

# ExpressionNet v1.0 Documentation

06.01.2005

Jingchun Zhu

<http://expressionnet.sourceforge.net>

# Table of Contents

<b>INSTALLATION.....</b>	<b>1</b>
<b>QUICK START.....</b>	<b>1</b>
1. LOADING A DEMO NETWORK.....	1
Figure 1. Graphic representation of a network .....	1
2. VIEWING THE NODE DEFINITION.....	2
Figure 2. Viewing a node definition .....	2
3. SCORING THE DEMO NETWORK .....	2
Table 1. Demo dataset (a sample).....	2
Figure 3. Network score.....	3
4. LEARNING A BETTER MODEL .....	3
Figure 4. A better-scored network using the demo data .....	4
<b>CONSTRUCTING A NETWORK GRAPHICALLY.....</b>	<b>4</b>
1. OPEN A NEW NETWORK DRAWING WINDOW .....	4
Figure 5. Selecting a new Bayesian network representation .....	4
Figure 6. A new Bayesian network.....	5
2. ADDING A NODE.....	5
Figure 7. Defining a node's property .....	6
3. ADDING EDGES .....	6
4. DELETING A NODE OR AN EDGE.....	6
<b>SCORING A NETWORK - HOW WELL DOES THE EXTERNAL DATA SUPPORT A NETWORK STRUCTURE .....</b>	<b>6</b>
1. FORMAT FOR EXTERNAL DATA FILE .....	6
2. LOADING EXTERNAL FILE TO SCORE A NETWORK.....	7
Figure 8. Load external data .....	7
3. THE CONDITIONAL PROBABILITY DISTRIBUTION OF A NODE .....	8
Figure 9. Conditional probability distribution table .....	8
<b>NETWORK LEARNING – FIND THE BEST MODEL BASED ON THE EXPERIMENTAL DATASET .....</b>	<b>8</b>
1. STARTING THE LEARNING PROCESS .....	9
Figure 10. Threshold for high-scoring networks .....	9
Figure 11. Output location .....	9
2. LEARNING.....	9
3. LEARNING RESULTS – BAYESIAN AVERAGE NETWORKS.....	10
Figure 12. Selecting a new average network representation.....	10
Figure 13. New Bayesian average network representation.....	10
Figure 14. Importing networks to construct a Bayesian average network....	11
Figure 15. A Bayesian average network.....	11
<b>REFERENCE.....</b>	<b>12</b>

## Installation

ExpressionNet runs under Windows<sup>®</sup> operating system. After downloading the program from <http://expressionnet.sourceforge.net>, you need to first decompress the .zip file, and then install the program by double clicking the *setup.exe* file. The program will be installed and ready to use. You will also find the demo data and this documentation in the directory where the program is installed.

To uninstall the program, you should use the Windows built-in “Add or Remove Programs” function.

## Quick Start

### 1. Loading a demo network

You can load the demo network by double-clicking the file “demo\_model.bn”. The graphic representation allows a user to move, add, or delete any nodes or edges.

