

## Classification Using Binned NMR Spectral Data

By Jianguo Xia ([jianguox@ualberta.ca](mailto:jianguox@ualberta.ca))

Last update: 4/15/2009

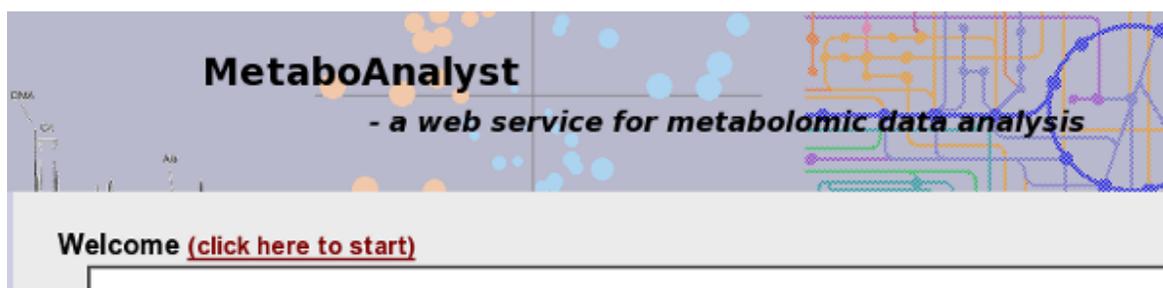
This tutorial shows how to perform classification using methods provided in MetaboAnalyst. The example used is from NMR spectral binning data published by [Psihogios NG, et al. \(PMID: 17705523\)](#). Proton-NMR spectra were collected from human urine samples of two groups - 25 healthy control and 25 patients with severe renal damage (tubulointerstitial lesions). The diagnoses were based on histopathological evaluation of renal biopsy. After removal of water regions and drug peaks, these spectra were binned into ~200 bins using a 0.04 ppm bin width. The purpose is to investigate whether we can discriminate healthy control from renal patients based only on the urine spectral binning data.

**Step 1.** Go to the “**Data Formats**” page, click the [download](#) link after the “Binned NMR/MS spectra data” option. Unzip the downloaded file and save it as “nmr\_bin.csv”.

**Comma Separated Values (.csv) format ([show details](#)), including :**

- Compound concentration data ([download](#))
- Binned NMR/MS spectra data ([download](#))
- Paired time-series concentration data ([download](#))
- Processed peak intensity table ([download](#))

**Step 2.** Go the MetaboAnalyst **Home** page and click “click here to start” to enter the data upload page.



**Step 3.** In the **Upload** page, go to the “Upload your data” panel, and select the options as indicated below, then click “Submit”

1) Upload your data : [data format](#)

**Comma Separated Values (.csv) :**

Data type :  Concentrations  Spectral bins  Peak intensity table

Format:

Data file :

**Note:** alternatively, you can directly select the second option in the “Try our test data” without downloading the example.

2) Try our test data : ( You can download these data [here](#) )

**Data Type**

Concentrations

[Tutorial](#) | [Report](#)

NMR spectral bins

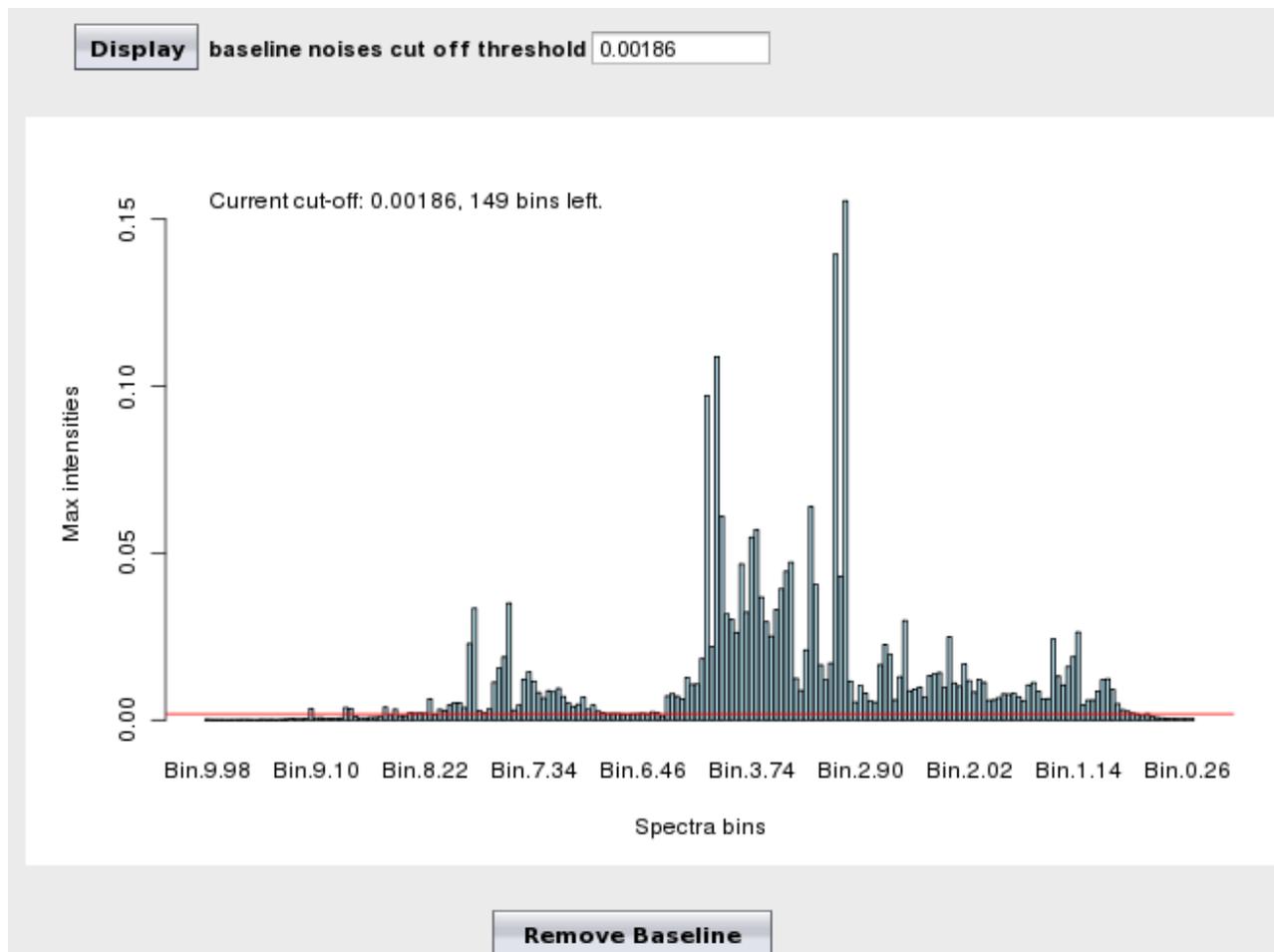
[Tutorial](#) | [Report](#)

**Description**

Urinary compound concentrations from 57 cancer patients measured by 1H NMR (unpublished data). Group 1- no weight loss; group 2 - severe weight loss

Binned 1H NMR spectra of 50 urine samples using 0.04 ppm constant width ([Psihogios NG, et al.](#)) Group 1- control; group 2 - severe kidney disease.

