NetLand: a software for quantitative modeling and visualization of Waddington's epigenetic landscape

Jing Guo[1,2] and Jie Zheng[1,3,4*]

[1]School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798, Singapore, [2]Bioinformatics Institute, Agency for Science, Technology, and Research (A*STAR), Singapore 138671, Singapore, [3]Genome Institute of Singapore, A*STAR, Singapore 138672, Singapore and [4]Complexity Institute, Nanyang Technological University, Singapore 637723, Singapore

Email: zhengjie@ntu.edu.sg

URL: http://netland-ntu.github.io/NetLand/

Last updated: 4 Apr, 2016

Contents

1. Introduction

Waddington's epigenetic landscape is a visual metaphor for cellular dynamics in embryonic development, proposed by Conrad Hal Waddington (1905 - 1975) in 1957 [1]. In this picture, the cells are marbles rolling down a hill along trajectories towards valleys representing cell fates. Despite its unifying power and unfading popularity, the original concept of Waddington's epigenetic landscape was only a qualitative metaphor that lacks rigorous interpretation of mechanisms. Nonetheless, it has inspired mathematical ideas in theoretical biology [2] [3]. Recently, quantitative computational models of Waddington's landscape have been proposed [4] [5] [6] [7]. Typically, these methods use concepts and mathematical models from Physics, as analogues for cellular dynamics. However, practical applications of these methods need address several limitations. First, there is a lack of user-friendly software tools to calculate and display the landscape. Second, most of the methods are still restricted to very small networks (e.g. the mutual inhibiting 2-gene toggle switch). Third, it is still difficult to visualize the 3-dimensional landscape for a regulatory network with more than two genes. Although a 52-gene network has been recently used to plot a potential landscape [8], their method either selects only two marker genes for the coordinates of state space, or requires the landscape to have only two attractors. The NetLand software introduced here can fill these gaps, so that the quantitative models of Waddington's epigenetic landscape can be widely used by researchers of diverse backgrounds.

NetLand is a software tool designed for modeling, simulations and visualization of gene regulatory networks and their corresponding Waddington's epigenetic landscapes. Users can import models of gene regulatory network from a file or database (e.g. BioModels), or manually edit their own network models. Then, NetLand will automatically encode differential equations for the kinetics of transcriptional regulation. The forms of equations and parameter values can be edited by the users. The network model will be used to simulate the dynamics of cell state transitions, which will be plotted as time-course trajectories. After that, NetLand will calculate the quasi-potential landscape based on efficient numerical solutions of the kinetic equations. Our tests have shown that NetLand can calculate landscape for up to 80 genes on a regular desktop. To display the 3-D landscape for a network of more than two genes, NetLand allows users to either choose two marker genes, or use a dimension reduction method called GPDM (Gaussian process dynamical model) [9] [10]. Therefore, NetLand can provide a global picture of cellular dynamics for a user-specified gene regulatory network. Although we designed the NetLand software originally for modeling stem cell fate, it can be also used to study other cellular phenotypes, such as cancer cell death, cellular ageing, etc.

The source code of NetLand was written in Java, and thus it can run under Microsoft Windows, Linux and Mac OS. Its graphical user interface (GUI) allows for interactive visualization. This user manual provides instructions about how to install, launch and use the NetLand software. It also includes running time assessment and an example of using NetLand for the study of stem cell differentiation and reprogramming.

2. External Libraries

A number of external open-source libraries are bundled with the NetLand program. The external library dependencies are listed in the following table.

Table 1. The list of external libraries.

Package	Description	License	Version
GeneNetWeaver (GNW)	NetLand adopted the network interface from GNW.	MIT license	3.1
libSDE	libSDE is an Java library for numerical integration of Stochastic Differential Equations (SDEs).	MIT license	1.0.3
JMathPlot	Jmathplot is a light java library designed to allow easy plotting of 2D and 3D datas in a java panel. It is used to plot the time courses.	BSD License	1.0.1
JLatexMath	A Java API to render LaTeX Project logo. It is used to render mathematical equations to user-friendly format.	GNU General Public License (GPL)	1.0.3
Dizzy	Dizzy is a chemical kinetics stochastic simulation software package. It is used to perform stochastic simulation using Gillespie algorithm.	GNU Library or "Lesser" General Public License (LGPL)	1.11.4
BeautyEye	Look and Feel for swing	Apache License 2.0	3.7
Jzy3d	Jzy3d is an open source java library that allows to easily drawing 3d scientific data.	The BSD 3-Clause License	0.9.1
JOGL 2 and GlueGen	JOGL 2 is designed to provide hardware-supported 3D graphics to applications written in Java. It is required by Jzy3d. GlueGen is a tool which automatically generates the Java and JNI code necessary to call C libraries.	The New BSD 2-clause license	2.1.5
GPDM (C++)	Gaussian process code in C++ including some implementations of GP-LVM and IVM.		0.001

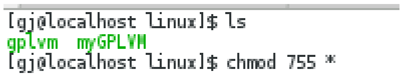
3. Installation

The Java Platform, Standard Edition Runtime Environment (JRE) is required to be installed (Java SE is available at http://java.com/en/download/inc/windows_upgrade_ie.jsp). The software NetLand can be downloaded for free at http://netland-ntu.github.io/NetLand/.

The current NetLand package includes the following files.

- Two shell scripts, "runNetLand.bat" and "runNetLand.sh", for launching NetLand in different operating systems.
- A jar file “main.jar”. It should be in the same directory as shell scripts.
- Two folders namely "toy models" and "saved results" which contain toy gene network files and pre-computed results respectively.
- A folder namely "lib". It contains the required libraries to launch NetLand.
- A folder namely "GPDM" which contains the executable files required for running "GPDM" program in NetLand. Make sure you have the executive permissions to the executable files. To check the permission of a file, the command is "*ls -l filename*". Use "*chmod 705 filename*" to gain the executive permission.

  Example:



```
[gj@localhost linux]$ ls
gplvm  myGPLVM
[gj@localhost linux]$ chmod 755 *
```

Note that the scripts must be in the same directory as the “GPDM” folder and "lib" folder.

To run the “GPDM” program, DLL (Dynamic Link Library) of BLAS, LAPACK and GFORTRAN are required. The DLL files for Windows OS are in the GPDM/win folder. For Mac OS, please check if libblas.dylib, liblapack.dylib and libgfortran.dylib are under /usr/lib/. Otherwise users should install the libraries themselves.

