The NWB Tool Basic Tutorial: Discrete Network Dynamics Tutorial

Table of Contents

Last Modified: Wednesday, July 2, 2008

Goals

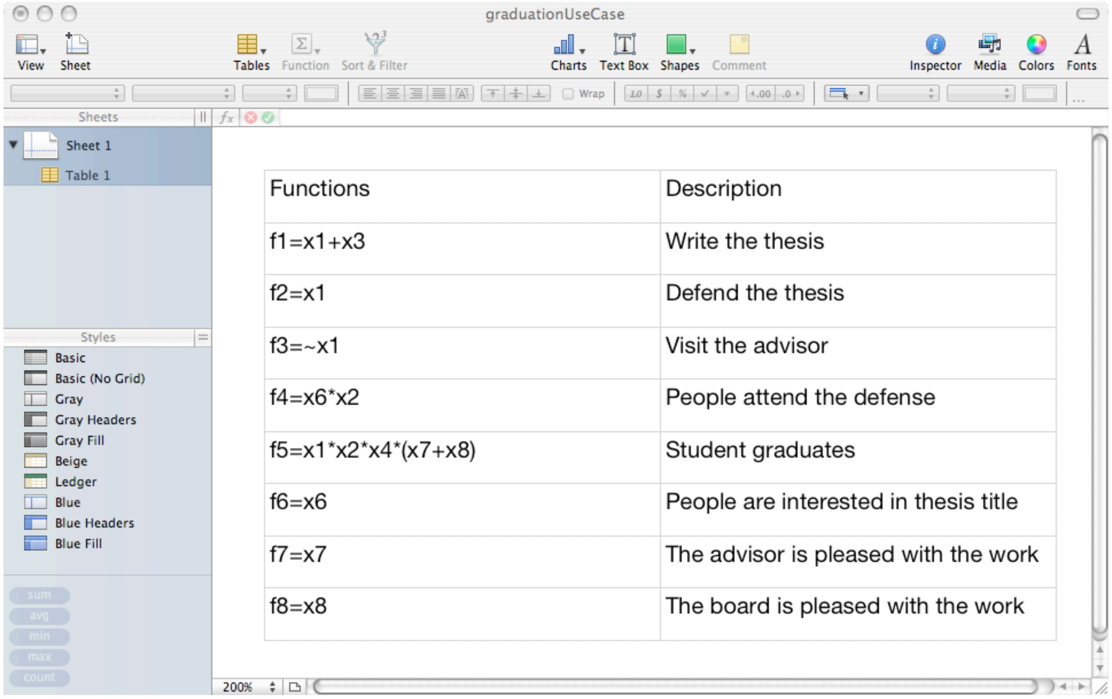
This tutorial introduces the Discrete Network Dynamics tool. You will learn how to perform the following:

- Load or create multi-state discrete network model.
- Generate the state space graph of the model using the Discrete Network Dynamics tool.
- Analyze the attractors of the state space.
- Generate a visualization of one of the attractor basins.

Create and/or Load a Model into the NWB Tool

Create a Model

You can create your own model or load one from our data sets. Creating a model requires some set of interactions you wish to explore using a set number of states and discrete time. A model is created by creating a set of equations that detail the interactions and saving them into a .csv file. The easiest way to do this is to open a spreadsheet program such as Microsoft Excel or OpenOffice Calc to do your editing (see Figure 1).



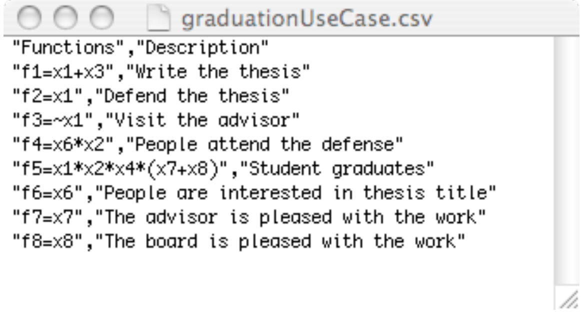
Functions	Description
f1=x1+x3	Write the thesis
f2=x1	Defend the thesis
f3=~x1	Visit the advisor
f4=x6*x2	People attend the defense
f5=x1*x2*x4*(x7+x8)	Student graduates
f6=x6	People are interested in thesis title
f7=x7	The advisor is pleased with the work
f8=x8	The board is pleased with the work

Figure 1: A sample multi-state network model opened in a spreadsheet program

1.1.1. Create a heading for each column in the table, at a minimum you need a column to hold your functions, but you can add additional information for the dependency graphs by including additional columns.

1.1.2. Each row represents a component you wish to examine in relationship to other components. You create a function for each component that describes the relationship. See the section **Creating Functions** for how to create a valid function.

1.1.3. After you have created your functions, export your spreadsheet to .csv format. Check the documentation for your spreadsheet program on how to do this. The final .csv file should look similar to Figure 2.



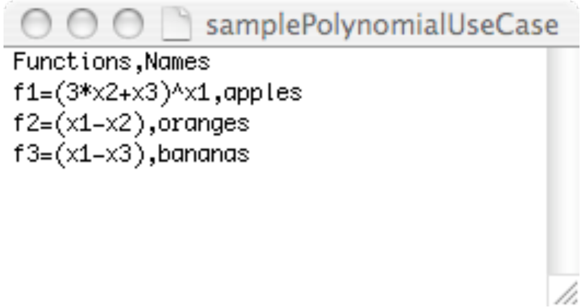
```
"Functions","Description"
"f1=x1+x3","Write the thesis"
"f2=x1","Defend the thesis"
"f3=~x1","Visit the advisor"
"f4=x6*x2","People attend the defense"
"f5=x1*x2*x4*(x7+x8)","Student graduates"
"f6=x6","People are interested in thesis title"
"f7=x7","The advisor is pleased with the work"
"f8=x8","The board is pleased with the work"
```

Figure 2: Final .csv file of some sample data.

Load a Model

It is assumed that you have installed the NWB tool and are familiar with the load operations described in **The NWB Tool Basic Tutorial: Getting Started**. To load a model, simply select **File → Load...** in the main menu and select a model file either from the sample data sets provided by the NWB tool at ***NWB_Installation_Directory*/sampledata/DND/filename** (see Figure 1) or from your own collection.

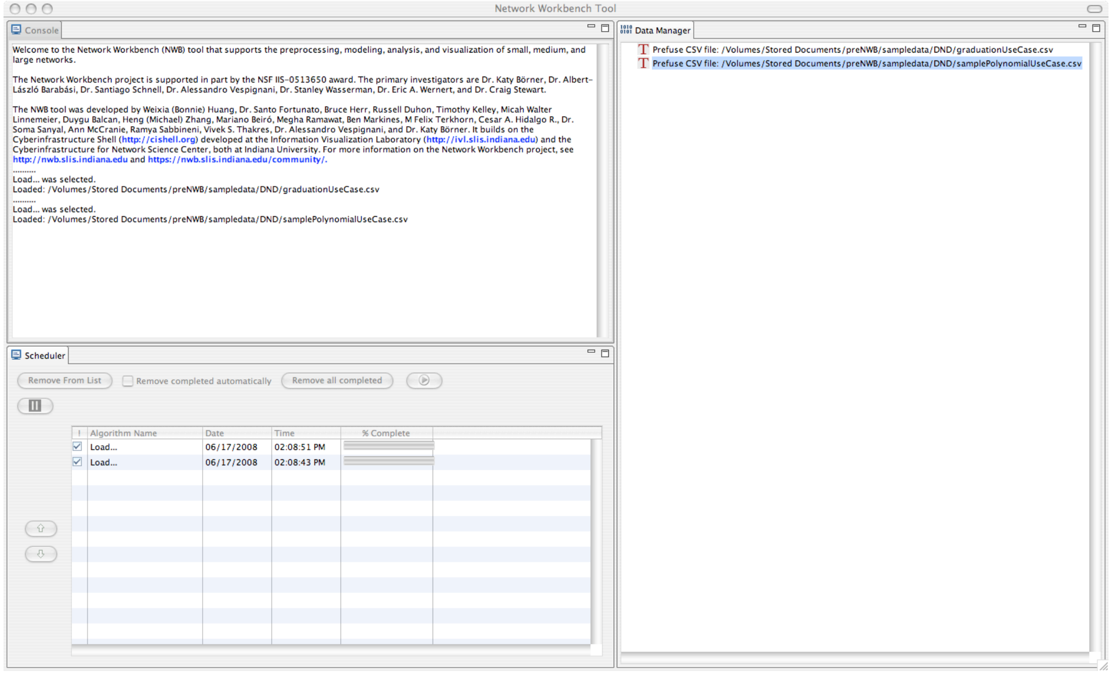
We will be examining two network models for this tutorial: the one described by Figure 2 and the one described by Figure 3. Both can be found in the sample data directory.



```
Functions,Names
f1=(3*x2+x3)^x1,apples
f2=(x1-x2),oranges
f3=(x1-x3),bananas
```

Figure 3: A polynomial model

After loading them both into NWB your workspace should look like Figure 4.



