Research Project:
"Bioconductor as a tool for gene expression analysis"

Presented by:
Luisa Mercado Mendoza

MATH 3210 Data Mining Foundations

December 12, 2016

Bioconductor as a tool for gene expression analysis

Executive Summary

The goal of this research project was to demonstrate how Bioconductor, which is a software project, can be used to facilitate the analysis of gene expression data. Gene expression analysis deals with the expression of thousands of genes, therefore, the analysis of this huge amount of data becomes extremely difficult without the help of appropriate computational tools. Bioconductor offers extensive and flexible tools to perform a variety of tasks in gene expression analysis. Three tasks were addressed using some of the tools provided by Bioconductor to demonstrate the capabilities of this software to facilitate gene expression analysis. The tasks that were addressed aimed to select from a dataset containing 12,625 genes, the ones that were differentially expressed between two samples. The results were satisfactory, the tools successfully addressed each of the tasks. Identifying 201 genes that were differentially expressed and leaving behind the genes that were not informative.

Problem Description

Gene expression analysis deals with datasets that contain information about the expression levels of thousands of genes. Therefore, the methods that are used to extract valuable information become nearly impossible to perform on such big amounts of data. This creates the need to develop appropriate computational tools to facilitate this labor.

Analysis Technique

Gene expression analysis consists of monitoring the expression levels of multiple genes simultaneously under a condition of interest. An example of this can be measuring the expression of a gene in a cancerous cell or in a cell under a particular medical treatment. These analyses are widely used in clinical medicine because the comparison of the level of expression of the genes could be used to identify diagnostic therapy or prognostic biomarkers, classify diseases, monitor the response to therapy, or understand the mechanisms involved in diseases (Tarca, Romero, & Draghici, 2006).

In recent years, comprehensive high-throughput methods for molecular biology experimentation have had a huge development. An example of this is DNA microarray technologies that allow measuring tens of thousands of genes simultaneously. Because of this technology, gene expression analyses can be performed in much more genes simultaneously than ever before. The way that this technology measures gene expression is by means of fluorescence intensity, in which the genes are labeled with a fluorescent protein and they are scanned by a machine that measures the intensity of the fluorescence light and captures it into images. The images are transformed into values to represent the level of expression of the genes. The data that is obtained can be viewed as a matrix of expression levels that display genes versus tissue samples. When the tissue sample corresponds to a single microarray experiments, the output from M experiments can be represented as N x M array (matrix). Each column of the matrix (the expression signature vector) contains the expression levels on the N genes monitored in the microarrays, while each row (expression profile) contains the expression levels of a gene as it varies over the M tissue sample (McLachlan, Do, & Ambroise, 2004).

The overall structure of gene expression analysis can be summarized into four stages. The first stage known as Data processing and Quality control is performed to generate consistent expression values based on the scanned microarrays. The second stage known as Differential Expression is performed to identify the genes that are differentially expressed with respect to the sample characteristic of interest. Both the third stage known as unsupervised clustering and Data visualization, and the fourth stage known as supervised classification and prediction are perform

to identify large-scale patterns of gene expression and utilize them to predict biological patterns (Hofmann, 2006). Each of these stages involves specific tasks to accomplish the objectives described above.

The big challenge in performing these analyses is the amount of data that should be analyzed. As mention above microarray experiments allow to obtain the expression levels of thousands of genes simultaneously. This has created the necessity of developing computing environments such as Bioconductor to facilitate these analyses.

Bioconductor is an open source and open development software project for the analysis of genomic data. It provides an extensive and flexible set of tools to address specific tasks and answer specific questions (Heydebrek, Wolfgang , & Gentleman, 2004). It has a wide collection of statistical tools to perform analytic calculations; additionally, it contains a collection of annotated data and experimental data.

Bioconductor uses the R programming language to design and distribute integrated and interoperable software modules, called packages to provide comprehensive software solutions to relevant problems (www.Bioconductor.org, 2016).There are currently around 2,000 packages in Bioconductor that can tackle a variety of tasks required in many biological related analyses such as RNA-seq analysis as well as the gene expression analysis. To perform gene expression analyses, Bioconductor provide packages to perform tasks that are required in all the stages of the analysis described above. For, the pre-processing stage of the data it provides packages such as the affy package and the vsn package that facilitate tasks such as creating diagnostic plots, performing background correction, and probe-level normalization. In the gene identification stage of the analysis, it provides packages such as genefilter, multttes, limma and ROC to perform tasks such as non-specific filtering, gene selection, and multiple testing correction procedures. The stages of the analysis that deal with clustering and prediction are normally performed using packages from CRAN, which is a software repository in R. However, Bioconductor provides uniform calling sequences and return values for all machine learning algorithms available in CRAN that are used for clustering and prediction purposes. Additionally, of the software packages, Bioconductor also offers utility packages such as annotation packages, experiment data packages and packages that allow to handle graphs and networks. The package annotate, ALL, and geneplotter are examples of these utility packages respectively.

