PattRun: QuickStart

Rev: 6 Dec 2006 – Michael Ochs, initial version
10 May 2007 – Michael Ochs, new screenshots and order new version

Downloading and Starting the Application

Download the appropriate version of PattRun from

http://www.cancerbiostats.onc.jhmi.edu/software.cfm

or from Fox Chase at the link provided on this page. To get the full version with BD, you must download from Fox Chase following completion of an MTA form. Unfortunately, this version is only available to academic users.

Unzip or untar the file, and move to the PattTools directory (or folder).

For Mac and Windows users, the PattRun.jar file can be double-clicked to start the GUI client and server on the host machine. Alternatively, remote nodes can be established with the command

java –cp PattRun.jar PattServ &

and the computer will now serve as remote node for calculations. For Linux and Solaris machines, please read the documentation on setting PATH variables. Then

java –cp PattRun.jar PattServ &

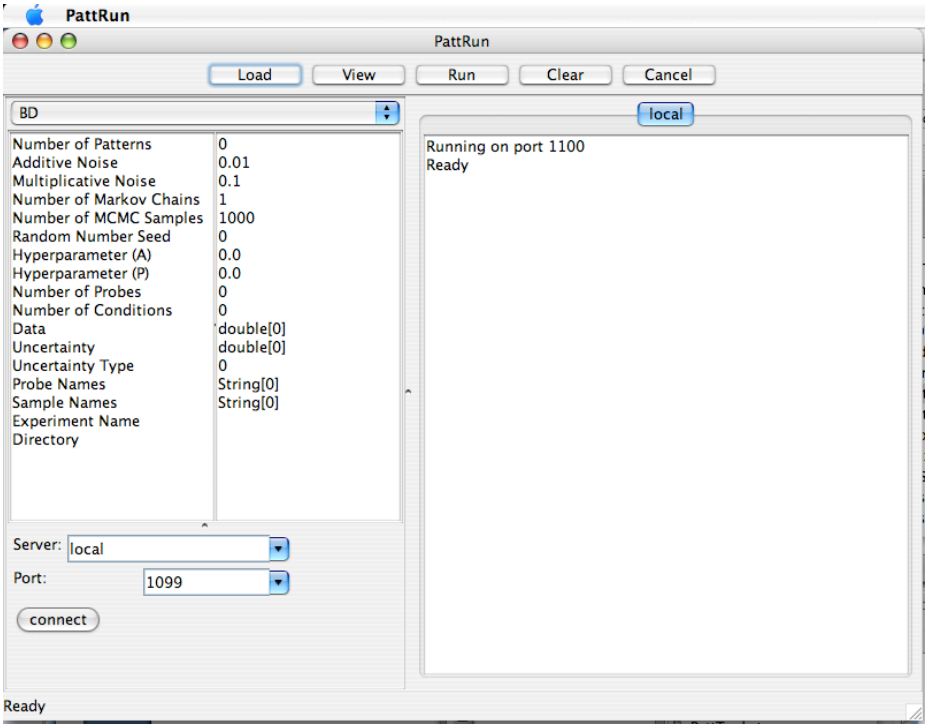
will start the server and

java –cp PattRun.jar PattRun &

will start the client and server on the local machine.

