

# POULIN • HUGIN

PATTERNS & PREDICTIONS

Patterns and Predictions is a simplified tool for predictive analysis.

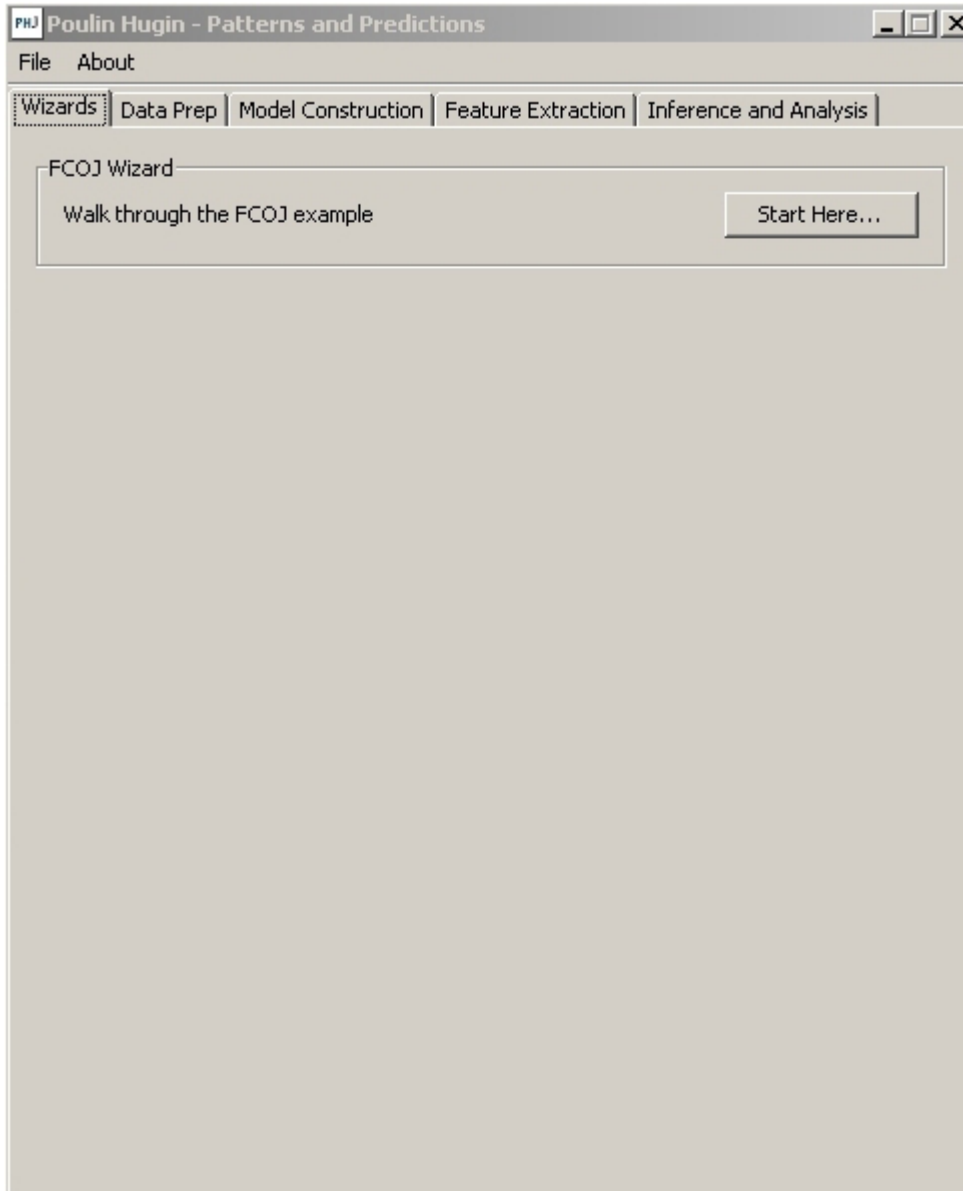
Utilizing technology based on Bayesian statistics, the system predicts future events when given a simple data set.

This same technology is used by many of the world's largest companies, and now a simplified version is available to you the individual researcher.