**Step 4.** This step tries to remove the baseline noises by applying a linear filter. Users can select various cut-off thresholds based on a visual evaluation of the graph of the binned data and the number of remaining bins. The default value will remove 25% of the lowest bins. Accept the default and click “Remove Baseline”.



**Step 5.** The data integrity check will run automatically and the result is shown below. After filtering the baseline noises from the last step, all the remaining values are positive. In addition, no missing values were detected. If missing values had been detected, then the most appropriate from a variety of methods provided by MetaboAnalyst could have been used to deal with this issue (for such an example, see MetaboAnalyst Tutorial 4). Click “Skip” to go to Normalization step.

**Note:** missing values are represented as NA (no quotes) or empty values.

**Data processing information**

Checking data content ...passed

Two groups were detected based on the sample labels.

Samples are not paired.

All data values are numeric.

All data values are non-negative.

A total of 0 , ( 0 %) zero values were detected

A total of 0 , ( 0 %) missing values were detected

By default, these values will be replaced by a small value

Click **Skip** button if you accept the default practice

Or click **Missing value imputation** to use other methods

**Step 6.** Now we arrive at the data normalization step. The internal data structure is transformed now to a table with each row representing a urine sample (from a patient) and each column representing a feature (a spectral bin). With the data structured in this format, two types of data normalization protocols - row-wise normalization and column-wise normalization -- may be used. These are often applied sequentially to reduce systematic variance and to improve the performance for downstream statistical analysis. Row-wise normalization aims to normalize each sample (row) so that they are comparable to each other. For row-wise normalization MetaboAnalyst supports normalization to a constant sum, normalization to a reference sample (probabilistic quotient normalization), normalization to a reference feature (creatinine or an internal standard) and sample-specific normalization (dry weight or tissue volume). In contrast to row-wise normalization, column-wise normalization aims to make each feature (column) more comparable in magnitude to each other. Four widely-used methods are offered in MetaboAnalyst - log transformation, auto-scaling, Pareto scaling, and range scaling. The binned urine spectra data are usually normalized by a constant sum. In this case, we choose “normalization by constant sum” for row-wise normalization and “Log normalization” for column-wise normalization.

### Row-wise normalization

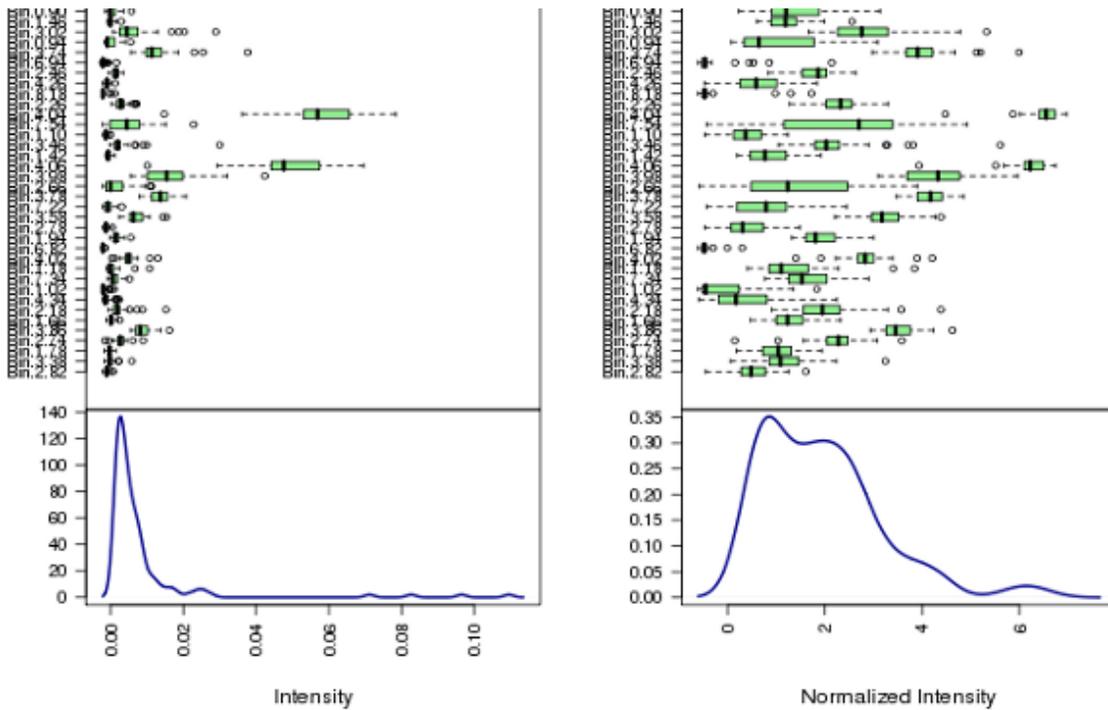
- None
- Normalization by sum
- Normalization by a reference sample <Not set> ▾
- Normalization by a reference feature <Not set> ▾
- Sample specific normalization (i.e. dry weight, volume) [Click here to specify](#)

### Column-wise normalization

- None
- Log (log<sub>2</sub> transformation)
- Autoscaling (mean-centered and divided by the standard deviation of each variable)
- Pareto Scaling (mean-centered and divided by the square root of standard deviation of each variable)
- Range Scaling (mean-centered and divided by the range of each variable)

