

## Quick exploRase tutorial

This guide will quickly introduce the features of exploRase through an example analysis session. It is assumed that the user has already installed exploRase and that the user is generally familiar with microarray data analysis. Please follow the numbered steps below in order. The physical action required for each step is italicized and is followed by further explanatory details. Please feel free to ask questions.

1. *Start R.* ExploRase is written in R, so that it may benefit from the various data analysis packages written in R for Bioinformatics, in particular those from the Bioconductor project. This means that R must be started before launching exploRase. To make this easier, Windows users can create a desktop shortcut as documented on the exploRase web page.
2. *Enter the following into the R console:*

```
> library(explorase)
> explorase()
```

The first command loads the *explorase* package into R. The second command displays the exploRase GUI, which should now be displayed on the screen. This GUI is written entirely from within R, using the RGtk2 package.

One will notice that the newly initialized GUI is empty. ExploRase requires some experimental data, as well as several types of metadata. The metadata includes the experimental design matrix, a matrix of annotations for the entities (e.g. genes) in the experiment and one or more lists of “interesting” entities. Each type of information is stored in a separate file. The most convenient way to organize an analysis project is to place all of the files in the same directory in the file system. This is called a “project” in exploRase.

3. *Download and unzip the sample project from <http://www.metnetdb.org/exploRase/files/mittler.zip>.* The archive contains a project directory called “mittler” that holds the experimental data as well as metadata of the types mentioned above. The data was provided by Ron Mittler at the University of Nevada - Reno.
4. *Load the sample project by clicking on the Open button in the toolbar and choosing the mittler directory in the File Open dialog.* ExploRase will automatically load all of the files in the directory. Note that this requires the file extensions to be named according to a convention specified in the exploRase manual, so that exploRase knows the type of information in each file.
5. *Take a moment to become familiar with the exploRase GUI.* The large table holds the annotations for every gene in the experiment. In this sample project, the information was derived from MetNetDB. To the left of the main table are two panels, the bottom one holds entity (gene) lists (more on this later) and the top one lists the chips from the experimental design matrix.
6. *Click on the “Details” button under the list of chips.* The “Details” button displays the experimental design matrix in a table. In this experiment, wildtype (control) and apx1 (ascorbate peroxidase 1, involved in response to oxidative stress) deficient Arabidopsis plants were exposed to moderate light stress for 24 hours, and Affymetrix microarrays were performed for 7 time points, 0m, 15m, 30m, 90m, 3h, 6h, and 24h, with two replicates for each time point, for a total of 28 chips.
7. *Find the GGobi control panel and scatterplot that appeared when the data was loaded.* ExploRase leverages GGobi for visualizing the experimental data and analysis results using interactive graphics. GGobi is a general tool for multivariate interactive graphics in support of exploratory data analysis. ExploRase always opens a GGobi scatterplot for the first two variables (in this case, the two replicates for wildtype at 0 minutes). In order to maximize the performance of GGobi (especially on Windows) it is best to work on a subset of the data.

8. *Go back to the *exploRase* window and from the *Tools* menu select *Subset*.* This launches the subset dialog. There are currently three methods for subsetting: by minimum value, minimum fold change, and maximum variance between replicates. Clicking on the “Show Slider” button displays a slider that allows one to adjust the cutoff values based on the percentage of the data that is retained by the filter.
9. *Enter 2 for the ‘at least one fold change should be greater than’ item and press the *Apply* button.* This will hide those genes that never change more than two-fold across all of the chips. The GGobi scatterplot will now look very different, as the number of visible genes has been reduced from ~22000 to ~1200. This helps focus the analysis on the most interesting subset of the data and also has the technical benefit of accelerating the drawing in GGobi.
10. *Return to the *Tools* menu and choose *Average replicates*.* Another common data preparation step is to average over the replicates. This helps the analyst concentrate on differences between genotype and/or treatment rather than between replicates.
11. *In the main *GGobi* window, select the *24 hour means (with one genotype as X and the other as Y)*.* The plotted variables in a GGobi display can be changed by clicking on the appropriate button next to the variable name. The GGobi scatterplot now compares the genotypes at the last time point of the experiment. This would not have been possible before collapsing the replicate pairs into single variables, as done by the averaging tool in *explorase*. It is important to visually explore the data using GGobi plots before delving too deeply into the analysis.
12. *Select the same two variables (the means at 24 hours) in the *exploRase* chip list.* Before any of the *exploRase* analysis methods are executed, the variables of interest must be selected in the *exploRase* list of conditions (chips). The Analysis menu holds the available analysis methods. They are categorized by purpose. Methods in “Find interesting entities” help identify those entities (genes) that change between two selected conditions. “Find similar entities” measures the correlation between a selected entity (in the annotation table) and the others along the selected conditions. Other methods include (hierarchical) clustering and “pattern finding” that classifies transitions between conditions (time points) as up, down, or same depending on a quantile test.
13. *Open the “Find interesting entities” submenu and choose *Difference*.* The sample dataset has been log transformed, so the simple difference (subtraction) is roughly equivalent to fold change. The result of the calculations was added to both the *exploRase* annotation table and the GGobi dataset. While looking at the raw numbers in the *exploRase* table is rarely useful, the table may be sorted according to the calculated values by clicking on the corresponding column header. This makes it easy to see the ranking of genes by a given statistic. In this case, the top-ranked genes are those with the most difference between genotypes after 24 hours of light stress.
14. *Select the top 10 or so genes in the annotation table and press the *Brush* button in the toolbar.* The Brush tool is the primary link between *exploRase* and GGobi. The *exploRase* brush tool changes the color of the selected entities in the annotation table, as well as in the GGobi plots. If one looks at the scatterplot of the means at 24 hours, the outliers (the most different between genotypes) are now colored using the current *exploRase* brush color. The differentially expressed genes have now been identified for a single time point, but it is not feasible to use the difference calculation to detect significant patterns of differential expression across multiple time points.
15. *Press the *Clear* button in the toolbar, which resets the colors in GGobi to their default.* It is now time to try a different approach.
16. *From the *Modeling* menu, select *Limma*.* Fitting a linear model to each gene is an efficient way to evaluate the significance of the effect of each factor (genotype and time) on each gene. This is the approach taken by the Limma package from Bioconductor. Limma goes a step further