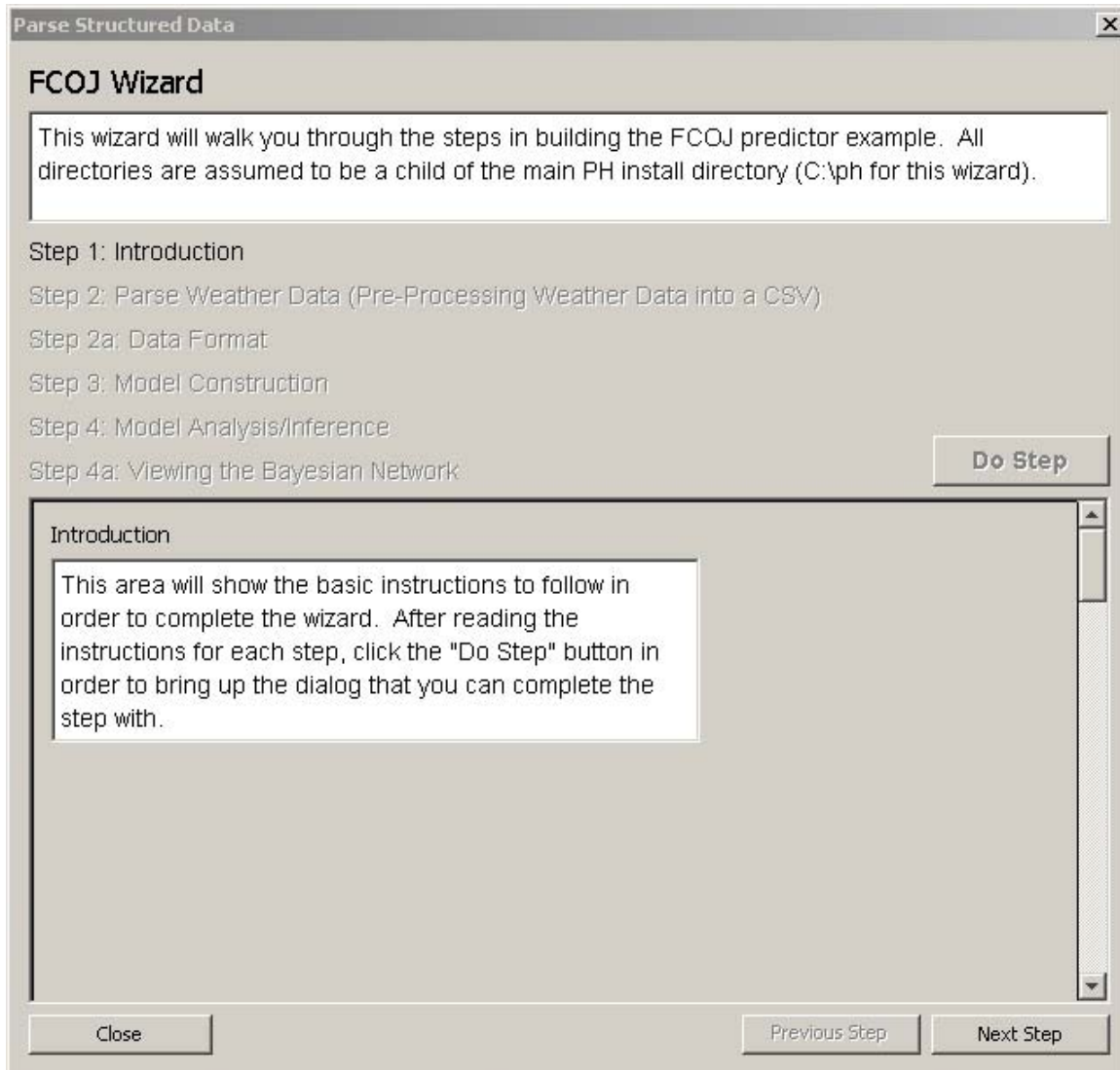
This tutorial will serve to instruct you on the usage of the system.



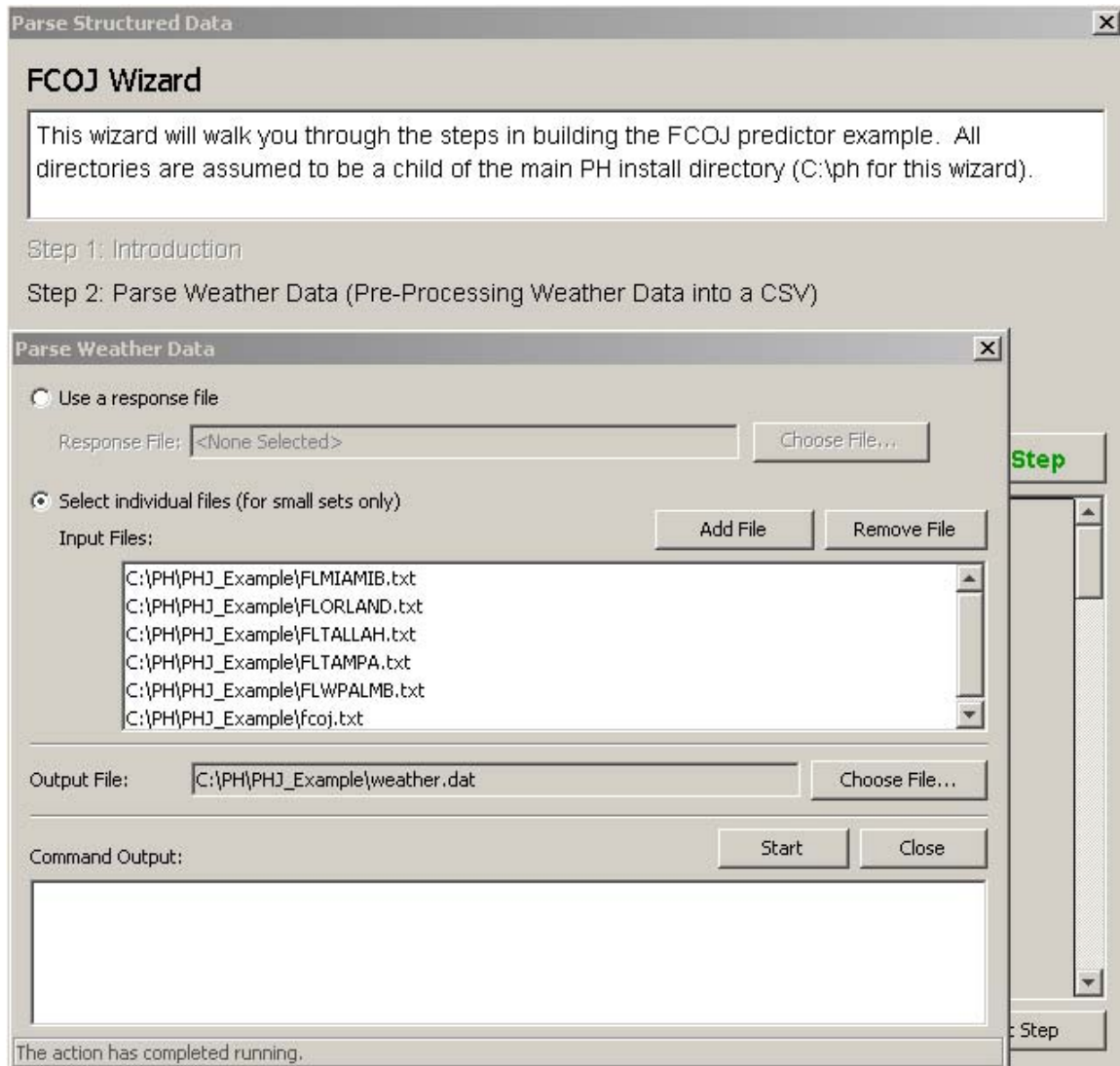
By accessing the PHJ Wizard application, the user has a graphical capability to learn the functions of the tool.