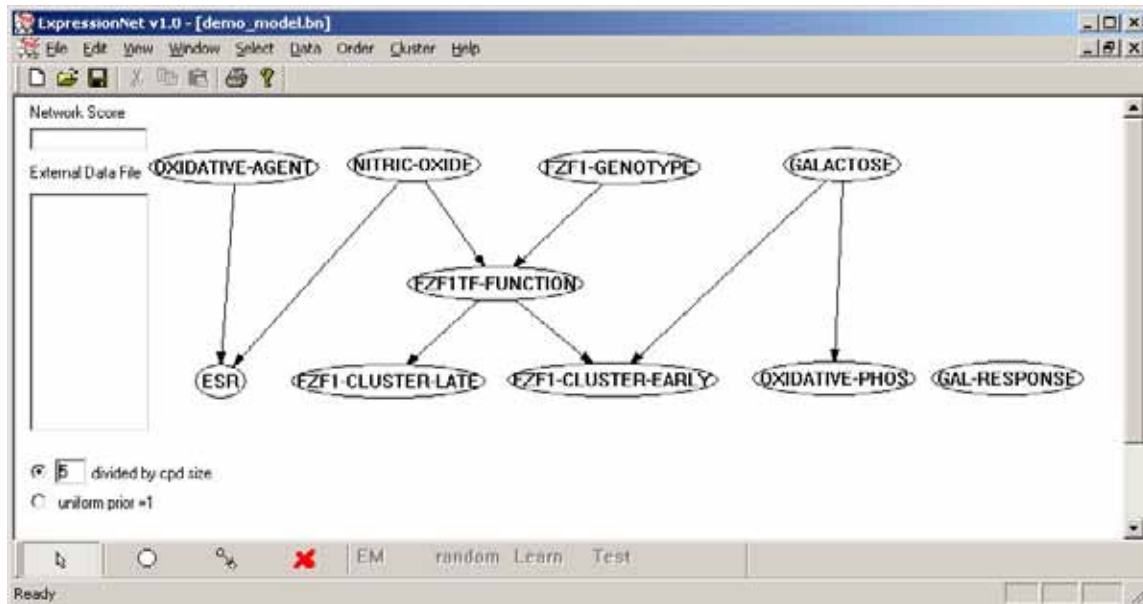



Figure 1. Graphic representation of a network

To add a node or an edge, you need to first select the new node  or the new edge



icon located on the bottom toolbar, and then click or drag the mouse through the main window to achieve the desired position.

You can change the position of an existing node by dragging it across the window. The edges connected to the node will be automatically repositioned.

## 2. Viewing the node definition

The node definition can be viewed by double clicking on a node. In this case is the node GALACTOSE.

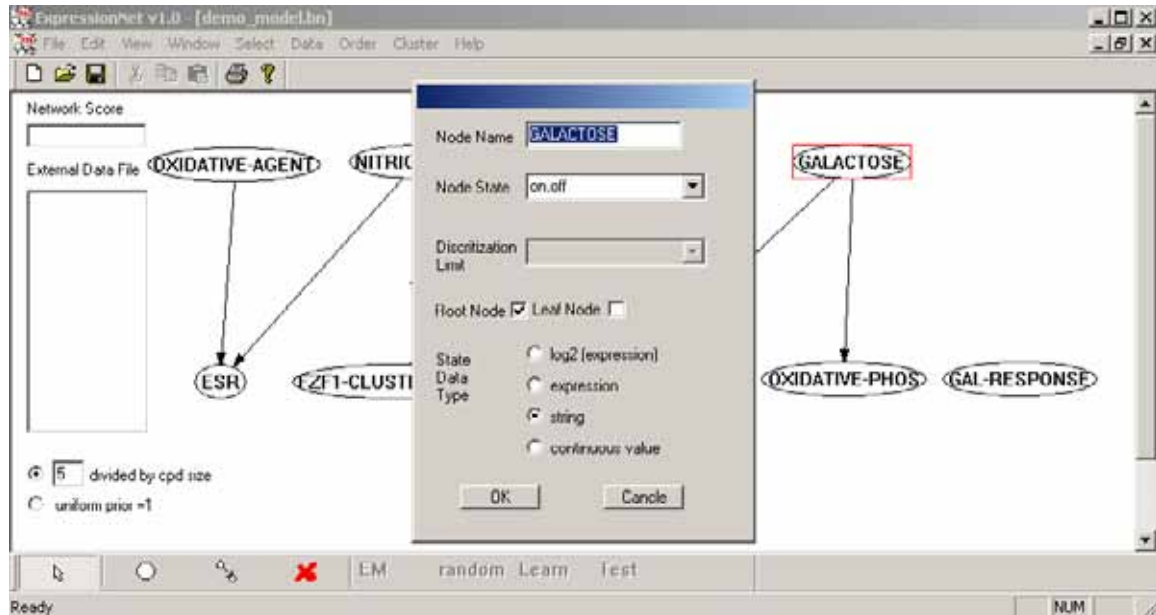


Figure 2. Viewing a node definition

## 3. Scoring the demo network

A network can be evaluated (scored) using an external dataset. The dataset is required to have a tab-delimited text file format and also matches the node definitions in the network model. A demo dataset has been supplied in the release.

