isomiRs

Lorena Pantano[1*], Georgia Escaramis[2*], Eulalia Martin[2*]
Harvard H Chan School of Public Health, Boston, US;
[2] Center of Genomic Regulation, Barcelona, Spain;
[*]lpantano (at) iscb.org

Modified: 3 Feb, 2015. Compiled: October 17, 2016

Contents

Introduction

miRNA 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) for each sample in the following format:

seq	name	freq	mir	start	end	mism	add	t5	t3	s5	s3	DB	am
TGTAAACATCCTACACTCAGCTGT		seq_100014_x23	23		hsa-miR-30b-5p	17		40		0		0	0
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"))
```

```
## DataFrame with 6 rows and 6 columns
##                                                nb1       nb2       nb3        o1
##                                          <numeric> <numeric> <numeric> <numeric>
```

```
## hsa-let-7a-5p.iso.t5:0.seed:0.t3:0.ad:A.mm:0       42075     46862     39425     39459
## hsa-let-7a-5p.iso.t5:0.seed:0.t3:0.ad:AA.mm:0       4202      2861      2923      1332
## hsa-let-7a-5p.iso.t5:0.seed:0.t3:0.ad:AAA.mm:0        56        50        73        46
## hsa-let-7a-5p.iso.t5:0.seed:0.t3:0.ad:AC.mm:0         57        13        32         0
## hsa-let-7a-5p.iso.t5:0.seed:0.t3:0.ad:AG.mm:0         44        40        37        23
## hsa-let-7a-5p.iso.t5:0.seed:0.t3:0.ad:AT.mm:0       1271      1139      1126       347
##                                                       o2        o3
##                                                <numeric> <numeric>
## hsa-let-7a-5p.iso.t5:0.seed:0.t3:0.ad:A.mm:0      25513     42805
## hsa-let-7a-5p.iso.t5:0.seed:0.t3:0.ad:AA.mm:0      1296      1277
## hsa-let-7a-5p.iso.t5:0.seed:0.t3:0.ad:AAA.mm:0       30        56
## hsa-let-7a-5p.iso.t5:0.seed:0.t3:0.ad:AC.mm:0         0        14
## hsa-let-7a-5p.iso.t5:0.seed:0.t3:0.ad:AG.mm:0        21        18
## hsa-let-7a-5p.iso.t5:0.seed:0.t3:0.ad:AT.mm:0       359       306
```

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 frontal cortex samples [5] to give some basic examples of isomiRs analyses.

In this data set we will find two groups:

- b: 3 individuals with less than a year
- o: 3 individuals in the elderly.

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

4.1 Reading input

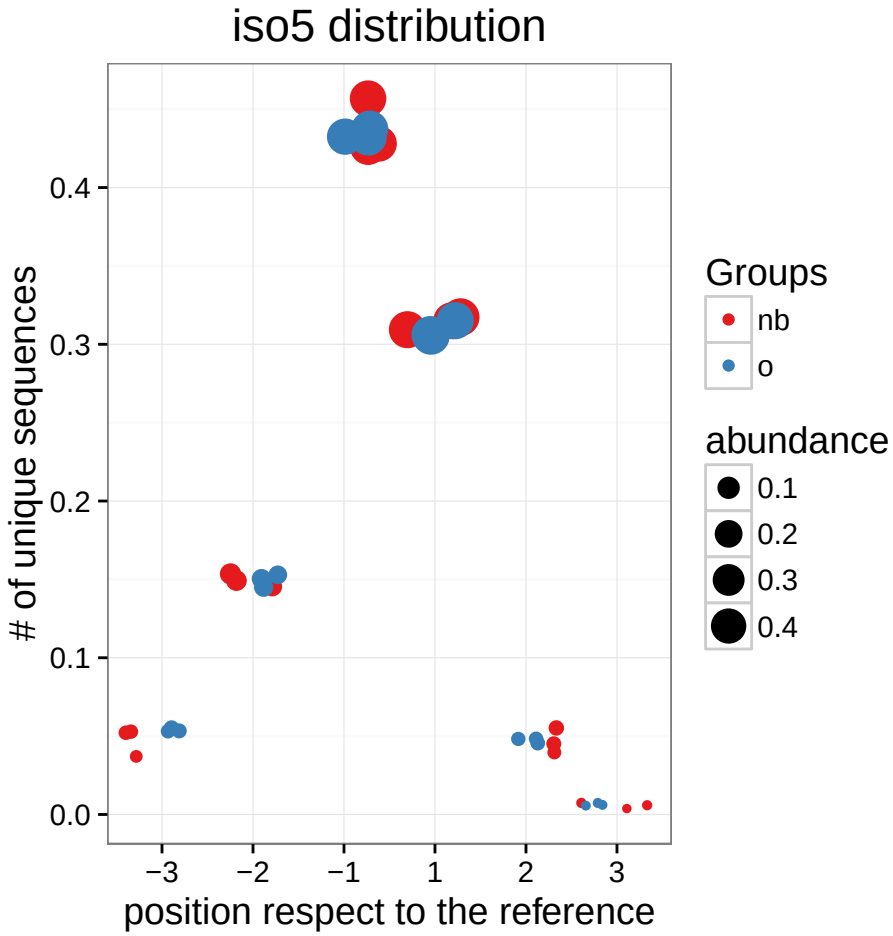
The function `IsomirDataSeqFromFiles` needs a vector with the paths for each file and a data frame with the design experiment similar to the one used for a mRNA differential expression analysis. Row names of data frame should be the names for each sample in the same order than the list of files.