The BLAS, LAPACK and GFORTRAN can be downloaded from

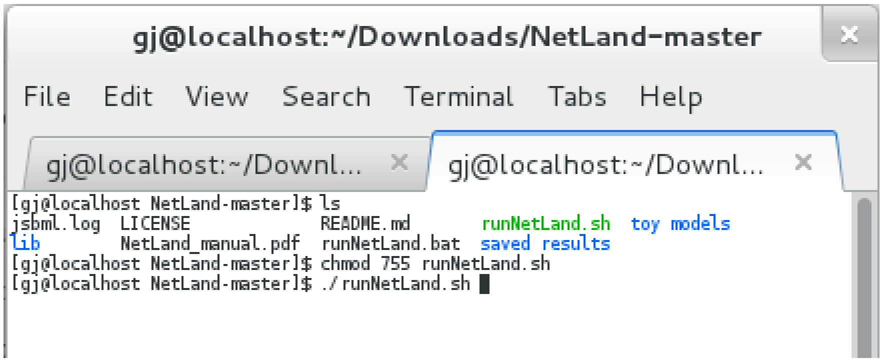
- LAPACK, see http://www.netlib.org/lapack/
- BLAS, see http://www.netlib.org/blas/
- GFORTRAN, see https://gcc.gnu.org/wiki/GFortranBinaries#MacOS

Then follow the install instructions in the package.
For Linux, users can use the command, e.g. yum, to install these packages.

We have tested NetLand under 64-bit version of Window 7/10, Linux Fedora 18 and Mac OS 10.8.

To launch NetLand:

- **In Windows, double click the “runNetLand.bat”.**
- **In Linux, run command “./runNetLand.sh” under a terminal.**



```
[gj@localhost NetLand-master]$ ls
jsbml.log  LICENSE             README.md       runNetLand.sh  toy models
lib        NetLand_manual.pdf  runNetLand.bat  saved results
[gj@localhost NetLand-master]$ chmod 755 runNetLand.sh
[gj@localhost NetLand-master]$ ./runNetLand.sh
```

- **In OSX, run command "./runNetLand.sh" under a konsole.**

Make sure you have the executive permissions to the scripts. To check the permission, the command is "*ls –l filename*". Use "*chmod 705 filename*" to gain the executive permission. For Win users, please check if "java" is in your system path. If not, you can either add it to the system path or add the full path of "java.exe" to the script.

Example:

"C:\Program Files\Java\jre1.8.0_31\bin\java.exe" -classpath "lib*;main.jar" WindowGUI.NetLand

For OSX and Linux users, the script should be run under a terminal or konsole using the command "bash runNetLand.sh" or "./runNetLand.sh".

The estimated RAM requirement for this software is 2G. NetLand uses the default settings of minimum and maximum RAM in JVM. To increase the RAM usage, users can add "-Xms2048m -Xmx4096m" to the shell scripts.

Example: java **-Xms2048m -Xmx4096m** -classpath "lib*;main.jar " WindowGUI.NetLand

This command will allocate at least 2G RAM and at most 4G RAM to run NetLand.

4. File operation

4.1. Load files

NetLand focuses on the transcriptional regulations within gene regulation networks (GRN) which contribute to the main force of the change of genetic patterns. It can load network structure files (TSV, GML, DOT) or dynamic models (SBML). The network interface is adopted from GeneNetWeaver [11].

TSV is a file extension for a tab-delimited file used with spreadsheet software. TSV stands for Tab Separated Values. TSV files are used for raw data and can be imported into and exported from spreadsheet software. TSV files are essentially text files, and the raw data can be viewed by text editors, though they are often used when moving raw data between spreadsheets.
Example:

```
A	B	+
B	A	-
```

Graph Modeling Language (**GML**) is a hierarchical ASCII-based file format for describing graphs. It has been also named Graph Meta Language.
Example:

```
graph [
	comment "Two-gene network"
	directed 1
	id 1
	label "Hello, I am a graph"
	node [
		id 1
		label "Gene A"
		thisIsASampleAttribute 42
	]
	node [
		id 2
		label "Gene B"
		thisIsASampleAttribute 44
	]
	edge [
		source 1
		target 2
		label "Edge from Gene A to Gene B"
	]
]
```

DOT is a plain text graph description language. It is a simple way of describing graphs that both humans and computer programs can use.
Example:

```
graph ethane {
   A -- B [type=s];
}
```

The Systems Biology Markup Language (**SBML**) is a representation format, based on XML, for communicating and storing computational models of biological processes. SBML can represent many different classes of biological phenomena, including metabolic networks, cell signaling pathways, regulatory networks, infectious diseases, and many others. For more information of SBML, please refer to http://sbml.org/Main_Page. NetLand can process SBML files that describe transcriptional regulations.

4.2. Save networks

The dynamic models can be saved in SBML format. And the structure of the network can be saved in TSV format.

4.3. Load saved results

The simulation results, e.g. time-course data and constructed landscapes, and the dynamic model can be saved when performing the simulation. Users can load not only the results, but also the dynamic model.

