1 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)
   library(CLL)
   data(sCLLex)
   sCLLex$SampleID <- Biobase::sampleNames(sCLLex)
   explorase(sCLLex)
   ```

   The first command loads the `explorase` package into R. The second command loads the `CLL` data package, and the third the *ExpressionSet* named `sCLLex`. We then tweak the object to help exploRase better guess the structure of the data. The final command displays the exploRase GUI, which should now be displayed on the screen, and loads the `sCLLex` dataset. 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, besides as an *ExpressionSet*, is to place all of the files in the same directory in the file system. This is called a "project" in exploRase.

3. *Take a moment to become familiar with the exploRase GUI.* The large table holds the annotations for every gene in the experiment. 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.

4. *Click on the `Details` button under the list of chips.* The `Details` button displays the experimental design matrix in a table. In this experiment, there are 24 samples, with two types of disease progression: stable and progressive. The replicate column is derived automatically, and, in this case, indicates biological replicates.

5. *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, CLL11 and CLL12). In order to maximize the performance of GGobi (especially on Windows) it is best to work on a subset of the data.

6. *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.

7. *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 12000 to 2500. 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.

8. *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.

9. *In the main GGobi window, select the means (with one disease type 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 disease types. 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.

10. *Select the same two variables (the means) 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 Difference` help identify those entities (genes) that change between two selected conditions. `Find Similar` 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, which classifies transitions

between conditions (time points) as up, down, or same depending on a quantile test.

11. *Open the `Find Difference` submenu and choose `Subtract`.* The sample dataset has been log transformed, so the simple difference (subtract ion) 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 the disease types.

12. *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, the outliers (the most different between the disease types) are now colored using the current exploRase brush color. The differentially expressed genes have now been identified according to the means, but we are not taking into account the variation within the disease types.

13. *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.

14. *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 disease type on each gene. This is the approach taken by the `Limma` package from Bioconductor. Optionally, exploRase goes a step further and adjusts the p-values returned from `Limma` 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.

15. *Select the genotype factor in the Limma dialog and click Apply.* In order to determine the disease-type-dependent genes across all time points, it is necessary only to select the `Disease` factor. If this was a time-series dataset, one could select time or the interaction between time and genotype. exploRase also offers a polynomial time model under the `Models` menu. Just as in the difference, the results (p-value and coefficient) 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 plot.

16. *Select every sample in the sample (chip) list and then choose Parallel Coordinates Plot from the Graphics menu.* A GGobi parallel coordinates plot

with every sample should now be displayed. Brusing a gene in exploRase or another plot will show its profile changing across disease type.

17. *Bring up the original GGobi scatterplot and change the axes to show coeff.Disease as X and p.Disease as Y.* The scatterplot of the p-value vs. the coefficient for disease type makes it easy to pick out those genes that are highly dependent on the type of disease, relative to the others.

18. *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 disease types.

19. *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.

20. *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.

21. *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 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 disease type. It has taken a good number of steps to reach this point, so it would be a good idea to save this list.

22. *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. 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.