NAME	nitric oxide gas wt (120 min)	DPTA del-fzf1 (120 min)	glucose to galactose 12	mM H2O2 (120 min) redo	Menadione (10 min) redo
ESR	1.59516	0.367379	-0.0327391	1.458	0.852
OXIDATIVE-PHOS	0.192507	0.962231	2.54073	0.275714	0.304286
GAL-RESPONSE	0.1658825	0.4220956	3.27862	0.1818055	0.197827
FZF1-CLUSTER-EARLY	3.838073	0.5030005	3.8089355	0.21	0.535
FZF1-CLUSTER-late	3.827986333	0.902787333	5.628219333	0.005	-0.165
NITRIC-OXIDE	120	120	0	0	0
GALACTOSE	off	off	on	off	off
FZF1TF-FUNCTION		off	on		
FZF1-GENOTYPE	wt	delete	overexpression	wt	wt
OXIDATIVE-AGENT	no	no	no	yes	yes

Table 1. Demo dataset (a sample)

The demo dataset can be loaded by selecting from the “data” menu on the main menu bar, and then followed by selecting “Load new data cases”, then selecting “TAB delimited text file” (Figure 8).

After the data file has been loaded, the network score and the data file will be shown in red on the left.

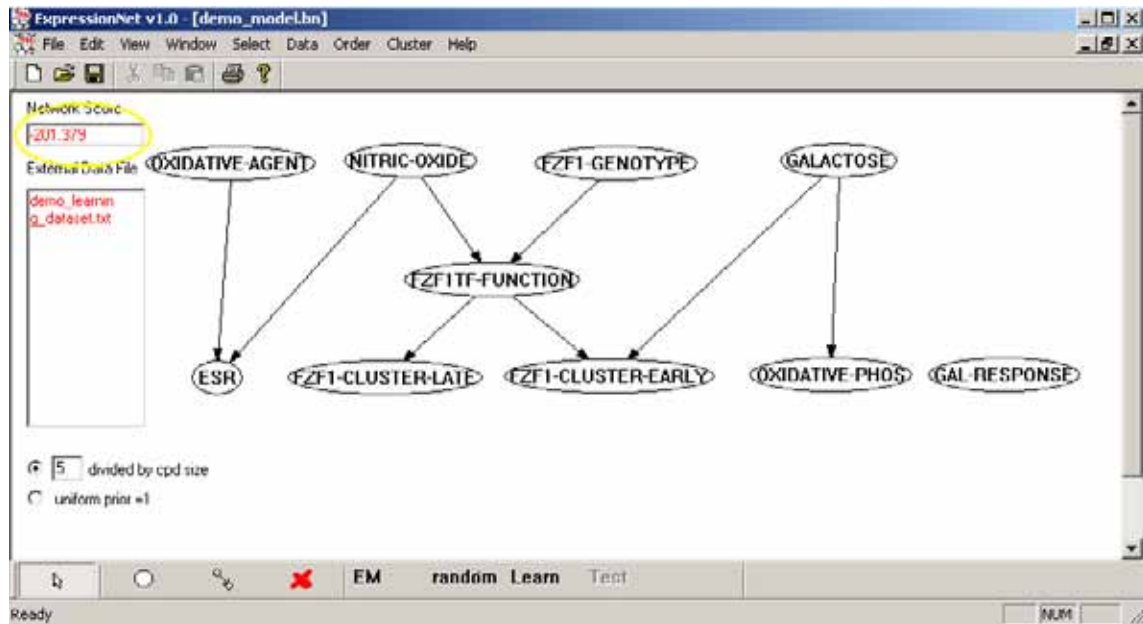


Figure 3. Network score

#### 4. Learning a better model

Given an observed data, a probability score can be assigned to a network. The score measures the likelihood of obtaining the observed data based on the assumed the network structure. The network score will change when the edge connection is modified or the dataset is altered. Since the score is calculated as  $\log_{10}$  of the likelihood, it is always a negative number ranging from *zero* to *-inf*. A smaller absolute value (higher score) means a better network. If we fix the dataset, we can search for a better model by searching for a different network structure (i.e. edge connection) that scores higher. In the case of the demo network, we can add a new edge from the node GALACTOSE to the node GAL-RESPONSE to obtain a better-scored network.