5. Modification of the network

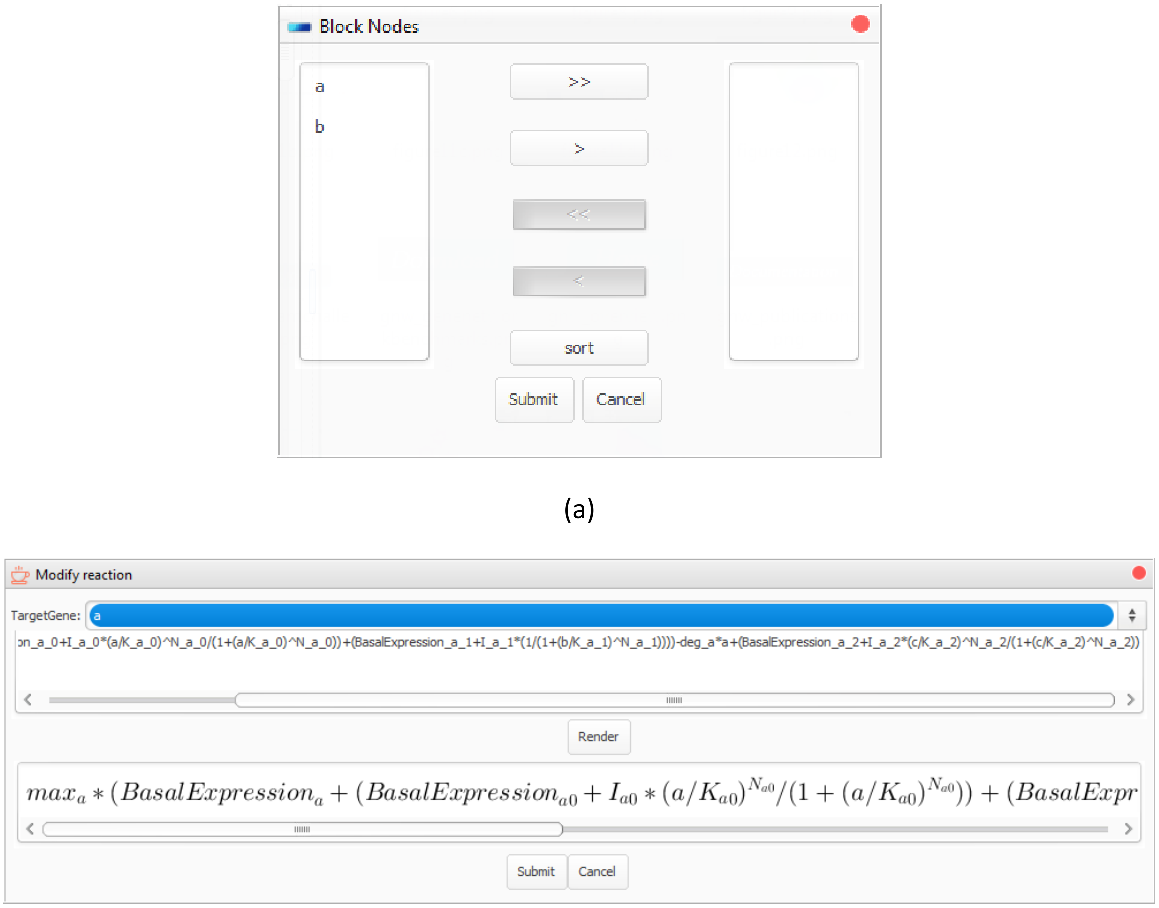
NetLand provides flexible ways to load and modify a transcriptional network. In this section, we describe how to use NetLand to create a network and modify the structure, e.g. deleting nodes, adding new nodes or edges.

5.1. Create a new network

By clicking the “New Network” button in the menu, a simple network with a demo node and self-loop is created. Then users can modify the network, e.g. adding new nodes or edges, defining regulatory equations.

5.2. Delete nodes

The nodes in a gene regulatory network can be blocked to simulate the gene knockout experiment. By clicking the “Block nodes” button in “Network” menu or the button in the top button bar, a dialog with a gene list will be pop-up (Figure 1a). After moving the blocked genes to the right list, the reactions of the remaining genes should be modified when they contain removed nodes as regulators.



(a)

TargetGene: a

```
on_a_0+I_a_0*(a/K_a_0)^N_a_0/(1+(a/K_a_0)^N_a_0))+(BasalExpression_a_1+I_a_1*(1/(1+(b/K_a_1)^N_a_1))))-deg_a*a+(BasalExpression_a_2+I_a_2*(c/K_a_2)^N_a_2/(1+(c/K_a_2)^N_a_2))
```

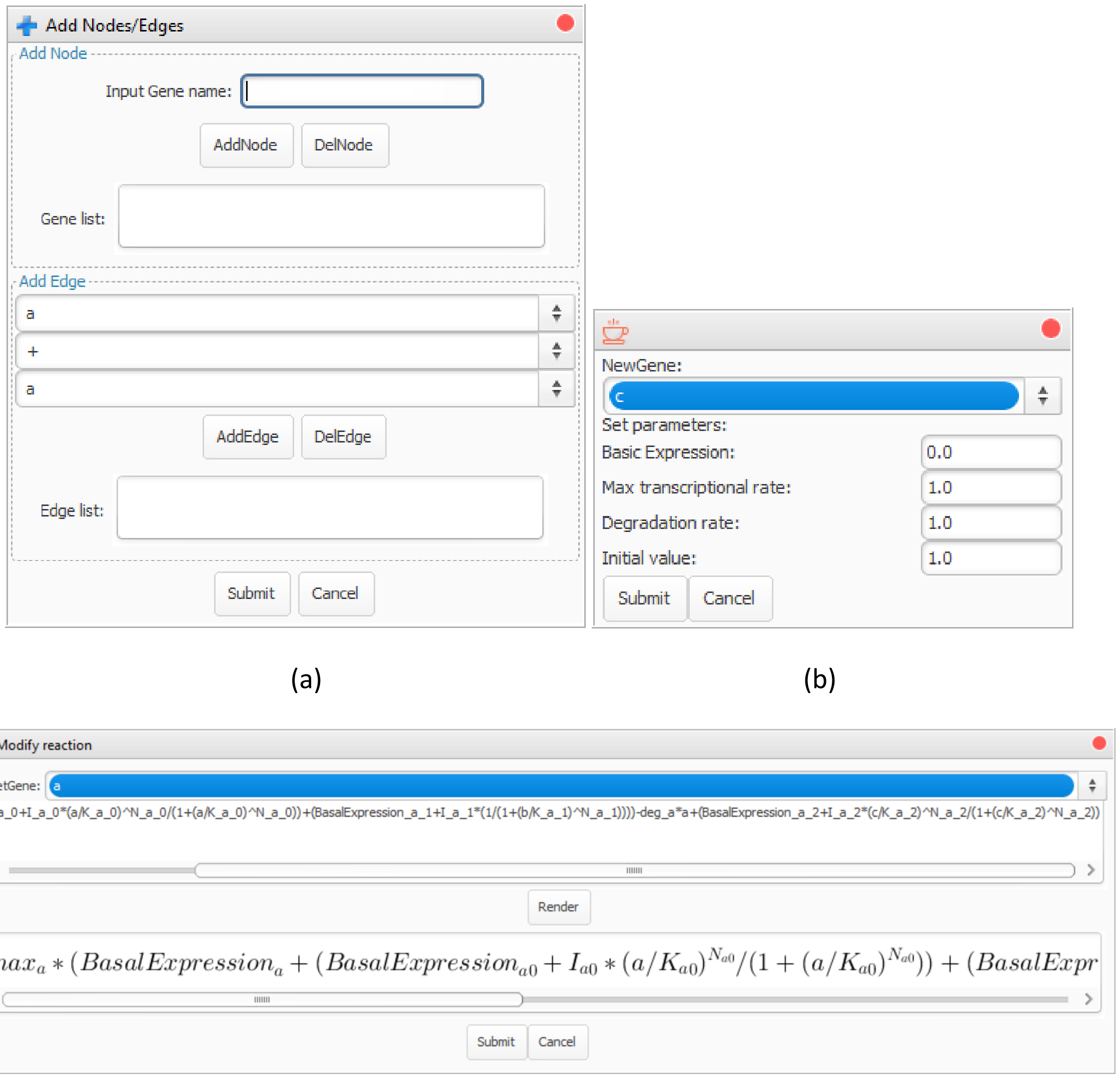
$max_a * (BasalExpression_a + (BasalExpression_{a0} + I_{a0} * (a/K_{a0})^{N_{a0}}/(1 + (a/K_{a0})^{N_{a0}})) + (BasalExpr$

(b)

Figure 1. The panel for blocking nodes.