Process

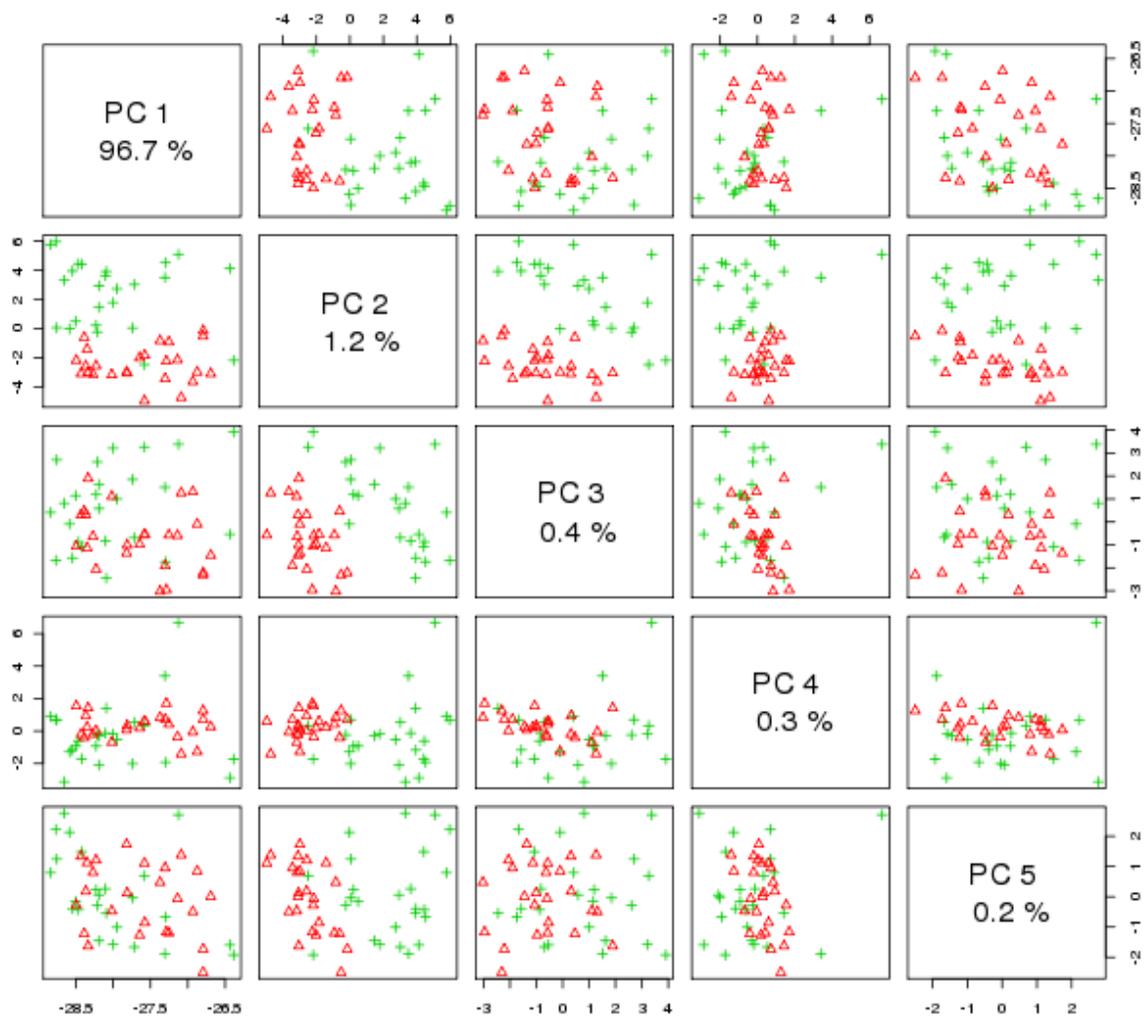
The result of normalization is shown below. On the left is a plot (box-whisker plot on top, linear distribution plot on the bottom) of the data prior to normalization. On the right is a plot (box-whisker plot on top, linear distribution plot on the bottom) of the data after normalization. As can be seen by comparing the linear concentration curve on the left (which has an exponential decay character) with the normalized curve on the right, the variables are now more comparable to each other. Note the peak on the left side of the normalized curve is caused by many close-to-zero values typical in binned spectra data.



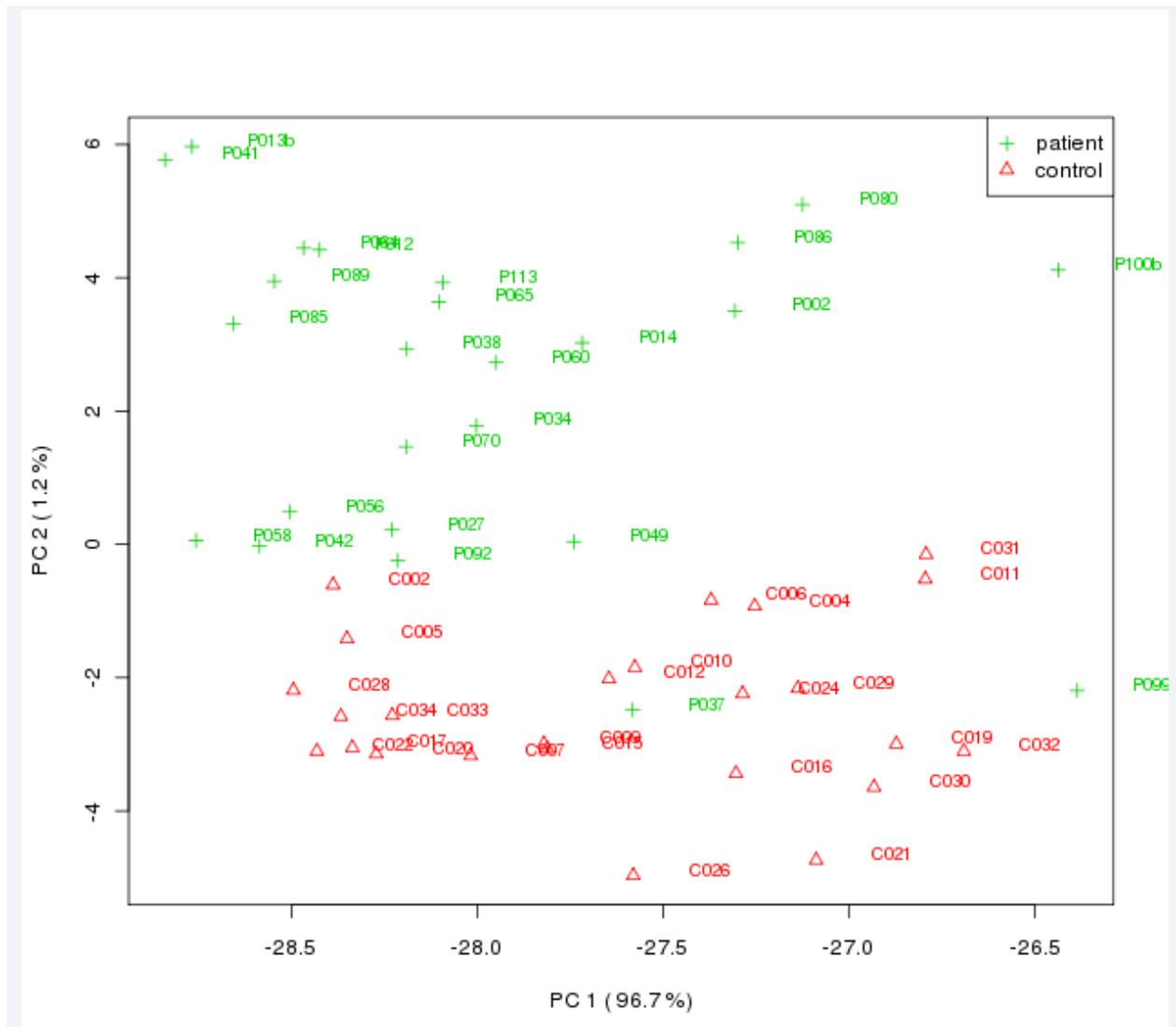
**Step 7.** We finished data processing and normalization and now the data is suitable for different statistical analysis. There are many methods available in MetaboAnalyst for classification (both supervised and unsupervised). Here we will only show results from two unsupervised (clustering) methods - PCA and heatmap, and two supervised methods - PLS-DA and random forest. The screen shot below shows the Analysis view. Please note the navigation panel on the left. A color change indicates the corresponding step has been successfully performed. All the data analysis methods can be directly accessed by clicking the corresponding link.