Welcome to the Network Workbench (NWB) tool that supports the preprocessing, modeling, analysis, and visualization of small, medium, and large networks.

The Network Workbench project is supported in part by the NSF IIS-0513650 award. The primary investigators are Dr. Katy Börner, Dr. Albert-László Barabási, Dr. Santiago Schnell, Dr. Alessandro Vespignani, Dr. Stanley Wasserman, Dr. Eric A. Wernert, and Dr. Craig Stewart.

The NWB tool was developed by Weixia (Bonnie) Huang, Dr. Santo Fortunato, Bruce Herr, Russell Duhon, Timothy Kelley, Micah Walter Linnemeier, Duygu Balcan, Heng (Michael) Zhang, Mariano Beiró, Megha Ramawat, Ben Markines, M Felix Terkhorn, Cesar A. Hidalgo R., Dr. Soma Sanyal, Ann McCranie, Ramya Sabbineni, Vivek S. Thakres, Dr. Alessandro Vespignani, and Dr. Katy Börner. It builds on the Cyberinfrastructure Shell (http://cishell.org) developed at the Information Visualization Laboratory (http://ivl.slis.indiana.edu) and the Cyberinfrastructure for Network Science Center, both at Indiana University. For more information on the Network Workbench project, see http://nwb.slis.indiana.edu and https://nwb.slis.indiana.edu/community/.

```
..........
Load... was selected.
Loaded: /Volumes/Stored Documents/preNWB/sampledata/DND/graduationUseCase.csv
..........
Load... was selected.
Loaded: /Volumes/Stored Documents/preNWB/sampledata/DND/samplePolynomialUseCase.csv
```

Prefuse CSV file: /Volumes/Stored Documents/preNWB/sampledata/DND/graduationUseCase.csv
Prefuse CSV file: /Volumes/Stored Documents/preNWB/sampledata/DND/samplePolynomialUseCase.csv

!	Algorithm Name	Date	Time	% Complete
☑	Load...	06/17/2008	02:08:51 PM	
☑	Load...	06/17/2008	02:08:43 PM	

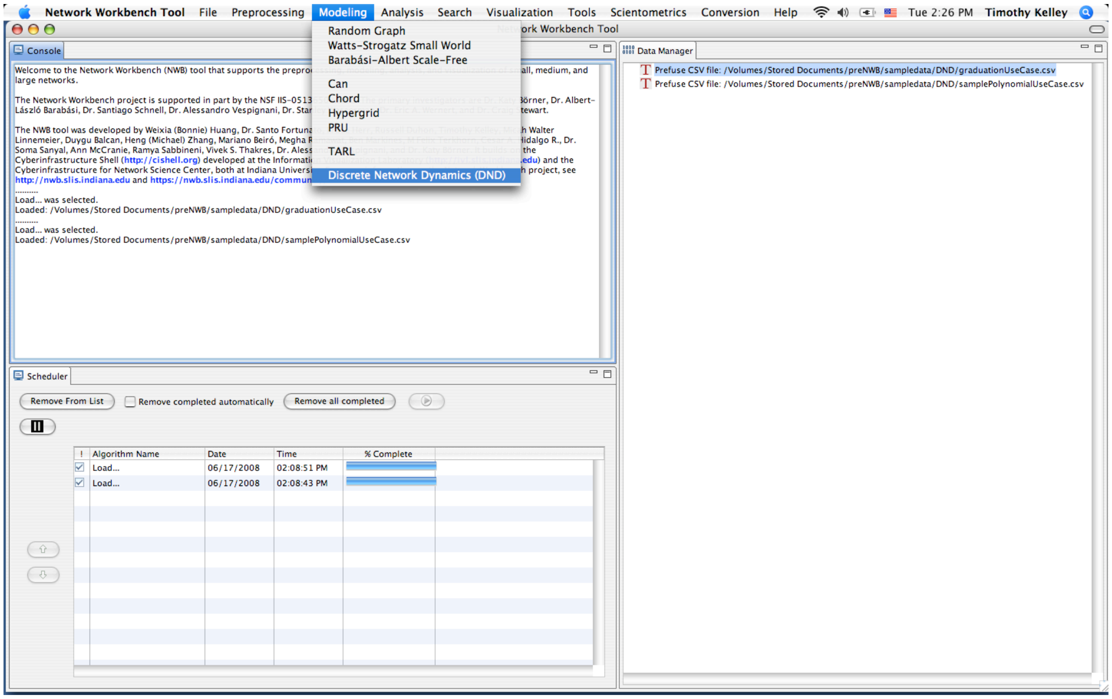
Figure 4: Workspace after loading in two sample models

Using the Discrete Network Dynamics (DND) Tool

Once the models are loaded, they show up in the Data Manager window (Figure 4) this allows you to manipulate them in some basic ways, such as viewing them as text files or discarding them. These operations are covered in The NWB Tool Basic Tutorial: Getting Started.

Evaluating a Boolean Network Model

We will start off by evaluating the **graduationUseCase.csv** model. This is a toy model that will tell us under what conditions a graduate student might graduate. Click on the model in the Data Manager window and select the Discrete Network Dynamics (DND) from the Modeling menu (Figure 5).

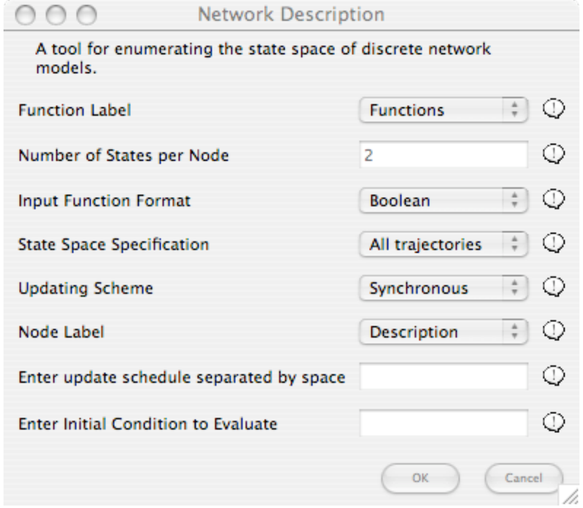


Prefuse CSV file: /Volumes/Stored Documents/preNWB/sampledata/DND/graduationUseCase.csv
Prefuse CSV file: /Volumes/Stored Documents/preNWB/sampledata/DND/samplePolynomialUseCase.csv

!	Algorithm Name	Date	Time	% Complete
☑	Load...	06/17/2008	02:08:51 PM	
☑	Load...	06/17/2008	02:08:43 PM	

Figure 5: Selecting the DND tool from the menu.

After selecting the DND modeling algorithm, it will bring up a window which displays the various parameters of the algorithm. For now we will just evaluate the network using the basic options. Set the **Function Label** to the column name that defines the functions and set the **Input Function Format** to Boolean as in Figure 6, then press **OK**.