5.3. Add nodes/Edges

Compared to blocking functions, users can add new genes and edges to build new regulatory interactions. By clicking the “Add nodes/edges” item in “Network” menu or the button in the top button bar, a dialog will be pop-up (Figure 2a). Once you add a new gene, a set of parameters should be verified including initial value, degradation rate *etc*. (Figure 2b). The new regulatory relations (edges) are encoded as Hill equations and added to the target genes’ original formulas (Figure 2c).



Input Gene name:

Gene list:

a

+

a

Edge list:

(a)

NewGene: c

Set parameters:

Basic Expression: 0.0

Max transcriptional rate: 1.0

Degradation rate: 1.0

Initial value: 1.0

(b)

TargetGene: a

```
on_a_0+I_a_0*(a/K_a_0)^N_a_0/(1+(a/K_a_0)^N_a_0))+(BasalExpression_a_1+I_a_1*(1/(1+(b/K_a_1)^N_a_1))))-deg_a*a+(BasalExpression_a_2+I_a_2*(c/K_a_2)^N_a_2/(1+(c/K_a_2)^N_a_2))
```

$$max_a * (BasalExpression_a + (BasalExpression_{a0} + I_{a0} * (a/K_{a0})^{N_{a0}}/(1 + (a/K_{a0})^{N_{a0}})) + (BasalExpr$$

(c)

Figure 2. The panel for adding nodes and edges.

6. Dynamic models

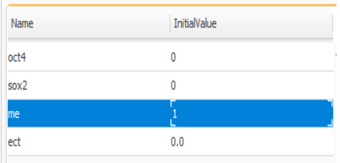
Species, reactions and parameters constitute three main components of a dynamic model. NetLand can automatically convert transcriptional regulations in a network into a dynamic model. During conversion of a transcriptional network to a dynamic model by manual editing or loading an SBML file, the initial values of genes and parameters in reactions are assigned by a default number or read from the SBML file (Figure 3a). The change of the concentration of each gene is encoded in the form of differential equations (Eq. 1).

$$\frac{d[\mathrm{Gene1}]}{dt} = K^{+}\varphi(\mathrm{Gene1}, \mathrm{Gene2}, \mathrm{Gene3} \ldots \mathrm{GeneN}) - K^{-}[\mathrm{Gene1}] + D \cdot \theta(t) \qquad (1)$$

where [Gene1] denotes the concentration of Gene1. K^{+} is the maximum transcriptional rate of Gene1, K^{-} is the degradation rate, $\varphi(\cdot)$ represents the cis-regulation of Gene1 by other nodes, $\theta(t)$ is the noise term, and D controls the strength of noises. For each gene, $\varphi(\cdot)$ is composed of two parts, i.e. an active component and an inhibitive component, denoting the regulation of activators and inhibitors by Hill equations (Eq. 2).

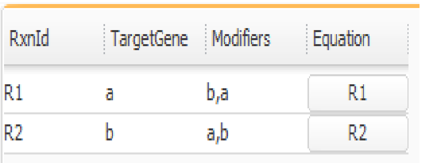
$$\Phi(\cdot) = a * \sum_{i=1}^{ma} \frac{\mathrm{Genei}^{na}}{\mathrm{Genei}^{na} + ka^{na}} + b * \sum_{i=1}^{mb} \frac{ki^{ni}}{\mathrm{Genei}^{n} + ki^{ni}} \qquad (2)$$

The reactions can be easily modified by clicking the button in the "Reaction" Panel (Figure 3bd). Similarly the values of genes and parameters can be directly modified in the panel view. Note that this conversion may not be appropriate to other types of networks, e.g. signaling networks, metabolic networks. For the analysis of networks or models other than GRN, we recommend that users upload an existing model saved in SBML format with reactions of each molecular. Or users can modify the reactions after loading the structure of the network.



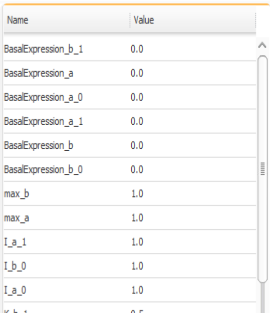
Name	InitialValue
oct4	0
sox2	0
me	1
ect	0.0

(a)



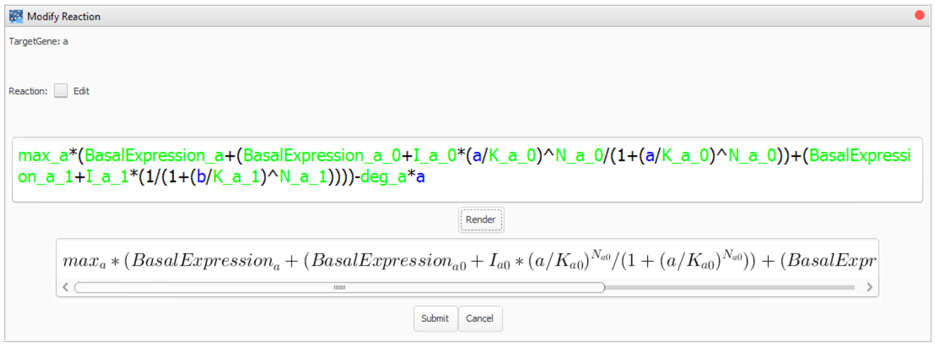
RxnId	TargetGene	Modifiers	Equation
R1	a	b,a	R1
R2	b	a,b	R2

(b)



Name	Value
BasalExpression_b_1	0.0
BasalExpression_a	0.0
BasalExpression_a_0	0.0
BasalExpression_a_1	0.0
BasalExpression_b	0.0
BasalExpression_b_0	0.0
max_b	1.0
max_a	1.0
I_a_1	1.0
I_b_0	1.0
I_a_0	1.0

(c)



TargetGene: a

Reaction: □ Edit

```
max_a*(BasalExpression_a+(BasalExpression_a_0+I_a_0*(a/K_a_0)^N_a_0/(1+(a/K_a_0)^N_a_0))+(BasalExpressi
on_a_1+I_a_1*(1/(1+(b/K_a_1)^N_a_1))))-deg_a*a
```

$$max_a * (BasalExpression_a + (BasalExpression_{a0} + I_{a0} * (a/K_{a0})^{N_{a0}}/(1 + (a/K_{a0})^{N_{a0}})) + (BasalExpr$$

(d)

Figure 3. The panels for species, reactions and parameters of the dynamic model.