This project will work on addressing three tasks that are performed during differential gene expression using the tools provided by Bioconductor in order to demonstrate how these tools can facilitate the analysis of gene expression. The first task that is going to be addressed is non-specific filtering, which consists of removing genes that seem to be never expressed under any of the conditions of interest. This is an important task because these genes will not contribute in any form to these types of analysis due to the lack of expression. Therefore, by removing these genes in advance, further analysis that are going to be performed on the data will speed up, since there will be less genes to be considered. This task will be addressed using the genefilter package. This package provides a variety of filtering mechanisms to filter genes from gene expression datasets. This package can be used to perform non-specific filtering by calculating the overall variability across arrays of each probe set regardless of which type of sample they belong to. Then the probes that display very low variability are removed, assuming that differential expression between samples are reflected as high variability. There are several functions in this package to perform this task.

The second task that is going to be addressed is gene selection. The objective of this task is to identify those genes that are differentially expressed between two samples. This task can also be addressed using the genefilter package which also provides a variety of statistical tests. This task can be addressed by using a statistical test in which each gene is treated as an independent experiment to look at the ability of the gene to be used as a marker for one of the groups that is being compared. This means that, in that group, a high or low expression of the gene is mainly seen (Heydebrek, Wolfgang , & Gentleman, 2004).

The third task that is going to be addressed is multiple testing procedures. The objective of this task is to correct the type I errors that are generated during the gene selection step. These type I errors are produced when thousands of null hypotheses are tested simultaneously. One of the approaches that had been proposed to solve this problem and appears to be appropriate in many microarray-related context is the false discovery rate (FDR), which is the expected proportion of false positives among the genes that are called differentially expressed. This algorithm makes the p-values shorter and sequentially rejects the hypotheses starting from the smaller p-value. The idea is to achieve the smallest possible fraction of false signals among all those that appear to be true. The package that is going to be used to address this task is the multtest package, which provides multiple resampling-based single step and stepwise multiple testing procedures (MPT) for controlling a variety of classes of type I error rates (Hahne, Wolfgang, Gentleman, & Falcon , 2008).

The dataset that is going to be used to address the tasks mentioned above is the ALL dataset. This dataset come from a study of acute lymphoblastic leukemia (ALL). It consists of microarrays from 128 different individuals with this type of disease. There are 95 samples with B-cell ALL and 33 with T-cell ALL, which refers to two different types of tumors among these samples. This project will use the B-cell ALL sample, which contains individuals carrying the BCR/ABL mutation and individuals that do not display a cytogenetic abnormality. This allow us to perform differential expression analysis between samples having the BCR/ABL mutation labeled as BCR/ABL, and the ones that do not labeled as NEG. The total number of genes found in the B-cell ALL sample is 12,625.

For the first task, which is non-specific filtering, the total number of genes presented in the B-cell sample will be reduced by removing those that are not expressed or have little variability, which cannot be used to discriminate between the BCR/ABL and the NEG samples. The function rowSds from the genefilter package is going to be used to calculate the overall variability across arrays of each probe set, independent to the sample they belong to. This function specifically calculates the standard deviation for each row. After the standard deviation is calculated, the shorth function from the same package can be used to calculate the midpoint of the shortest interval containing half of the data (shorth), which in many cases can be used to estimate the “peak” of a distribution. The genes that display lower variability than the shorth are filtered out, leaving only the genes that have a higher variability.

The resulting set of genes from this filtering procedure are going to be used to perform the second task, which is gene selection. This task also can be performed using the genefilter package because it is also a filtering task. In this case, the function rowttests is going to be used to perform probe-by-probe tests for differential expression. It performs a t-test row by row, to detect significant differences in the location distribution of the expression level of the two samples. The null hypothesis is that the genes are not differentially expressed and the alternative hypothesis is that they are expressed differently between samples. After this procedure is performed, the genes whose are differentially expressed between the samples (p-values < 0.05) are going to be identified.

In the third task, which is multiple testing correction, the raw p-values obtained from the t-test are going to be adjusted since there were performed thousands of t-tests, causing a high rate of false-positives (type I) errors. The function mt.rawp2adjp from the multtest package is going to be used to reduce the rate of type I errors through the false discovery rate (FDR) implementing the Benjamini and Hochberg procedure (BH). This function adjusts the raw p-values and returns the adjusted ones. After this procedure the genes that are differentially expressed are identified using the adjusted p-values.