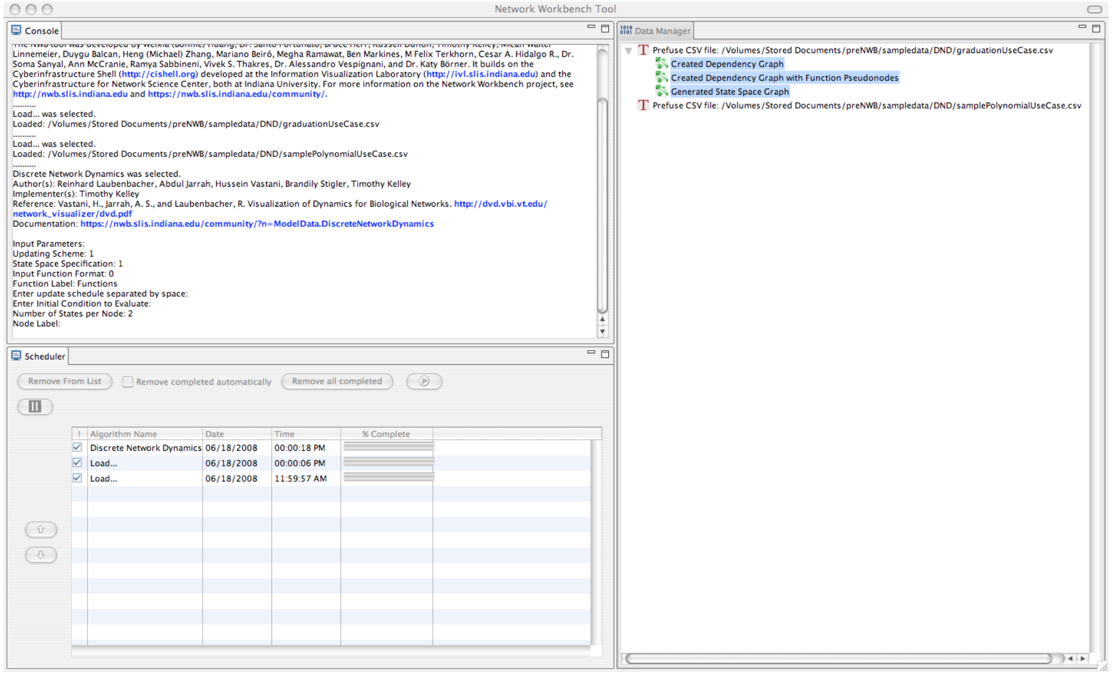


A tool for enumerating the state space of discrete network models.

Function Label: Functions
Number of States per Node: 2
Input Function Format: Boolean
State Space Specification: All trajectories
Updating Scheme: Synchronous
Node Label: Description
Enter update schedule separated by space:
Enter Initial Condition to Evaluate:

Figure 6: Setting the parameters

The algorithm will run and it should produce the results found in Figure 7. There are three graphs that are generated, the dependency graph, the dependency graph with pseudo-nodes, and a state space graph. The dependency graph and the dependency graph with pseudo-nodes are representations of the model, while the state space graph is an exploration of the model. Visualizing the state space graph using one of the visualization algorithms under the **Visualization** menu, you can identify important components in the model.



```
The NWB tool was developed by Weixia (Bonnie) Huang, Dr. Santo Fortunato, Bruce Herr, Russell Duhon, Timothy Kelley, Micah Walter
Linnemeier, Duygu Balcan, Heng (Michael) Zhang, Mariano Beiró, Megha Ramawat, Ben Markines, M Felix Terkhorn, Cesar A. Hidalgo R., Dr.
Soma Sanyal, Ann McCranie, Ramya Sabbineni, Vivek S. Thakres, Dr. Alessandro Vespignani, and Dr. Katy Börner. It builds on the
Cyberinfrastructure Shell (http://cishell.org) developed at the Information Visualization Laboratory (http://ivl.slis.indiana.edu) and the
Cyberinfrastructure for Network Science Center, both at Indiana University. For more information on the Network Workbench project, see
http://nwb.slis.indiana.edu and https://nwb.slis.indiana.edu/community/.
..........
Load... was selected.
Loaded: /Volumes/Stored Documents/preNWB/sampledata/DND/graduationUseCase.csv
..........
Load... was selected.
Loaded: /Volumes/Stored Documents/preNWB/sampledata/DND/samplePolynomialUseCase.csv
..........
Discrete Network Dynamics was selected.
Author(s): Reinhard Laubenbacher, Abdul Jarrah, Hussein Vastani, Brandily Stigler, Timothy Kelley
Implementer(s): Timothy Kelley
Reference: Vastani, H., Jarrah, A. S., and Laubenbacher, R. Visualization of Dynamics for Biological Networks. http://dvd.vbi.vt.edu/
network_visualizer/dvd.pdf
Documentation: https://nwb.slis.indiana.edu/community/?n=ModelData.DiscreteNetworkDynamics

Input Parameters:
Updating Scheme: 1
State Space Specification: 1
Input Function Format: 0
Function Label: Functions
Enter update schedule separated by space:
Enter Initial Condition to Evaluate:
Number of States per Node: 2
Node Label:
```

- Prefuse CSV file: /Volumes/Stored Documents/preNWB/sampledata/DND/graduationUseCase.csv
  - Created Dependency Graph
  - Created Dependency Graph with Function Pseudonodes
  - Generated State Space Graph
- Prefuse CSV file: /Volumes/Stored Documents/preNWB/sampledata/DND/samplePolynomialUseCase.csv

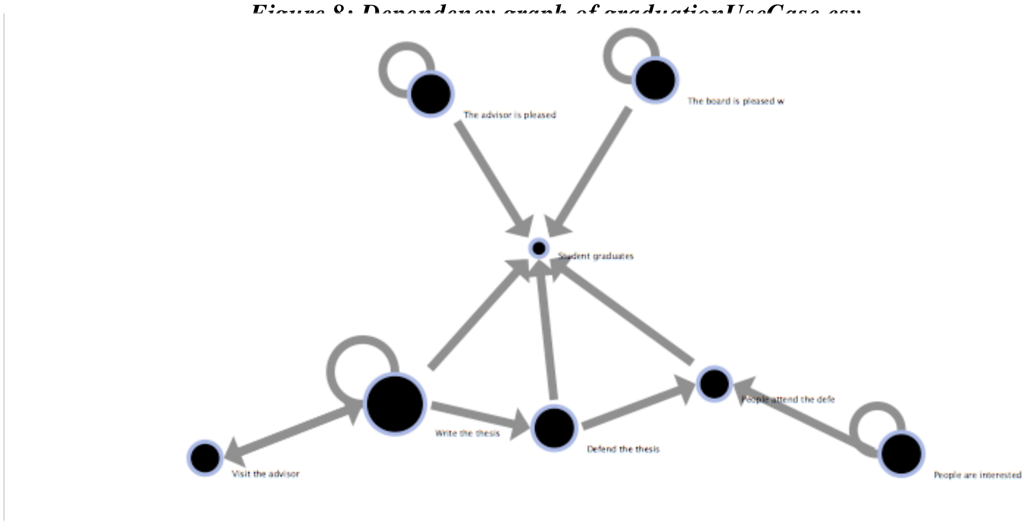
□ Remove completed automatically

!	Algorithm Name	Date	Time	% Complete
☑	Discrete Network Dynamics	06/18/2008	00:00:18 PM	
☑	Load...	06/18/2008	00:00:06 PM	
☑	Load...	06/18/2008	11:59:57 AM	

Figure 7: Data Manager after analysis of the Boolean network model.

Examining the dependency graph of the Boolean network model described the **graduationUseCase.csv** file using the GUESS visualization with some transformations of the graph[1], we see that writing the thesis is the most important component of the model in that it influences the most components (Figure 8). The commands used to generate the visualization found in Figure 8 are found in the GUESS Commands section and correspond to commands 1, 2, and 3.

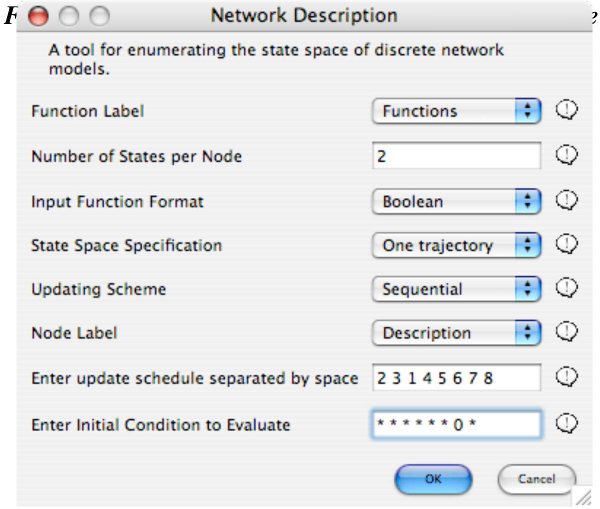
Figure 8: Dependency graph of graduationUseCase.csv



Additional Parameters

Now we are going to consider some of the additional parameters for this model. Let us assume we know that the student visits their advisor first, then writes the thesis, then people show up for the defense, then we check to see if the student graduates. We are using the last three columns as parameters and so can update those last. Also, given the above update sequence, let us also say we are interested in the situations where the advisor is not pleased with the work. We can perform these calculations by filling in the GUI as shown in Figure 9. It helps to examine the function file when setting these additional parameters.

