Bayesian Network Wizard: user-friendly Bayesian networks learning

Fulvia Ferrazzi and Riccardo Bellazzi

Dipartimento di Informatica e Sistemistica, Università degli Studi di Pavia, Pavia, Italy

Bayesian Network Wizard is a tool to learn different types of Bayesian networks (static/dynamic) with continuous or discrete variables. The software is simple to use thanks to a wizard-based user-friendly interface that guides the user through all phases of network learning, from data loading to model selection, and from learning the network to its graphical visualization.

Availability: Compiled software for Windows-based systems and Matlab source files are downloadable at http://aimed11.unipv.it/BayesianNetworkWizard

Contact: fulvia.ferrazzi@unipv.it

Bayesian Network Wizard is described in Ferrazzi and Bellazzi, Bioinformatics, 2009.

Introduction

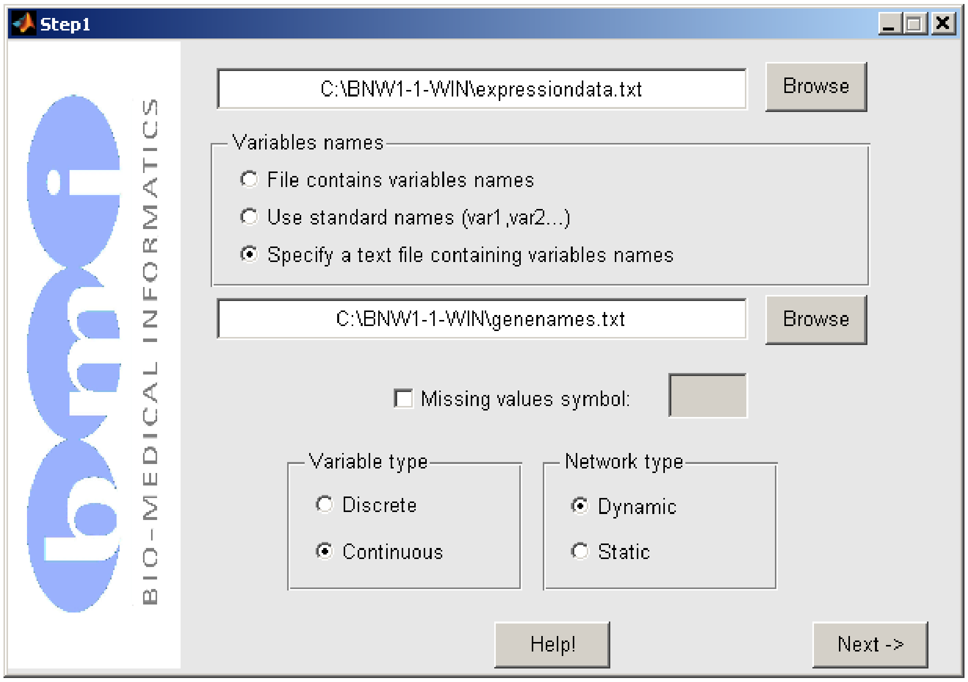
With this tutorial we will describe a typical use of Bayesian Network Wizard.
We will learn a dynamic Bayesian network with continuous variables from temporal expression data relative to 20 well characterized cell cycle genes. The data come from a study performed at Stanford University to identify genes periodically expressed during the human cell cycle (Whitfield ML et al., Mol Biol Cell. 2002 Jun; 13(6):1977-2000).

To use the compiled version of the software, follow the installation instructions contained in the readme file. If you have Matlab and want to use the program from the source files, start the application by typing “main” at the command prompt (please note that the tool has been developed with Matlab R2007b). In this case, in order to enable the graphical visualization of the networks, you need to add to the Matlab classpath the directory containing the source files as well as the jar files contained in it.

Step1

The dataset is contained in the file expressiondata.txt and gene names are contained in genenames.txt. In order to load the dataset file, let's click on Browse, choose the file expressiondata.txt and then click on ‘Open’. In order to also load the gene names file, let's select the option ‘Specify a text file containing variables names’: in this way another Browse button will be activated, allowing us to load the file genenames.txt.
The data file contains no missing values; thus we can avoid specifying the symbol employed to indicate them. As the network we are going to learn is dynamic with continuous variables, we should choose Continuous as Variable type and Dynamic as Network type.