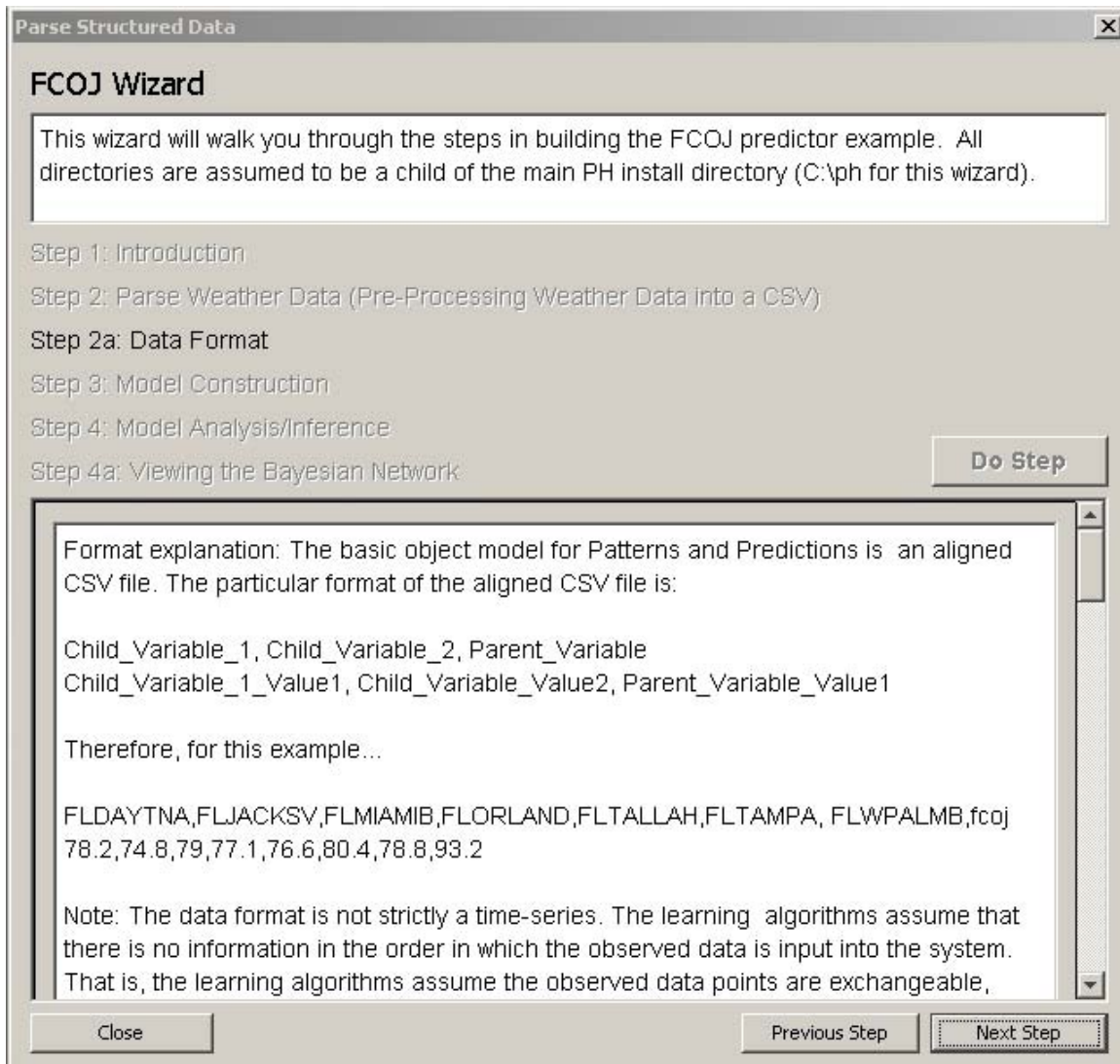
‘Start Here’. We will then be walked through the process of building and using a predictive model for the price of Frozen Concentrated Orange Juice using Florida weather data.