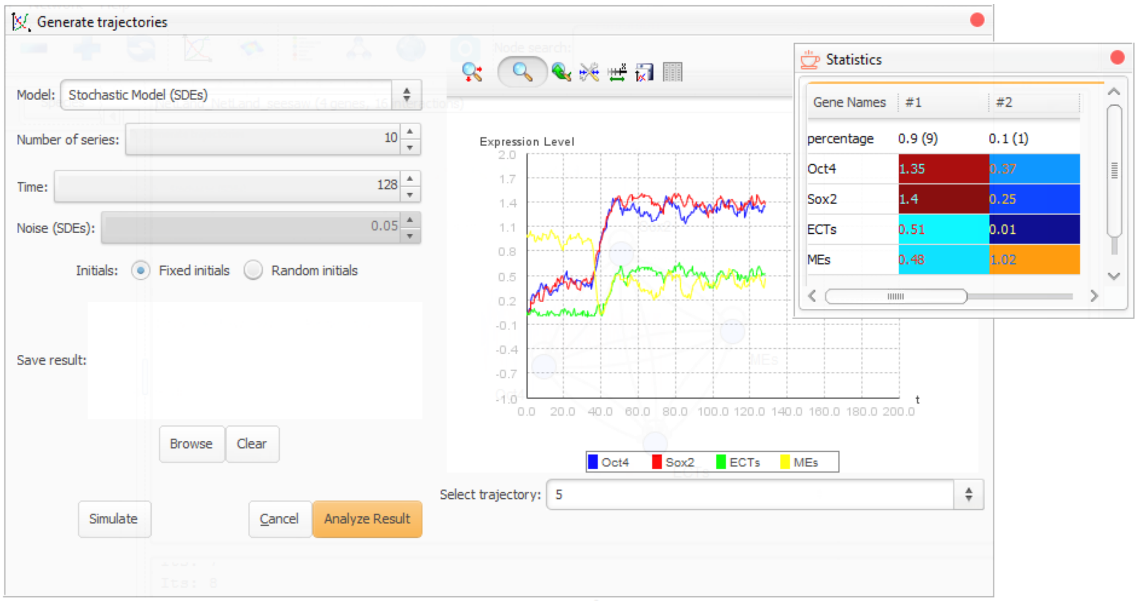
7. Simulation analysis

NetLand is designed to study GRN kinetics. Through *in silico* simulation, the dynamics of the GRN can be visualized in Waddington's epigenetic landscape.

7.1. Generation of trajectories

The transcriptional network can be encoded in deterministic models (ODEs) or stochastic models (SDEs) (Figure 4). The Runga-Kuta method (RKF45) is used to solve ODEs. Gillespie algorithm is used to simulate the stochastic processes [12]. SDEs are numerically solved by the Euler-Maruyama method.

In the stochastic model, the change of concentrations is described by Langevin equations (SDE). By control of the noise strength in SDEs, users can study the influence of noises on the transcriptional regulation. NetLand supports batch simulations by setting the number of time series. The symbol "Time" is the elapsed time of each simulation with the same time unit as the parameters and values of species. For the simulation, there are two choices for setting of the initial values of genes, i.e. fixed or random. The fixed initial values are defined in the species panel. For the random initial settings, users can define the range of random numbers as initial values. The simulated data can be retrieved by the function buttons on the top of the figure. The steady states of multiple trajectories can be analyzed by clicking the "Analyze result" button. It will show the steady states and percentages of trajectories that end at each steady state.



Model: Stochastic Model (SDEs)

Number of series: 10

Time: 128

Noise (SDEs): 0.05

Initials: ● Fixed initials ○ Random initials

Select trajectory: 5

Gene Names	#1	#2
percentage	0.9 (9)	0.1 (1)
Oct4	1.35	0.37
Sox2	1.4	0.25
ECTs	0.51	0.01
MEs	0.48	1.02

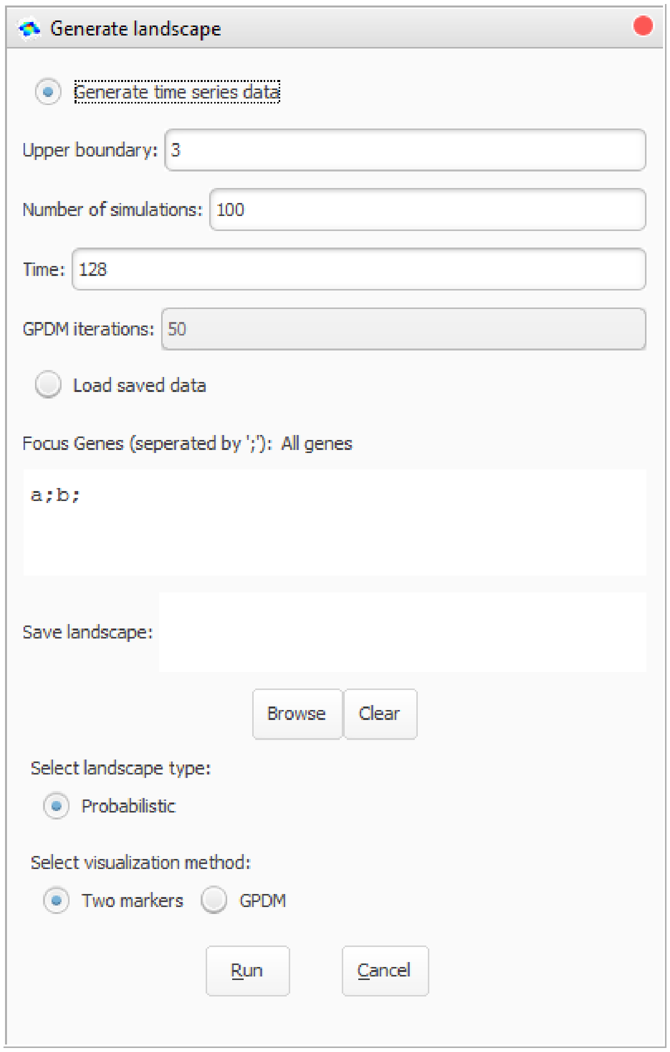
Figure 4. The panel for generation of trajectories.