```
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")
```



4.3 Count data

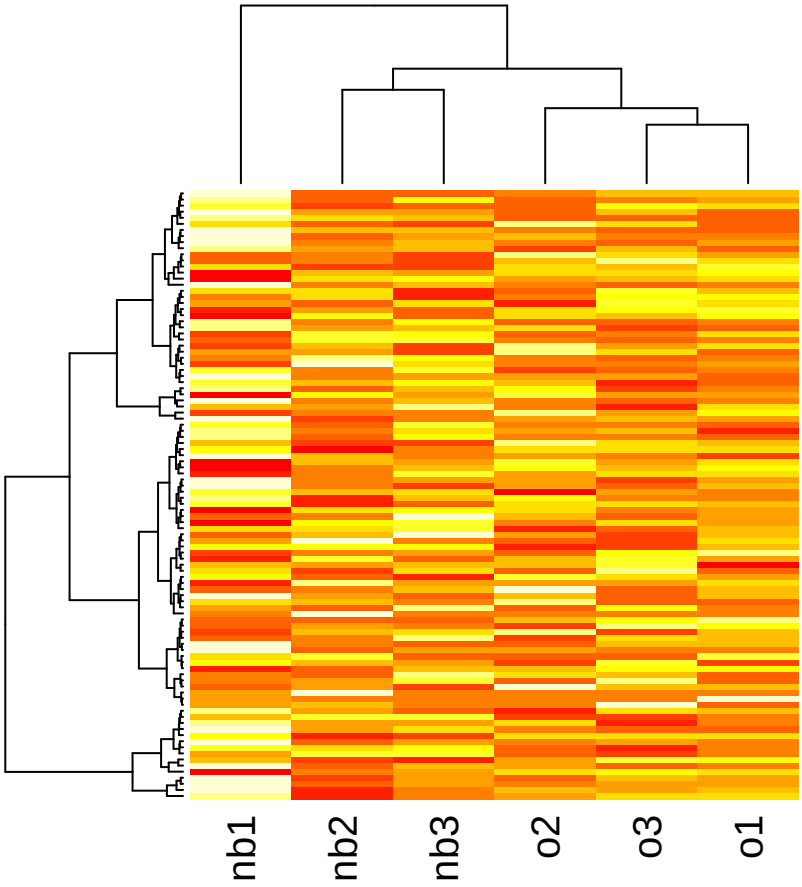
`isoCounts` gets the count matrix that can be used for many different downstream analyses changing the way isomiRs are collapsed. The following command will merge all isomiRs into one feature: the reference miRNA.

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