and adjusts the p-values to account for the possibility that an effect will be found significant by random chance alone, due to the large number of genes being tested. The limma dialog allows the user to specify the factors to consider as well as the results that will be added to *exploRase* and *GGobi*.

17. *Select the genotype factor in the Limma dialog and click Apply.* In order to determine the genotype-dependent genes across all time points, it is necessary only to select the genotype factor. If one was also interested in time dependence, one could select time or the interaction between time and genotype. *exploRase* also offers a polynomial time model under the *Models* menu, but the current question does not require that. Just as in the difference, the results (p-value and F statistic) have been added to *exploRase* and *GGobi*. It would be very inconvenient to try to interpret the results across every time point using scatterplots. Thus, a different type of plot is necessary: the parallel coordinates or “profile” plot.
18. *Select all of the means in the sample (chip) list and then choose Parallel Coordinates Plot from the Graphics menu.* A *GGobi* parallel coordinates plot for every mean should now be displayed. Brushing a gene in *exploRase* or another plot will show its profile across time and genotype.
19. *Bring up the original GGobi scatterplot and change the axes to show p.genotype as X and F.genotype as Y.* The scatterplot of the p-value vs. the F statistic for genotype makes it easy to pick out those genes that are highly significant for genotype, relative to the others.
20. *Switch the scatterplot to brush mode by choosing Brush from the Interaction menu in the GGobi window. With the profile plot visible, brush the outliers in the top-left of the scatterplot.* As the brush moves over the outliers in the scatterplot, the significant profiles are highlighted in the profile plot. One should observe that these profiles show a major difference between genotypes.
21. *Switch the brush to “persistent” mode by checking the Persistent box in the main GGobi window and brush just a few of the most obvious outliers.* Making the brush persistent is an easy way to mark interesting genes in *GGobi*.
22. *Click the Sync Colors button in the exploRase toolbar.* This transfers the brushed colors from *GGobi* to the *exploRase* annotation table and completes the loop that integrates *GGobi* and the *exploRase* GUI. It is not likely, however, that the colored rows will be visible in the annotation table. It is possible to sort the table by color, but it would be even better if the uninteresting rows were filtered out.
23. *Click on the Filter button that is just above the annotation table, select the Yellow color from the drop down box on the right, and press Apply.* The annotation table in *exploRase* may be filtered in many different ways. In this case a filter rule has been created that only passes the rows that have the yellow color. It is possible to filter by any column in the annotation table, including analysis results, as well by presence in an entity list and other criteria. Multiple filter rules may be applied simultaneously. The visible list now consists solely of genes that are highly dependent on genotype across the entire time course. It has taken a good number of steps to reach this point, so it would be a good idea to save this list.
24. *Select all of the entities in the (filtered) table and press the Create List button in the toolbar. A new row appears in the list panel and requires a name to be entered.* The list of significant genes has now been saved (in memory) as a list. If this were a real project, it would be beneficial to save the list to disk via the *Save Project* or *Export List* options in the *File* menu. For the rest of this session, clicking on the list item will automatically select the listed entities in the annotation table. It is also possible to filter the table by the list.
25. *With the significant genes selected, click on the AtGeneSearch button in the toolbar.* This will launch a web browser to *AtGeneSearch*, the web interface to the *MetNetDB*. This will give further

details on the selected genes, which might help a biologist form hypotheses to be tested in future experiments.

For the sake of brevity and simplicity, this tutorial has only scratched the surface of the potential of *exploRase*. Feel free to explore this sample dataset further or get started with your own data. Below are some suggested tasks to motivate further exploration.

1. Find some genes that are significantly dependent on time. The *Temporal Modeling* tool in the *Modeling* menu would likely be helpful for this.
2. Choose one of the time-dependent genes with an interesting pattern. Try to find other genes with the same pattern, perhaps using the *Find Similar Entities* and/or *Pattern Finder* tools. One might view the expression profiles to graphically evaluate the similarity of the patterns.
3. Filter the annotation table so that only the genes with the pattern of interest are shown.