7.2. Construction of landscapes

For a comprehensive understanding of the determination of cellular phenotypes (e.g. cell fate decision in differentiation and reprogramming), the Waddington's epigenetic landscape has been quantified to analyze and visualize the global dynamics of gene regulatory networks [7, 13, 14]. Based on the essential idea of Waddington's epigenetic landscape, changes of phenotypes are represented by transitions overcoming energy barriers between different attractors. Thus the quantified epigenetic landscape is the stage on which the play of cell fate decisions and cell type maturation is choreographed according to physical theories [15].

The quantitative modeling and visualization of Waddington's epigenetic landscape is implemented in NetLand. We adopted the self-consistent mean field approximation method in [8] to construct the probabilistic landscape, based on the notion that the probability of gene expression states determines the stability [13, 8]. It is assumed that the probabilities of cell states (vectors of gene expression levels) are independent and follow the Gaussian distribution. For a system without stable states in multiple simulations, NetLand cannot construct the landscape of the system. The steady state is a situation in which all state variables are constant in spite of ongoing processes that strive to change them. For example, a system with its variables exponentially increasing can never reach a steady state. Instead a warning message will be shown. Then users may consider modify the model in order to reach a stable system. And the current version of NetLand cannot construct the landscape of a system with stable oscillations.

To visualize the landscape of a network with more than 2 genes, NetLand provides two options. The first option is to select two marker genes whose expression levels are the x-axis and y-axis in the landscape. However, the two marker genes may not represent the global dynamics in many cases. To avoid the loss of information, NetLand offers the second option as follows. A probabilistic dimension reduction method, Gaussian process dynamical model (GPDM) is employed to project high dimensional gene expression time-series data to a low dimensional latent space with a dynamical model [10, 16, 17]. In this case, the x-axis and y-axis are two constructed components, representing the dynamics of the whole network. For both options, the z-axis is the quasi-potential U (Figure 5). The upper boundary defines the maximum gene expression value that controls the selection of initial states. For networks with large number of nodes, normally more iterations and executive time are required to get their stable states. The parameters are sensitive to the network's dynamics. Once a landscape is constructed, the set of trajectories will be stored. Users can load the pre-computed data to construct another landscape.



Generate landscape

● Generate time series data

Upper boundary: 3

Number of simulations: 100

Time: 128

GPDM iterations: 50

○ Load saved data

Focus Genes (seperated by ';'): All genes

```
a;b;
```

Save landscape:

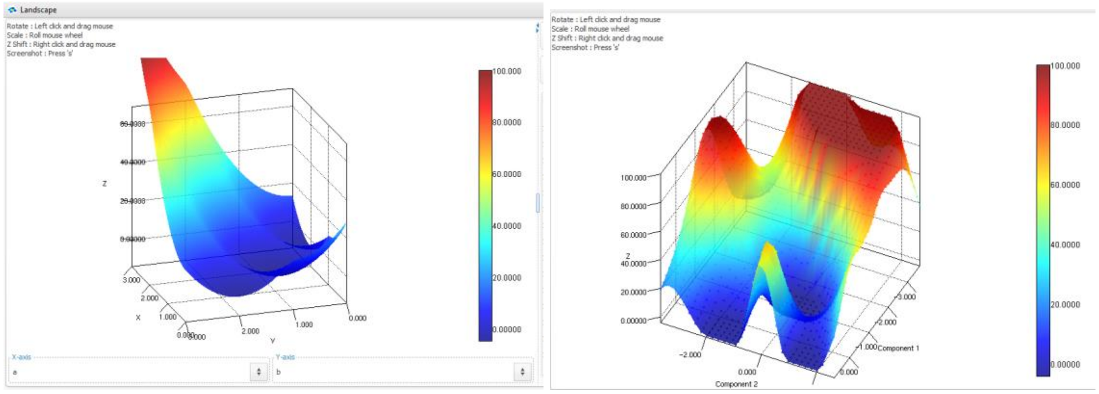
Select landscape type:

● Probabilistic

Select visualization method:

● Two markers ○ GPDM

Figure 5. The panel for the construction of landscape.



Landscape

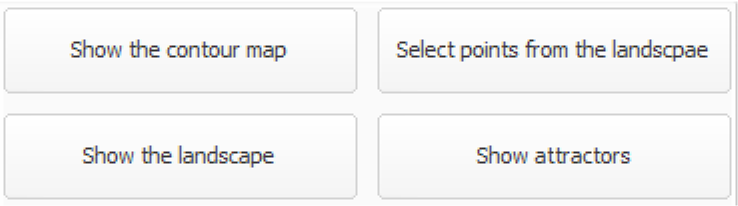
Rotate : Left click and drag mouse
Scale : Roll mouse wheel
Z Shift : Right click and drag mouse
Screenshot : Press 's'

X-axis
a

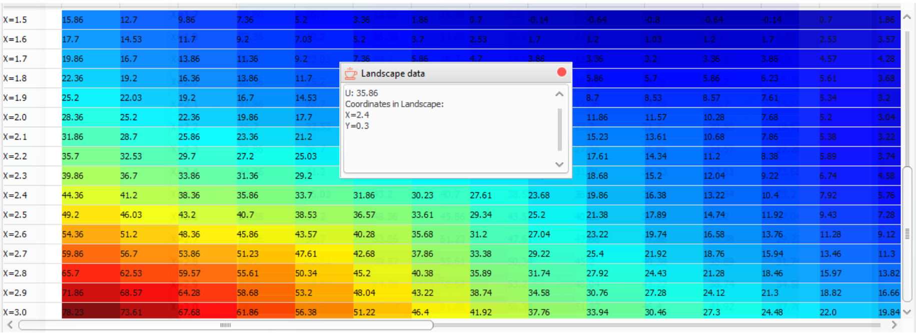
Y-axis
b

Rotate : Left click and drag mouse
Scale : Roll mouse wheel
Z Shift : Right click and drag mouse
Screenshot : Press 's'