Running PattRun on a Single Computer

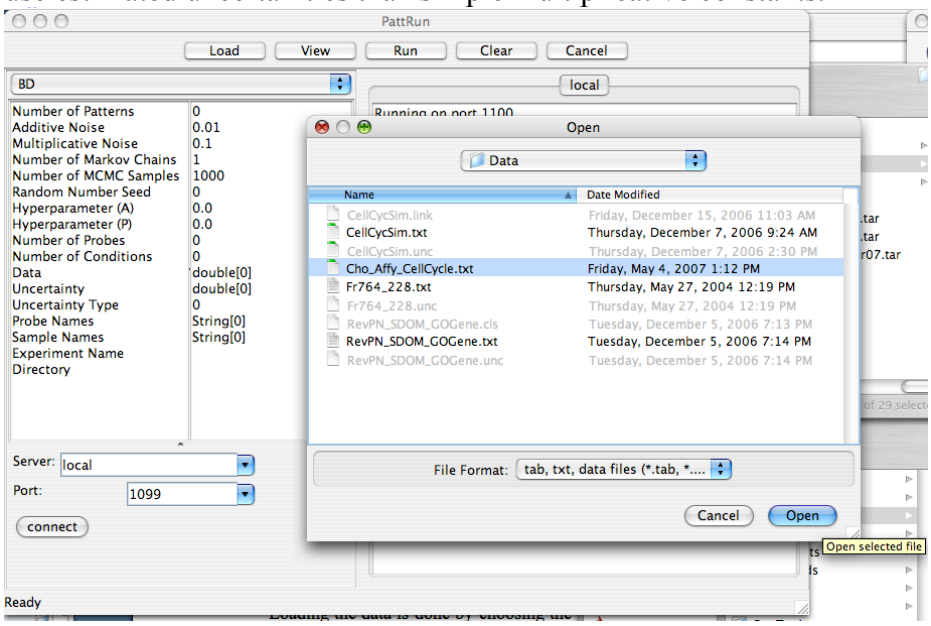
We will describe running Bayesian Decomposition (BD) and note changes for LS-NMF. To run the client and server locally, double-clicking on the PattRun.jar file will complete the set-up. The interface should appear as (although you will not have the same servers and may not have the same libraries available)



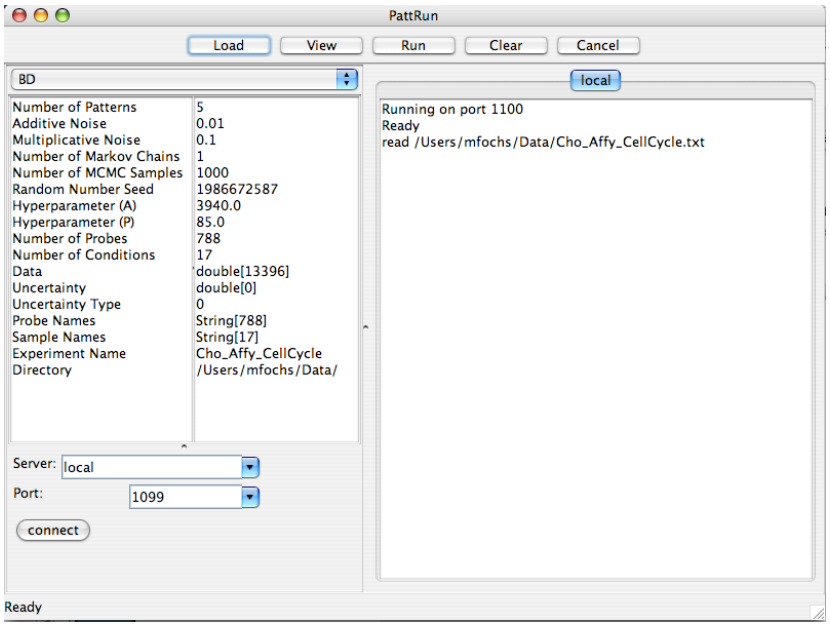
The information on the left contains a pop-up to choose different algorithms, parameters, a pop-up for the servers available based on the PattServers.list file and the Port in use by the client (contact is only made when one is picked, so servers that are not online will still be listed), and status of the local server. In the new version, a PattServers.list file is no longer required, as the name of the server can be typed directly into the pop-up. The parameters include

Number of Patterns	number of patterns to use
Additive Noise	factor for additive noise if no .unc file
Multiplicative Noise	factor for multiplicative noise if no .unc file
Number of Markov Chains	always 1 for 1 Markov chain
Number of MCMC Samples	number of iterations in equilibration and sampling
Random Number Seed	random number seed (automatically generated)
Hyperparameter (A)	hyperparameter on number of atoms in A domain
Hyperparameter (P)	hyperparameter on number of atoms in P domain
Number of Probes	number of probe sets or genes in data (rows of data)
Number of Conditions	number of conditions (columns of data)
Data	will show number of data points
Uncertainty	will show number of uncertainty estimates
Uncertainty Flag	=1 if .unc file is read and used
Probe Names	will show data structure for probe names
Sample Names	will show data structure for condition names
Experiment Name	a root name for output files
Directory	location for output files

Pressing on the Load button and moving to the data directory in the PattTools directory, you can load the Cho_Affy_CellCycle.txt data file (Cho et al, *Mol Cell*, **2**, 65, 1999). Since there are no uncertainty estimates on this data, PattRun will calculate uncertainties based on the additive and multiplicative terms appearing in the interface (default is 0.01 additive and 0.1, i.e. 10%, multiplicative). For new experiments with replication, it is better to use estimated uncertainties than simple multiplicative constants.

