isomiRs: miRNAoma analysis from small-RNAseq data

Lorena Pantano [*1], ***Georgia Escaramis*** [2], ***and Eulalia Martin*** [2]

[1]Harvard TH Chan School of Public Health, Boston, US
[2]Center of Genomic Regulation, Barcelona, Spain

*lorena.pantano@gmail.com

October 30, 2017

Package

isomiRs 1.6.0

Contents

Introduction

miRNAs are small RNA fragments (18-23 nt long) that influence gene expression during development and cell stability. Morin et al [1], discovered isomiRs first time after sequencing human stem cells.

IsomiRs are miRNAs that vary slightly in sequence, which result from variations in the cleavage site during miRNA biogenesis (5'-trimming and 3'-trimming variants), nucleotide additions to the 3'-end of the mature miRNA (3'-addition variants) and nucleotide modifications (substitution variants)[2].

There are many tools designed for isomiR detection, however the majority are web application where user can not control the analysis. The two main command tools for isomiRs mapping are SeqBuster and sRNAbench[3]. *isomiRs* package is designed to analyze the output of SeqBuster tool or any other tool after converting to the desire format.

1 Citing isomiRs

If you use the package, please cite this paper [4].

2 Input format

The input should be the output of SeqBuster-miraligner tool (*.mirna files). It is compatible with mirTOP tool as well, which parses BAM files with alignments against miRNA precursors.

For each sample the file should have the following format:

seq	name	freq	mir	start	end	mism	add	t5	t3
TGTAAACATCCTACACTCAGCT	seq_100014_x23	23	hsa-miR-30b-5p	17	40	0	0	0	GT
TGTAAACATCCCTGACTGGAA	seq_100019_x4	4	hsa-miR-30d-5p	6	26	13TC	0	0	g
TGTAAACATCCCTGACTGGAA	seq_100019_x4	4	hsa-miR-30e-5p	17	37	12CT	0	0	g
CAAATTCGTATCTAGGGGATT	seq_100049_x1	1	hsa-miR-10a-3p	63	81	0	TT	0	ata
TGACCTAGGAATTGACAGCCAGT	seq_100060_x1	1	hsa-miR-192-5p	25	47	8GT	0	c	agt

This is the standard output of SeqBuster-miraligner tool, but can be converted from any other tool having the mapping information on the precursors. Read more on miraligner manual

3 IsomirDataSeq class

This object will store all raw data from the input files and some processed information used for visualization and statistical analysis. It is a subclass of *SummarizedExperiment* with `colData` and `counts` methods. Beside that, the object contains raw and normalized counts from miraligner allowing to update the summarization of miRNA expression.

3.1 Access data

The user can access the normalized count matrix with `counts(object, norm=TRUE)`.

You can browse for the same miRNA or isomiRs in all samples with `isoSelect` method.

```
library(isomiRs)
data(mirData)
head(isoSelect(mirData, mirna="hsa-let-7a-5p", 1000))
```

```
## DataFrame with 6 rows and 15 columns
##                                                id       pc1       pc2
##                                         <character> <numeric> <numeric>
## 1  hsa-let-7a-5p 0 0 0 0 : TGAGGTAGTAGGTTGTATAGTT    382703    259187
## 2 hsa-let-7a-5p 0 0 0 T : TGAGGTAGTAGGTTGTATAGTTT     14582      9490
## 3   hsa-let-7a-5p 0 0 0 gtt : TGAGGTAGTAGGTTGTATA      1355      1036
## 4   hsa-let-7a-5p 0 0 0 t : TGAGGTAGTAGGTTGTATAGT     76284     65140
## 5   hsa-let-7a-5p 0 0 0 tt : TGAGGTAGTAGGTTGTATAG      7582      5884
## 6 hsa-let-7a-5p 0 A 0 0 : TGAGGTAGTAGGTTGTATAGTTA     15438      7826
##         pc3       pc4       pc5       pc6       pc7       pt1       pt2
##   <numeric> <numeric> <numeric> <numeric> <numeric> <numeric> <numeric>
## 1    279317    353169    337896    157358    247664    111195    239647
## 2     10487     13063     12455      5908      9233      4481      8640
## 3      1097      1482      1297       673      1022       370       986
## 4     62420     91323     89100     39450     63273     25631     57218
## 5      6201      9535      8264      3808      5963      2745      5242
## 6     10425     12032     10865      5021      8075      3677      7523
##         pt3       pt4       pt5       pt6       pt7
##   <numeric> <numeric> <numeric> <numeric> <numeric>
## 1    363483    321629    110483    222561    391118
## 2     14828     12396      4467      8337     15646
## 3      1173       853       448       917      1305
## 4     90108     60010     27788     50366     79196
## 5      8086      5455      2899      5300      7485
## 6     13486     13765      3728      7498     15605
```