The screen will appear as in the following screenshot:



C:\BNW1-1-WIN\expressiondata.txt

Variables names

○ File contains variables names
○ Use standard names (var1,var2...)
● Specify a text file containing variables names

C:\BNW1-1-WIN\genenames.txt

□ Missing values symbol:

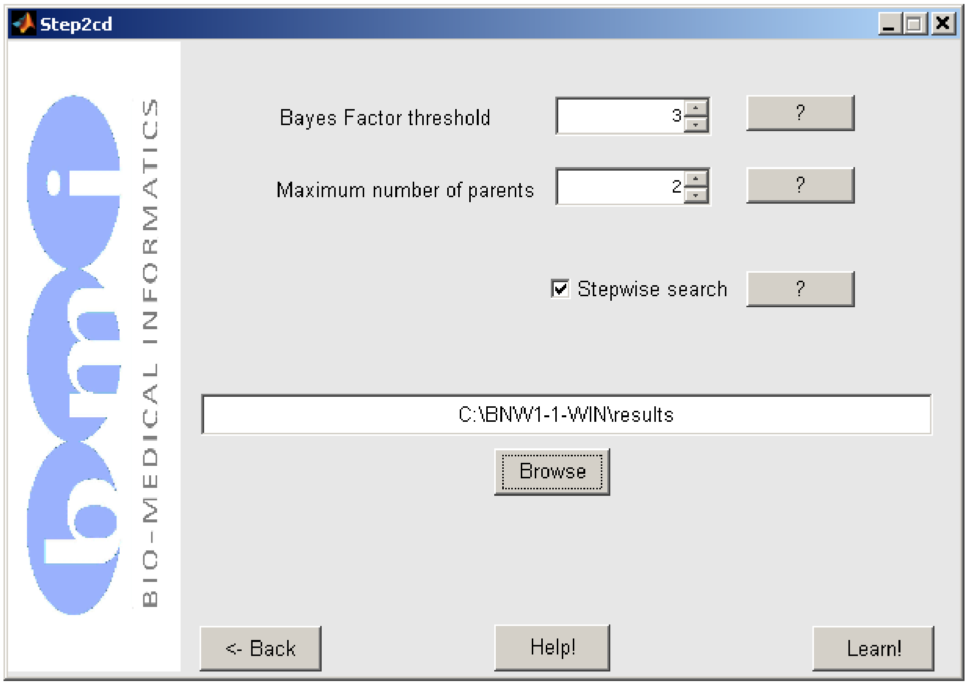
Variable type
○ Discrete
● Continuous

Network type
● Dynamic
○ Static

Let's click on Next and proceed to the second Step.

Step2

At this step, the screen relative to continuous dynamic networks will appear and the user needs to choose the values for some algorithmic parameters. The buttons indicated with “?” provide an explanation for each parameter. It is also necessary to choose the directory in which a text based representation of the network structure (GML file) will be saved, together with a text file containing the learned network parameters.