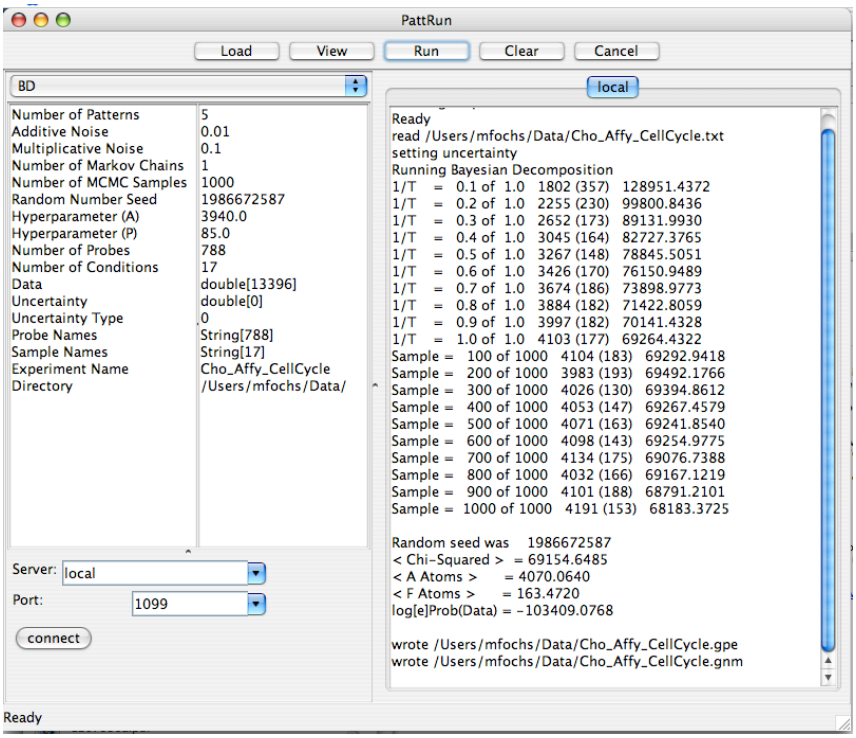


Loading the data is done by choosing the .txt file containing tab-delimited data, with the first column being gene names and the first row condition names. If .unc, .cls, and .link files of the same root name exist, these will also be loaded. The .unc contains uncertainty estimates and matches the format of the .txt file, while the .link file provides the ability to link genes according to annotation, such as by transcription factor binding sites. These files are discussed in more detail below.



Once the Cho data is loaded, the algorithm is ready to be run. After loading the Cho data, the interface indicates the parameters, including the number of conditions (columns in the data file, here 17 time points), the number of probes (here 788), and the total number of data points (13396). The Uncertainty is shown as having no size, indicating there was no .unc file, and the Uncertainty Type is set to 0, so that the uncertainties for each point will be calculated using additive and multiplicative terms. Following loading, the parameters on the left can be adjusted.

The algorithm is started by pressing the Run button. After completion (which may take a considerable time on older machines, but on a 2.16 GHz dual-core Intel processor takes only about 10 minutes), the display shows



The results show two phases: equilibration with simulated annealing and sampling. The equilibration shows each step in temperature as 1/T, the number of atoms in A(P) domains, and the chi-squared of the present sample. Sampling gives updates every 100 samples, including atom numbers and chi squared.