`metadata(mirData)` contains two lists: `rawList` is a list with same length than number of samples and stores the input files for each sample; `isoList` is a list with same length than number of samples and stores information for each isomiR type summarizing the different changes for the different isomiRs (trimming at 3', trimming a 5', addition and substitution). For instance, you can get the data stored in `isoList` for sample 1 and 5' changes with this code `metadata(ids)[["isoList"]][[1]]["t5sum"]`.

3.2 isomiRs annotation

IsomiR names follows this structure:

- miRNA name
- type: ref if the sequence is the same than the miRNA reference. 'iso' if the sequence has variations.

- t5 tag: indicates variations at 5' position. The naming contains two words: 'direction - nucleotides', where direction can be UPPER CASE NT (changes upstream of the 5' reference position) or LOWER CASE NT (changes downstream of the 5' reference position). '0' indicates no variation, meaning the 5' position is the same than the reference. After 'direction', it follows the nucleotide/s that are added (for upstream changes) or deleted (for downstream changes).
- t3 tag: indicates variations at 3' position. The naming contains two words: 'direction - nucleotides', where direction can be LOWER CASE NT (upstream of the 3' reference position) or UPPER CASE NT (downstream of the 3' reference position). '0' indicates no variation, meaning the 3' position is the same than the reference. After 'direction', it follows the nucleotide/s that are added (for downstream changes) or deleted (for upstream chanes).
- ad tag: indicates nucleotides additions at 3' position. The naming contains two words: 'direction - nucleotides', where direction is UPPER CASE NT (upstream of the 5' reference position). '0' indicates no variation, meaning the 3' position has no additions. After 'direction', it follows the nucleotide/s that are added.
- mm tag: indicates nucleotides substitutions along the sequences. The naming contains three words: 'position-nucleotideATsequence-nucleotideATreference'.
- seed tag: same than 'mm' tag, but only if the change happens between nucleotide 2 and 8.

In general nucleotides in UPPER case mean insertions respect to the reference sequence, and nucleotides in LOWER case mean deletions respect to the reference sequence.

4 Quick start

We are going to use a small RNAseq data from human brain samples [5] to give some basic examples of isomiRs analyses.

In this data set we will find two groups:

- pc: 7 control individuals
- pt: 7 patients with Parkinson's Disease in early stage.

```
library(isomiRs)
data(mirData)
```

4.1 Reading input

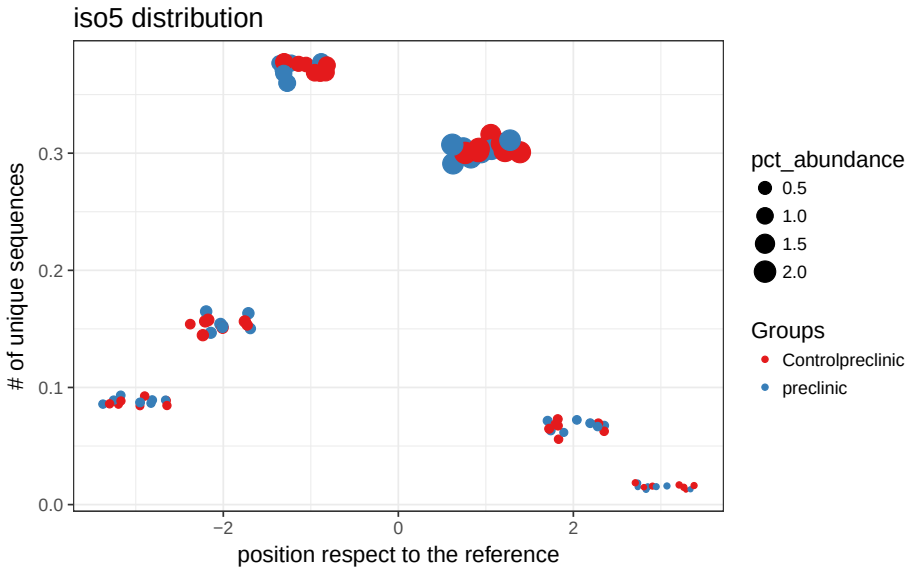
The function `IsomirDataSeqFromFiles` needs a vector with the paths for each file and a data frame with the design experiment similar to the one used for a mRNA differential expression analysis. Row names of the data frame should be the names for each sample in the same order than the list of files.

```
ids <- IsomirDataSeqFromFiles(fn_list, design=de)
```