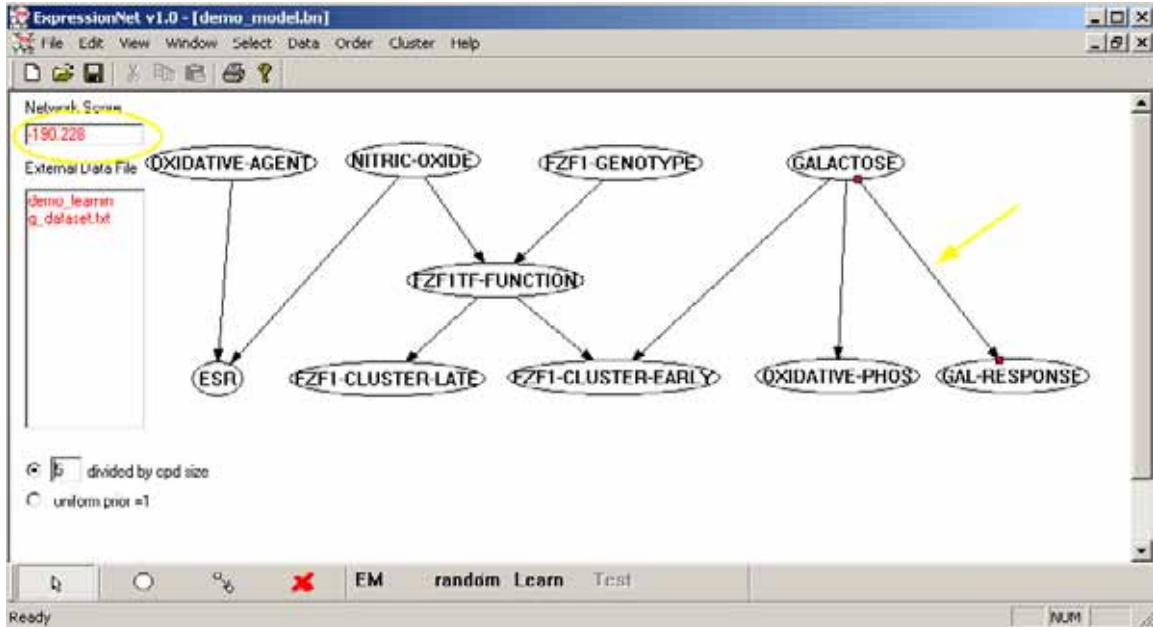


Figure 4. A better-scored network using the demo data

## Constructing a network graphically

ExpressionNet provides users graphing functionalities to construct a Bayesian network in the main window.

### 1. Open a new network drawing window

You can open a new network using “File” -> “New”.