F ***e.***



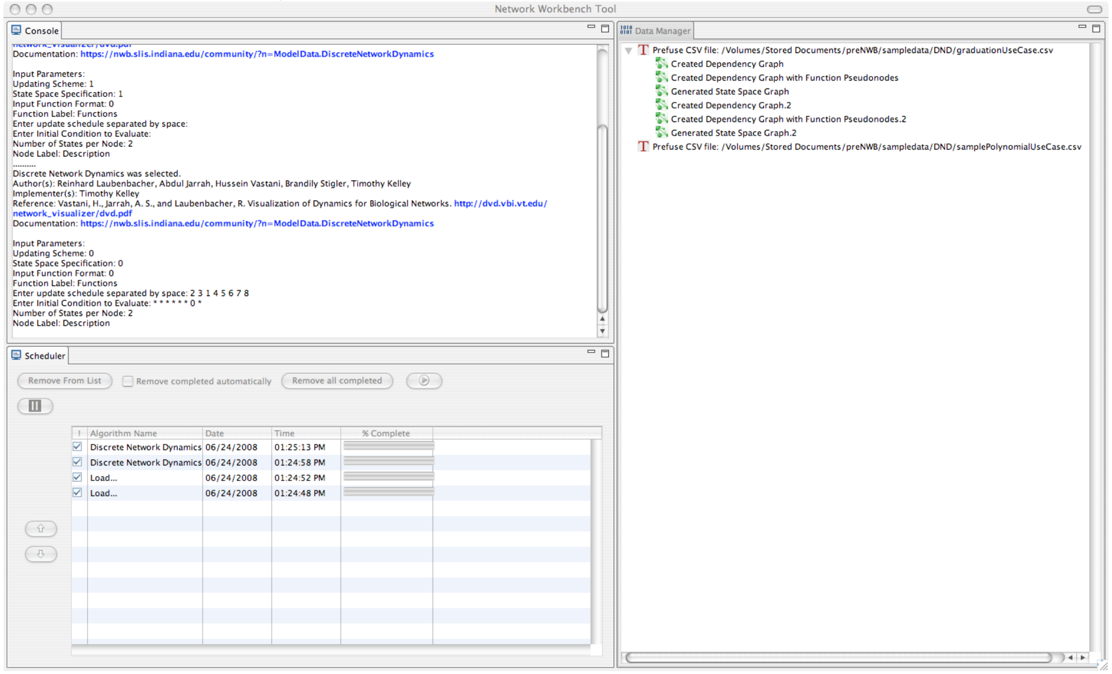
Network Description

A tool for enumerating the state space of discrete network models.

Function Label	Functions
Number of States per Node	2
Input Function Format	Boolean
State Space Specification	One trajectory
Updating Scheme	Sequential
Node Label	Description
Enter update schedule separated by space	2 3 1 4 5 6 7 8
Enter Initial Condition to Evaluate	* * * * * * 0 *

The new Dependency Graphs along with the new state space graph should show up underneath the main file as shown in Figure 10.

[1] A manual for using GUESS can be found at http://guess.wikispot.org/manual



```
network_visualizer/dvd.pdf
Documentation: https://nwb.slis.indiana.edu/community/?n=ModelData.DiscreteNetworkDynamics

Input Parameters:
Updating Scheme: 1
State Space Specification: 1
Input Function Format: 0
Function Label: Functions
Enter update schedule separated by space:
Enter Initial Condition to Evaluate:
Number of States per Node: 2
Node Label: Description
..........
Discrete Network Dynamics was selected.
Author(s): Reinhard Laubenbacher, Abdul Jarrah, Hussein Vastani, Brandily Stigler, Timothy Kelley
Implementer(s): Timothy Kelley
Reference: Vastani, H., Jarrah, A. S., and Laubenbacher, R. Visualization of Dynamics for Biological Networks. http://dvd.vbi.vt.edu/
network_visualizer/dvd.pdf
Documentation: https://nwb.slis.indiana.edu/community/?n=ModelData.DiscreteNetworkDynamics

Input Parameters:
Updating Scheme: 0
State Space Specification: 0
Input Function Format: 0
Function Label: Functions
Enter update schedule separated by space: 2 3 1 4 5 6 7 8
Enter Initial Condition to Evaluate: * * * * * * 0 *
Number of States per Node: 2
Node Label: Description
```

!	Algorithm Name	Date	Time	% Complete
☑	Discrete Network Dynamics	06/24/2008	01:25:13 PM	
☑	Discrete Network Dynamics	06/24/2008	01:24:58 PM	
☑	Load...	06/24/2008	01:24:52 PM	
☑	Load...	06/24/2008	01:24:48 PM	

```
▼ T Prefuse CSV file: /Volumes/Stored Documents/preNWB/sampledata/DND/graduationUseCase.csv
      Created Dependency Graph
      Created Dependency Graph with Function Pseudonodes
      Generated State Space Graph
      Created Dependency Graph.2
      Created Dependency Graph with Function Pseudonodes.2
      Generated State Space Graph.2
  T Prefuse CSV file: /Volumes/Stored Documents/preNWB/sampledata/DND/samplePolynomialUseCase.csv
```

Figure 10: Data Manager after analyzing network for the second time.

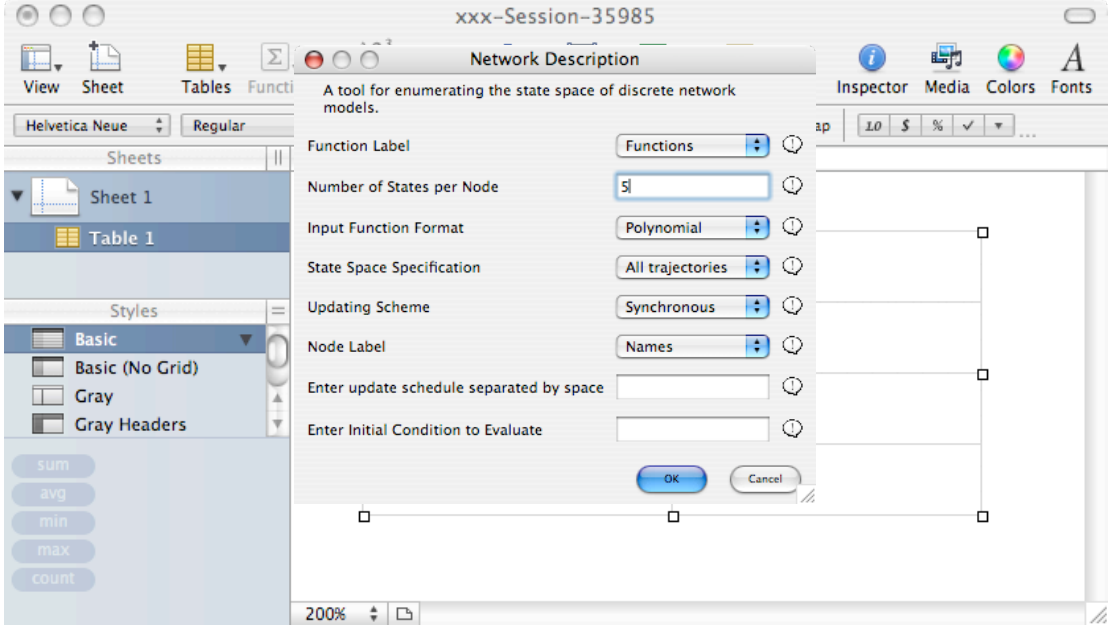
This part of the tutorial has covered the necessary steps to enumerate the state space of a Boolean network model. We have done the following:

1. Loaded a .csv file describing the model
2. Selected the DND tool from the Modeling menu
3. Analyzed the network using default parameters
4. Analyzed the network using additional information provided in the optional parameters

The next step of the tutorial will be brief and use the second file that we loaded for simple analysis using the default parameter settings for discrete Polynomial network models.