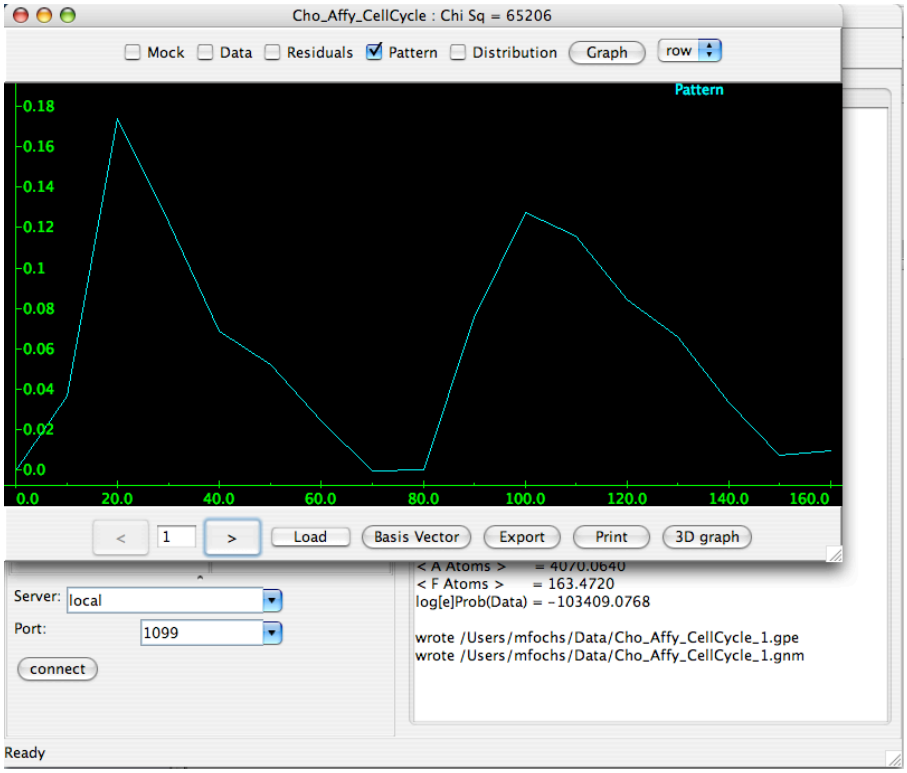
When using BD, it is important during sampling that the chi-squared value not be consistently decreasing. If it continues to improve, it shows inadequate equilibration and the algorithm should be rerun increasing the Number of MCMC Samples variable. The final lines note what output files have been written. The .gpe file is a serialized data object containing A and P matrices, uncertainty estimates on these matrices generated during MCMC sampling, the fitted and initial data matrices, and residuals. The .gnm file contains the probe names. These files are used by the viewer (see below) and by our annotation analysis tool, ClutrFree.

We have published details of the BD analysis of the Cho data (Moloshok et al, *Bioinformatics*, **18**, 566, 2002).

Visualizing the Results

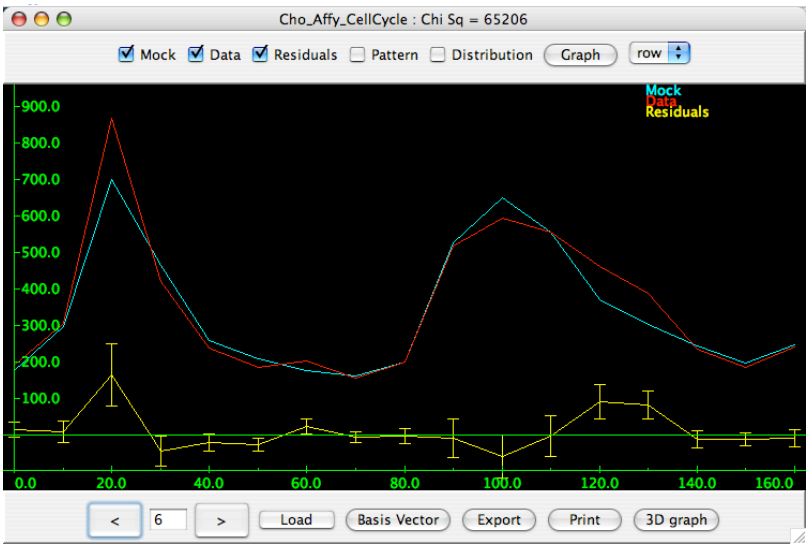
As noted above, the interface confirms the writing of the .gpe and .gnm files. Pressing the View button will now allow the user to view the results summarized in these files.

The first thing to note is the chi-squared value. It should be lower than the final chi squared based on the scalar average across samples, as it is the chi squared of the mean result. If the space is well sampled, then this should be better than the individual samples. Next, marking the Pattern box and clicking Graph will show the first pattern. The arrow keys on the bottom left allow you to step through the patterns.

