miRtest v. 1.5 Package Vignette

Stephan Artmann <stephan-artmann@gmx.net>,
Klaus Jung,
Tim Beißbarth

Göttingen 2012

Contents

1 Introduction

High-throughput measurements of gene expression are gaining popularity. So are microRNA analyses. The 'miRtest' package (Artmann *et˜al.*, 2012) intends to help researchers find differentially expressed miRNAs between two groups.

'miRtest' tries to improve power when testing for differentially regulated miRNAs by incorporation of their regulated gene sets' expression data.

miRNA-wise testing is done with the linear models implemented in the 'limma' package (Smyth, 2004). For gene set testing, different procedures can be chosen from: the self-contained tests 'globaltest' (Goeman *et˜al.*, 2004), 'GlobalAncova' (Mansmann and Meister, 2005; Hummel *et˜al.*, 2008), 'RepeatedHighDim' (Jung *et˜al.*, sub; Brunner, 2009), the rotation tests 'ROAST' (Wu *et˜al.*, 2010) and 'Romer' (Majewski *et˜al.*, 2010) as well as non-rotation enrichment tests.

2   Simple Example

The main function of 'miRtest' is 'miR.test'. It requires the user to supply an expression matrix X of miRNAs with miRNAs in its rows and microarray samples in its columns. Additionally, the procedures require an analogous matrix Y of mRNA expression values. Finally, a data.frame A is necessary: it defines which mRNAs are attacked by which miRNA. To begin with, we will generate random expression data: miRNA expression matrix X, mRNA expression matrix Y and an allocation data.frame A.

```
> ########################################
> ### Generate random expression data ###
> ########################################
> # Generate random miRNA expression data of 3 miRNAs
> # with 8 replicates
> set.seed(1)
> X = rnorm(24);
> dim(X) = c(3,8);
> rownames(X) = 1:3;
```

In this synthetic experiment, 8 microarray replicates are present with three miRNAs on each. Additionally, we need a corresponding matrix Y for mRNAs. Here we assume we have 20 mRNAs and 10 microarray replicates:

```
> # Generate random mRNA expression data with 20 mRNAs
> # and 10 replicates
> Y = rnorm(200);
> dim(Y) = c(20,10);
> rownames(Y) = 1:20;
```

Now we need to define what we want to test for. We shall concentrate on two-group testing, i. e. the search for miRNAs differentially expressed between two groups 1 and 2. For other designs see Section~5. Let's say both groups are of equal sample size in miRNA and mRNA microarrays:

```
> # Let's assume that we want to compare 2 miRNA groups, each of 4 replicates:
> group.miRNA = factor(c(1,1,1,1,2,2,2,2));
> # ... and that the corresponding mRNA experiments had 5 replicates in each group
> group.mRNA = factor(c(1,1,1,1,1,2,2,2,2,2));
```

Next, we need the allocation information. In most databases it is provided as a data.frame A, where the first column contains mRNAs and the second miRNAs. Each row of A indicates which mRNA is targeted by which miRNA. Let's say that miRNA 1 has nine target genes and miRNA 2 the remaining ones. The gene set of miRNA 3 will be empty.

```
> ####################
> ### Perform Test ###
> ####################
> library(miRtest)
> #Let miRNA 1 attack mRNAs 1 to 9 and miRNA 2 attack mRNAs 10 to 17.
> # mRNAs 18 to 20 are not attacked. miRNA 3 has no gene set.
> miR = c(rep(1,9),c(rep(2,8)));
> mRNAs = 1:17;
> A = data.frame(mRNAs,miR); # Note that the miRNAs MUST be in the second column!
> A

   mRNAs miR
1      1   1
2      2   1
3      3   1
4      4   1
5      5   1
6      6   1
7      7   1
8      8   1
9      9   1
10    10   2
11    11   2
12    12   2
13    13   2
14    14   2
15    15   2
16    16   2
17    17   2
```

Finally, the function 'miR.test' is called which does the testing.

```
> set.seed(1)
> P = miR.test(X,Y,A,group.miRNA,group.mRNA)

[1] "Assuming that group names of miRNA samples are the same as of mRNA samples!"

> P

    miRtest
1 0.7208223
2 0.3157580
3       NA

>
>
```

Note that for the empty gene set 'NA' was returned.