4.2 Descriptive analysis

You can plot isomiRs expression with `isoPlot`. In this figure you will see how abundant is each type of isomiRs at different positions considering the total abundance and the total number of sequences. The `type` parameter controls what type of isomiRs to show. It can be trimming (iso5 and iso3), addition (add) or substitution (subs) changes.

```
ids <- isoCounts(mirData)
isoPlot(ids, type="iso5", column = "group")
```



4.3 Count data

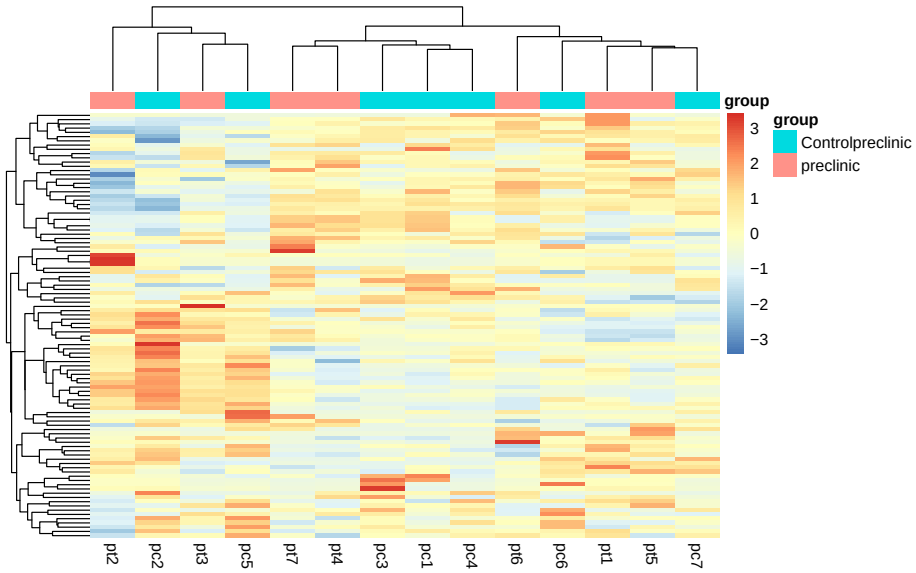
`isoCounts` gets the count matrix that can be used for many different downstream analyses changing the way isomiRs are collapsed. The following command will merge all isomiRs into one feature: the reference miRNA.

```
head(counts(ids))
```

```
##                    pc2    pt2    pt7    pc1    pt6    pc3    pt3    pt5
## hsa-let-7a-2-3p     11      7     10     13      4     13      9      3
## hsa-let-7a-3p      928    745   1159   1293    613    973   1361    433
## hsa-let-7a-5p   355578 324134 517950 507046 299028 375836 500423 152191
## hsa-let-7b-3p     1971   1410   1595   1646   1055   1267   1997    566
## hsa-let-7b-5p    77274  65928  92828 114643  53345  78586  96965  28974
## hsa-let-7c-3p       26     20     76     68     49     53     39     21
##                    pt4    pc5    pc4    pc7    pc6    pt1
## hsa-let-7a-2-3p      0     14     20      6     10      2
## hsa-let-7a-3p      978   1614   1050   1219    637    542
## hsa-let-7a-5p   419754 468792 489195 340782 215635 150421
## hsa-let-7b-3p     1148   2852   1986   1724    875    760
## hsa-let-7b-5p    71768  93764  97902  68304  43050  29572
## hsa-let-7c-3p       52     45     54     56     27     22
```

The normalization uses `rlog` from *DESeq2* package and allows quick integration to another analyses like heatmap, clustering or PCA.