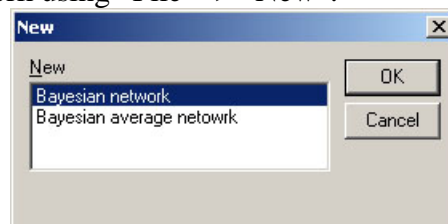


Figure 5. Selecting a new Bayesian network representation

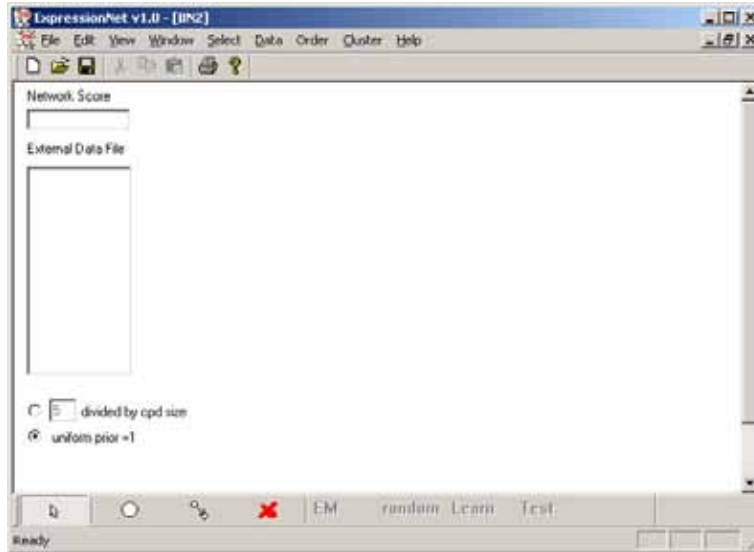


Figure 6. A new Bayesian network

## 2. Adding a node

Users can start constructing the network by adding nodes. To add a node, you first select




from the bottom toolbar, and then click on the main window. The node definition dialog box will appear, which is used to define or view the node name, state values and etc (Figure 7). Node name is the label that will be shown graphically. Node states are the labels of the discrete state values, which should be a list of text separated by comma. If a node takes continuous values, they need to be converted into discrete values using a series of cutoffs, which should be specified in the “Discretization Limit” box. In addition, one of the three appropriate “state data type”: log<sub>2</sub> (expression), expression, or continuous value should be checked for continuous value nodes. If the node takes discrete values, check the “state data type” as “string” and specify the state values. If the node must be root or leaf node, you need to check the corresponding box; otherwise, leave them unchecked.


**Figure 7. Defining a node's property**

### 3. Adding edges



To add an edge, you first select  from the bottom toolbar, and then simply drag your mouse from the parent node to the child node.

### 4. Deleting a node or an edge

Nodes and edges can be deleted by first select the objects and then using “Delete” key or clicking  from the bottom toolbar.

## Scoring a network - How well does the external data support a network structure

A network can be scored (or evaluated) using an external dataset. The score measures the likelihood of deriving the observed external data based on the network structure. The value of the score is  $\log_{10}$  of the likelihood, which ranges from zero to  $-\infty$ . A smaller absolute value means that the corresponding network structure matches the external data better; therefore it is a better-scored network.

### 1. Format for external data file

The external file format is required to be TAB-delimited text file. Table 1 illustrates a sample file. The first column specifies the node name. The order of the names is not important, but they must match the node name in the graphic representation. Each remaining columns represents an independent experiment.



The first row records the name of the experiments; the names are not important for ExpressionNet. Each remaining row has data for a specific node, which must match the node states defined for the corresponding node (section Constructing a network graphically). If the node is defined as taking discrete string values, the data must match one of the state values in the node definition. For example, row GALACTOSE must have either “on”, “off” in Table 1. If the node takes continuous values (a gene expression value, a  $\log_2$  (expression) value, or generic continuous value) the data are required to be a number, such as the row ESR.

Incomplete data can be left as empty in the data file.

If a data point is set by intervention, a “\*” character should be added at the end the data point, such as “overexpression\*”.

If the data file does not match the network node definition, the file will be not be accepted by the network.

## 2. Loading external file to score a network

If the format of the data file matches the nodes definition, it can be loaded into the network. Otherwise, error message will be generated during loading. Users need to exam both the file format and node definitions to solve the problem.

Two methods are available to load the data. The first method is to use the “Data” menu on the main menu toolbar (Figure 8).

