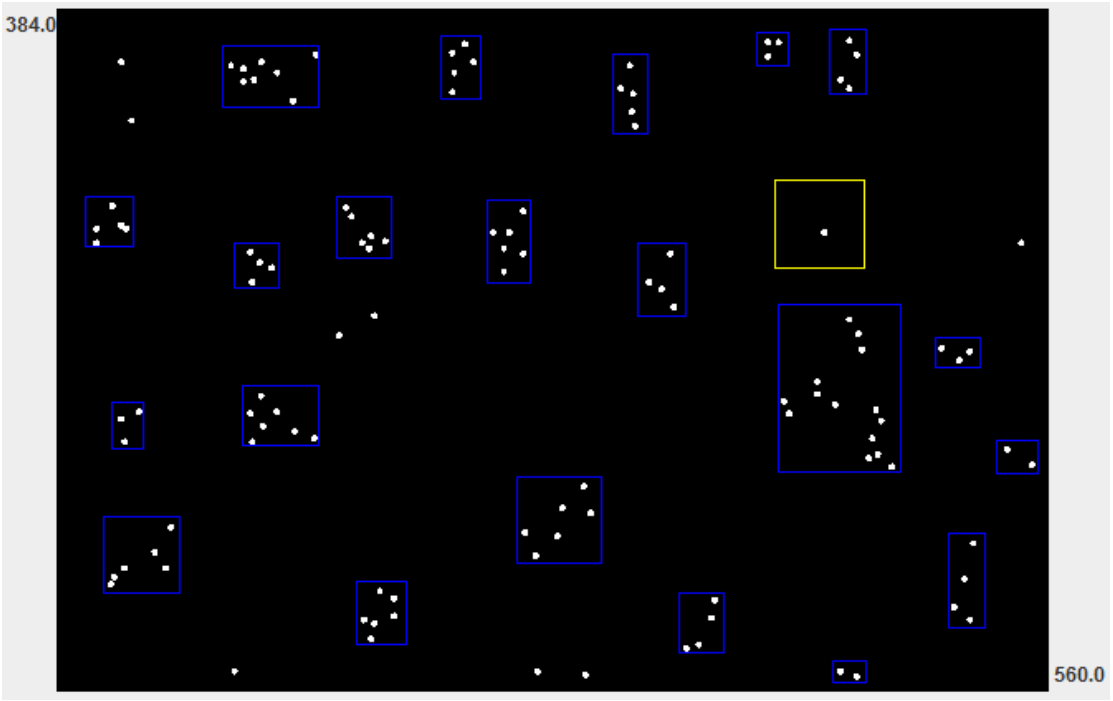
Set all Diff_coef: defines an identical diffusion coefficient for all species.

Load Diff_coef: imports diffusion coefficients from a file. The file contains two columns with species ID in the first column and the numerical value its diffusion coefficient in the second.

Set all P_exit: defines an identical exit probability for all species.

2.5 Definition of domains

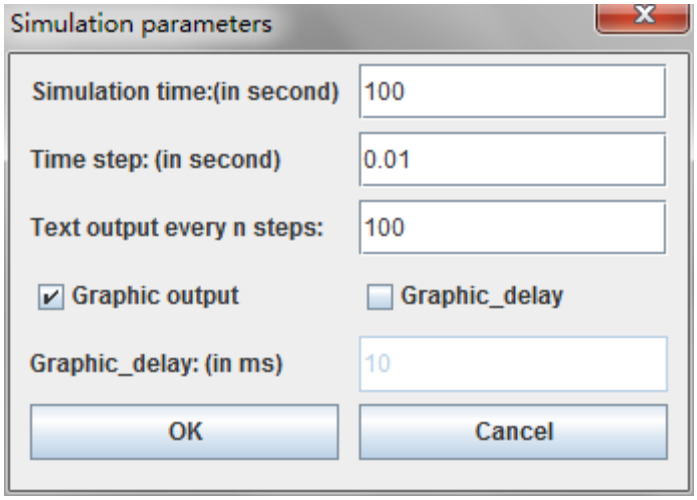
A domain is a small region that confines the diffusion of particles. Domains can be defined by pressing the "domain" button with two options: read the domain information from a file or draw domains based on assumptions. If the first option is selected, choose the correct domain file with four columns corresponding to x, y, width and height. If the 2nd option is selected, make sure the domain button is selected and then first click on the panel at the top-left corner, move the mouse to the bottom right, and click again, the drawing box is in the color of yellow, and after the second click, it will turn into blue as the same as other existed domains.