(a)



(b)



X=1.5	15.86	12.7	9.86	7.36	5.2	3.36	1.86	0.7	-0.14	-0.64	-0.8	-0.64	-0.14	0.7	1.86
X=1.6	17.7	14.53	11.7	9.2	7.03	5.2	3.7	2.53	1.7	1.2	1.03	1.2	1.7	2.53	3.57
X=1.7	19.86	16.7	13.86	11.36	9.2	7.36	5.86	4.7	3.86	3.36	3.2	3.36	3.86	4.57	4.28
X=1.8	22.36	19.2	16.36	13.86	11.7					5.86	5.7	5.86	6.23	5.61	3.68
X=1.9	25.2	22.03	19.2	16.7	14.53					8.7	8.53	8.57	7.61	5.34	3.2
X=2.0	28.36	25.2	22.36	19.86	17.7					11.86	11.57	10.28	7.68	5.2	3.04
X=2.1	31.86	28.7	25.86	23.36	21.2					15.23	13.61	10.68	7.86	5.38	3.22
X=2.2	35.7	32.53	29.7	27.2	25.03					17.61	14.34	11.2	8.38	5.89	3.74
X=2.3	39.86	36.7	33.86	31.36	29.2					18.68	15.2	12.04	9.22	6.74	4.58
X=2.4	44.36	41.2	38.36	35.86	33.7	31.86	30.23	27.61	23.68	19.86	16.38	13.22	10.4	7.92	5.76
X=2.5	49.2	46.03	43.2	40.7	38.53	36.57	33.61	29.34	25.2	21.38	17.89	14.74	11.92	9.43	7.28
X=2.6	54.36	51.2	48.36	45.86	43.57	40.28	35.68	31.2	27.04	23.22	19.74	16.58	13.76	11.28	9.12
X=2.7	59.86	56.7	53.86	51.23	47.61	42.68	37.86	33.38	29.22	25.4	21.92	18.76	15.94	13.46	11.3
X=2.8	65.7	62.53	59.57	55.61	50.34	45.2	40.38	35.89	31.74	27.92	24.43	21.28	18.46	15.97	13.82
X=2.9	71.86	68.57	64.28	58.68	53.2	48.04	43.22	38.74	34.58	30.76	27.28	24.12	21.3	18.82	16.66
X=3.0	78.23	73.61	67.68	61.86	56.38	51.22	46.4	41.92	37.76	33.94	30.46	27.3	24.48	22.0	19.84

Landscape data

U: 35.86
Coordinates in Landscape:
X=2.4
Y=0.3

(c)

Figure 6. The table of the grid data of the constructed landscape colored by the value of quasi-potential.

The detailed elevations, coordinates on the landscape and gene expression values can be accessed by double click the cells in the table at the bottom-right corner (Figure 6). By clicking the buttons on the top-right panel, users can plot the contour map or show the points on the landscape.

8. Running time

To test the performance of NetLand, we generated 51 random networks with 2 to 52 genes. The experiments were implemented on a PC of 8-core CPU 3.60 GHz Intel Xeon Win 7 Systems. The parameters were fixed. The max expression value was set 3. Time duration was 128. Totally 100 trajectories were generated. Figure 7 shows the time consumed to construct the landscape.

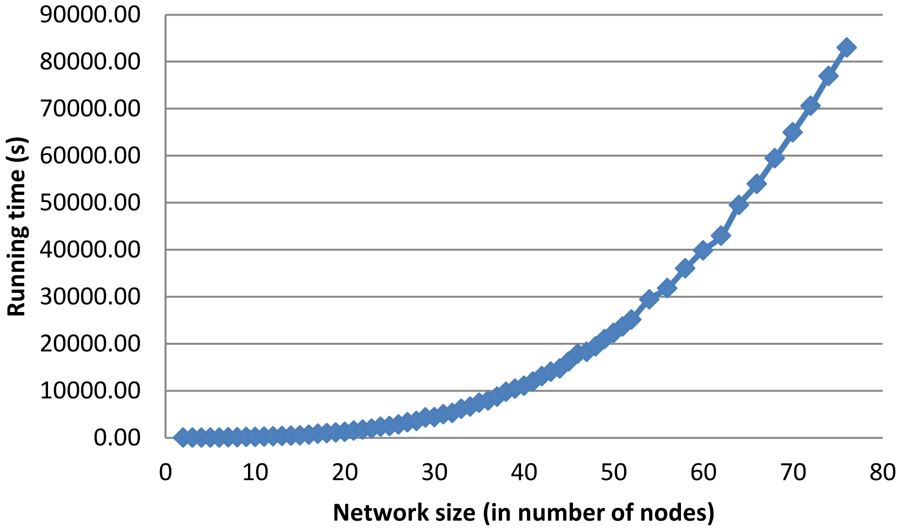


Figure 7. Time consumption during the construction of probabilistic landscapes.

9. Example

To study the stem cell reprogramming process, a computational model, 'seesaw model' with 4 genes (Figure 8) was constructed in [18]. NetLand was applied to simulate and analyze the model. First, the regulatory relations between genes were extracted into a TSV format file. After loading it to NetLand, the differential equations would be updated according to the paper including the parameters. Once the dynamic model was settled, further analysis and simulation can be performed based on this model. Figure 9 shows the simulation result and landscape constructed by NetLand.

More examples can be found at http://netland-ntu.github.io/NetLand/model.html.