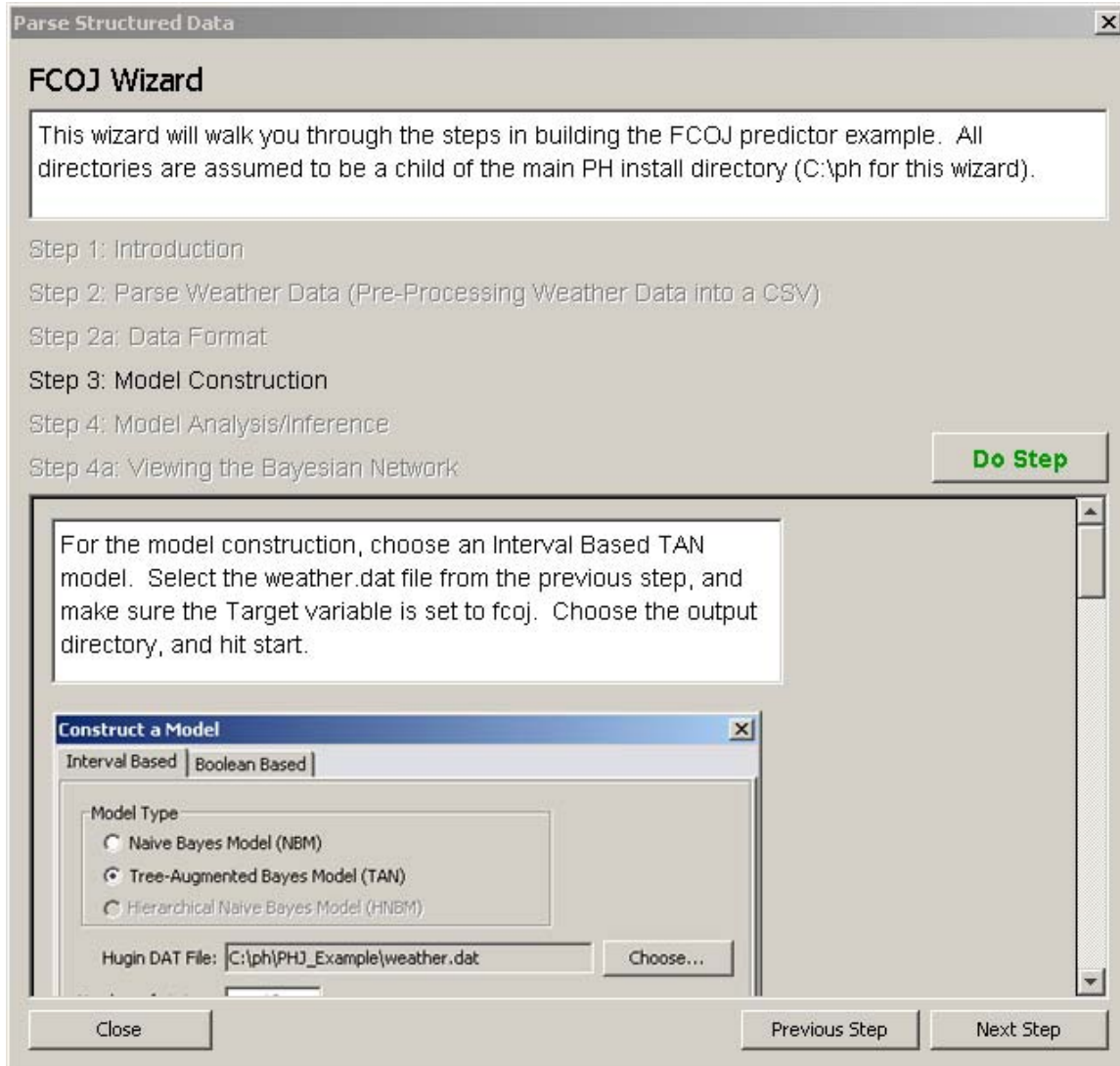
The initial wizard view explains that our starting path is the subfolder C:/PH/. Meanwhile for each step, the written instructions will be in the window pane. Use the scrollbar to view all content. Finally when appropriate, the wizard will prompt the user with a DO STEP in the mid right corner of the box.