The screenshot displays the MetaboAnalyst web interface. On the left is a vertical 'Steps' navigation panel with a blue background. The steps are: 1. Upload, 2. Process, 3. Normalize, 4. Analyze (highlighted in a darker blue), 5. Peak Search, 6. Pathway Mapping, 7. Download, and Log Out. Under '4. Analyze', several analysis methods are listed: Univariate, PCA, PLSDA, SAM, EBAM, Tree & heatmap, Kmean & SOM, RandomForest, and R-SVM. The main content area on the right has a light gray background and is titled 'Select an analysis path to explore :'. It contains several categories of analysis methods, each with a bold heading and a list of links: 'Univariate Analysis' (Fold Change, t-Tests, and Volcano plot), 'Chemometrics' (Principal Component Analysis (PCA), Partial-Least Square - Discriminant Analysis (PLS-DA)), 'Significant Feature Identification' (Significance Analysis of Microarray (SAM), Empirical Bayesian Analysis of Microarray (EBAM)), 'Cluster Analysis' (Hierarchical Clustering - Dendrogram and Heatmap, Partitional Clustering - K-Means and Self Organizing Map (SOM)), and 'Classification & Feature Selection' (Random Forest, Support Vector Machine (SVM)).

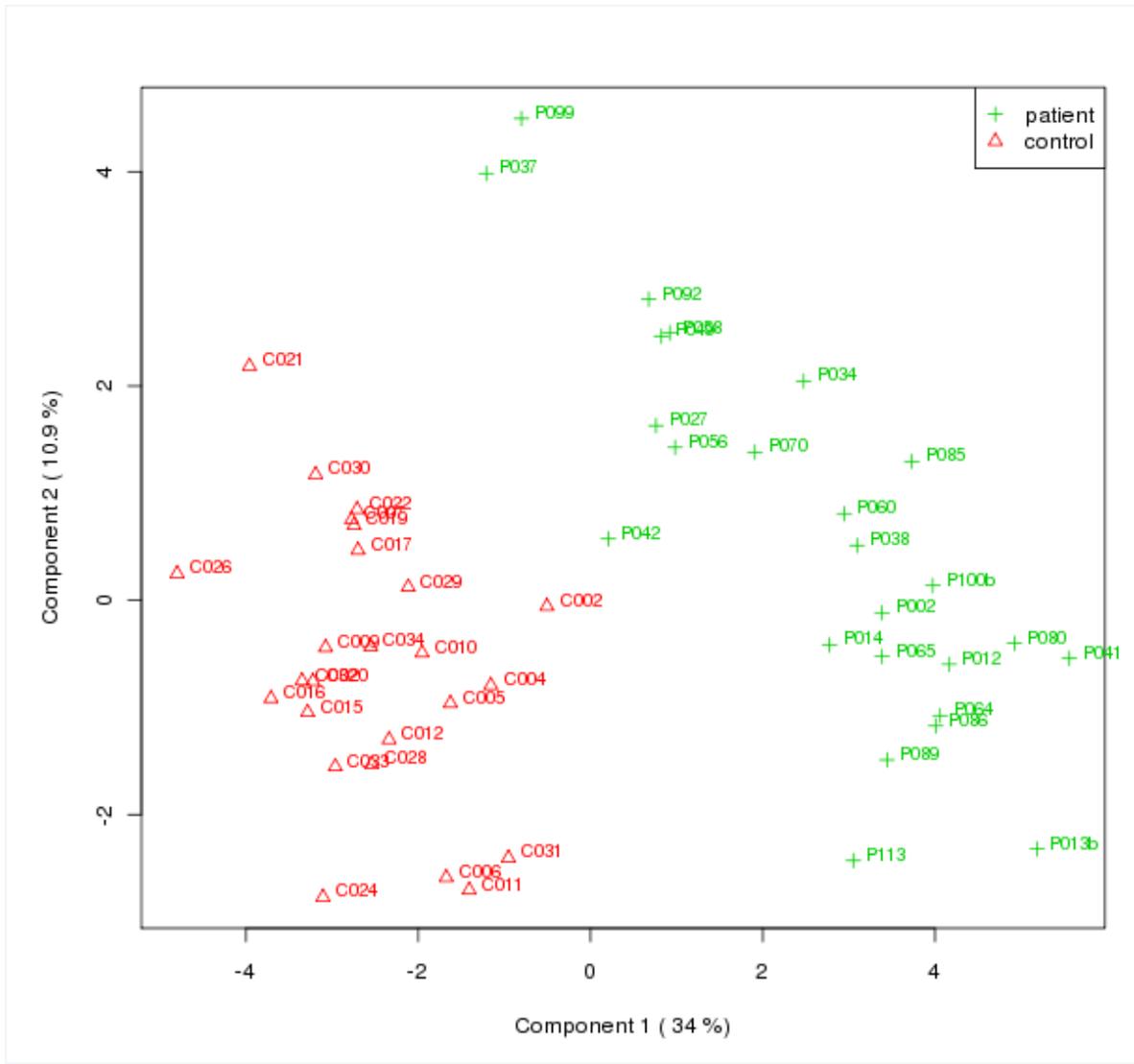
**Step 8.** We first want to see if there are inherent group patterns with the data structure without using the class labels (unsupervised clustering). Principal Component Analysis (PCA) provides an excellent visualization tool of high-dimensional data by projecting the data into low-dimensional space (usually 2D or 3D). Click the “PCA link” on the navigation panel and you will see the following overview of pairwise score plots from the top five PCs:



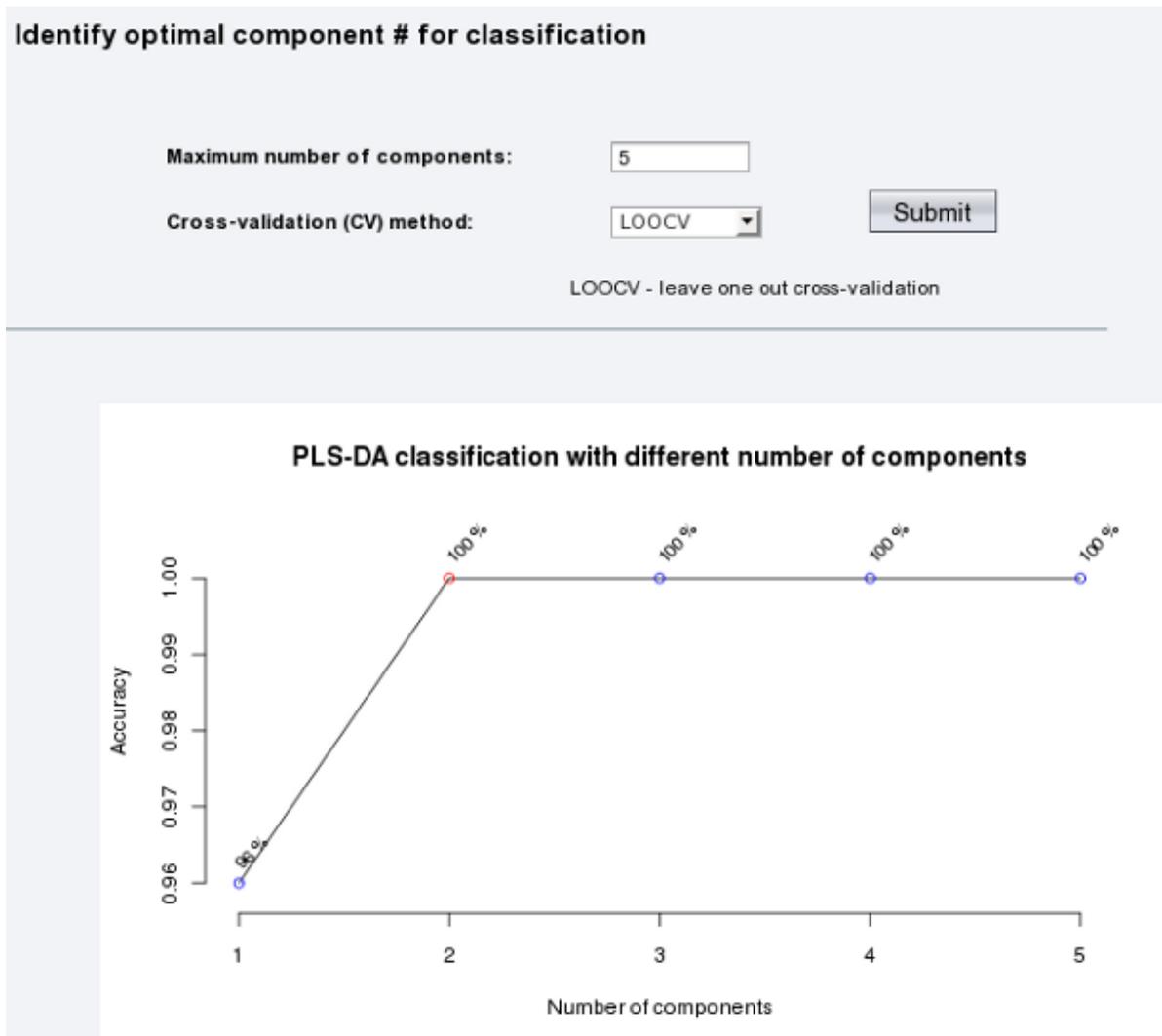
Click the “2D score plot” tab, where you can see a detailed score plots between the control and renal patients using PC1 and PC2. A good group pattern was detected, although there are several samples C002, P037 and P099 that could not be separated by using the first two components. Users can view the score plot between other PCs by entering a different PC index.



**Step 9.** Click the “PLSDA” link on the navigation panel; the default is an overview of score plots using the top 5 components. Click the “2D score plot”. The following view is shown. As you will notice, a complete separation was achieved using first two PLS components.

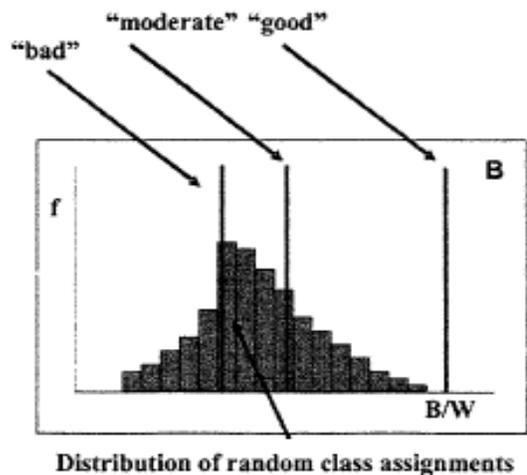


The PLS-DA classification performance can be seen by clicking the “Classification” tab. The performance using the top 5 components (latent variables) is plotted as shown below. As you can see, using the top 2 latent variables, 100% classification accuracy can be achieved. The default evaluation scheme is based on leave-one out cross validation (LOOCV).