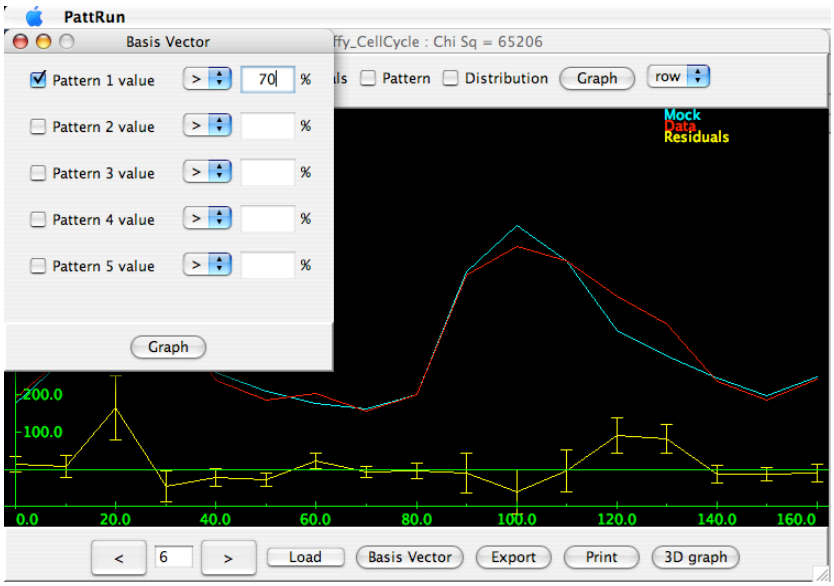


Technical Note: Ideally the chi-squared value should equal the number of degrees of freedom, which is approximated by the number of data points – the number of atoms in the model. However, this assumes very precisely measured uncertainty estimates as well as normally distributed variables. We have found Bayesian Decomposition to be robust to misestimation of noise, although gross overestimation will lead to loss of structure and gross underestimation will lead to an explosion in the number of atoms as the algorithm attempts to fit the model to the noise. For this data set, the uncertainty is underestimated (our paper used about double this estimate), however the patterns are as in our paper. However, if the chi squared is grossly off the predicted value, it is usually a sign of error in the input files or an attempt to fit the data with too few or too many patterns.

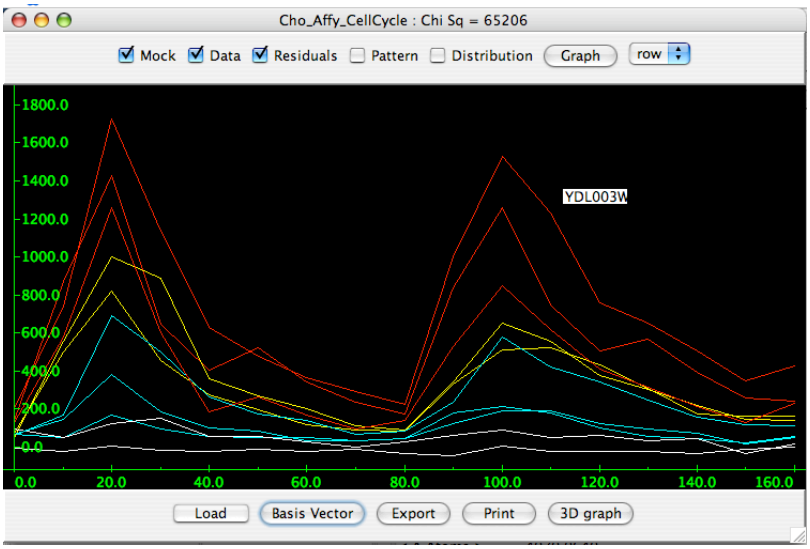
The viewer also allows users to explore the assignment of individual genes to patterns (Distribution) and the fit of the reconstructed data (Mock) and the data. Checking the Mock, Data, and Residuals boxes simultaneously and pressing the Graph button allows users to view how the reconstructed data performs in reproducing the data. It is important to remember that in a global fit of hundreds or thousands of genes, not every gene will be fit well, and outlying points are expected naturally even in a good data set. (Naturally, we show a nice fit however). The residuals appear together with estimated uncertainties from the .unc file or from the calculation based on additive and multiplicative noise terms. This approach can be used to identify genes that are not fit well by the model. For instance, a gene that not only has some data points outside the error bars, but also has a form that is not matched to any possible mixture of patterns.