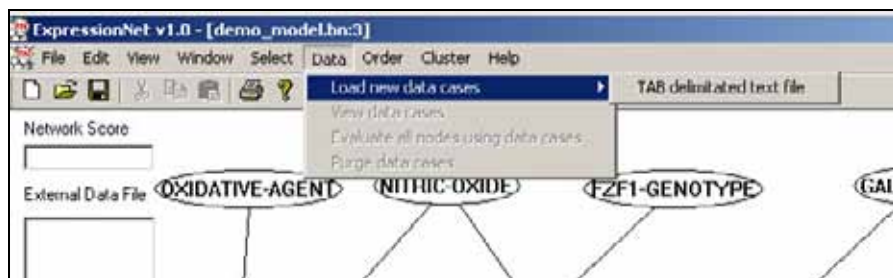


Figure 8. Load external data

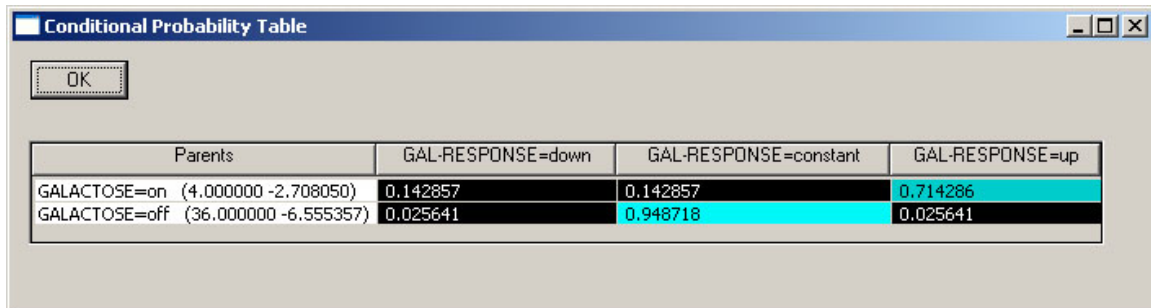
The second method is to click on radio selection buttons for prior settings.

After the data is successfully loaded, the network score and the external data file are shown on the left side of the window (Figure 3). The data file can also be purged from the network using the “Data” menu in a similar fashion as the first loading method.

### 3. The conditional probability distribution of a node

In addition to obtaining a network score using an external data set, we also obtain the conditional probability distribution (CPD) of each node. To view a node's CPD, a user can right click the node and then select "View Data CPD".

Figure 9 shows the CPD of the "GAL-RESPONSE" node in the demo network. Each probability parameter occupies a cell, as for a specific parent-child states combination. The first column shows the parent node states. The first row specifies the child node state. For example, given GALACTOSE =ON, the probability of "GAL-REPOSE"=up is 0.74.



The screenshot shows a window titled "Conditional Probability Table" with an "OK" button. The table contains the following data:

Parents	GAL-RESPONSE=down	GAL-RESPONSE=constant	GAL-RESPONSE=up
GALACTOSE=on (4.000000 -2.708050)	0.142857	0.142857	0.714286
GALACTOSE=off (36.000000 -6.555357)	0.025641	0.948718	0.025641

Figure 9. Conditional probability distribution table

Sometimes, the dataset is incomplete. ExpressionNet is still able to estimate the parameters under those conditions using EM algorithms (1). To illustrate the difference, ExpressionNet uses the background color to code for the percentage of actual data used to estimate the parameters. A brighter background color represents a higher percentage of actual data. The black background represents the parameter is entirely estimated computationally. In Figure 9, the probability of GAL-RESPONSE = down given the condition of GALACTOSE=on is estimate computationally ( $P = 0.12$ ).

## Network Learning – Find the best model based on the experimental dataset

We used a Bayesian scoring function to assign a probability score for a network model and the clique-tree and the variable elimination algorithms for efficient inference and learning (2, 3). The learning process started with random edge combinations, gradually improving the network topology using a greedy search strategy until the score reached a local maximum. The greedy search was iterated to generate a collection of high scoring networks. High scoring networks were subjected to further small topology changes (single edge addition, deletion or reversion) to expand the collection. Learning was repeated using two different prior probability distributions of the network parameters (priors), both set as a Dirichlet distribution:  $Dir(1,1, \dots, 1)$  and  $Dir(P_0 \cdot \alpha, P_0 \cdot \alpha, \dots, P_0 \cdot \alpha)$ , where  $P_0$  is a uniform distribution over the probability space of each CPD and  $\alpha=5$ . Networks scoring within a percentile cutoff using both priors were used to construct

derived Bayesian average networks model. Missing values was handled using a Structural Expectation-Maximization algorithm (1).

## 1. Starting the learning process

After the nodes have been constructed and data file has been successfully loaded, a user can start the learning procedure by clicking the learning icon **Learn** located on the bottom toolbar.

The learning procedure starts by asking the user to decide the percentile cutoff within which high scoring networks will be defined. The percentage threshold is defined as the top percentage of the all networks evaluated by the learning process sorted by their network scores in descending order.