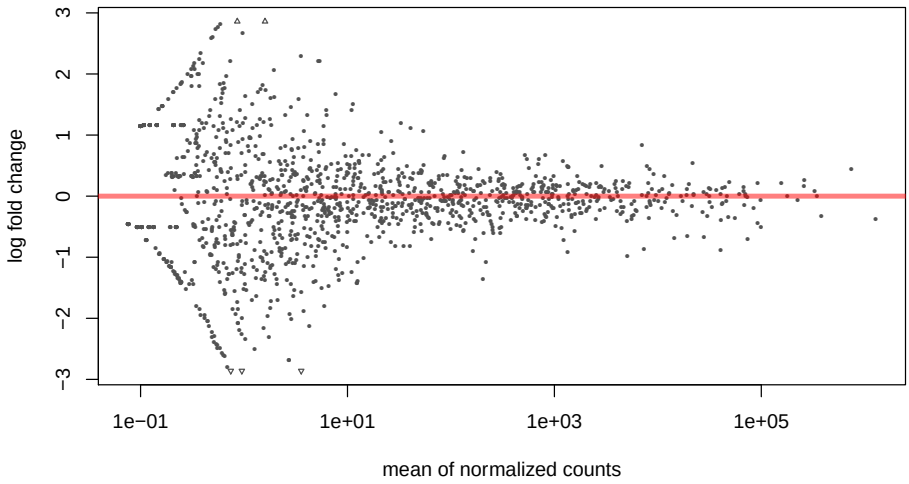
```
library(pheatmap)
ids = isoNorm(ids, formula = ~ group)
pheatmap(counts(ids, norm=TRUE)[1:100,],
         annotation_col = data.frame(colData(ids)[,1,drop=FALSE]),
         show_rownames = FALSE, scale="row")
```



4.4 Differential expression analysis

The `isoDE` uses functions from *DESeq2* package. This function has parameters to create a matrix using only the reference miRNAs, all isomiRs, or some of them. This matrix and the design matrix are the inputs for DESeq2. The output will be a DESeqDataSet object, allowing to generate any plot or table explained in DESeq2 package vignette.

```
dds <- isoDE(ids, formula=~group)
library(DESeq2)
plotMA(dds)
```




```
head(results(dds, format="DataFrame"))
```

```
## log2 fold change (MLE): group preclinic vs Controlpreclinic
## Wald test p-value: group preclinic vs Controlpreclinic
## DataFrame with 6 rows and 6 columns
##                     baseMean log2FoldChange     lfcSE        stat     pvalue
##                    <numeric>      <numeric> <numeric>   <numeric>  <numeric>
## hsa-let-7a-2-3p 8.282474e+00   -1.034311579 0.5180708 -1.99646767 0.04588304
## hsa-let-7a-3p   9.346179e+02   -0.164169458 0.2420068 -0.67836707 0.49753898
## hsa-let-7a-5p   3.467309e+05   -0.002840299 0.2177472 -0.01304402 0.98959267
## hsa-let-7b-3p   1.475014e+03   -0.316417693 0.3152244 -1.00378553 0.31548200
## hsa-let-7b-5p   6.872642e+04   -0.143770326 0.2306833 -0.62323671 0.53312898
## hsa-let-7c-3p   3.978041e+01    0.048300096 0.2145063  0.22516869 0.82184805
##                       padj
##                  <numeric>
## hsa-let-7a-2-3p 0.9852389
## hsa-let-7a-3p   0.9852389
## hsa-let-7a-5p   0.9971689
## hsa-let-7b-3p   0.9852389
## hsa-let-7b-5p   0.9852389
## hsa-let-7c-3p   0.9852389
```

You can differentiate between reference sequences and isomiRs at 5' end with this command:

```
dds = isoDE(ids, formula=~group, ref=TRUE, iso5=TRUE)
head(results(dds, tidy=TRUE))
```

```
##                         row    baseMean log2FoldChange     lfcSE       stat
## 1 hsa-let-7a-2-3p.iso.t5:0   3.3721956     -1.8884006 0.7912017 -2.3867498
## 2 hsa-let-7a-2-3p.iso.t5:A   0.1684532     -1.0125876 3.0746413 -0.3293352
## 3 hsa-let-7a-2-3p.ref.t5:0   4.6743318     -0.4022899 0.6242767 -0.6444096
## 4   hsa-let-7a-3p.iso.t5:0 633.9291305     -0.1123118 0.2165499 -0.5186417
## 5   hsa-let-7a-3p.iso.t5:A   1.8192053      1.1303400 0.9964880  1.1343238
## 6 hsa-let-7a-3p.iso.t5:TAA   0.2865428     -1.0504155 3.0735687 -0.3417576
##       pvalue      padj
## 1 0.01699806 0.9835941
## 2 0.74190234 0.9835941
```

```
## 3 0.51930985 0.9835941
## 4 0.60401061 0.9835941
## 5 0.25665876 0.9835941
## 6 0.73253331 0.9835941
```