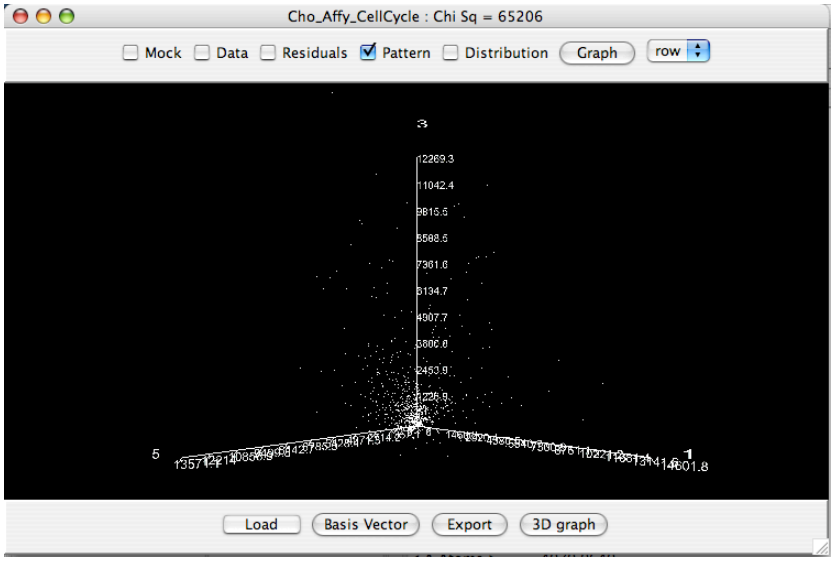
The Basis Vector button will allow exploration of genes associated strongly with a single pattern or combination of patterns. Clicking it brings up a dialog box allowing the user to specify the strength of association (as how much of a gene's behavior is explained by a pattern) and the patterns of interest.



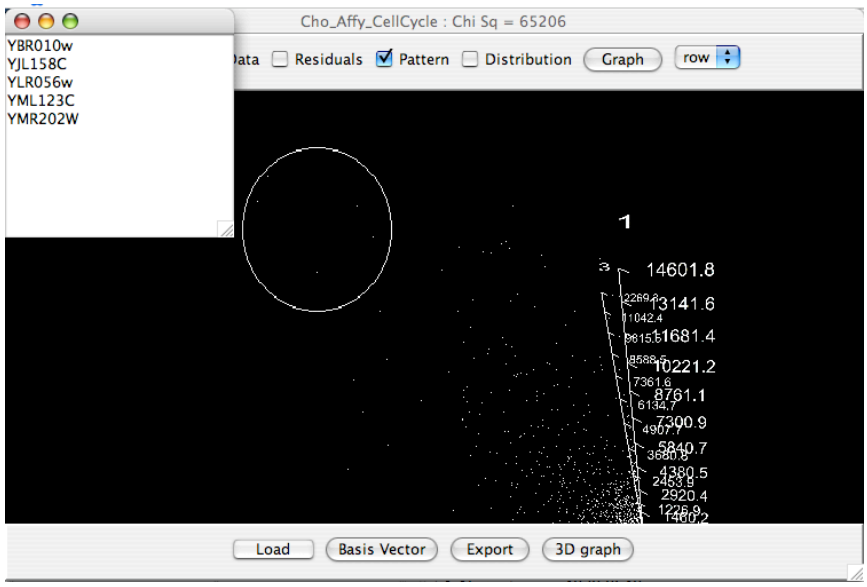
Checking a box for a pattern and then providing a percentage (here first pattern at 70%), allows viewing the genes linked to a pattern. For instance, for the G1 pattern in this data these are all genes linked at 70% or more of their behavior explained by this pattern. Mousing over a line pops up the gene name.



The viewer also allows three dimensional views of the distribution of genes in patterns. Pressing the 3D graph button will provide a dialog to choose three of the patterns. All genes in the data set will then be plotted based on their strength in each of these patterns.

