BioSys PhD | Earthsystems PhD

Statistics 1

Statistical Methods for Microarray Data

Lisete Sousa
lmsousa@fc.ul.pt
Room: 6.4.25

Departamento de Estatística e Investigação Operacional
Faculdade de Ciências da Universidade de Lisboa

2017

Summary

1. Overview

In the first part of this tutorial you have learned how to pre-process microarray data using R. Now you will be presented three packages which aim at identifying differentially expressed (DE) genes: up- or down-regulated genes, under one condition against another condition, e.g. two different treatments, two different tissue types, etc.

- An aspect to consider is the fact that in almost all experiments involving microarrays, the number of replicates is considerably small, often not more than four.
- Hence, when dealing with microarray data, not only multiple testing is a problem but also the lack of robustness constitutes an important aspect to consider.
- In such small samples, outliers have enormous effect on the results since both very large and very small values for the mean can be driven by the presence of outliers in the data, which occurs quite frequently in this context (Lönnstedt and Speed, 2002).
- Although normalizing the data along the slides helps to reduce discrepancies, this procedure does not reduce substantially the effect of such observations in the results.

This section focuses on three packages for detecting DE genes, corresponding to distinct methodologies:

- Package `limma`: empirical Bayesian method;
- Package `RankProd`: non-parametric method;
- Package `nudge`: normal uniform mixture model.

2. limma Package: Empirical Bayesian Method

Introduction

- Package `limma` uses linear models to analyze microarray data.
- The approach requires the specification of one or two matrices:
  (i) the design matrix - *indicating which RNA samples have been applied to each array*;
  (ii) the contrast matrix - *specifies which comparisons should be made between the RNA samples*.
- For very simple experiments the contrast matrix may not be needed.

- The method starts by fitting a linear model in order to estimate the variability in the data.
- For statistical analysis and assessing differential expression, limma uses an empirical Bayesian method (Lönnstedt and Speed, 2002; Lin et al., 2003; Smyth, 2004) to moderate the standard errors of the estimated log-fold changes.
- This results in more stable inference and improved power, especially for experiments with small numbers of arrays.

Theoretical Aspects

- An approach to improving on the t-statistic-based methods is the empirical Bayes method for analyzing replicated two-channel microarray data (paired data) proposed by Lönnstedt and Speed (2002), which will be introduced here.
- For general microarray experiments with arbitrary numbers of treatments see Lin et al. (2003) and Smyth (2004).
- Log ratio of gene expression:

$$M_{ij} = \log_2 R_{ij}$$

Gene i $(i = 1, \ldots, m)$ Individual j $(j = 1, \ldots, J)$

- Regard the components of M_i as random variables from Normal distribution with mean μ_i and variance σ_i^2, so that independently:

$$M_{ij}|\mu_i, \sigma_i \sim N(\mu_i, \sigma_i)$$

- Let I_i be the indicator for whether a gene is differentially expressed or not,

$$I_i = \begin{cases} 0, & \mu_i = 0 \\ 1, & \mu_i \neq 0 \end{cases}$$

- The authors propose a target measure - a logarithm of posterior odds, for being differentially expressed for each gene i:

$$B_i = \ln \frac{P(I_i{=}1|\mathbf{M}_i)}{P(I_i{=}0|\mathbf{M}_i)} = \ln \frac{p\, f_{I_i=1}(\mathbf{M}_i)}{(1-p)\, f_{I_i=0}(\mathbf{M}_i)};$$

  - p is the proportion of differentially expressed genes;
  - $\mathbf{M}_i = (M_{i1}, \ldots, M_{iJ})$.
  - Gene i is differentially expressed if $B_i > 0$.