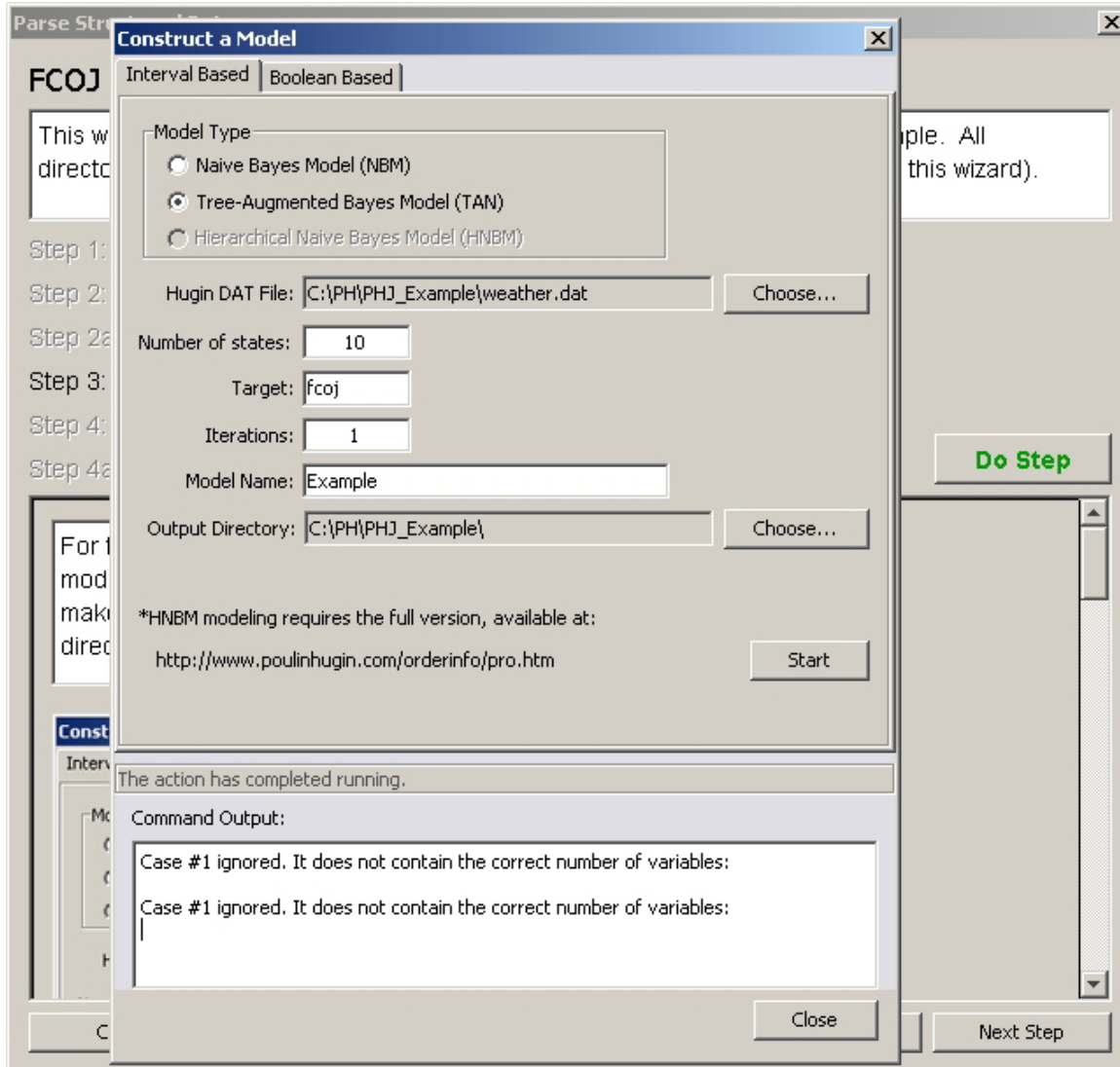
In this next step, we begin the process of building a data set. In this case, the weather data must be processed and combined with the FCOJ price information to produce a data file (DAT). You accomplish this by adding a collection of weather data files and the fcoj prices file all in NCDC format. In this case, when we generate the DAT file the appended the fcoj prices file is the last entry.



In this next step, we momentarily divert from the process to explain what happened in the last data processing step. The DAT file produced is essentially a CSV file, native to Microsoft's Excel. Specifically, the Child (Predictor) variables are used to predict the attributes of the Parent (Target) variable. The Child variables are the weather data values, while the Parent is the FCOJ price. Please note that any set of Child variables or single Parent variable could be used in this way interchangeably.