Alternative, for more complicated cases or if you want to control more the differential expression analysis paramters you can use directly *DESeq2* package feeding it with the output of `counts(ids)` and `colData(ids)` like this:

```
dds = DESeqDataSetFromMatrix(counts(ids),
                             colData(ids), design = ~ group)
```

4.5 Supervised classification

Partial Least Squares Discriminant Analysis (PLS-DA) is a technique specifically appropriate for analysis of high dimensionality data sets and multicollineality [6]. PLS-DA is a supervised method (i.e. makes use of class labels) with the aim to provide a dimension reduction strategy in a situation where we want to relate a binary response variable (in our case young or old status) to a set of predictor variables. Dimensionality reduction procedure is based on orthogonal transformations of the original variables (isomiRs) into a set of linearly uncorrelated latent variables (usually termed as components) such that maximizes the separation between the different classes in the first few components [7]. We used sum of squares captured by the model (R2) as a goodness of fit measure. We implemented this method using the *DiscriMiner* into `isoPLSDA` function. The output p-value of this function will tell about the statistical significant of the group separation using miRNA expression data. Moreover, the function `isoPLSDAplot` helps to visualize the results. It will plot the samples using the significant components (t1, t2, t3 ...) from the PLS-DA analysis and the samples distribution along the components.

```
ids = isoCounts(ids, iso5=TRUE, minc=10, mins=6)
ids = isoNorm(ids, formula = ~ group)
pls.ids = isoPLSDA(ids, "group", nperm = 2)
df = isoPLSDAplot(pls.ids)
```

The analysis can be done again using only the most important discriminant isomiRS from the PLS-DA models based on the analysis. We used Variable Importance for the Projection (VIP) criterion to select the most important features, since takes into account the contribution of a specific predictor for both the explained variability on the response and the explained variability on the predictors.

```
pls.ids = isoPLSDA(ids,"group", refinment = FALSE, vip = 0.8)
```

Session info

Here is the output of `sessionInfo` on the system on which this document was compiled:

- R version 3.4.2 (2017-09-28), `x86_64-pc-linux-gnu`
- Locale: `LC_CTYPE=en_US.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=en_US.UTF-8`, `LC_COLLATE=C`, `LC_MONETARY=en_US.UTF-8`, `LC_MESSAGES=en_US.UTF-8`, `LC_PAPER=en_US.UTF-8`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=en_US.UTF-8`, `LC_IDENTIFICATION=C`
- Running under: `Ubuntu 16.04.3 LTS`
- Matrix products: default
- BLAS: `/home/biocbuild/bbs-3.6-bioc/R/lib/libRblas.so`
- LAPACK: `/home/biocbuild/bbs-3.6-bioc/R/lib/libRlapack.so`
- Base packages: base, datasets, grDevices, graphics, methods, parallel, stats, stats4, utils
- Other packages: Biobase 2.38.0, BiocGenerics 0.24.0, DESeq2 1.18.0, DelayedArray 0.4.0, DiscriMiner 0.1-29, GenomeInfoDb 1.14.0, GenomicRanges 1.30.0, IRanges 2.12.0, RcppEigen 0.3.3.3.0, S4Vectors 0.16.0, SummarizedExperiment 1.8.0, TMB 1.7.11, bindrcpp 0.2, isomiRs 1.6.0, knitr 1.17, matrixStats 0.52.2, pheatmap 1.0.8
- Loaded via a namespace (and not attached): AnnotationDbi 1.40.0, BiocParallel 1.12.0, BiocStyle 2.6.0, DBI 0.7, Formula 1.2-2, GGally 1.3.2, GenomeInfoDbData 0.99.1, Hmisc 4.0-3, KernSmooth 2.23-15, MASS 7.3-47, Matrix 1.2-11, R6 2.2.2, RColorBrewer 1.1-2, RCurl 1.95-4.8, RSQLite 2.0, Rcpp 0.12.13, XML 3.98-1.9, XVector 0.18.0, acepack 1.4.1, annotate 1.56.0, assertthat 0.2.0, backports 1.1.1, base64enc 0.1-3, bindr 0.1, bit 1.1-12, bit64 0.9-7, bitops 1.0-6, blob 1.1.0, caTools 1.17.1, checkmate 1.8.5, cluster 2.0.6, colorspace 1.3-2, compiler 3.4.2, data.table 1.10.4-3, digest 0.6.12, dplyr 0.7.4, evaluate 0.10.1, foreign 0.8-69, gamlss 5.0-4, gamlss.data 5.0-0, gamlss.dist 5.0-3, gdata 2.18.0, genefilter 1.60.0, geneplotter 1.56.0, ggplot2 2.2.1, glue 1.2.0, gplots 3.0.1, grid 3.4.2, gridExtra 2.3, gtable 0.2.0, gtools 3.5.0, highr 0.6, hms 0.3, htmlTable 1.9, htmltools 0.3.6, htmlwidgets 0.9, labeling 0.3, lattice 0.20-35, latticeExtra 0.6-28, lazyeval 0.2.1, lme4 1.1-14, locfit 1.5-9.1, magrittr 1.5, memoise 1.1.0, minqa 1.2.4, munsell 0.4.3, nlme 3.1-131, nloptr 1.0.4, nnet 7.3-12, pkgconfig 2.0.1, plyr 1.8.4, purrr 0.2.4, readr 1.1.1, reshape 0.8.7, rlang 0.1.2, rmarkdown 1.6, rpart 4.1-11, rprojroot 1.2, scales 0.5.0, splines 3.4.2, stringi 1.1.5, stringr 1.2.0, survival 2.41-3, tibble 1.3.4, tidyr 0.7.2, tidyselect 0.2.2, tools 3.4.2, xtable 1.8-2, yaml 2.1.14, zlibbioc 1.24.0