Evaluating a Polynomial Network Model

In this section we evaluate a simple polynomial network model. This is a toy model, similar to the previous one, but even more simple. It details a theoretical person's decisions about how much of a specific fruit to eat based on relationships between other fruit. Select the file **simplePolynomialUseCase.csv** that we loaded and select **View** or **View With...** as described in **The NWB Tool Basic Tutorial: Getting Started**. Doing so should bring up something like Figure 11. Notice that there are more operators available for analyzing polynomial network models. The possible operators are discussed in the section Creating Functions. Now we select the DND tool from the Modeling menu and select a number of states per node, say five, since our theoretical person does not want to eat any more than five of each of the fruits per day. The GUI should look like Figure 12. Press the **OK** button to analyze the model. This should give you results similar to Figure 13.



Network Description

A tool for enumerating the state space of discrete network models.

Function Label: Functions

Number of States per Node: 5

Input Function Format: Polynomial

State Space Specification: All trajectories

Updating Scheme: Synchronous

Node Label: Names

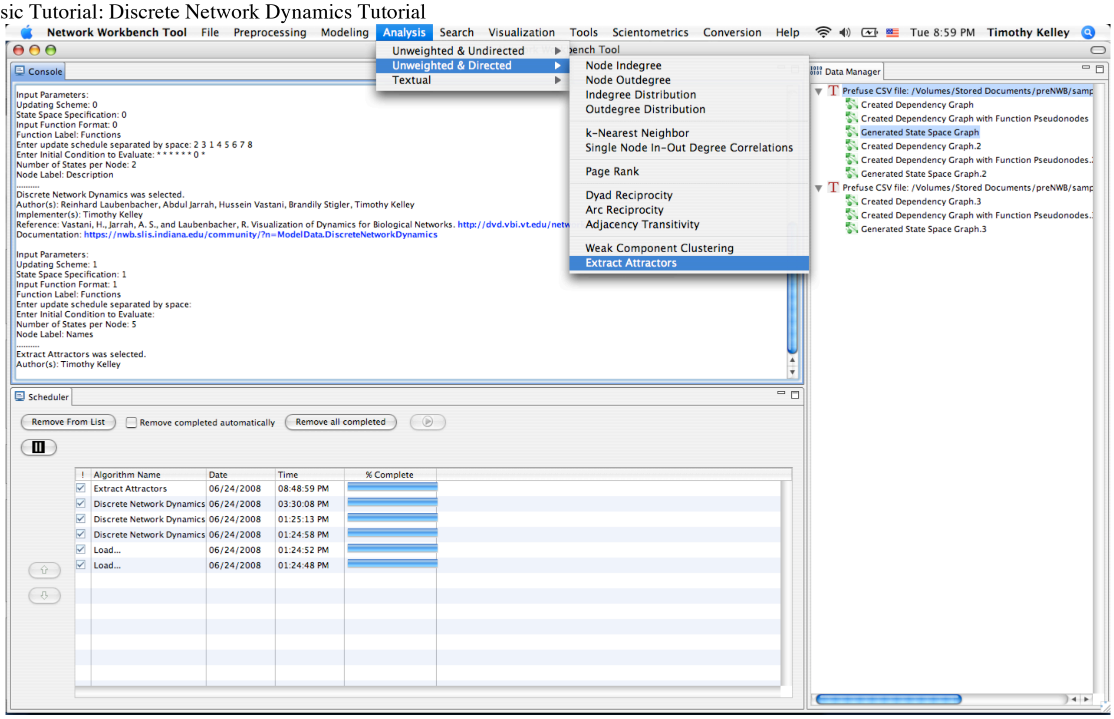
Enter update schedule separated by space:

Enter Initial Condition to Evaluate:

Figure 12: Parameter selection for a polynomial network.

Analyzing the Attractors

After we have generated the state space, we want to be able to see what sort of behavior our model has. We do this by analyzing the attractors. Analyzing the attractors will tell us what the steady states of the model are and allow us to compare our model with experimental data. To do this we must first select the state space graph and the original model in the data manager and select **Extract Attractors** from the **Analysis → Unweighted & Directed** menu (Figure 13).


Network Workbench Tool
File
Preprocessing
Modeling
Analysis
Search
Visualization
Tools
Scientometrics
Conversion
Help
Tue 8:59 PM
Timothy Kelley
Unweighted & Undirected
Unweighted & Directed
Textual
Node Indegree
Node Outdegree
Indegree Distribution
Outdegree Distribution
k-Nearest Neighbor
Single Node In-Out Degree Correlations
Page Rank
Dyad Reciprocity
Arc Reciprocity
Adjacency Transitivity
Weak Component Clustering
Extract Attractors
Console
Data Manager
Prefuse CSV file: /Volumes/Stored Documents/preNWB/samp
Created Dependency Graph
Created Dependency Graph with Function Pseudonodes
Generated State Space Graph
Created Dependency Graph.2
Created Dependency Graph with Function Pseudonodes.
Generated State Space Graph.2
Prefuse CSV file: /Volumes/Stored Documents/preNWB/samp
Created Dependency Graph.3
Created Dependency Graph with Function Pseudonodes.
Generated State Space Graph.3
Scheduler
Remove From List
Remove all completed

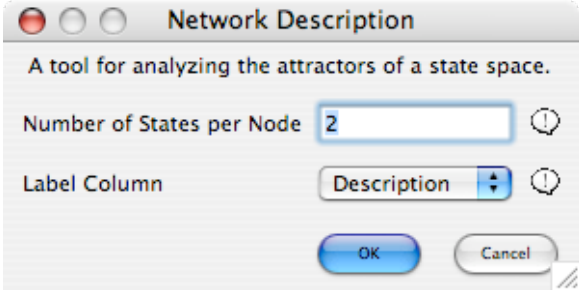
```
Input Parameters:
Updating Scheme: 0
State Space Specification: 0
Input Function Format: 0
Function Label: Functions
Enter update schedule separated by space: 2 3 1 4 5 6 7 8
Enter Initial Condition to Evaluate: * * * * * * 0 *
Number of States per Node: 2
Node Label: Description
..........
Discrete Network Dynamics was selected.
Author(s): Reinhard Laubenbacher, Abdul Jarrah, Hussein Vastani, Brandily Stigler, Timothy Kelley
Implementer(s): Timothy Kelley
Reference: Vastani, H., Jarrah, A. S., and Laubenbacher, R. Visualization of Dynamics for Biological Networks. http://dvd.vbi.vt.edu/netw
Documentation: https://nwb.slis.indiana.edu/community/?n=ModelData.DiscreteNetworkDynamics

Input Parameters:
Updating Scheme: 1
State Space Specification: 1
Input Function Format: 1
Function Label: Functions
Enter update schedule separated by space:
Enter Initial Condition to Evaluate:
Number of States per Node: 5
Node Label: Names
..........
Extract Attractors was selected.
Author(s): Timothy Kelley
```

□ Remove completed automatically

!	Algorithm Name	Date	Time	% Complete
☑	Extract Attractors	06/24/2008	08:48:59 PM	
☑	Discrete Network Dynamics	06/24/2008	03:30:08 PM	
☑	Discrete Network Dynamics	06/24/2008	01:25:13 PM	
☑	Discrete Network Dynamics	06/24/2008	01:24:58 PM	
☑	Load...	06/24/2008	01:24:52 PM	
☑	Load...	06/24/2008	01:24:48 PM	

Figure 13: Preparing to analyze the attractors of the Boolean network model.

The GUI for **Extract Attractors** should pop up and we optionally select the column in our model file that contains the labels we want to use for our attractors (Figure 14). Currently, the Number of States per Node is not functional but will be used in future versions of the algorithm. Press **OK** to complete the extraction and analysis.



A tool for analyzing the attractors of a state space.

Number of States per Node: 2

Label Column: Description

Figure 14: GUI for Extract Attractors.