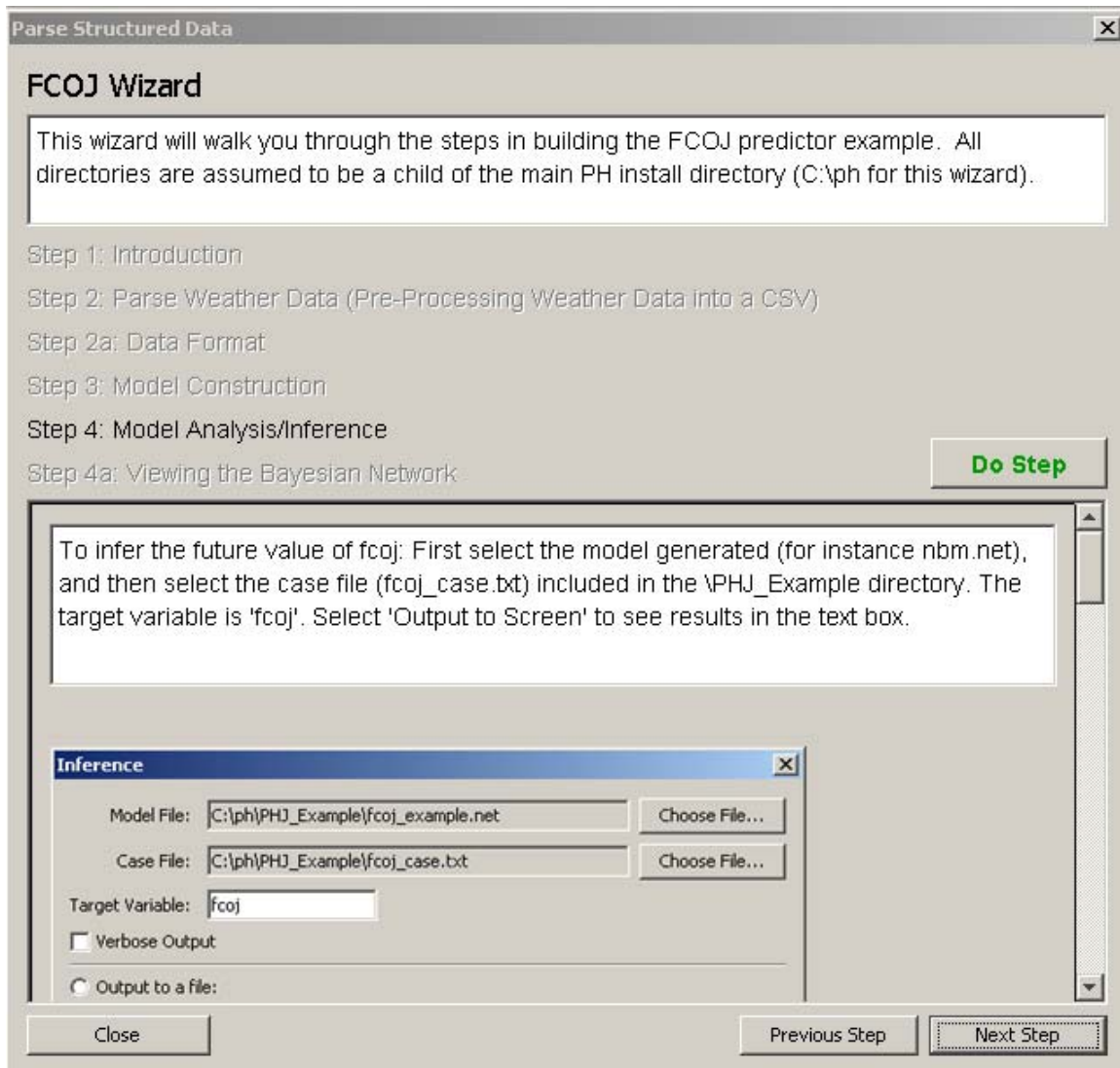


In this next step, we actually build a predictive model based on the data we generated. In this case, we provide the ability to build an NBM (or Naive Bayesian Model) or a TAN (Tree-Augmented Bayesian Model). The difference between the two deals mainly with the optimization of the model structure. NBM models generally have faster build times and query performance, while TAN models are sometimes slightly more accurate. Note: the Hierarchical Bayesian model listed is only available in the full professional version. Hierarchical models are an advanced model type and allow for such things as the discovery of ‘hidden’ or ‘dark’ patterns in data.

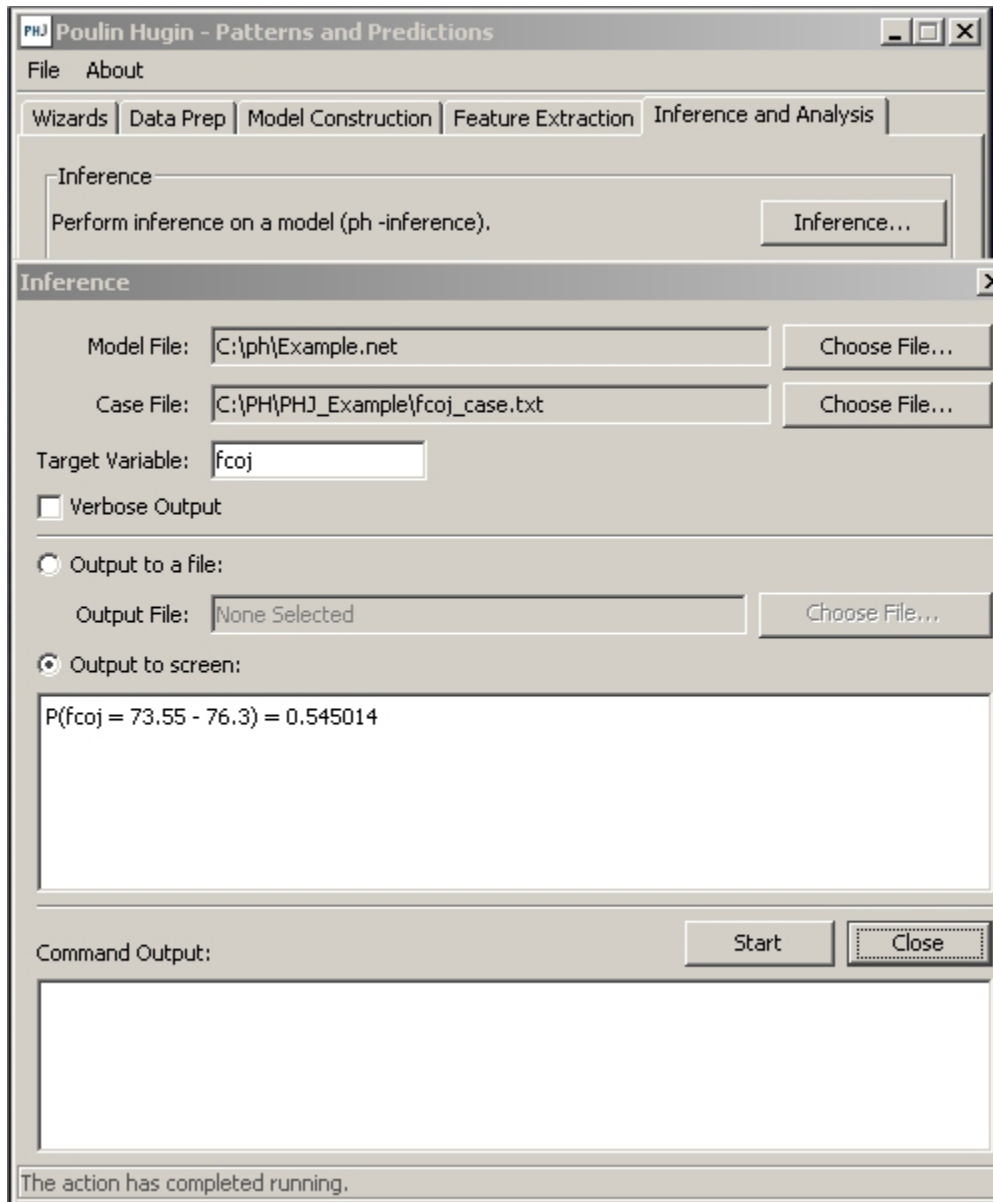


When you click 'DO STEP' we are prompted to choose the model type, the DAT file input, the number of states, iterations, model name, and output directory. The model type is again NBM or TAN. The DAT file is simply the file that we created in the previous steps. The 'number of states' is the number of divisions, increments or 'bins'. I.e. the total range is a number line where we set the number of divisions of the line as states. 'Target' must be the Parent variable that we generated in our DAT file, and is the variable we want to know the predicted value of. 'Iterations' is simply an input for the number of times that we would like to run through the generation of the model. Finally, you can select a custom Model Name and Output directory, though the program will have default values. Note: Any warning about Cases is merely to state the alignment of the actual file.