Modification and deletion of the selected domain can be done as described in Section 2. In this example, the domains are input from the file "./Example/mmm.bb".

2.6 Simulation parameters

After all the preparing steps mentioned above, a dialog with simulation parameters is shown by clicking the "Start" button.



Simulation time: the total time for the simulation.

Time step: the time interval between two iterations in Smoldyn. one of the important parameter that is related with the calculation of binding radius.

Text output every n steps: Definition of the round for text output in order to reduce the waste of time for redundant I/O output.

Graphic output: Output the simulation results with a canvas showing all the particles

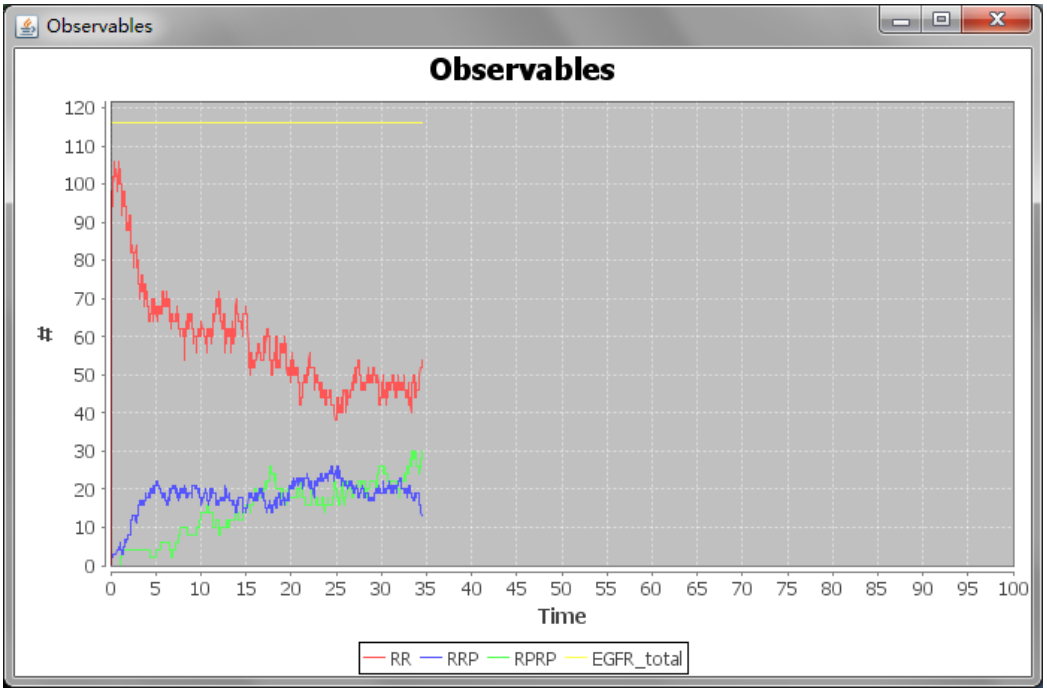
in corresponding colors and its hops if this option is selected.

Graphic_delay: A delay will be added between the frames of graphic output if it is selected, and the actual delay is in the unit of ms inputted in the text field.

Attention! If the option of 'Graphic output' is selected, The pop-up canvas of Smoldyn will stay in the ready-to-start state with all the defined particles staying in the initial positions. After the check of the initial state, the simulation can be started by pressing the 'space' key with focus on that canvas. At the same time, you can also pause/continue the simulation by the 'space' key. If you are so sure about your initial condition and want to get a faster results without graphical output, the simulation will auto started without the option of 'Graphic output' selected.

2.7 Output

A real time plot of observables will display beside the top right of the main window, showing all the observables defined in the BioNetGen file.



The entire text output is in the file named 'molc.txt' under the home folder, with all the species listed for the further analysis.

As the achievement of this function is reading the output file of Smoldyn instead of accessing the memory, This output plot is not so 'real time' sometimes due to the writing buffer of C language. For example, if you run the example of 'EGFR.bngl' in the example folder and use the default setting of 100s simulation time, 0.01s time step and output every 100 steps, the whole simulation will finished in 2 minutes without any curves on the output windows before you quit Smoldyn. That's the buffer issue mentioned above, the output file 'molc.txt' should be 0 bytes before you quit Smoldyn, and after you press the capical 'Q' key on the canvas to quit Smoldyn,

this 'molc.txt' should contain some results now and the output curve will appear. To overcome this problem, especially for the simulation that is quite fast, the parameter of 'Text output every n step' should be decreased so that the writing buffer can be full filled during the simulation.