```
##                  nb1    nb2    nb3     o1     o2     o3
## hsa-let-7a-3p     24     70     23     47     26     65
## hsa-let-7a-5p 427615 544663 427219 556660 325845 625602
## hsa-let-7b-3p     12     38     17     24     27     33
## hsa-let-7b-5p 109767 188394 125986 150227 104593 160253
## hsa-let-7c-3p      0      1      2      0      0      3
## hsa-let-7c-5p 481931 462630 363116 425470 272375 434007
```

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

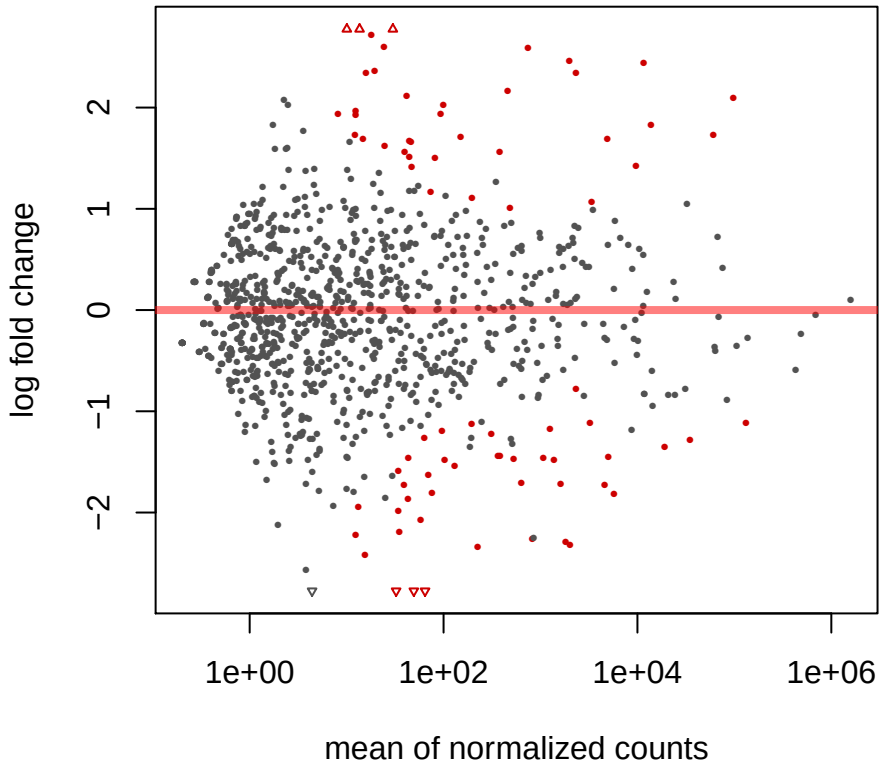
```
ids = isoNorm(ids)
heatmap(counts(ids, norm=TRUE)[1:100,], labRow = "")
```



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=~condition)
library(DESeq2)
plotMA(dds)
```




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

```
## log2 fold change (MAP): condition o vs nb
## Wald test p-value: condition o vs nb
## DataFrame with 6 rows and 6 columns
##                    baseMean log2FoldChange     lfcSE        stat    pvalue      padj
##                   <numeric>      <numeric> <numeric>   <numeric> <numeric> <numeric>
## hsa-let-7a-3p 3.877756e+01     0.10979062 0.4887353  0.22464230 0.8222576 0.9223461
## hsa-let-7a-5p 4.870364e+05    -0.23729093 0.4242072 -0.55937503 0.5759058 0.8166049
## hsa-let-7b-3p 2.348284e+01     0.25357968 0.5095023  0.49770070 0.6186950 0.8484268
## hsa-let-7b-5p 1.377705e+05    -0.27447071 0.3684119 -0.74501051 0.4562654 0.7169356
## hsa-let-7c-3p 8.503548e-01    -0.06368802 0.8554810 -0.07444703 0.9406547        NA
## hsa-let-7c-5p 4.309203e+05    -0.59300080 0.4723486 -1.25543043 0.2093225 0.5076990
```

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

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

```
##                       row  baseMean log2FoldChange     lfcSE        stat    pvalue
## 1  hsa-let-7a-3p.iso.t5:0 18.174659     -0.0147846 0.5485291 -0.02695318 0.9784971
```

```
## 2  hsa-let-7a-3p.iso.t5:c  3.482281      0.7353251 0.8043144  0.91422589 0.3605982
## 3 hsa-let-7a-3p.iso.t5:ct  4.148260      0.4996872 0.7783157  0.64201102 0.5208660
## 4  hsa-let-7a-3p.ref.t5:0  7.052740     -0.2545893 0.6761882 -0.37650660 0.7065403
## 5  hsa-let-7a-3p.ref.t5:c  4.231561      0.2260381 0.7555882  0.29915509 0.7648217
## 6 hsa-let-7a-3p.ref.t5:ct  1.192198     -0.3630919 0.8902994 -0.40783118 0.6833976
##        padj
## 1 0.9912659
## 2        NA
## 3 0.7652165
## 4 0.8787074
## 5 0.9025462
## 6        NA
```