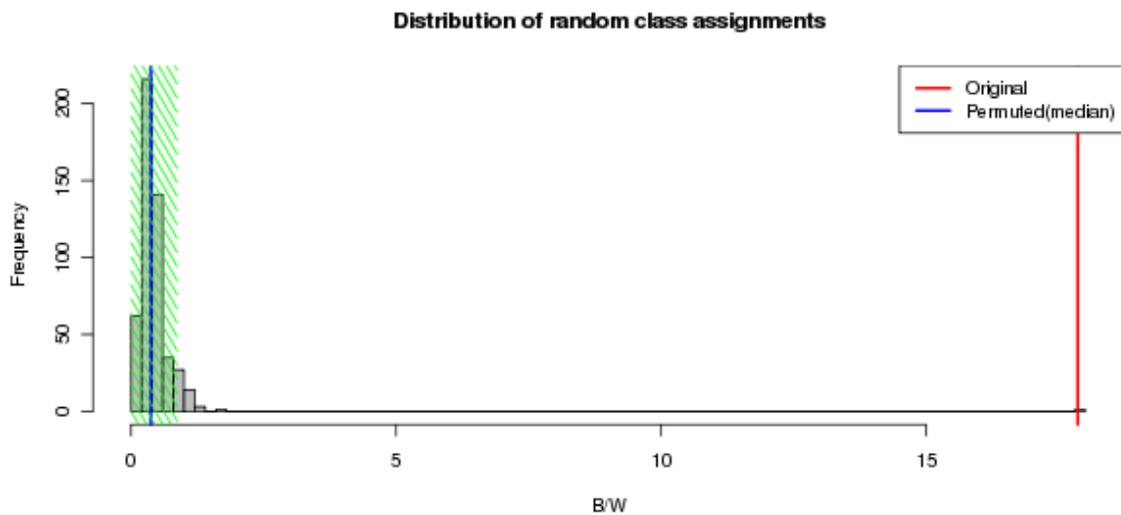


PLS-DA tends to overfit the data and therefore the model needs to be validated to see whether the separation is statistically significant or is due to random noise. This is done using permutation tests. In each permutation, a PLS-DA model is built between the data (X) and the permuted class labels (Y) using the optimal number of components determined by cross validation for the model based on the original class assignment. The ratio of the between sum of the squares and the within sum of squares (B/W-ratio) for the class assignment prediction of each model is calculated. If the B/W ratio of the original class assignment is a part of the distribution based on the permuted class assignments, the contrast between the two class assignments cannot be considered significant from a statistical point of view.

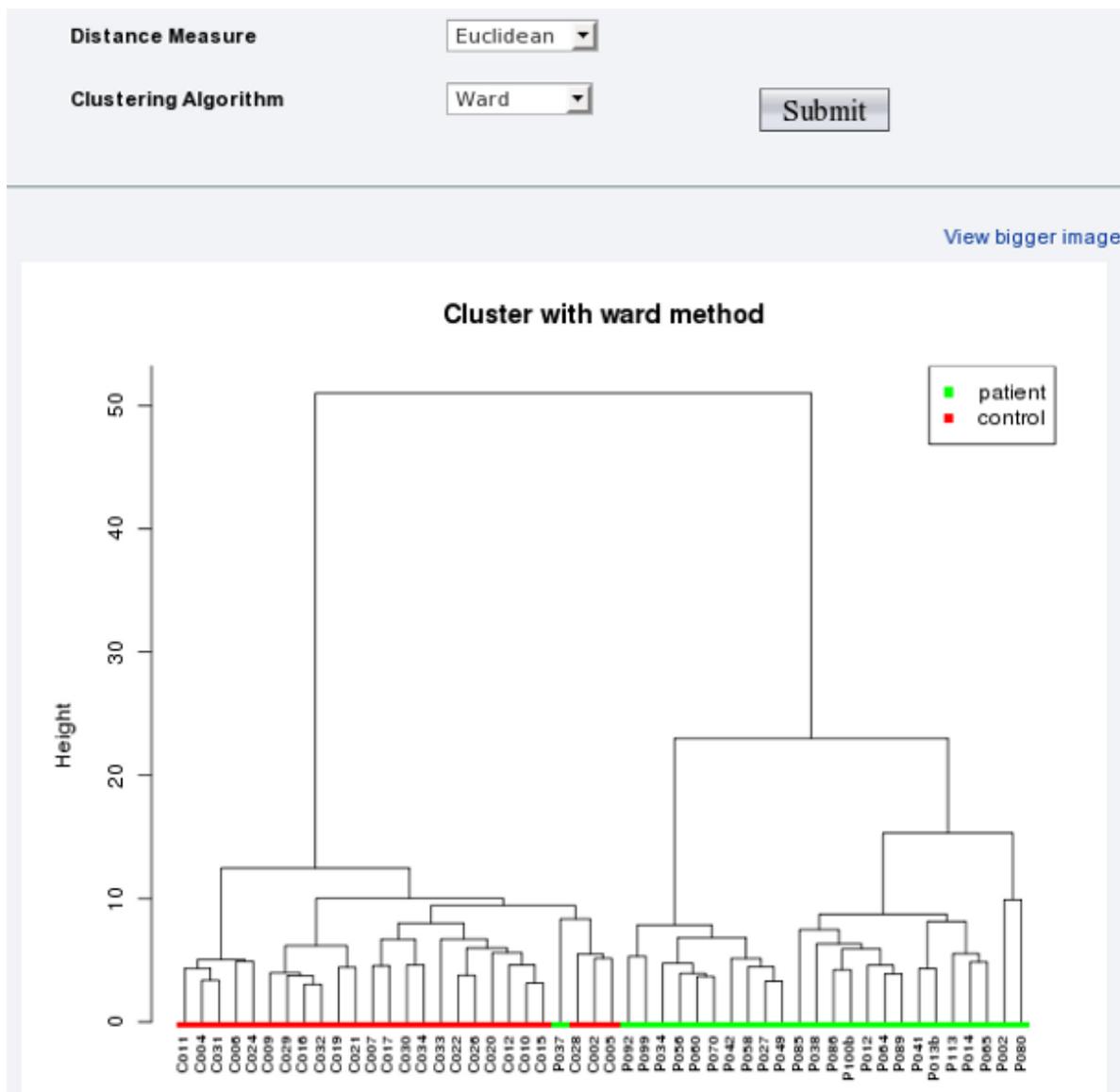
This following graph is the suggested by Bijlsma *et al.* (PMID: 16408941) on how to evaluate whether a class assignment is good or bad. The histogram shows the distribution formed by the permuted samples. The bar represents the original sample. The further away to the right of the distribution, the more significant the separation between the two groups is.