References

[1] R. D. Morin, M. D. O'Connor, M. Griffith, F. Kuchenbauer, A. Delaney, A.-L. Prabhu, Y. Zhao, H. McDonald, T. Zeng, M. Hirst, C. J. Eaves, and M. A. Marra. Application of massively parallel sequencing to microRNA profiling and discovery in human embryonic stem cells. *Genome Res.*, 18:610–621, 2008. doi:10.1101/gr.7179508, PMID:18285502.

[2] Eulàlia Martí, Lorena Pantano, Mónica Bañez Coronel, Franc Llorens, Elena Miñones Moyano, Sílvia Porta, Lauro Sumoy, Isidre Ferrer, and Xavier Estivill. A myriad of miRNA variants in control and Huntington's disease brain regions detected by massively parallel sequencing. *Nucleic Acids Res.*, 38:7219–35, 2010. doi:10.1093/nar/gkq575, PMID:20591823.

[3] Barturen Guillermo, Rueda Antonio, Hamberg Maarten, Alganza Angel, Lebron Ricardo, Kotsyfakis Michalis, Shi BuJun, KoppersLalic Danijela, and Hackenberg Michael. sRNAbench: profiling of small RNAs and its sequence variants in single or multi-species high-throughput experiments. *Methods in Next Generation Sequencing*, 1(1):2084–7173, 2014. doi:10.2478/mngs-2014-0001.

[4] Estivil X Pantano L, Marti E. SeqBuster. *Nucleic Acids Res.*, 38:e34, 2010. doi:10.1093/nar/gkp1127, PMID:20008100.

[5] Lorena Pantano, Marc R Friedlander, Georgia Escaramis, Esther Lizano, Joan Pallares-Albanell, Isidre Ferrer, Xavier Estivill, and Eulalia Marti. Specific small-RNA signatures in the amygdala at premotor and motor stages of Parkinson's disease revealed by deep sequencing analysis. *Bioinformatics (Oxford, England)*, nov 2015. URL: http://www.ncbi.nlm.nih.gov/pubmed/26530722, doi:10.1093/bioinformatics/btv632.

[6] Miguel Pérez-Enciso and Michel Tenenhaus. Prediction of clinical outcome with microarray data: a partial least squares discriminant analysis (PLS-DA) approach. *Human genetics*, 112:581–592, 2003. doi:10.1007/s00439-003-0921-9, PMID:12607117.

[7] Jianguo Xia and David S Wishart. Web-based inference of biological patterns, functions and pathways from metabolomic data using MetaboAnalyst. *Nature protocols*, 6:743–760, 2011. doi:10.1038/nprot.2011.319, PMID:21637195.