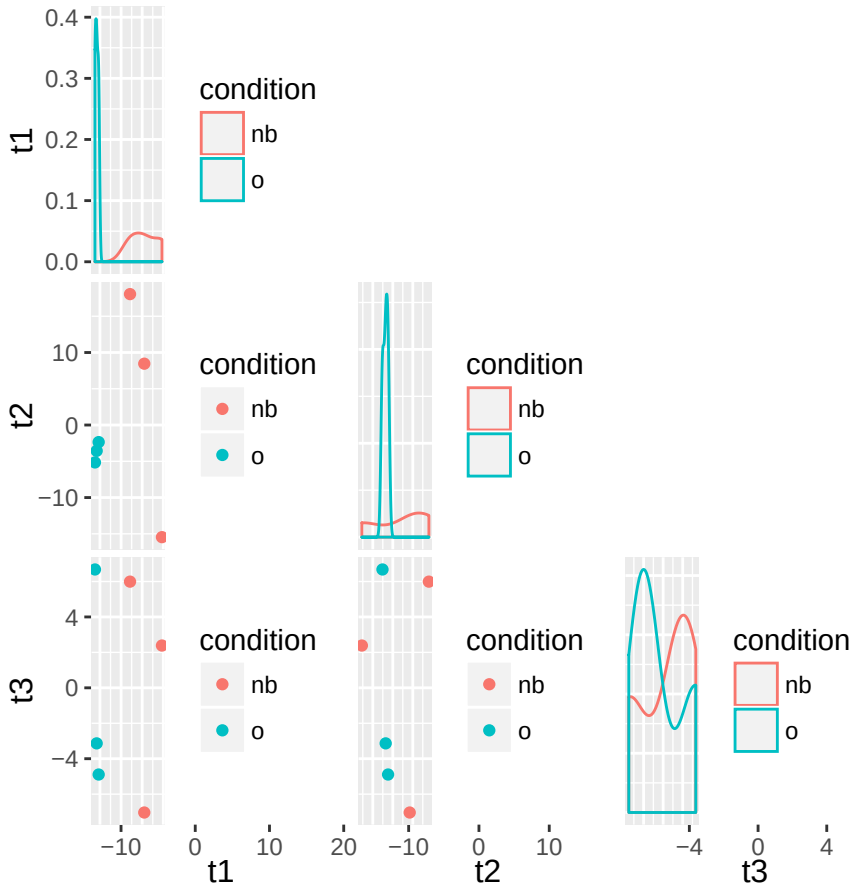
Alternative, for more complicated cases or if you want to control more the differential expression analysis paramters you can use directly *DESeq2* package feeding it with the output of `counts(ids)` and `colData(ids)` like this:

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

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

Session info

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

- R version 3.3.1 (2016-06-21), `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`
- Base packages: base, datasets, grDevices, graphics, methods, parallel, stats, stats4, utils
- Other packages: Biobase 2.34.0, BiocGenerics 0.20.0, DESeq2 1.14.0, DiscriMiner 0.1-29, GenomeInfoDb 1.10.0, GenomicRanges 1.26.0, IRanges 2.8.0, S4Vectors 0.12.0, SummarizedExperiment 1.4.0, isomiRs 1.2.0, knitr 1.14
- Loaded via a namespace (and not attached): AnnotationDbi 1.36.0, BiocParallel 1.8.0, BiocStyle 2.2.0, DBI 0.5-1, Formula 1.2-1, GGally 1.2.0, Hmisc 3.17-4, KernSmooth 2.23-15, Matrix 1.2-7.1, R6 2.2.0, RColorBrewer 1.1-2, RCurl 1.95-4.8, RSQLite 1.0.0, Rcpp 0.12.7, XML 3.98-1.4, XVector 0.14.0, acepack 1.3-3.3, annotate 1.52.0, assertthat 0.1, bitops 1.0-6, caTools 1.17.1, chron 2.3-47, cluster 2.0.5, colorspace 1.2-7, data.table 1.9.6, digest 0.6.10, dplyr 0.5.0, evaluate 0.10, foreign 0.8-67, formatR 1.4, gdata 2.17.0, genefilter 1.56.0, geneplotter 1.52.0, ggplot2 2.1.0, gplots 3.0.1, grid 3.3.1, gridExtra 2.2.1, gtable 0.2.0, gtools 3.5.0, highr 0.6, labeling 0.3, lattice 0.20-34, latticeExtra 0.6-28, lazyeval 0.2.0, locfit 1.5-9.1, magrittr 1.5, munsell 0.4.3, nnet 7.3-12, plyr 1.8.4, readr 1.0.0, reshape 0.8.5, rpart 4.1-10, scales 0.4.0, splines 3.3.1, stringi 1.1.2, stringr 1.1.0, survival 2.39-5, tibble 1.2, tidyr 0.6.0, tools 3.3.1, xtable 1.8-2, zlibbioc 1.20.0

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] Mehmet Somel, Song Guo, Ning Fu, Zheng Yan, Hai Yang Hu, Ying Xu, Yuan Yuan, Zhibin Ning, Yuhui Hu, Corinna Menzel, Hao Hu, Michael Lachmann, Rong Zeng, Wei Chen, and Philipp Khaitovich. MicroRNA, mRNA, and protein expression link development and aging in human and macaque brain. *Genome Research*, 20(9):1207–1218, 2010. doi:10.1101/gr.106849.110.

[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.