Assumptions

It was assumed that the dataset that was used was appropriately pre-processed.

Results

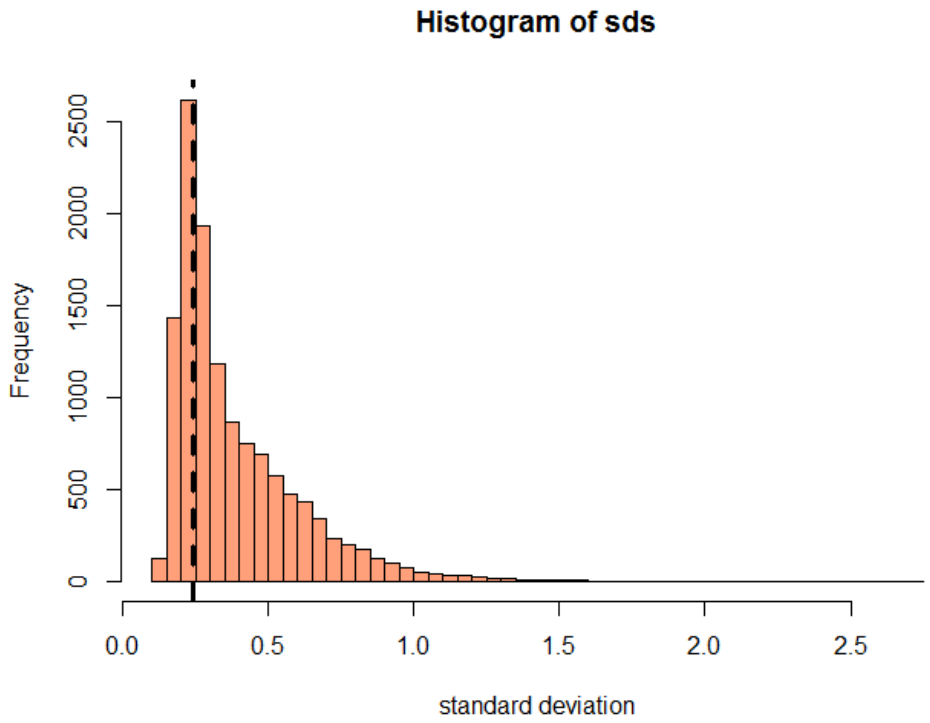

Figure 1: Histogram of standard deviations.

Figure 1 indicates the histogram obtained from the non-specific filtering task. The value of the shorth is represented by the vertical dashed line at 0.242. The genes that are below this point are not considered to have high variability; therefore, they are not considered expressed. These genes were removed from the analysis. The rest of the genes that are above this point, were selected to perform the gene selection task. The number total genes before performing this task was 12,625 and after performing this task was 8,812. This means that using this tools we could remove 3,813 uninformative genes.

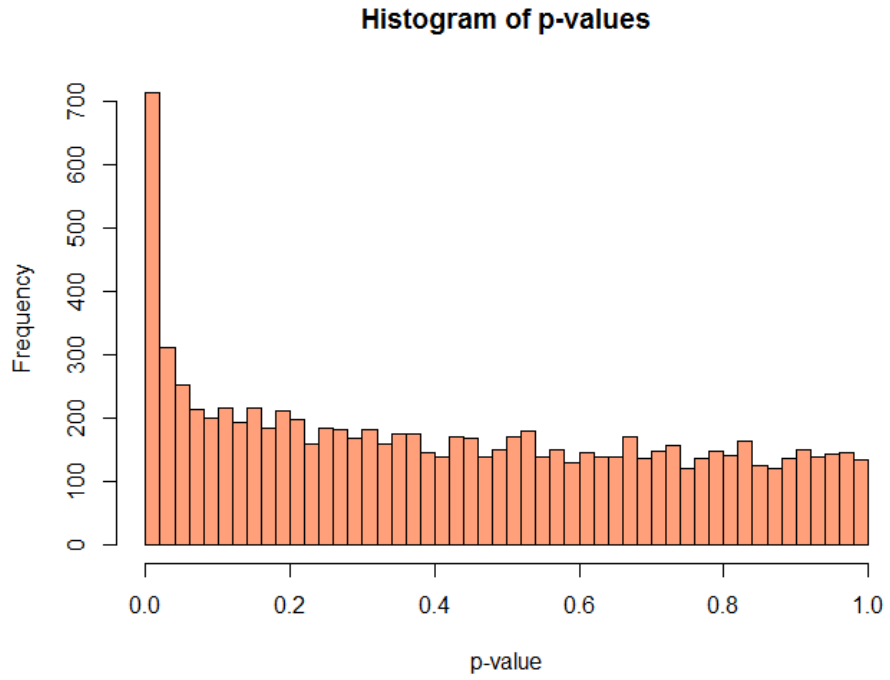

Figure 2: Histogram of p-values.