- Prior distributions:

$$\tau_i \sim Gamma(\nu, 1)$$

$$\mu_i|\tau_i \begin{cases} = 0, & I_i = 0 \\ \sim N(0, cJa/2\tau_i), & I_i = 1 \end{cases}$$

$$\tau_i = J\,a/2\sigma_i^2$$
$$\nu, a, c > 0$$

- The integration of the joint densities is performed, and hence the logarithm of posterior odds is:

$$B_i = \ln \frac{p}{(1-p)\sqrt{1+Jc}} + \left(\nu + \frac{J}{2}\right) \ln \frac{a+\sigma_i^2+\overline{M}_i^2}{a+\sigma_i^2+\overline{M}_i^2/(1+Jc)}$$

- Estimation of hyperparameters ν, a and c:
  - Fix p (usually 0.01).
  - The difficulty in estimating c is that it only occurs in the distribution of genes which are differentially expressed ($I_i = 1$) and those genes are not known. Given the relation $var(\mu_i|\sigma_i^2, I_i = 1) = c\, var(M_{ij}|\mu_i, \sigma_i^2)$, c is estimated from calculating the variance of M_i and the variance of M_{ij}.
  - Estimate ν and a using the moments method.

Advantages and Disadvantages

limma is a very complete package. Some advantages:

- Fast;
- Can be applied to both two-color (with a common reference or direct two-color design) and single-channel microarray data;
- Performs background correction and data normalization;
- For arrays with within-array replicate spots, limma uses a pooled correlation method to make full use of the duplicate spots;
- Allows the comparison of more than two groups;
- Designed for specific and complex designs.

The disadvantages of this method are mainly related to failures in multiple testing adjustment for different contrasts.

Getting Started with limma

- Package limma should be installed and loaded according to the instructions given ahead (see also ex. 5).
- The model is specified by the design matrix. Each row of the design matrix corresponds to an array in the experiment and each column corresponds to a coefficient which is used to describe the RNA sources in the experiment.
- The package estimates the fold changes and standard errors by fitting a linear model for each gene. The design matrix indicates which arrays are dye swaps.

Required functions

Some of the main functions are briefly described here. Only arguments used for both datasets will be indicated. For more information see limma vignette (Smyth et al., 2013).

- `lmFit(object,design,...)`
- This function is used in order to fit a linear model for each gene given a series of arrays. Some of the arguments required for this function are:

  object - *may be a matrix or an object of class `MAlist` (limma) or `marrayNorm` (marray), containing log ratios or log values of expression for a series of microarrays.*

  design - *the design matrix of the microarray experiment, with rows corresponding to arrays.*

- `eBayes(fit,proportion=0.01,...)`
- Computes moderated t-statistics and log-odds of differential expression by empirical Bayes method. Some of the arguments required for this function are:
  - fit - *a list object produced, for example, by `lmFit`.*
  - proportion - *proportion (number between 0 and 1) of genes which are DE.*

- `topTable(fit,number,adjust="BH",sort.by="B",...)`
- Produces a table of the top ranked genes from the fitted model. Some of the arguments required for this function are:
  - fit - *an object of class `MArrayLM` produced, for example, by `lmFit`.*
  - number - *number of genes to pick out.*
  - adjust - *method used to adjust p-values for multiple testing. The "BH" method (Benjamini and Hochberg, 1995), which controls the expected false discovery rate (FDR) below the specified value, is the default adjustment method because it is the most likely to be appropriate for microarray studies.*
  - sort.by - *statistic to sort the selected genes: "M", "A", "T" (t-statistic), "P" (p-value) and or "B" (log-odds).*

- `volcanoplot(fit,highlight=0,...)`
- Creates a volcano plot of log-fold changes (M) versus log-odds (B) of differential expression. Some of the arguments required for this function are:
  - fit - *an object of class `MArrayLM` produced by `lmFit`.*
  - highlight - *number of top genes to be highlighted.*