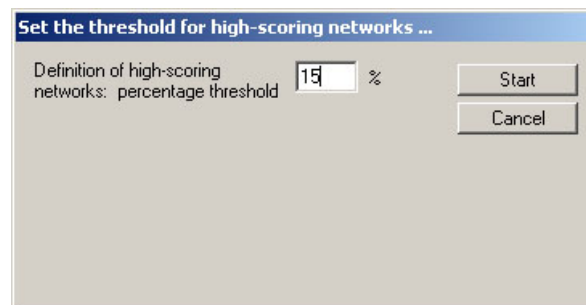


Figure 10. Threshold for high-scoring networks

The next step is to assign the output folder where the high scoring networks, log and intermediate files will be stored.



Figure 11. Output location

After a few more confirmation steps, the learning process will then start.

## 2. Learning

The first phase of the learning process performs greedy search (using both prior parameters) to generate an initial set of high-scoring networks. It followed by the second phase to expand the collection by make local structural modifications to the existing

high-scoring networks. If the process is cancelled during the first phase, no results will be generated.

The second phase can be cancelled at any time, and the resulting high-scoring networks will be stored in the series of subfolders located in the output directory (Figure 11). Those subfolders are named as integers, such as 15, 13 and 11. The integers represent the various percentile cutoffs ranging from the user-defined cutoff to zero.

### 3. Learning results – Bayesian average networks

Using the high-scoring network collection in each of the subfolders named with integers, users can generate Bayesian average network using various percentile cutoffs.

To do so, the user need to first initiate a new network in the average network representation (Figure 12, Figure 13).

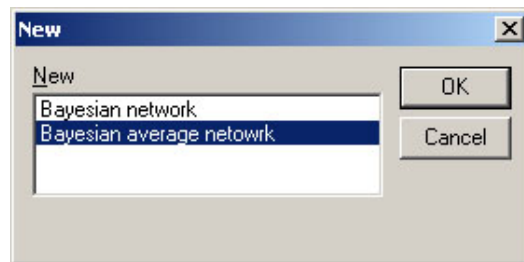


Figure 12. Selecting a new average network representation

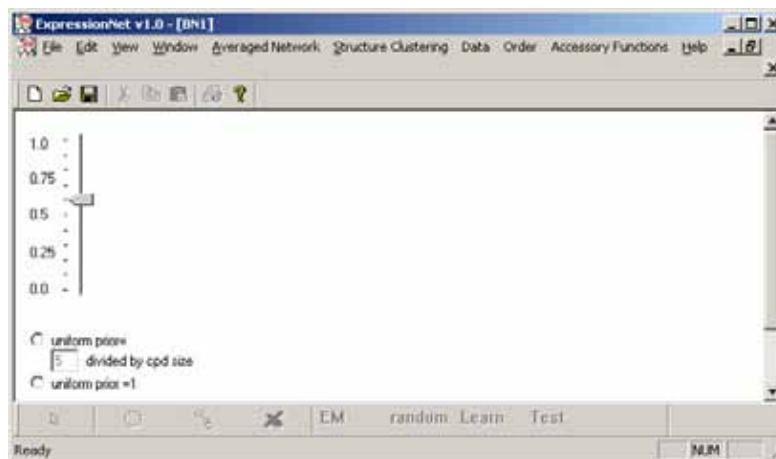
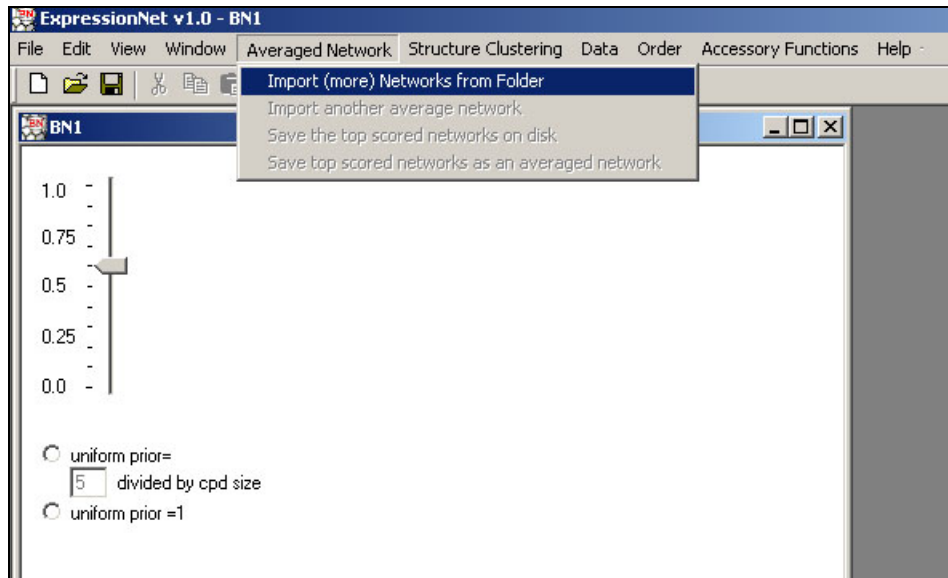