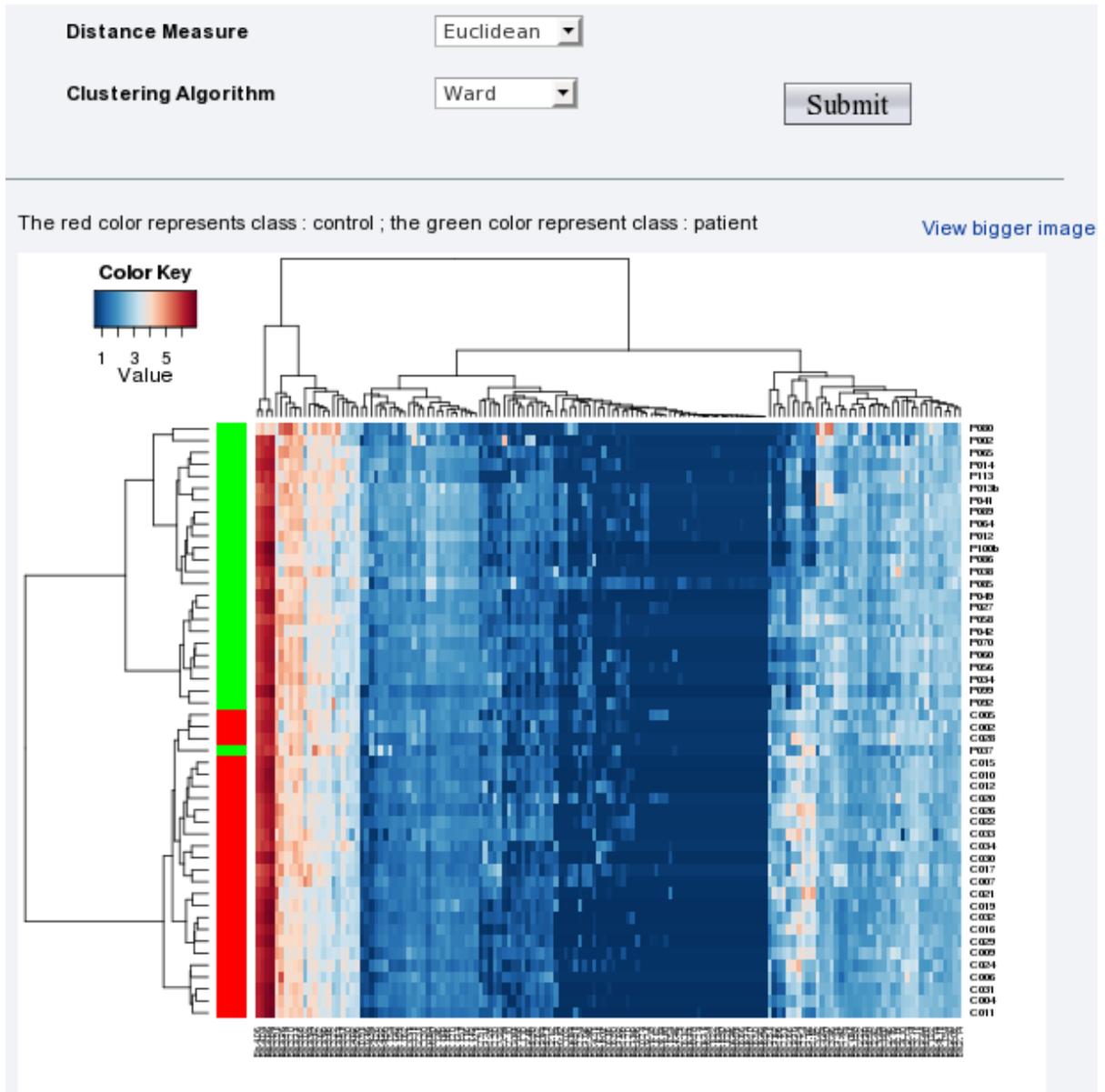
Click the “Permutation” button to view the permuted B/W vs the original value. The following graph shows a graph after 500 permutations. The top graph compares the original B/W values to that of the permuted ones. The bottom graph shows the relative location of the original B/W on the distribution of the permuted B/W values. The green line (top) and green area (bottom) mark the 95% confidence region of the B/W for the permuted data. As you can see, the original class assignment is very significant and not part of the distribution we obtained using the permuted data.



**Step 10.** Hierarchical clustering is commonly used for unsupervised clustering. Agglomerative hierarchical clustering begins with each sample as separate cluster and then proceeds to combine them until all samples belong to one cluster. Users need to specify a dissimilarity measure (Euclidean distance, Pearson's correlation, or Spearman's rank correlation) and a clustering method (average linkage, complete linkage, single linkage, or Ward's linkage). The result is usually presented as a dendrogram or heatmap; both have been implemented in MetaboAnalyst. Click the “Tree & heatmap” link on the navigation panel, select “Euclidean” in the “Distance Measure” and click “Submit”. The image below shows the resulting dendrogram.