Application: Swirl Zebrafish Dataset

The data

- Swirl zebrafish data is a direct two-color design, and is available in `limma`.
- The main goal of the Swirl experiment is to identify genes with altered expression in the Swirl mutant compared to wild-type (wt) zebrafish.
- Two of the three packages described here (`limma` and `RankProd`) will use the data preprocessed according to the following:

```
> source("http://bioconductor.org/biocLite.R")
> biocLite("limma")
> library(limma)
>
> swirl <- readTargets("SwirlSample.txt")
> RG <- read.maimages(swirl$FileName, source="spot")
> RG$genes <- readGAL("fish.gal")
> RG$printer <- getLayout(RG$genes)
>
> MA <- backgroundCorrect(RG,method="sub")
> MA <- normalizeWithinArrays(MA,method="printtiploess")
> MA <- normalizeBetweenArrays(MA,method="scale")
```

Interpreting the output

The appropriate design matrix is created, as well as a linear model fitting for each gene using,

```
> design <- c(-1,1,-1,1)
> fit <- lmFit(MA,design)
```

The negative numbers in the design matrix indicate the dye swaps: swirl 1 and 3.

Now, follows the computation of the empirical Bayes statistics for differential expression,

```
> fit <- eBayes(fit)
> fit
```

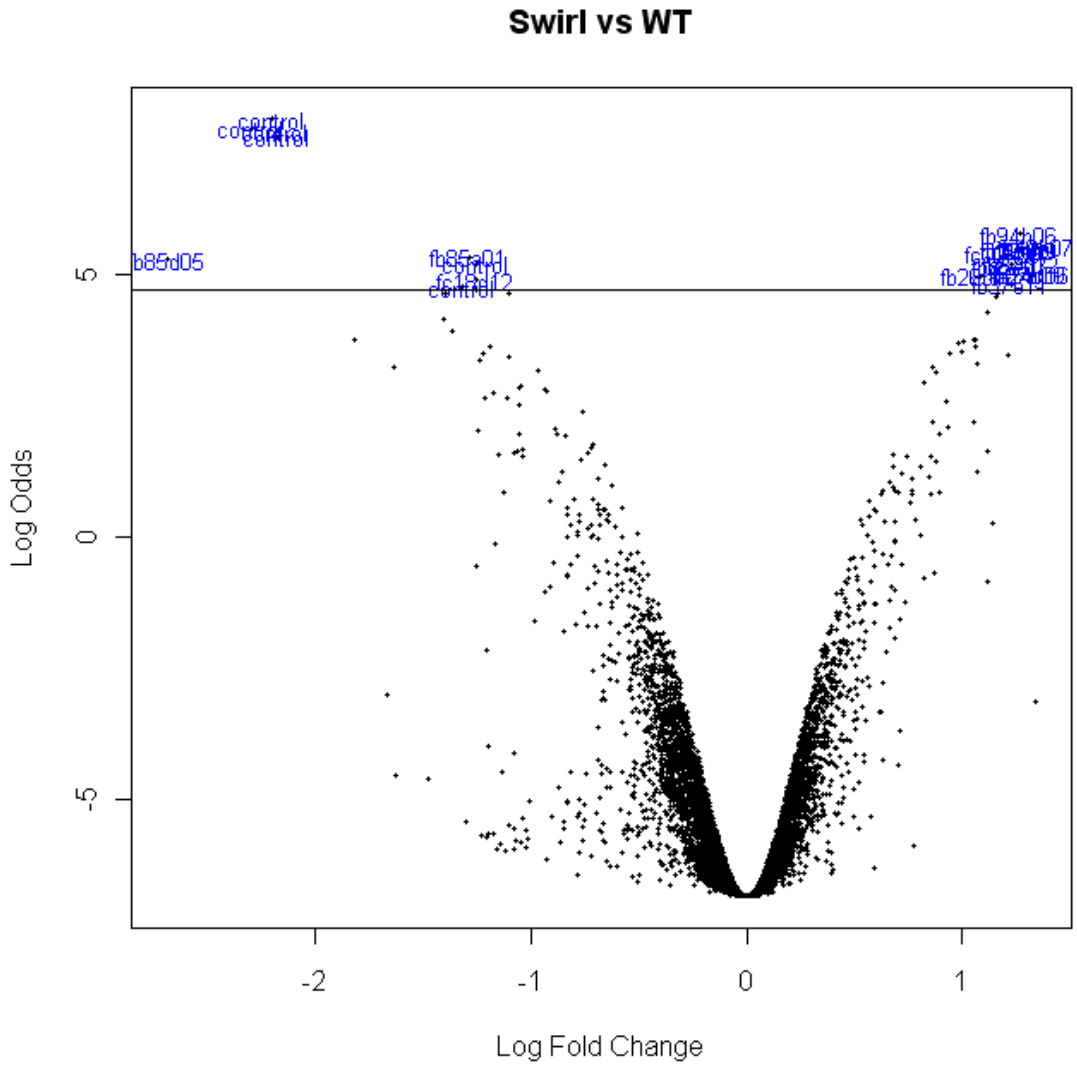
Next, you may obtain a summary table of some key statistics for the top genes,

```
> table <- topTable(fit,number=20,adjust="BH")
> table
```

```
   Block  Row  Column  ID  Name  logFC  AveExpr  t
3721  8  2  1  control  BMP2  -2.205288  12.10451  -21.06952
1609  4  2  1  control  BMP2  -2.296045  13.14286  -20.28697
3723  8  2  3  control  Dlx3  -2.184900  13.28808  -20.01066
1611  4  2  3  control  Dlx3  -2.180471  13.49555  -19.63599
8295 16  16  15  fb94h06  20-L12  1.271119  12.03651  14.08467
...
    P.Value   adj.P.Val   B
3721  1.028468e-07  0.0003572816  7.960750
1609  1.343812e-07  0.0003572816  7.778330
3723  1.480433e-07  0.0003572816  7.710959
1611  1.691674e-07  0.0003572816  7.617005
8295  1.735790e-06  0.0020666932  5.779021
...
```