2.8 Single molecular tracking

As Smoldyn is a species-based modeling tool, different from an agent-based approach, it does not explicitly track information of individual molecules in time. However, it is possible to use BioNetGen to label single molecules for tracking, at a cost of generating a larger reaction network for Smoldyn.

Consider a simple reversible binding reaction

A + B <-> C ,

in which molecule A binds to molecule B to form a complex C. Besides the monomeric A molecule, complex C also contains molecule A. If we want to track a single molecule A, the program must monitor all chemical species that might potentially contain the A molecule for tracking, which is similar to scenario where one tracks single molecules in an experiment using a label. If A' is used to indicate the labeled A, and A' functions exact the same as A, and we observe the detailed particles with the 'prime', such as A' and C', we can achieve the same tracking results as the experiments. As a consequence, more reactions should be added into the reaction network, for example:

A' + B <-> C'

In this simple case, only one more reaction is needed to track A. However, an original reaction network might have more reactions involves molecule A. For example, complex C can further react into other species D, E, and so on. We then need to track these D', E' to trace the reaction pathway of the tracked A molecule. Systematically, it requires a traversal of the whole reaction network to label all of these species and reactions that involve molecule A. BioNetGen can help to modify the network with tracking information.

The BioNetGen file without tracking looks as follows.

```
begin molecule types
    A(b)
    B(a)
end molecule types

begin seed species
    A(b)            20
    B(a)            20
end seed species

begin reaction rules
```

```
    A(b) + B(a) <-> A(b!1).B(a!1) kp0, km0
end reaction rules
```

In comparison, the BioNetGen file with tracking looks as follows

```
begin molecule types
    A(b,track~0~1)
    B(a)
end molecule types

begin seed species
    A(b,track~0)    19
    A(b,track~1)     1
    B(a)            20
end seed species

begin reaction rules
    A(b) + B(a) <-> A(b!1).B(a!1) kp0, km0
end reaction rules

begin observables
    Molecules   track_1 A(track~1)
end observables
```

As shown in the second file, a few modifications of the BNG code are needed to achieve single-molecule tracking.

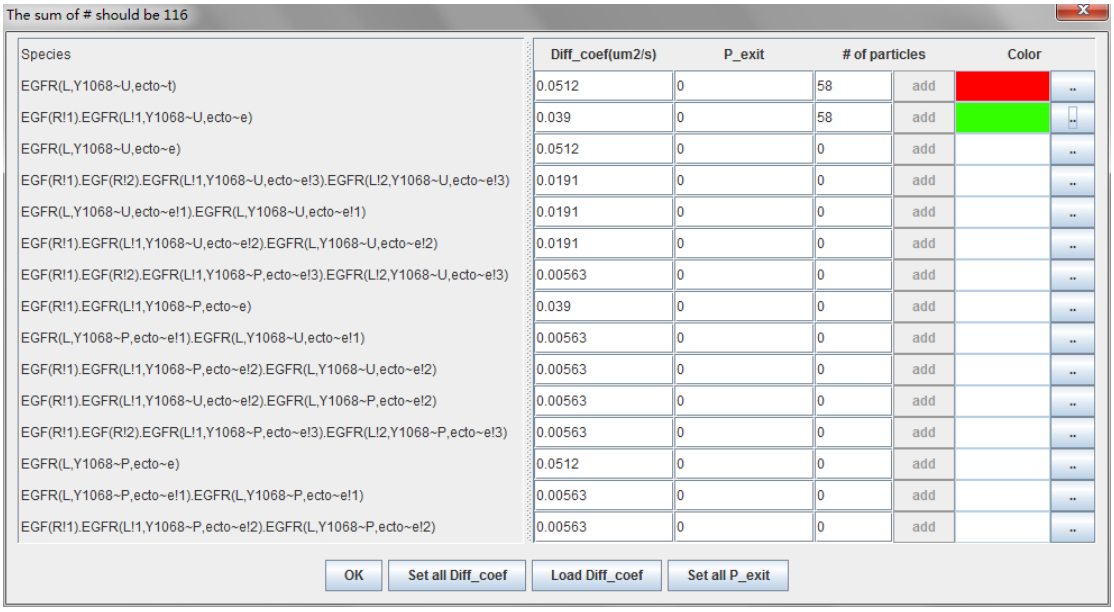
1) Add tracking component into the molecules intended for tracking. For example, to track an A molecule, a new component "track~0~1" should be added to molecule type A. To track multiple A molecules, "track~0~1~2~3" should be added to the molecule type A. As another example, to track two A molecules and 1 B, "track~0~1~2" to A and "track~0~3" to B should be added. The state of track~0 is the unlabeled state, whereas the other nonzero states serve distinctive labels.