Step2cd

Bayes Factor threshold 3

Maximum number of parents 2

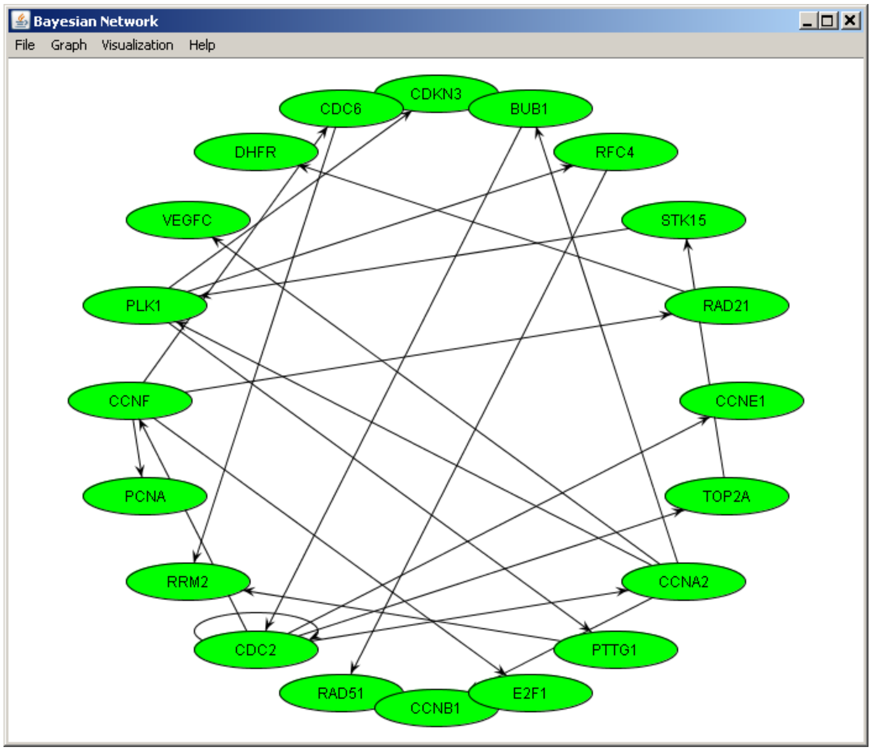
☑ Stepwise search

C:\BNW1-1-WIN\results

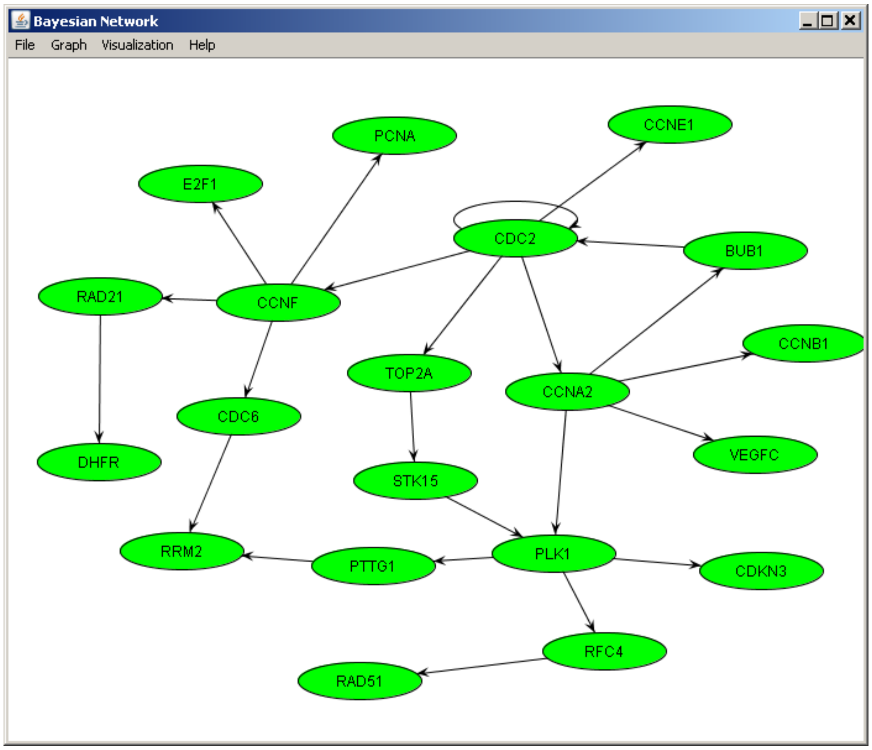
Both the window relative to Step1 and the one relative to Step2 contain a “Help!” button that provides indications about the use of the tool.