- Genes are ranked according to B-statistics.
- Down-regulated (up-regulated) genes are those whose log fold change (logFC) is negative (positive).
- We have selected the top 20 genes, corresponding to $B > 4.76$.
- If we had chosen genes with $B > 0$, 141 genes would be selected.
- These genes can be highlighted in the volcano plot,

```
> volcanoplot(fit,highlight=20,names=fit$genes$NAME, main="Swirl vs WT")
> abline(4.7,0)
```



3. RankProd Package: Non-Parametric Method

Introduction

- Breitling et al. (2004) present a technique for identifying differentially expressed genes that originates from an analysis of biological reasoning.
- The technique is based on calculating rank products (RP) from replicate experiments.
- At the same time, it provides a statistical way to determine the significance level for each gene and allows for the flexible control of the false-detection rate (FDR).

Theoretical Aspects

The assumptions made for RP method are relatively weak. It is assumed that

(1) relevant expression changes affect only a minority of genes,

(2) measurements are independent between replicate arrays,

(3) most changes are independent of each other,

(4) measurement variance is about equal for all genes.

- *RP* is a non-parametric statistic used to detect genes that are consistently highly ranked (strongly up-regulated/down-regulated) in a number of replicate experiments.

- It is assumed that, under the null hypothesis that the order of all genes is random, the probability of finding a specific gene among the top r of n genes in a replicate is