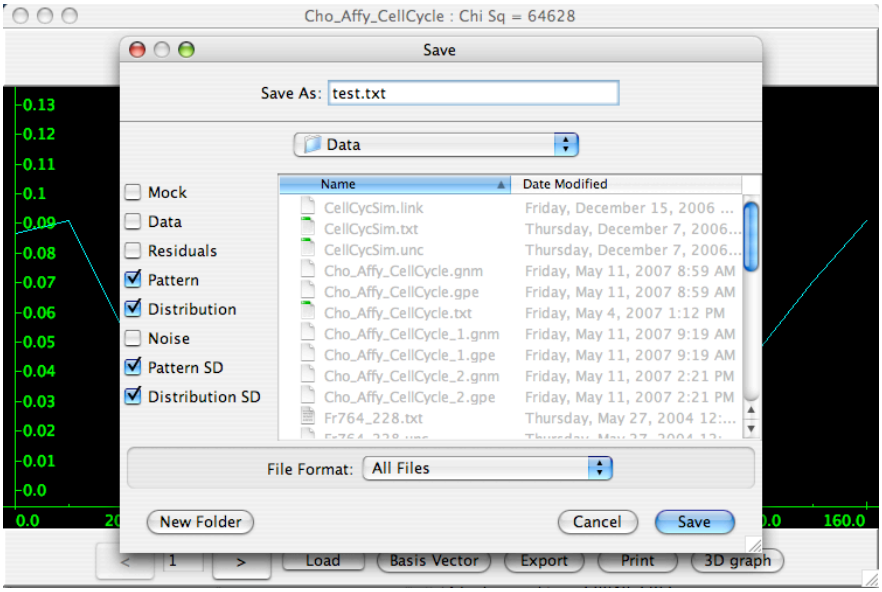


The arrow keys allow you to rotate the image around all axes (you may have to click in the draw area and the dismiss a dialog box first). Once you have isolated genes of interest, you can draw an oval around them and a dialog box with the genes contained will appear.



Exporting Data

PattView allows data to be exported into a tab-delimited file. Pressing the Export button brings up a dialog box where you can choose items to export. In this example, we are exporting the A matrix (Distribution) and P matrix (Pattern) together with the associated standard deviations measured at each point.



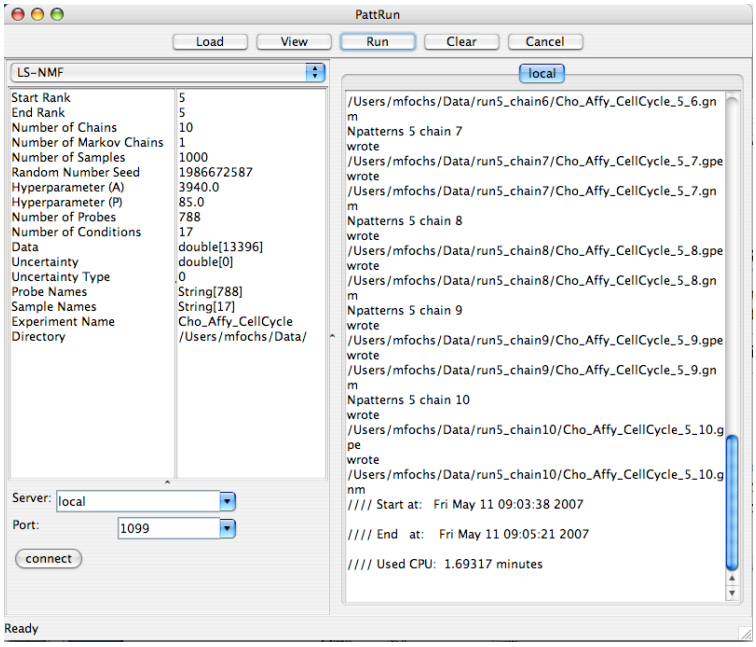
The export options include

Mock	The reconstucted data matrix (=**AP**)
Data	The input data matrix
Residuals	The misfit between the data and the mock data
Pattern	The **P** matrix
Distribution	The **A** matrix
Noise	The input uncertainty estimates on the data
Pattern SD	The standard deviation of the **P** matrix
Distribution SD	The standard deviation of the **A** matrix

If you are displaying a set of genes linked to a Basis Vector when you hit the Export button, an additional choice to export this list of genes and data point values is included.