In this step, we have built a model and we can now perform an 'Inference' or Prediction. We will have to predict the value of FCOJ based upon the value of a 'Case'. A case file is simply a set of future/present conditions that are similar to the historical data that we used to build our model. As such, when the Case is compared against the existing model the system derives the prediction.



When clicking 'DO STEP', you enter the Model file name, and a Case file name. You must then select the Parent/Target variable, which in this case it is fcoj (lowercase) and then we can output the predictive results to a screen or txt file. In this example, we see that the value of fcoj within P is the price range in cents per pound, and the right hand value is the percentage likelihood of the occurrence.

**Parse Structured Data**

### FCOJ Wizard

This wizard will walk you through the steps in building the FCOJ predictor example. All directories are assumed to be a child of the main PH install directory (C:\ph for this wizard).

Step 1: Introduction  
 Step 2: Parse Weather Data (Pre-Processing Weather Data into a CSV)  
 Step 2a: Data Format  
 Step 3: Model Construction  
 Step 4: Model Analysis/Inference  
 Step 4a: Viewing the Bayesian Network

**Do Step**

To graphically view and modify your Bayesian network for advanced features such as Expert analysis and input, please upgrade to Hugin Explorer:  
[http://www.hugin.com/Products\\_Services/Products/Commercial/Explorer/](http://www.hugin.com/Products_Services/Products/Commercial/Explorer/)

```

graph TD
  fcoj((fcoj)) --> FLDAYTNA((FLDAYTNA))
  FLDAYTNA --> FLORLAND((FLORLAND))
  FLDAYTNA --> FLJACKSV((FLJACKSV))
  fcoj --> FLORLAND
  fcoj --> FLJACKSV
  
```

**Close** **Previous Step** **Next Step**

Finally in this step we illustrate how the model looks visually using the full commercial software suite provided by the Hugin Explorer tool. In this case, we have used Hugin Explorer to export a visual representation of a TAN prediction network.

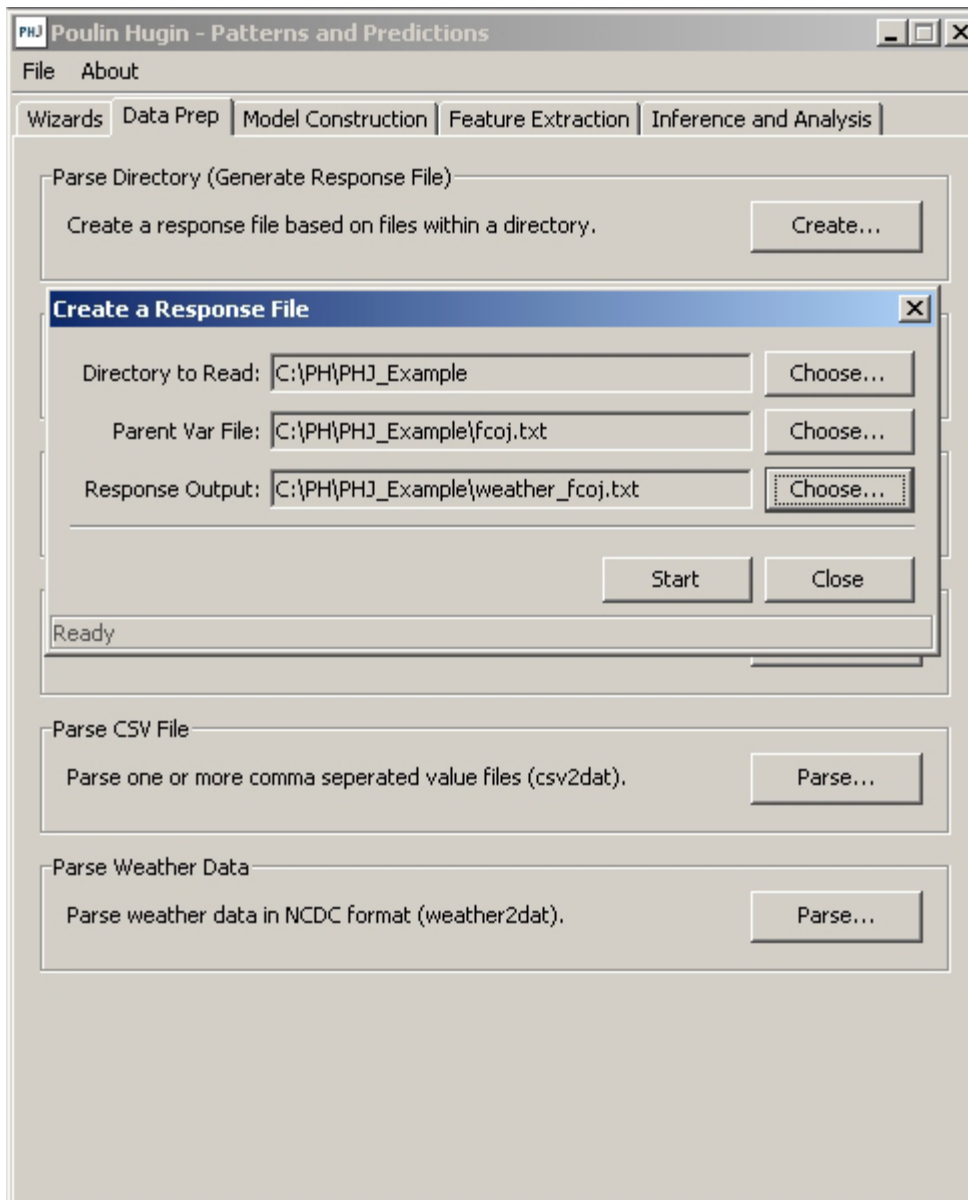
## ADVANCED FEATURES

- Response file generator

Response files are simply the contents of a directory with the Parent variable appended to the end of the document. They are useful for speeding up data pre-processing by allowing the automation of data collection.