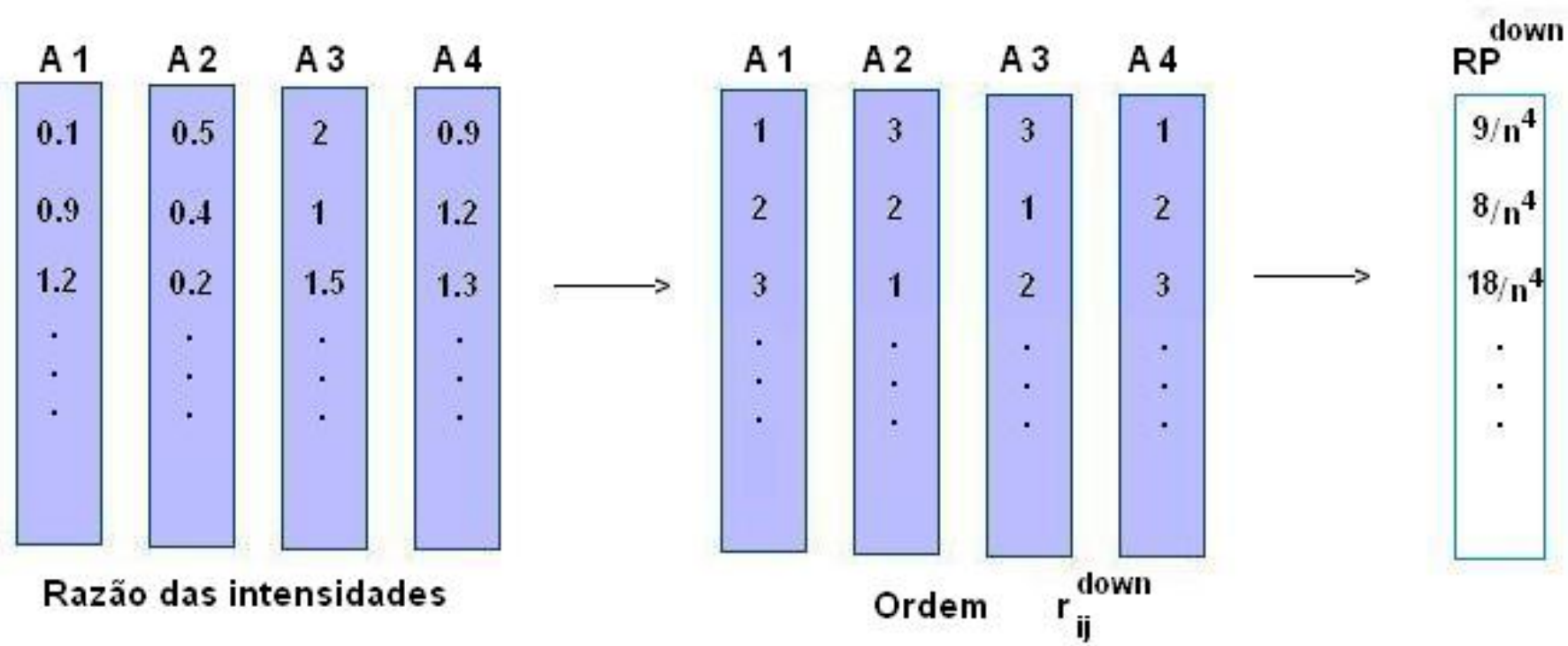
$$p = r/n.$$

- Multiplying these probabilities allows the calculation of the corresponding combined probability as a rank product

$$RP = \prod_i r_i/n_i,$$

where r_i is the position of a specific gene in the i-th replicate and n_i is the total number of genes in the i-th replicate sorted by increasing/decreasing (up-regulated/down-regulated) values.

- The smaller the *RP* value, the smaller the probability that the observed position of the gene at the top of the lists is due to chance.
- A simple permutation-based estimation procedure provides a very convenient way to determine how likely it is to observe a given *RP* value, or better, in a random experiment.
- If there is high variability in gene-specific variances, RP tends to give overly optimistic p-values. The average ranks may constitute an alternative (Breitling and Herzyk, 2006).


A 1
A 2
A 3
A 4
0.1
0.9
1.2
0.5
0.4
0.2
2
1
1.5
0.9
1.2
1.3
Razão das intensidades
A 1
A 2
A 3
A 4
1
2
3
3
2
1
3
1
2
1
2
3
Ordem
r_{ij}^{down}
RP down
$9/n^4$
$8/n^4$
$18/n^4$

Advantages and Disadvantages

Rank Products method has several advantages:

- Fast and simple;
- Results are reliable in highly noisy data;
- Result in an increased power and accuracy at small number of replicates;
- Able to combine data sets from different laboratories into one analysis to increase the power of the identification;
- Analyzes both Affymetrix and cDNA microarrays (designed with a common reference or a direct two-color design).