LS-NMF Differences

Unlike BD, LS-NMF cannot sample the probability distribution, as it is designed, as other NMF algorithms, to reach a local maximum. Instead, the algorithm runs multiple passes and places the results in a subdirectory. If there is no uncertainty file, LS-NMF reduces to standard NMF, and no uncertainties are calculated. Choosing LS-NMF from the drop down menu, changing Start Rank and End Rank both to 5, and pressing the Run button will run NMF on the Cho data. LS-NMF will automatically run with different random seeds a number of times equal to the Number of Chains variable. If Start Rank and End Rank are different, it will run this number of simulations for each possible number of patterns between Start Rank and End Rank.



NMF will produce multiple .gpe and .gnm output files. They are written into a directory structure suitable for direct loading into ClutrFree, however they can be loaded individually into the viewer program as well. To look at the individual files, you use the Load button in PattView to choose each .gpe file in turn. Generally, NMF requires many attempts and then a choice by the user of the result that provides the best fit (i.e., lowest chi squared). Alternatively, use of ClutrFree allows users to focus on patterns where genes are consistently assigned. Unfortunately, NMF processes in the complex space of most microarray experiments tend to lead to highly different sets of patterns, so such comparisons may not always be of use.

We have published details on the analysis of microarray data using LS-NMF (Wang et al, *BMC Bioinformatics*, **7**, 175, 2006).

Advanced Options for BD

Supervised Learning

For data that includes knowledge of classes, the .cls file is used, and it is a tab-delimited file containing a single line.

Number of Classes	3	Elements per class	6	6	4

This permits using the Supervised BD option, which enforces class information in the first *N* patterns, where *N* is specified here as the Number of Classes. The conditions are divided between classes in order as specified by Elements per class. For this example, there are 3 classes comprising the first 6, second 6, and third 4 (= 16) conditions. Additional conditions of unknown class can be included after these elements.

The RevPN_SDOM_GOGene.txt file contains a subset of the Project Normal data (Pritchard et al, *Proc Natl Acad Sci U S A*, **98**, 13266, 2001). The .unc file includes uncertainties estimated from the four replicates in each condition, and the .cls file (shown above) links the kidney, liver, and testis samples respectively for the different mice.

When this data is run, the analysis will enforce step functions for the first three patterns, matching their class. In PattView, these will appear as step functions linking the appropriate conditions. For a discussion of BD analysis of this data, reprints can be requested from our web site (http://www.cancerbiostats.onc.jhmi.edu/) for our publications (Moloshok et al, in Johnson and Lin (eds), *Methods of Microarray Data Analysis III*, 2003; Ochs et al, *Annals of the New York Academy of Sciences*, **1022**, 212, 2004).

Linking Genes through Annotations

A third version of BD, called BD-TF, allows linking of genes through annotations, such as genes that would be coregulated by a transcription factor or complex. The algorithm relies on a file listing linked genes in an ASCII file. The file begins with three header lines, which provide respectively a label after a # character, a fraction for the splitting of the atomic domain between coordinated atoms (those that use the annotation information in the .link file) and uncoordinated atoms. The third line indicates the type of preprocessing to run, and using BD is standard (3). The # character serves to mark end of reads so comments can be added.

The fourth row contains the key information for the linking of genes. The first number given the number of *a priori* sets of coregulated genes. In the CellCycSim.link file (the start of which appears below), there are 5 coregulated sets of 126 genes, 122 genes, 116 genes, 115 genes, and 64 genes. Beginning with row 5, there are the indices of the genes in the data matrix in C style (i.e., first element is element 0).

```
# Regulons
0.5    # influence: ratio of regulon/point mapping
3 # BD2
5 126 122 116 115 64
0
1
2
...
```

Genes will generally appear in more than one coregulation set due to multiple regulation. The algorihtm retains the ability to ignore coregulation information, as is essential given that genes may serve unknown cellular functions.

Analysis proceeds as with BD, only because the analysis must first run BD a number of times to normalize the spreading of atoms among coregulated genes, it takes considerably longer than the standard BD. The data included in CellCycSim.txt is a low-noise simulation of the cell cycle, and it can be used to explore the function of BD-TF. A good measure of how the prior knowledge is being used is to compare the number of atoms needed to model the A matrix with coregulation information and without If coregulation information is being applied successfully, there should be on average fewer atoms when BD-TF is used than when BD is used.