Graph

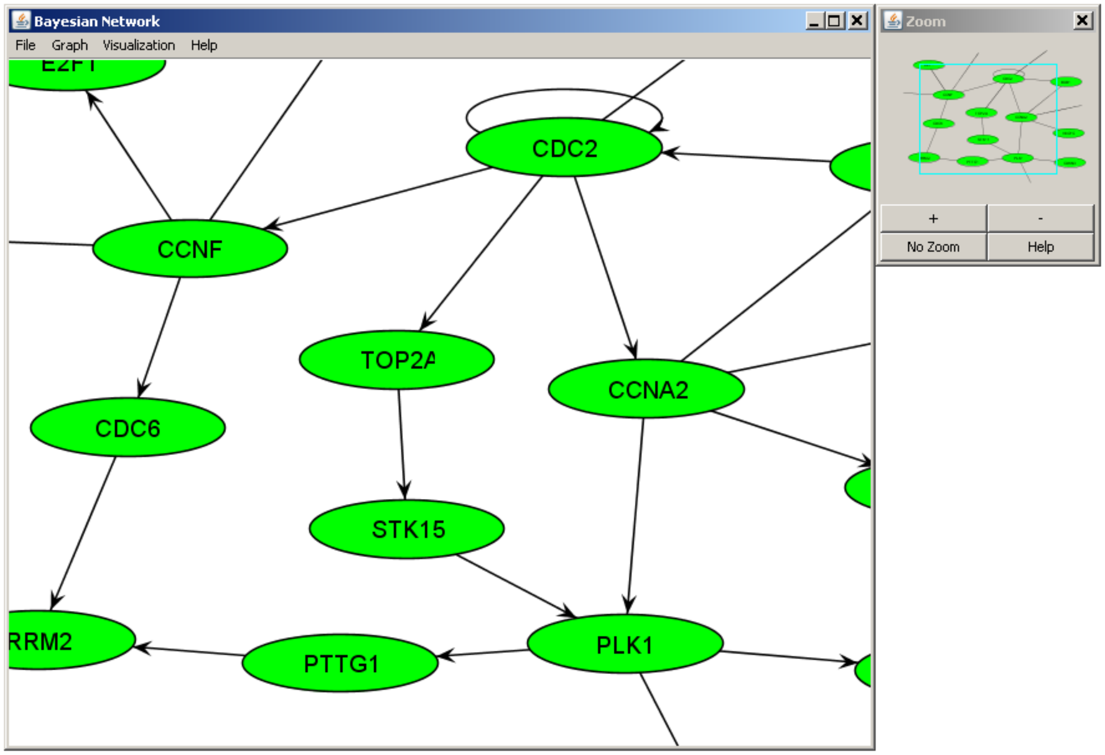
Once learning has been completed, a window containing a visual representation of the network opens.



In order to have a clearer visualization of the links between nodes, it is possible to choose a different layout. Let's click on the menu ‘Graph’ and from the submenu ‘Layout’ let's choose ‘KK Layout’. The result of the operation will be analogous to the following:



It can happen that also with this different layout it is not possible to well distinguish incoming and outgoing arcs from a node. A valid help in this case is given from the zoom functionality: clicking on ‘Visualization’ and then ‘Zoom’, a small window will appear on the right of the graph, with the commands to enlarge, make smaller or revert the graph to default size. In addition, the small window shows a preview of the graph, in which the graph area interested by the zoom operation is enclosed in a blue rectangle. By dragging the rectangle the involved region changes accordingly, and by clicking on the rectangle edges it is possible to resize the zoomed area.



Saved results

The text based representation of the learned network (GML file) saved in the Results folder can be used to visualize the network with powerful graphical tools such as Yed (http://www.yworks.com/en/products_yed_about.html).

In the same folder the file network_parameters.txt is saved. This tab delimited file contains the learned network parameters corresponding to each variable, given as in the following example relative to gene PLK1:

```
Variable: PLK1
                  Variance:     0.285
                  Beta:        -0.011     0.683     0.43
                  Parents      constant   CCNA2     STK15
```

The algorithm employed for continuous variable networks (see References) learns a regression model for each variable, estimating its Variance (first line) and regression parameters (Beta, second line), corresponding to the variable's Parents in the network (third line). In correspondence of the constant regression term, 'constant' is written in the Parents line.

<u>Note</u>:

In the case in which a Bayesian network with discrete variables is learned (for example from the file 'SNP_data.txt'), the network parameters are conditional probabilities. In this case the network parameters file contains tables analogous to the following example (referring to node SNP30):

Node: SNP30

SNP26	SNP28	0	1	2
0	0	0.817	0.176	0.007
1	0	0.992	0.004	0.004
0	1	0.564	0.435	0.001
1	1	0.968	0.016	0.016

The parents of SNP30 are SNP26 and SNP28. Each row in the table contains the conditional probability of SNP30 taking on value '0', '1' or '2' in correspondence of all possible combinations of parent values.

References

Algorithms employed for network learning:

- for discrete variable networks, BNW employs the functions made available in the Bayesian network toolbox developed by Kevin Murphy (http://www.cs.ubc.ca/~murphyk/Software/BNT/bnt.html);
- for continuous variable networks, BNW employs the linear Gaussian algorithm described in Ferrazzi F, Sebastiani P, Ramoni MF, Bellazzi R. Bayesian approaches to reverse engineer cellular systems: a simulation study on nonlinear Gaussian networks. BMC Bioinformatics. 2007 May 24;8 Suppl 5:S2. (http://www.biomedcentral.com/1471-2105/8/S5/S2).