The main disadvantage of RP method is that there is a significant loss of performance, when the equal-variance assumption is seriously violated and the number of replicates is higher than three.

Getting Started With RankProd

Package RankProd is installed and loaded as follows,

```
> source("http://bioconductor.org/biocLite.R")
> biocLite("RankProd")
> library(RankProd)
```

The method can combine data sets from different origins (meta-analysis) to increase the power of the identification.

This package is able to analyze normalized Affymetrix Genechip data and normalized spotted cDNA array data.

Required functions

Some of the main functions are briefly described here. Only arguments used in both datasets will be indicated. For more information see RankProd vignette (Hong and Wittner, 2010).

- `RP(data,c1,num.perm=100,logged=TRUE,...)`
- Performs rank product method to identify DE genes. Some of the arguments required for this function are:
  - data - *the data set; rows correspond to genes and columns to samples.*
  - c1 - *vector containing the class labels of the samples.*
  - num.perm - *number of permutations.*
  - logged - *"TRUE" if the data is logged.*

- `plotRP(x,cutoff=NULL)`
- Plots the estimated pfp *vs* number of identified genes.
  - x - *value returned by function RP.*
  - cutoff - *threshold used to select genes according to pfp.*

- `topGene(x,cutoff=NULL,num.gene=NULL, method="pfp",logged=TRUE,...)`
- Identifies DE genes. Some of the arguments required for this function are:
  - x - *value returned by function RP.*
  - cutoff - *threshold used to select genes.*
  - num.gene - *number of candidate genes of interest; ignored if cutoff is provided.*
  - method - *method selected to identify genes if cutoff is provided; "pfp" for percentage of false prediction and "pval" for p-value.*
  - logged - *"TRUE" if the data is logged.*

Application: Swirl Zebrafish Dataset

Interpreting the output

Log ratios (wt/swirl) matrix for each gene and slide are set into `data`.

For replicates 2 and 4 the ratio corresponds to (swirl/wt), therefore you need to make a correction previously:

```
> MA$M[,2] <- (-1)*MA$M[,2]
> MA$M[,4] <- (-1)*MA$M[,4]
> data<-MA$M
```

The number of samples is printed, in order to fix the dimension of c1.

Here, c1 contains only 1's as each sample (column) corresponds to expression ratios of two channels (paired samples).