Figure 2 shows the histogram of the p-values obtained after t-tests were performed for each gene during the gene selection task. It shows a large number of probe sets that have low p-values, which represent expressed genes that are significantly different among samples. Moreover, there is a large range of genes that have large p-values, which indicates that they are not significantly different among the samples. This means that most of the genes are not useful to discriminate between the ACR/ABL samples and the NEG samples. The number of genes that have a p-value below 0.05, is 1,155. This means that from the 8,812 genes that are expressed between the ACR/ABL and the NEG samples only 1,155 are the ones that are useful to discriminate between the samples.

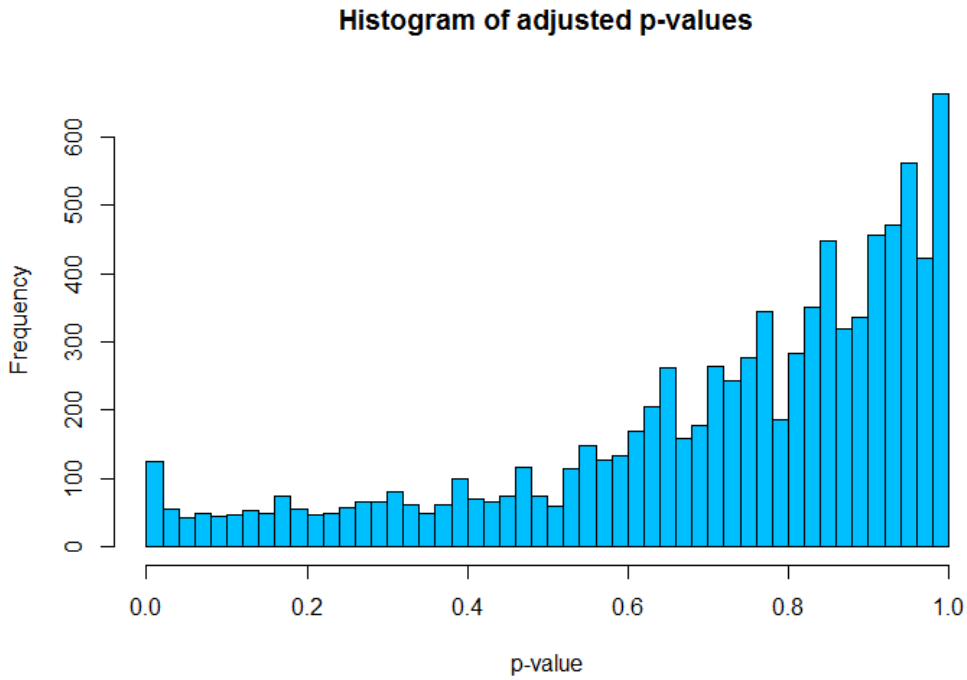

Figure 3: Histogram of adjusted p-values.

Figure 3 shows the histogram of the p-values after they were adjusted using the FDR procedure. It can be observed that the number of genes that were considered differentially expressed decreased compared to the histogram shown in figure 2. This indicates that many of the genes that were considered significantly different at first were indeed false positives. The number of genes that were considered differentially expressed using the raw p-values were 1,155, after the p-values were adjusted only 201 genes are the ones that are considered differentially expressed. This means that the mt.rawp2adjp function identified 954 genes that were false positives.

After these tasks were performed, 201 out of the 12,625 genes were identified as differentially expressed. These tasks were successfully performed using the tools provided by Bioconductor.

In conclusion, Bioconductor offers tools to address specific tasks involved in gene expression analysis. These tasks can be performed much faster an efficient. These tools facilitate manipulating large amount of data and help to perform the tasks that are required in gene expression analysis. There are much more packages offered by Bioconductor to address different kind of tasks. Therefore, it is necessary to choose which tools would be more appropriate for the domain of the analysis.

References

Hahne, F., Wolfgang, H., Gentleman, R., & Falcon , S. (2008). *Bioconductor Case Studies.* Springer.

Heydebrek, A. v., Wolfgang , H., & Gentleman, R. (2004). Differential Gene Expression with the Bioconductor Project. *Bioconductor Working Papers*.

Hofmann, W.-K. (2006). *Gene Expression Profiling by Microarrays.* Cambridge University Press.

McLachlan, G. J., Do, K. A., & Ambroise, C. (2004). *Analyzing Microarray Gene Expression.* John Wiley & Sons, Inc. .

Tarca, A. L., Romero, R., & Draghici, S. (2006). Analysis of microarray experiments of gene expression profiling. *National Institute of Health-Public Access*, 373-388.