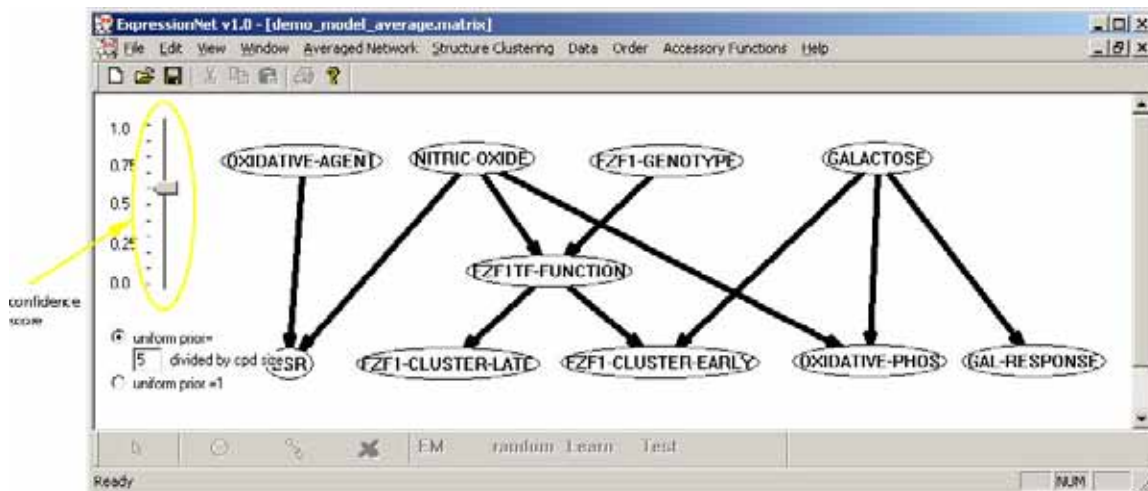
Figure 13. New Bayesian average network representation

After a new average network representation is open, the user can import Bayesian networks from a folder by selecting “Averaged Network” from the main menu, and then select “Import (more) networks from folder” (Figure 14).



**Figure 14. Importing networks to construct a Bayesian average network**

After selecting the folder for import, A Bayesian average network will now be constructed using networks in the selected folder (Figure 15).



**Figure 15. A Bayesian average network**

A confidence score is associated with each edge, which is represented graphically by the thickness of the edge. The slider bar on the left side of the window shows the confidence score cutoff. Edges with greater confidence scores than the cutoff will be shown. By dragging the slider up and down, less confident edges will disappear or appear in the window. The average network can also be saved to disk.

## REFERENCE

1. Friedman, N. (1998) in *Uncertainty in Artificial Intelligence. Proceedings of the Fourteenth Conference*, ed. S., C. G. M. (Morgan Kaufmann Publishers, Madison, WI, USA), pp. 129-38.
2. Jordan, M. I. (1999) *Learning in graphical models* (MIT Press, Cambridge, Mass.).
3. Heckerman, D., Geiger, C. & Chickering, D. (1995) *Machine Learning* **20**, 197-243.