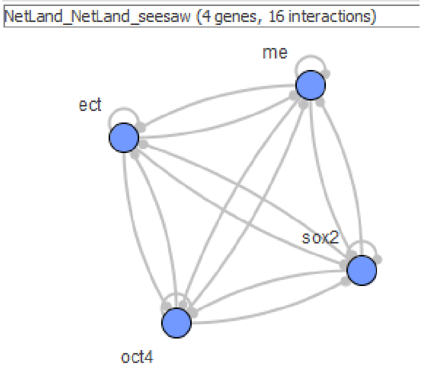
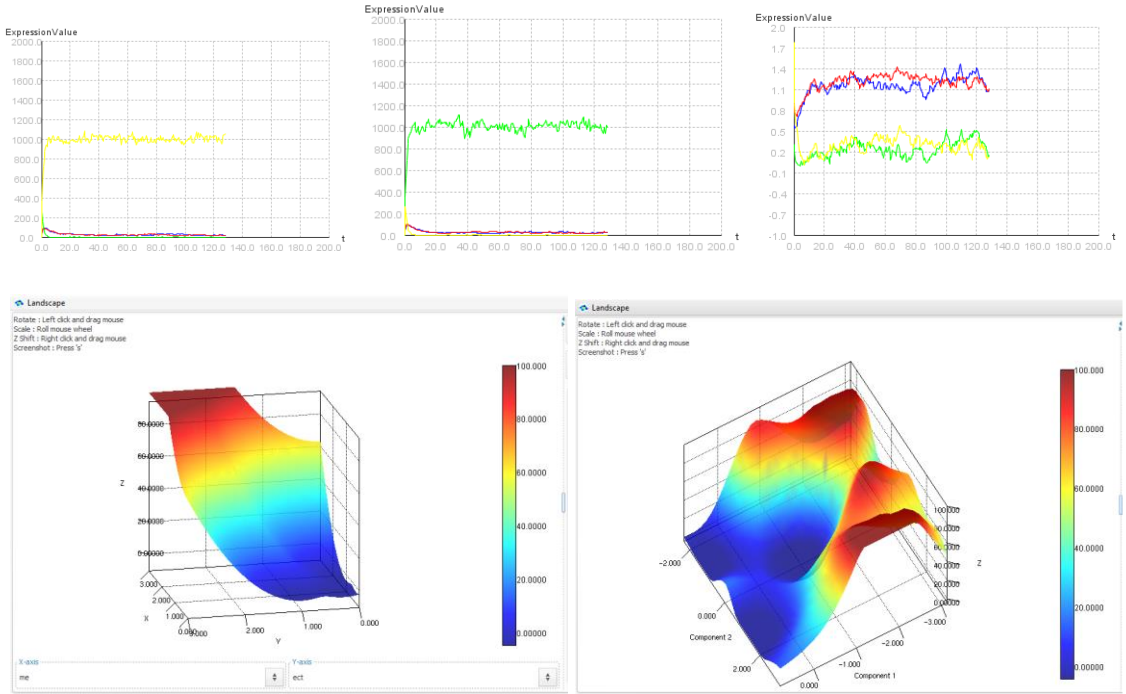

NetLand_NetLand_seesaw (4 genes, 16 interactions)

Figure 8. A seesaw model used to simulate differentiation and reprogramming.


Rotate : Left click and drag mouse
Scale : Roll mouse wheel
Z Shift : Right click and drag mouse
Screenshot : Press 's'

X-axis
me
Y-axis
ect

Rotate : Left click and drag mouse
Scale : Roll mouse wheel
Z Shift : Right click and drag mouse
Screenshot : Press 's'

Figure 9. Simulation results of the seesaw model.

10. Bugs and idiosyncrasies

The authors have carried out extensive testing and debugging of the program, which should be generally stable and functional if the instructions in this manual are followed. It is our hope that, with the GUI, users without much computational sophistication can also apply our algorithm to analyze their network smoothly. However, since the authors are not professional programmers, NetLand does not behave or look like a commercial software. Please send your suggestions, comments, reports of bugs and errors to *Jie Zheng* at zhengjie@ntu.edu.sg. Thank you!

References

1. Waddington, C.H., *The strategy of the genes; a discussion of some aspects of theoretical biology* 1957, London,: Allen & Unwin.

2. Kauffman, S.A., *Metabolic stability and epigenesis in randomly constructed genetic nets.* Journal of theoretical biology, 1969. **22**(3): p. 437-67.

3. Thom, R., *An inventory of Waddingtonian concepts.* Theorical Biology: Epigenetic and Evolutionary Order from Complex Systems, 1989: p. 1-7.

4. Sasai, M. and P.G. Wolynes, *Stochastic gene expression as a many-body problem.* Proc Natl Acad Sci U S A, 2003. **100**(5): p. 2374-9.

5. Huang, S., *The molecular and mathematical basis of Waddington's epigenetic landscape: a framework for post-Darwinian biology?* BioEssays : news and reviews in molecular, cellular and developmental biology, 2012. **34**(2): p. 149-57.

6. Wang, J., L. Xu, and E. Wang, *Potential landscape and flux framework of nonequilibrium networks: robustness, dissipation, and coherence of biochemical oscillations.* Proc Natl Acad Sci U S A, 2008. **105**(34): p. 12271-6.

7. Bhattacharya, S., Q. Zhang, and M. Andersen, *A deterministic map of Waddington's epigenetic landscape for cell fate specification.* BMC systems biology, 2011. **5**: p. 85.

8. Li, C. and J. Wang, *Quantifying cell fate decisions for differentiation and reprogramming of a human stem cell network: landscape and biological paths.* PLoS Comput Biol, 2013. **9**(8): p. e1003165.

9. Lawrence, N., *Probabilistic Non-linear Principal Component Analysis with Gaussian Process Latent Variable Models.* J. Mach. Learn. Res., 2005. **6**: p. 1783-1816.

10. Wang, J.M., D.J. Fleet, and A. Hertzmann, *Gaussian process dynamical models for human motion.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2008. **30**(2): p. 283-298.

11. Schaffter, T., D. Marbach, and D. Floreano, *GeneNetWeaver: in silico benchmark generation and performance profiling of network inference methods.* Bioinformatics, 2011. **27**(16): p. 2263-2270.

12. Ramsey, S., D. Orrell, and H. Bolouri, *Dizzy: stochastic simulation of large-scale genetic regulatory networks.* Journal of bioinformatics and computational biology, 2005. **3**(02): p. 415-436.

13. Zhou, J.X., et al., *Quasi-potential landscape in complex multi-stable systems.* Journal of The Royal Society Interface, 2012. **9**(77): p. 3539-3553.

15. Huang, S., I. Ernberg, and S. Kauffman. *Cancer attractors: a systems view of tumors from a gene network dynamics and developmental perspective*. in *Seminars in cell & developmental biology*. 2009. Elsevier.

16. Lawrence, N., *Probabilistic non-linear principal component analysis with Gaussian process latent variable models.* The Journal of Machine Learning Research, 2005. **6**: p. 1783-1816.

17. Gopalan, N. and T. IAS, *Gaussian Process Latent Variable Models for Dimensionality Reduction and Time Series Modeling.*

18. Shu, J., et al., *Induction of pluripotency in mouse somatic cells with lineage specifiers.* Cell, 2013. **153**(5): p. 963-975.