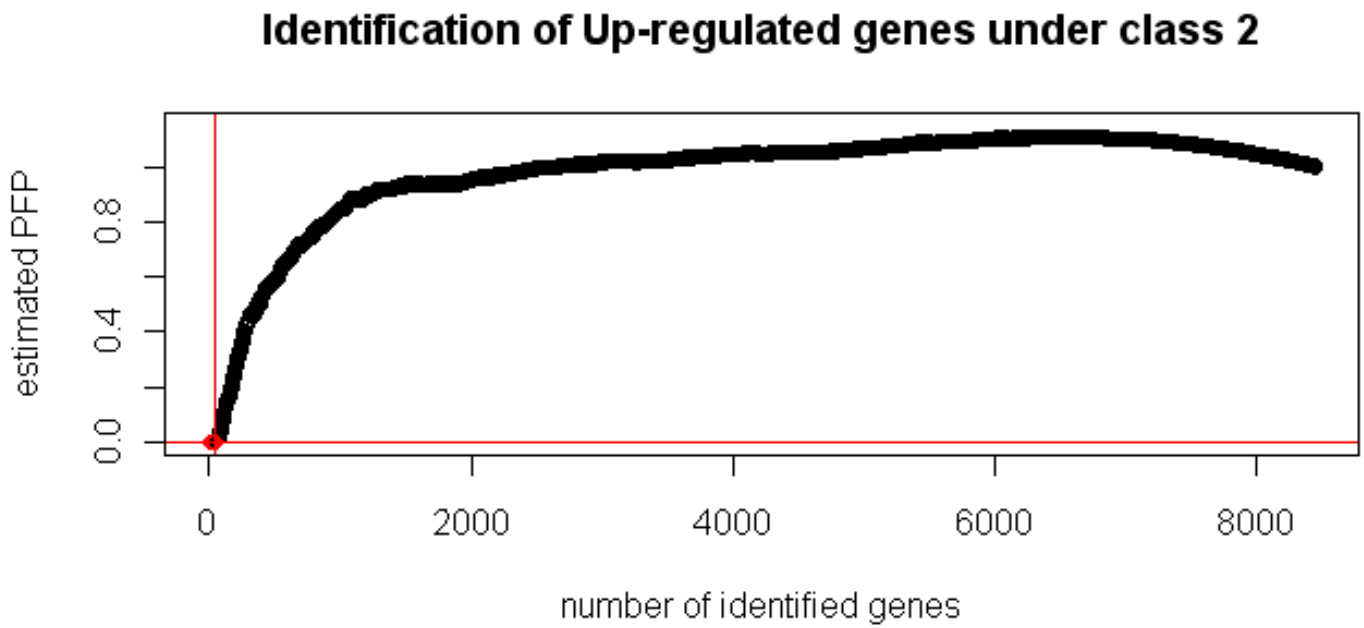
```
> k <- dim(data)[2]
> k
[1] 4
> c1 <- c(rep(1,k))
> c1
[1] 1 1 1 1
```

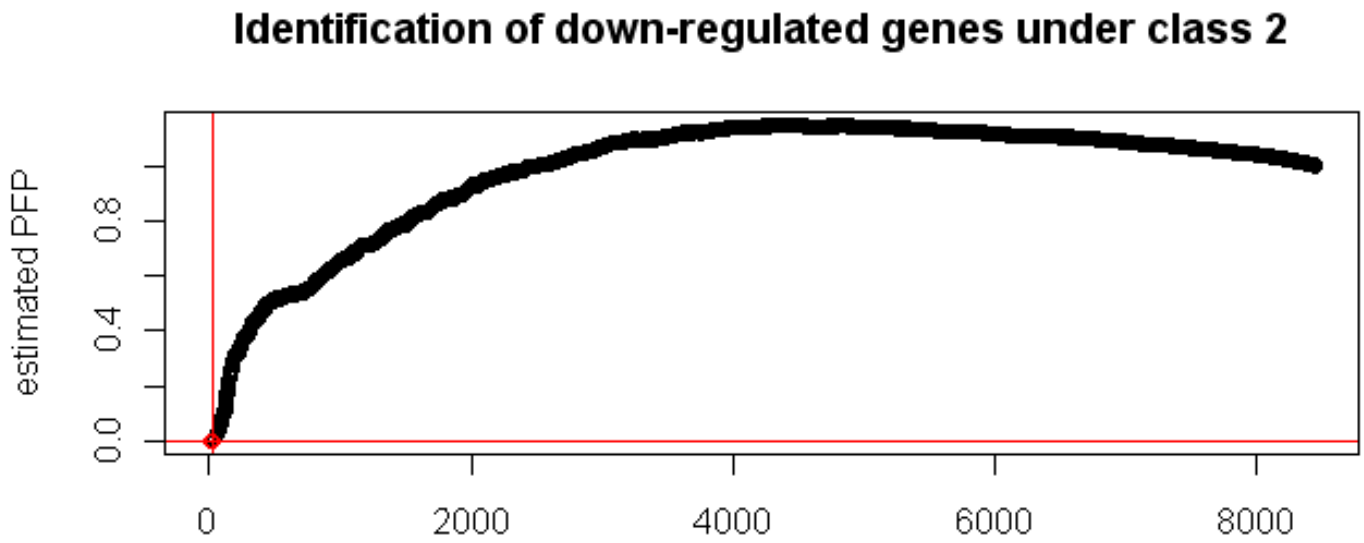
The analysis is done using,

```
> RP.out <- RP(data,c1,num.perm=100,logged=TRUE)
```