After the analysis is complete there should be a list of attractor basins and table views of the attractor for each basin. Before looking at the attractors individually, we want to examine the entire state space. Select the **Network Analysis Toolkit** under the **Tools** menu, as described in **The NWB Tool Basic Tutorial: Getting Started**. This will give us an overview of the state space graph and tell us where the single node attractors are by listing the self-loops. In this case, every attractor has a single node attractor. If we examine the self-loops from the Network Analysis Toolkit, we see that there are three attractors where our theoretical student graduates. We can choose to view the attractors by selecting a Table View of Attractor # and choosing to **View** or **View With...** to see what the final attractor looks like. This becomes more useful in real models such as the model found in the section **A Boolean Network Model of Colorectal Cancer**. The **View** or **View With...** presents the attractor in table form with the labels we selected in the GUI. This enables us to see what states our attractors take and, based on our knowledge of the model, draw conclusions on how the system behaves. A sample view of an attractor table can be found in Table 1.

Write the thesis	Defend the thesis	Visit the advisor	People attend the defense	Student graduates	People are interested in thesis title	The advisor is pleased with the work	The board is pleased with the work
1	1	0	0	0	0	0	0

Write the thesis	Defend the thesis	Visit the advisor	People attend the defense	Student graduates	People are interested in thesis title	The advisor is pleased with the work	The board is pleased with the work
1	1	0	0	0	0	0	0

We can see that in this case, while the student has written and presented the thesis, no one was interested and neither the board nor the advisor was pleased with the work and so the student fails to graduate. A more interesting behavior can be seen if we analyze the attractors of our polynomial model. Table 2 shows the behavior of the first attractor in the polynomial case.

apples	oranges	bananas
1	2	2
3	4	4
1	4	4

apples	oranges	bananas
1	2	2
3	4	4
1	4	4

Here we have a loop in our attractor rather than a steady node. Attractors in these types of discrete networks will always be nodes or loops.

A biological application can be found in the section **A Boolean Network Model of Colorectal Cancer**, where we look at what conclusions can be drawn from the table view of the attractors. Another way of analyzing the system qualitatively is to visualize the entire system. While visualizing large systems is computationally intensive, after separating the attractor basins will hopefully allow us to view each basin individually without taxing our resources too much.

Visualizing the Attractor Basins

Visualizing the attractor basins gives a qualitative overview of the basins. Visualization can give insight into critical transitions in the graph as well as the likelihood that a given initial condition will reach the attractor given random perturbations. There are several visualization algorithms available in Network Workbench, but for the following visualizations we will be using GUESS. The code used to generate the visualizations is found in GUESS Commands section.

We will visualize two attractor basins, one from the Boolean model and one from the polynomial model. We start with the Boolean network model.

1. Select **Attractor Basin 8 of 32 nodes** under the **graduationUseCase.csv** model.

2. Select **GUESS** from the **Visualization** menu. This should display a new window that looks something like Figure 15.

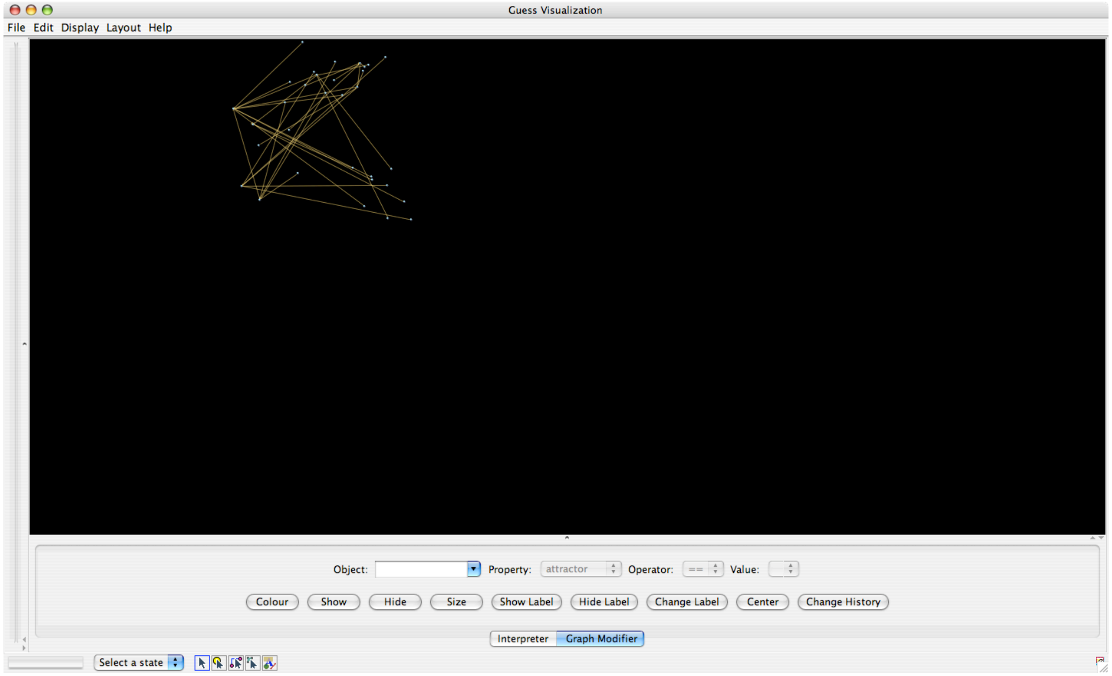
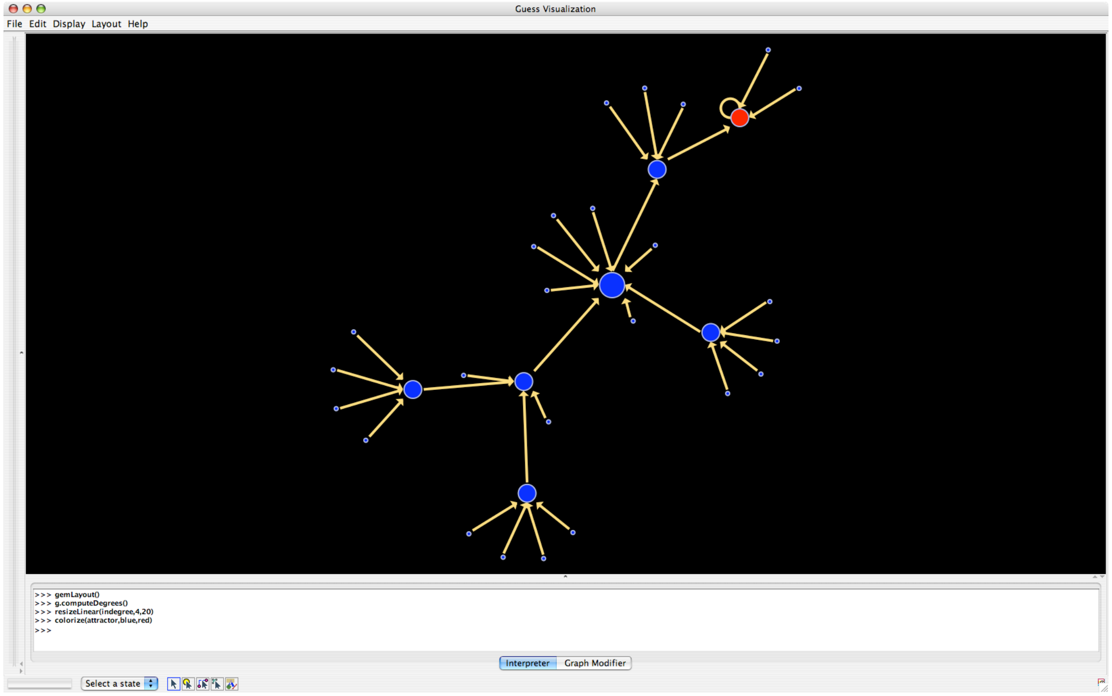


Figure 15: Guess Visualization application

3. At the bottom of the GUESS window, select the **Interpreter**. This will pull up a console in which we can enter commands.

4. Enter commands 1 and 4 from the GUESS Commands section. This should generate a visualization that looks similar to Figure 16.