2) To generate the reaction network with tracking information, the tracked molecules should be added into the seed species. In the example, A(b, track~1) is added in the above example.

3) Tracking observables is also needed to add into the observables part of BNG file, to indicating all possible species of the label component could be. By following the syntax of "Molecules track_# Particle (track~#)", where '#' is the index of tracked particles, and 'Particle' are the name of tracked particles.

After modifying the BNG file, we can read in the BNG file as shown in section 2.2. The

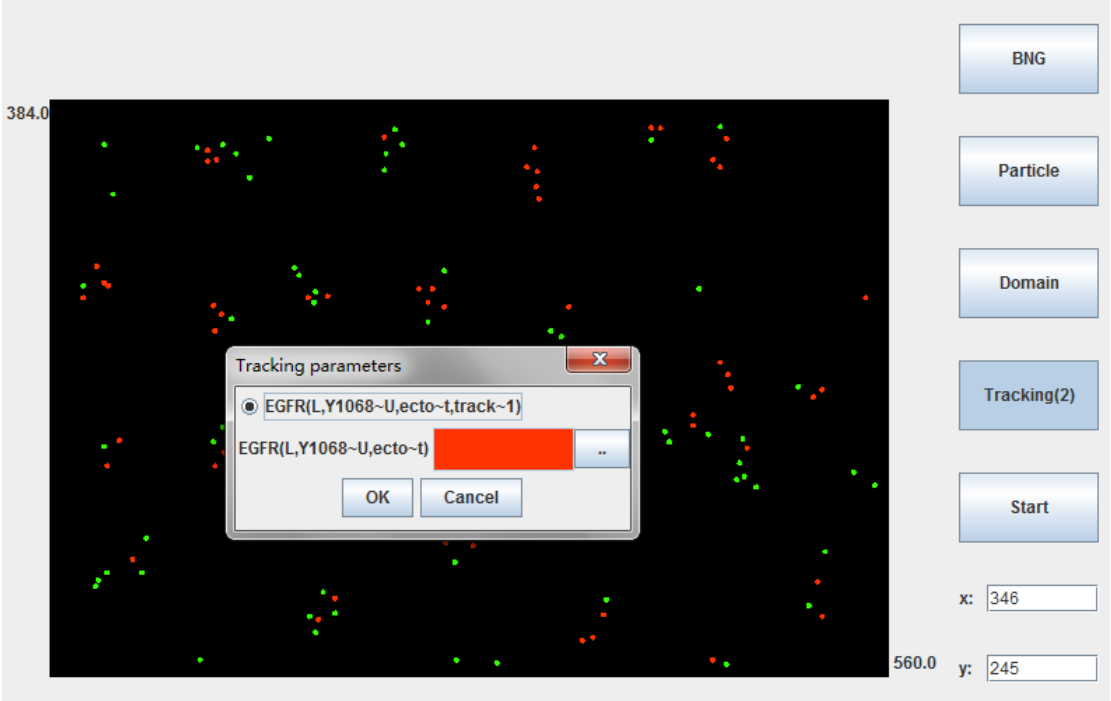
"Tracking" button will be enabled after setting up the parameters for particles. There will also be a number indicating how many particles are tracked in the BNG file. If we read in the tracking BNG file ".\Example\EGFR_track2.bngl", the coordinate file ".\Example\mmm_cood.txt" and the diffusion coefficient file ".\Example\difc.txt", assign 58 species_1 and 58 species_2 as the figure below:



A half of the particles read in are randomly assigned into species_1 which is the unliganded monomer. The other species is liganded monomer. As the BNG file tracks two particles, the first is an unliganded monomer and the second is a liganded monomer. The following steps show how to select initial particles for tracking particle on the canvas.

1) Press down the "Tracking" button;

2) Click on one particle, following the tracking index, the dialog with selected species and color is shown as the figure below:



If the selected species is consistent with the species containing the molecule type for tracking, the possible labeled species is listed, otherwise, the "OK" button will become disabled. After setting up the parameters, the tracked particle appears to be larger than other untracked particles.

A simulation stores the tracking information in a file named as " list_track#.txt", '#' is the index of tracked particle. In each file, the 1st column is the number of iteration. The 2nd column is the state of the tracked particle corresponding to the species list. The 3rd column records the state of the tracked particle, which is usually 0 in the solution. The rest columns record the particle coordinates.