In the output, the channel used as the numerator is called as class 1 (wt) and the channel used as denominator as class 2 (swirl). Two plots for identification of up- and down-regulates genes under class 2, are generated using

```
> plotRP(RP.out,cutoff=0.001)
```


Identification of Up-regulated genes under class 2
estimated PFP
0.0
0.4
0.8
0
2000
4000
6000
8000
number of identified genes


Identification of down-regulated genes under class 2
estimated PFP
0.0
0.4
0.8
0
2000
4000
6000
8000

Considering a cutoff of 0.001 for pfp, 58 genes are selected to be DE:

```
> length(RP.out$pfp[RP.out$pfp<0.001])
[1] 58
```

The list of selected up- and down-regulated genes is based on the estimated percentage of false positive predictions (pfp), which is also known as false discovery rate (FDR).

```
> table <- topGene(RP.out,num.gene=20,logged=TRUE,
+ logbase=2,method="pfp")
```

Top 20 genes selected to be up-regulated for Swirl mutation (up-regulated under class 2):

```
> table$Table1
```

```
  gene.index  RP/Rsum  FC:(class1/class2)  pfp  P.value
 [1,]   7036   9.3246              0.3931    0        0
 [2,]   7491  11.7725              0.3990    0        0
 [3,]   4546  12.0935              0.4257    0        0
 [4,]    683  12.1175              0.4040    0        0
 [5,]   5075  14.1421              0.4130    0        0
...
[16,]   4623  28.3417              0.4618    0        0
[17,]   7542  31.7084              0.4611    0        0
[18,]   1697  34.3205              0.4711    0        0
[19,]   6449  34.3649              0.4767    0        0
[20,]   2945  36.5476              0.4829    0        0
```

Top 20 genes selected to be down-regulated for Swirl mutation (down-regulated under class 2):

```
> table$Table2
```

	gene.index	RP/Rsum	FC:(class1/class2)	pfp	P.value
[1,]	2961	4.3004	6.4590	0.0000	0
[2,]	1609	7.9922	4.9111	0.0000	0
[3,]	1611	9.3977	4.5330	0.0000	0
[4,]	3723	10.2761	4.5470	0.0000	0
[5,]	3721	10.6168	4.6117	0.0000	0
...					
[16,]	2679	55.2085	2.3799	0.0006	0
[17,]	5265	62.9521	2.3157	0.0006	0
[18,]	4454	64.0852	2.3282	0.0006	0
[19,]	3200	65.1399	2.2839	0.0016	0
[20,]	1782	66.1038	2.3614	0.0015	0

Fixing the percentage of false predictions (FDR) to be 0.0001, at least 20 genes are selected to be up-regulated and 12 to be down-regulated.

Bibliography

Breitling, R., Armengaud, P., Amtmann, A., and Herzyk, P. (2004). Rank Products: A simple, yet powerful, new method to detect differentially regulated genes in replicated microarray experiments. *FEBS Letter* 57383–92.

Breitling, R. and Herzyk, P. (2006). Rank based methods as a non-parametric alternative of the t-statistic for the analysis of biological microarray data. *Journal of Bioinformatics and Computational Biology* 3(5): 1171–1189.

Hong, F. and Wittner, B. (2010). Bioconductor RankProd Package Vignette. Bioconductor.

Loennstedt, I. and Speed, T.P. (2002). Replicated microarray data. *Statistica Sinica* 12, 31–46.

Smyth, G.K., Ritchie, M., Thorne, N., Wettenhall, J. and Shi, W. (2010). User's Guide. limma: Linear Models for Microarray Data. Bioconductor.