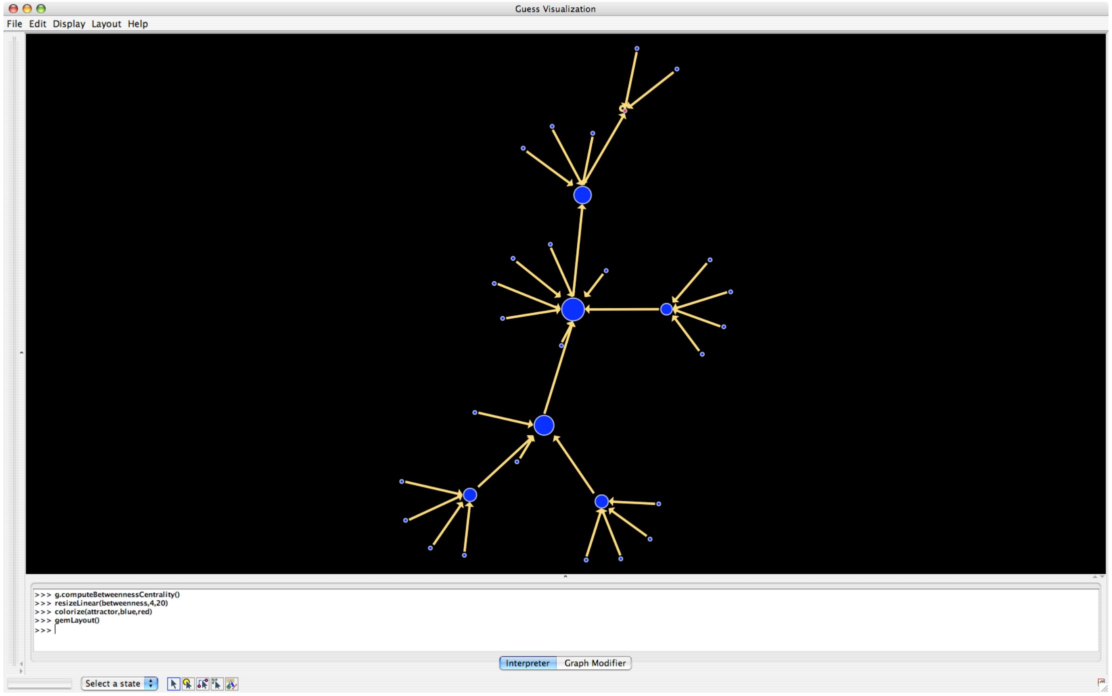


```
>>> gemLayout()
>>> g.computeDegrees()
>>> resizeLinear(indegree,4,20)
>>> colorize(attractor,blue,red)
>>>
```

Figure 16: Visualization of Attractor Basin 8 with GEM layout.

The attractor is colored red and the nodes are resized based on the number of incoming transitions. The size of the node indicates the importance to the transition space based on number of transitions coming into the specific state. Random perturbations at these larger nodes may cause transitions to jump between attractor basins.

Another metric is Betweenness Centrality, which can demonstrate the centrality of the attractor. States that are the components of many short paths have a higher Betweenness Centrality score and have a higher likelihood of being reached in the presence of noise. Attractor basin 8 rescaled on Betweenness Centrality is found in Figure 17, notice the attractor on the top of the graph and how small it is. This suggests that, if there is noise, this attractor would unlikely be reached.



```
>>> g.computeBetweennessCentrality()
>>> resizeLinear(betweenness,4,20)
>>> colorize(attractor,blue,red)
>>> gemLayout()
>>>
```

Figure 17: Attractor Basin 8 with Nodes Rescaled on Betweenness Centrality

Summary

In this tutorial we have explored some basic analysis of multi-state discrete network models using Network Workbench.

- We have created or loaded some simple models into Network Workbench.
- Have calculated the models' state spaces using the Discrete Network Dynamics Tool.
- Extracted the attractor basins and annotated the attractors using the Extract Attractor algorithm.
- Visualized the attractor basins using two different metrics: the in-degree of a transition state and the Betweenness Centrality of the a transition state.

There is some additional information about writing functions in the Creating Functions section. We also went through some basic commands using GUESS in the GUESS Commands section. **Something about the Biology Section here.**

Acknowledgments

The Network Workbench project (http://nwb.slis.indiana.edu/) is supported in part by the NSF IIS-0513650 award. This tutorial was compiled by Timothy Kelley, Weixia (Bonnie) Huang, and Katy Börner. For comments, questions or suggestions, please post to the nwb-helpdesk@googlegroups.com mailing list.

Creating Functions

The most important aspect of a multi-state discrete network model is its update functions. An update function determines the value of a variable at the next time step. The Discrete Network Dynamics Tool uses a specific format and a small set of operators to define an update function. There are different operators available based on whether you choose to evaluate Boolean expressions or polynomial expressions. Table 3 details the operations that are supported for the two different types of expressions.

Symbol	Polynomial	Boolean
*	$(x_1 \text{ mod } m * x_2 \text{ mod } m) \text{ mod } m$	x_1 **AND** x_2
/	$(x_1 \text{ mod } m * x_2^{-1} \text{ mod } m) \text{ mod } m$	Not Defined
^	$((x_1 \text{ mod } m)^{(x_2 \text{ mod } m)}) \text{ mod } m$	Not Defined
+	$(x_1 \text{ mod } m + x_2 \text{ mod } m) \text{ mod } m$	x_1 **OR** x_2
-	$(x_1 \text{ mod } m - x_2 \text{ mod } m) \text{ mod } m$	Not Defined
~	$-x_1 \text{ mod } m$	If x_1 == 1, x_1 = 0. If x_1 == 0, x_1 = 1

Table 3: Supported Operations by Expression Type

The m in the operators description represents the polynomial field used for operations. It is equal to the number of states per node set in the GUI as shown in Figure 12. Using the / operator can cause errors unless you are operating over a prime field, so be careful when using it.

Update functions follow a specific format that allows them to be interpreted by the program. After the heading row, each row must contain a function. Functions are defined by placing an f, followed by a row number, starting with 1. Subsequent functions increase their row numbers in a consecutive manner. After the row number use the equal sign to denote the definition of the function. An x, followed by a number indicates the variable associated with the number's row number. Numeric constants may also be added. For example, f1 = x3+3 is a valid expression for polynomial evaluation. Parentheses can also be included to specify order of operations. Figures 1, 2, and 3 have several sample functions. Additional information can be found at https://nwb.slis.indiana.edu/community/?n=ModelData.DiscreteNetworkDynamics.

GUESS Commands

GUESS is a visualization tool that is encapsulated into Network Workbench. This is a quick explanation of GUESS for generating the visualizations found in this tutorial. For a more in depth treatment of GUESS please see the documentation on GUESS which can be found at http://graphexploration.cond.org/documentation.html.

The first step to using GUESS is to select the **Interpreter** rather than the **Graph Modifier**. Once you have changed from **Graph Modifier** mode to **Interpreter** mode, you can enter commands into the command line. The commands used in this tutorial are found below.

1. **General Layout**

```
gemLayout()
center()
g.computeDegrees()
resizeLinear(indegree,4,20)
```

2. **Changing Node and Edge Colors**

```
g.nodes.color=black
g.edges.color=gray
```

3. **Changing Background Color**

```
setDisplayBackground(white)
```

4. **Colorizing the Attractor**

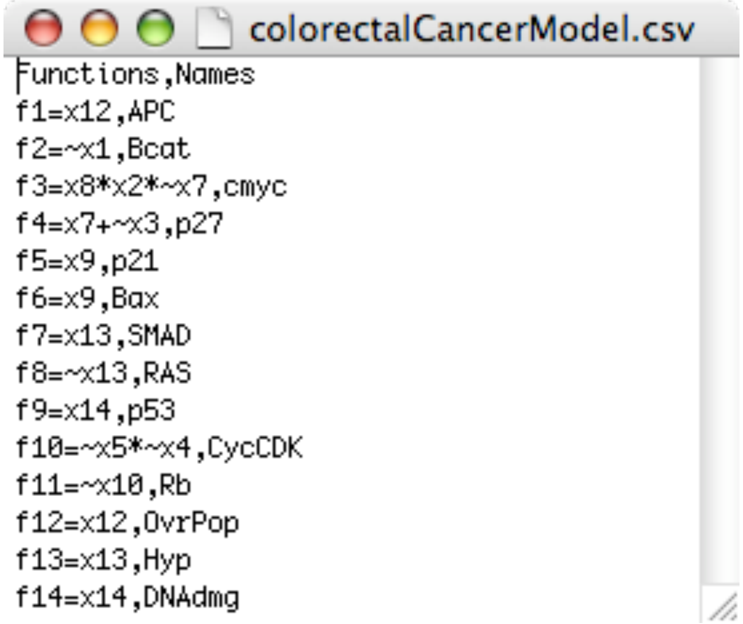
```
colorize(attractor,blue,red)
```

5. **Computing Betweenness Centrality and Resizing Nodes**

```
g.computeBetweennessCentrality()
resizeLinear(betweenness,4,20)
```