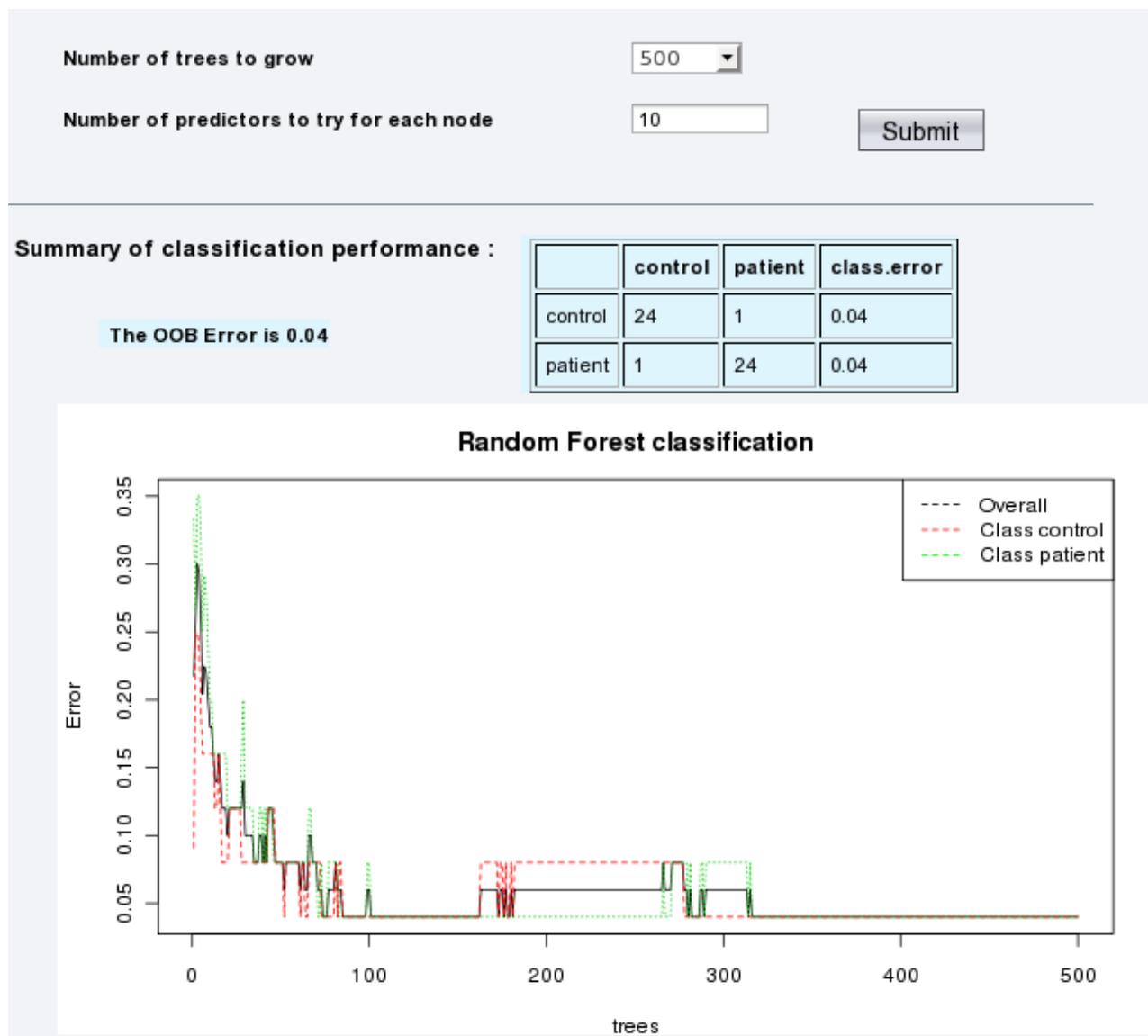


Click the “Heatmap” tab to see a default heatmap view. Select “Euclidean” in the “Distance Measure” and click “Submit” to generate the best separation, as shown in the image below. Users can choose different distance measures or clustering algorithms to visually explore the results.



**Step 11.** Random Forests is a supervised classification algorithm well-suited for high dimensional data analysis. It uses an ensemble of classification trees, each of which is grown by random feature selection from a bootstrap sample at each branch. Class prediction is based on the majority vote of the ensemble. Click the “RandomForest” link; the classification result is shown below. The default parameters achieve 0.04 classification error and 0.04 out-of-bag (OOB) error. The graph below shows the cumulative error rates for the prediction of each class as well as the overall prediction error rate.

**Note:** the error rates converges to 0.04 after trees grow over 320 (You may get different results since these trees are generated by random feature selections from bootstrap samples)



For information about random forest and OOB error, you can place your mouse over the “About Random Forest” link and a help text will pop up. More information about Random Forest can be obtained in MetaboAnalyst's FAQ page.

The screenshot shows the MetaboAnalyst web interface. At the top, there are two tabs: "Classification" and "Var. Importance". A link labeled "About Random Forest" with a question mark icon is visible. A help popup window is open, displaying the following text:

Random Forests is a supervised learning algorithm well-suited for high dimensional data analysis. It uses an ensemble of classification trees, each of which is grown by random feature selection from a bootstrap sample at each branch. Class prediction is based on the majority vote of the ensemble. RF also provides other useful information such as OOB (out-of-bag) error and variable importance measure. During tree construction, about one-third of the instances are left out of the bootstrap sample. This out-of-bag (OOB) data is then used as test sample to obtain an unbiased estimate of the classification error (OOB error). Variable importance is evaluated by measuring the increase of the OOB error when it is permuted.

In the background, there is a "Submit" button and a table showing "class.error" with two rows, both containing the value "0.04". Below the popup, a legend indicates three data series: "Overall" (dashed black line), "Class control" (dashed red line), and "Class patient" (dashed green line). A partial bar chart is visible at the bottom left, with a y-axis ranging from 0.20 to 0.30.

**Step 12.** Now, assume we have finished the analysis. Click the “Download” link on the left panel. A detailed analysis report will be generated (MetaboAnalystReport.pdf) containing introductions and results for every step we have performed. Now, you can directly click and download the “Download.zip” file which includes all the processed data, images, and the PDF report. Alternatively, you can ask MetaboAnalyst to send you the result via email by entering your email address. The data will remain on the server for 72 hours before being automatically deleted.

Email address :

Files in your home directory	
<a href="#">Download.zip</a>	<a href="#">MetaboAnalystReport.pdf</a>
<a href="#">big-hm-clust.png</a>	<a href="#">big-tree-clust.png</a>
<a href="#">conc-norm.png</a>	<a href="#">data_normalized.csv</a>
<a href="#">data_original.csv</a>	<a href="#">data_processed.csv</a>
<a href="#">hm-clust.png</a>	<a href="#">nmr-bin.png</a>
<a href="#">nmr_bins.csv</a>	<a href="#">pca_3d.png</a>
<a href="#">pca_biplot.png</a>	<a href="#">pca_loading.png</a>
<a href="#">pca_pair.png</a>	<a href="#">pca_score.png</a>
<a href="#">pca_scree.png</a>	<a href="#">pls_class.png</a>
<a href="#">pls_loading.png</a>	<a href="#">pls_pair.png</a>
<a href="#">pls-permut.png</a>	<a href="#">pls_score2d.png</a>
<a href="#">pls_score3d.png</a>	<a href="#">pls_VIP.png</a>
<a href="#">Rhistory.R</a>	<a href="#">tree-clust.png</a>

-----End of tutorial-----