3. 3-D Simulation

The simulations on 2-D canvas with domain information such as the membrane proteins are well supported by current version of BNG2Smoldyn. As it is not intuitive to use to GUI to define the initial positions of particles in 3-D, the spatial definition interface was not developed yet. We can still use BNG2Smoldyn to create the reaction network using rule-based definition and give the value of diffusion coefficient in step 2.4. After adding the other spatial information into Smoldyn manually, we can achieve the 3-D simulation under command line.

In order to get the correct configuration of Smoldyn, the configuration file was separated into 5 files here: USER_DEFINED.txt, para_spe.txt, domain.txt, mol.txt and para_time.txt.

USER_DEFINED.txt: Including the entire species list and reaction list in the gamma of Smoldyn. the species were defined as s1, s2, t1, t2 for short. The corresponding representation of BioNetGen is in the file named 'species.list' and 'track.list'. This file name is user defined, as in the example of EGFR, this file will be 'EGFR.txt' corresponding to input file of 'EGFR.bngl'. For 3-D simulation, if species 's2' is surface

bound, it should be represent as s2(front) or s2(back) to indicate the side it bound. You can refer to the user guide of Smoldyn for the detail of the definition of surface reactions.

para_spe.txt: This file contains all the diffusion coefficient and the color for graphical display for each species in the species list.

domain.txt: Definition of the simulation volume, surfaces and domains. As the surfaces and domains here are all rectangles, you can follow the 'panel' function in Smoldyn to define proper domains both in 2-D and 3-D.

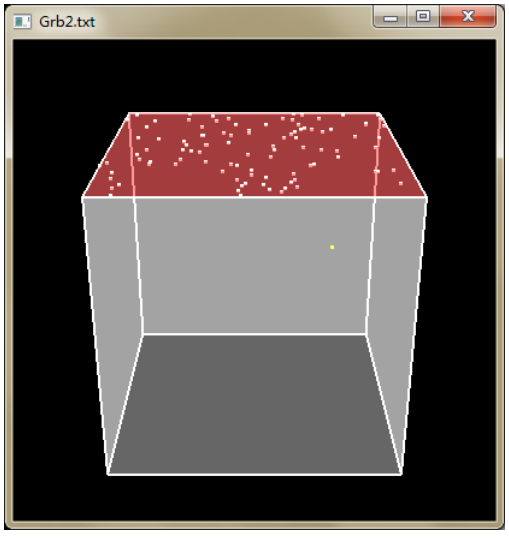
mol.txt: The initial condition for the particles. This file should also be manually defined for 3-D simulation. The function of 'mol' and 'surface_mol' are used to define the molecules in the 3-D space and 2-D surface.

para_time.txt: The information of simulation time, time step and the output setup for track information are recorded in this file. This can also be done in the BNG2Smoldyn by the step of 2.6.

With the 5 files defined in the correct way, 3-D simulation can be done. Some knowledge of Smoldyn is required for the file of domain.txt and mol.txt. Here is a simple toy model of p-tyr and SH2 rebinding in the fold of 'example/3D'.

The origin input file is 'Grb2.bngl'. There's only one reversible reaction that is the phosphor site of EGFR binds with the SH2 domain of Grb2 and the bond breaks. After reading in this BNG file, it was translated into the file 'Grbt.txt'. The diffusion coefficients of the EGFR, Grb2 and the EGFR-Grb2 complex are 0.0512μ m^2/s, 15μ m^2/s and 0.05μ m^2/s separately (which is defined in step 2.4 and stored in the file of para_spe.txt). In the file of 'domain.txt', the simulation volume was defined as 0.5*0.5*0.5μ m^3, the top and bottom surfaces are reflective and the other four are periodical, and the top surface are defined as 'membrane'. The simulation was initiated as the file of mol.txt with 100 s1 on the membrane (white) and only 1 t1 in the 3-D volume (yellow). and followed the simulation time and time step in the file of para_time.txt.

Simulation can be started by running the command of 'smoldyn Grb2.txt' if Smoldyn is included in the environment path. Otherwise, just call the full path of Smoldyn.



The full extension of BNG2Smldyn for 3-D simulation will be further developed.

References:

1. M. L. Blinov, J. R. Faeder, B. Goldstein and W. S. Hlavacek, *Bioinformatics*, 2004, **20**, 3289-3291.
2. S. S. Andrews and D. Bray, *Phys Biol*, 2004, **1**, 137-151.