A Boolean Network Model of Cancer

Much of the work in discrete network work analysis takes place within the field of System's Biology and deal with gene and protein interaction networks. The models discussed in the beginning of this tutorial are meant to familiarize a user with the capabilities of Network Workbench for dynamic network analysis and how one might go about analyzing a multi-state discrete network model, but do not give an example of a real world use. In this section we present the Boolean network model described in Ribba et.al.'s paper *A multiscale mathematical model of cancer, and its use in analyzing irradiation therapies*[2]. The file for this model is available in Network Workbench's sample data under the name **colorectalCancerModel.csv**. The file should something like Figure 18.



```
Functions,Names
f1=x12,APC
f2=~x1,Bcat
f3=x8*x2*~x7,cmyc
f4=x7+~x3,p27
f5=x9,p21
f6=x9,Bax
f7=x13,SMAD
f8=~x13,RAS
f9=x14,p53
f10=~x5*~x4,CycCDK
f11=~x10,Rb
f12=x12,OvrPop
f13=x13,Hyp
f14=x14,DNAdmg
```

Figure 18: Cancer protein interaction model

This model describes mutations to common genes when they are exposed to the following three parameters:

- The presence of the **Overpopulation Signal**
- The presence of the **Hypoxia Signal**
- The presence of **DNA damage**

The presence of DNA damage activates *p53* which leads to apoptosis, or programmed cell death. The hypoxia signal is activated when there is a lack oxygen in the cell environment. This triggers *TGFβ/SMAD* which inhibits cell proliferation. The overpopulation signal causes the activate of *APC* which also inhibits cell proliferation. These interactions are evaluated given the parameters and then fed into higher scale continuous models.

We follow the steps given above to analyze a single attractor basin of this system. We will

- Load the model
- Enumerate the state space
- Extract the attractors
- Visualize one of the attractors

The results are found below.

[2] Ribba, B., Colin, T., Schell, S. "A multiscale mathematical model of cancer, and its use in analyzing irradiation therapies," *Theoretical Biology and Medical Modelling*. 2006, 3:7. doi:10.1186/1742-4682-3-7. Available from: http://www.tbiomed.com/content/3/1/7

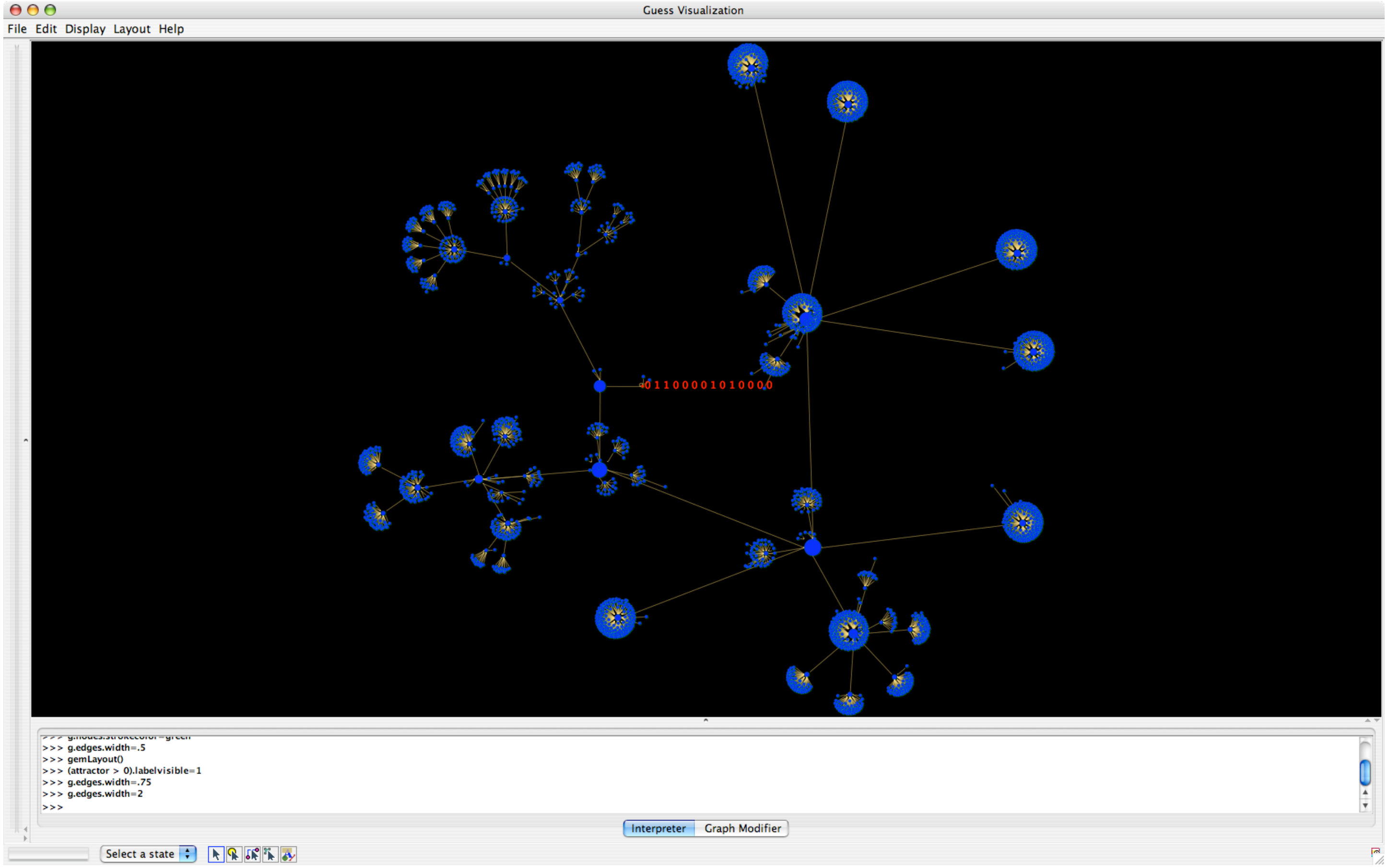
Results

After evaluating the model we examine the first attractor. This attractor corresponds to the situation in which there is no overpopulation, hypoxia, and DNA damage. Comparing our results with the description of the model in the paper we find that *Bax* is not activated and so there is no cell death. Furthermore, *Rb* is not activated so we have normal cell proliferation. This is the behavior we expect to see given normal cell conditions.

APC	Bcat	cmyc	p27	p21	Bax	SMAD	RAS	p53	CycCDK	Rb	OvrPop	Hyp	DNAdmg
0	1	1	0	0	0	0	1	0	1	0	0	0	0

Table 4: Attractor 1 of Colorectal Cancer Model

The state space graph of the first attractor basin in the colorectal cancer genetic regulation model is found in Figure 19. The nodes have been scaled by their betweenness centrality and the attractor has been highlighted in red. We've also added the label to the attractor for visibility's sake.



```
>>> g.nodes.strokecolor=green
>>> g.edges.width=.5
>>> gemLayout()
>>> (attractor > 0).labelvisible=1
>>> g.edges.width=.75
>>> g.edges.width=2
>>>
```

Figure 19: State Space Graph of Attractor Basin 1 of the Colorectal Cancer Model

This graph represents a qualitative analysis of one of the attractor basins. Examining the resizing, we can see that the attractor is not central to the basin, suggesting that this behavior is fragile. Further understanding of the experimental data should help clarify the behavior.

These models are very useful for visualizing the dynamics of biological systems, but they require knowledge of how the system is supposed to behave before they can be fully utilized.