For our FCOJ example, a small response file would be 'response.txt' illustrated here:

```
C:\PH\PHJ_Example\FLDAYTNA.txt  
C:\PH\PHJ_Example\FLJACKSV.txt  
C:\PH\PHJ_Example\FLMIAMIB.txt  
C:\PH\PHJ_Example\FLORLAND.txt  
C:\PH\PHJ_Example\FLTALLAH.txt  
C:\PH\PHJ_Example\FLTAMPA.txt  
C:\PH\PHJ_Example\FLWPALMB.txt  
C:\PH\PHJ_Example\fcoj.txt
```



To use the Response File Generator, under both the 'Data Prep' and File drop down selections you can select 'Generate a Response File'. You will then be prompted to select the directory you want to scan, the parent variable you want to define for observation, and finally the resulting file name. This process will then scan the directory, exclude the parent (observed) variable and then append it to the end of the file, and finally it will output a valid response file to be used in other data parsing steps of the application.

- Command line access

```

c:\ Command Prompt
C:\>cd ph
C:\PH>dir
Volume in drive C has no label.
Volume Serial Number is 4C17-95D2

Directory of C:\PH

12/27/2006  12:29 PM    <DIR>          .
12/27/2006  12:29 PM    <DIR>          ..
11/19/2004  07:42 PM             230 BayesTheorem.html
11/23/2004  10:16 AM        423,428 Brochure.pdf
11/30/2006  10:23 AM         24,576 class2dat.exe
11/30/2006  10:23 AM        28,672 class2net.exe
11/30/2006  10:23 AM         49,152 csv2dat.exe
11/30/2006  10:23 AM        57,344 dat2hcs.exe
12/27/2006  12:25 PM    <DIR>          Data
11/30/2006  08:29 PM         6,299 Eval License.txt
11/30/2006  10:23 AM        94,208 feature.exe
11/30/2006  10:24 AM        49,152 fit2net.exe
07/10/2003  12:24 PM         3,310 hugin.ico
03/27/2006  07:33 PM        532,480 hugincpp2.dll
11/06/2004  09:56 AM       147,456 odbcf2file.exe
11/06/2004  09:56 AM       147,456 orcl2file.exe
11/30/2006  10:23 AM        143,360 ph.exe
11/27/2006  08:53 PM     1,367,040 phj.exe
05/17/2006  09:49 AM        25,214 phj.ico
12/27/2006  05:39 PM    <DIR>          PHJ_Example
11/26/2006  10:01 AM       194,127 phmanual.pdf
11/23/2004  10:49 PM       126,976 pull.exe
11/30/2006  08:46 PM       15,305 README.txt
11/30/2006  08:29 PM         5,686 Research License.txt
11/30/2006  10:23 AM        28,672 struct2dat.exe
11/30/2006  10:23 AM        24,576 ustruct2dat.exe
11/30/2006  10:23 AM        24,576 ustruct2hcs.exe
11/30/2006  10:23 AM        24,576 weather2dat.exe
          24 File(s)      3,543,871 bytes
          4 Dir(s)    23,577,952,256 bytes free

C:\PH>

```

Patterns and Predictions™ was initially developed as a command line intensive automation tool. Therefore, all functions that are exposed in the PHJ tool are present as command line operations. In fact, there are many more options included within the command line, as we would assume advanced usage on large systems.

## Features in Patterns and Predictions Professional™

The following features are included in the professional version of the software.

- Hierarchical (HNBM) Modeling – the discovery of latent ‘hidden’ variables.
- Accuracy function – Using the k-fold (e.g leave-one-out) approach you can validate the accuracy of your model.
- Unstructured data parsing – Utilizing our Patent Pending element counting techniques, you can convert your unstructured data elements to a structured data set and optimize the accuracy of your models.
- Other platforms – Patterns and Predictions™ Pro is available to users for Windows, Solaris, Linux, Mac, and 64 bit platforms/systems.
- Support – 1 year of full support (both email and phone) is included with the product. Extended support is available for 20% of the total purchase price.

For more information on our predictive analytic solutions in Healthcare, Finance, and everyday life:

- Healthcare such as Health risk predictions (see Poulin-Hugin's work with the CDC at <http://www.poulinhugin.com/casestudy/Influenza.htm> )
- Finance such as Commodity price predictions (see Poulin-Hugin's work with FCOJ at <http://www.poulinhugin.com/fcoj.pdf> )
- Everyday life such as the outcome of Sporting events (see Poulin-Hugin's upcoming work)

### Lead Developers

Chris Poulin : Anders Madsen  
([chris@poulinholdings.com](mailto:chris@poulinholdings.com)), ([anders.l.madsen@hugin.com](mailto:anders.l.madsen@hugin.com))

### Marketing

Anne-Mette Christensen

### Supporting Code

Frank Jensen

### UI Design

Jason Nichols