3 Choice of Gene Set Tests

The 'gene.set.test' argument in miR.test takes a vector of strings. These are the gene set tests that shall be applied. The default is the 'romer' test as it is competitive and compensates for inter-gene correlations. The different gene set tests available are:

~

Test	Name in miR.test
Self-contained	
'globaltest' (Goeman et~al., 2004)	"globaltest"
'GlobalAncova'	"GA"
(Mansmann and Meister, 2005; Hummel et~al., 2008)	
'RepeatedHighDim' (Brunner, 2009)	"RHD"
Competitive	
Kolm. Smirnov test on gene ranks	"KS"
Wilcoxon test on gene ranks	"W"
Fisher's exact test on gene ranks with 5 % FDR threshold	"Fisher"
'ROAST' (Wu et~al., 2010)	"roast"
'romer' (Majewski et~al., 2010)	"romer"

3.1 Faster Algorithm

The specification of other gene set tests in miR.test is therefore rather simple. To obtain faster results than with the default 'romer' rotation test, the Wilcoxon two-sample test based on gene ranks is recommended:

```
> ######################################################
> ### For a faster result: use other gene set tests ###
> ######################################################
> # Wilcoxon two-sample test is recommended for fast results
> # Note that results may vary depending on how much genes correlate
>
> P.gsWilcox = miR.test(X,Y,A,group.miRNA,group.mRNA,gene.set.tests="W")

[1] "Assuming that group names of miRNA samples are the same as of mRNA samples!"

> P.gsWilcox

    miRtest
1 0.7099849
2 0.3076669
3        NA
```

4   Other Input Formats of Allocation Data

To make 'miR.test' run faster one can specify an allocation matrix instead of the allocation data.frame. Its columns stand for the miRNAs and its rows for the mRNAs. If a mRNA is a target of a miRNA, the corresponding entry is 1, else it is 0. An easy way to generate allocation matrices is the 'generate.A' function:

```
> ##############################################
> ### We can use an allocation matrix as A ###
> ##############################################
> A = generate.A(A,X=X,Y=Y,verbose=FALSE);
> A

   1 2 3
1  1 0 0
2  1 0 0
3  1 0 0
4  1 0 0
5  1 0 0
6  1 0 0
7  1 0 0
8  1 0 0
9  1 0 0
10 0 1 0
11 0 1 0
12 0 1 0
13 0 1 0
14 0 1 0
15 0 1 0
16 0 1 0
17 0 1 0
18 0 0 0
19 0 0 0
20 0 0 0
```

To use the allocation matrix, we need to set 'allocation.matrix=TRUE' in 'miR.test':

```
> # Now we can test as before
> set.seed(1)
> P = miR.test(X,Y,A,group.miRNA,group.mRNA,allocation.matrix=TRUE)

[1] "Assuming that group names of miRNA samples are the same as of mRNA samples!"

> P
```

```
    miRtest
1 0.7208223
2 0.3157580
3        NA

>
>
```

5 Other Designs than Two-Group Design

Primarily, 'miRtest' has been designed for two-group comparisons. However, 'miRtest' accepts design matrices as used in 'limma' (Smyth, 2004). The only limitation is that 'miRtest' takes the second column from 'limma's 'eBayes' function to calculate final *p*-values. This already allows designs including

- covariables and

- continuous group/response vectors.

Other designs will be implemented in future versions. Regard the following example which shows how to use 'miRtest' on such designs. First we create the design matrices

```
> ####################
> ### Other Designs ###
> ####################
>
> # Some more complicated designs are implemented, check the vignette "miRtest" fo
> group.miRNA = 1:8
> group.mRNA = 1:10
> covariable.miRNA = factor(c(1,2,3,4,1,2,3,4))    ### A covariable in miRNAs.
> covariable.mRNA = factor(c(1,2,3,4,5,1,2,3,4,5)) ### A covariable in mRNAs.
> library(limma)
> design.miRNA = model.matrix(~group.miRNA + covariable.miRNA)
> design.mRNA =  model.matrix(~group.mRNA + covariable.mRNA)
>
```

which then we use in 'miR.test'

```
> P = miR.test(X,Y,A,design.miRNA=design.miRNA,design.mRNA=design.mRNA,allocation.
> P

    miRtest
1 0.5797948
2 0.4981197
3        NA
```

Note that so far this works only with the competitive gene set tests and 'ROAST'.

6   Analysis of Data from Nielsen *et˜al.* (2009)

In Artmann *et˜al.* (2012), we analysed the neurogenesis data from Nielsen *et˜al.* (2009), where rat Early Neuronal Progenitors from embryonic day 11 and day 13 were compared in miRNA and mRNA microarrays.
We use part of this dataset as an example on how to analyse data with 'miRtest'.
First, we load the package and the miRNA and mRNA data tables:

```
> library(miRtest)
> data("X",package="miRtest")  ### this is the miRNA expression data of the ten mo
> data("Y",package="miRtest")  ### this is the mRNA expression data of the genes t
> data("A",package="miRtest")  ### allocation data from TargetScan
```

Note that A is a data.frame with mRNAs in its first and miRNAs in its second column.
This data is part of the allocation data from TargetScan 4.1.
Next, we define the groups (here we have two different ages of rats the cells were extracted from):

```
> group.mRNA = factor(c(1,1,1,1,2,2,2,2));
> group.miRNA = factor(c(1,1,1,2,2,2));
```

Finally, we perform the analysis:

```
> P = miR.test(X=X,Y=Y,A=A,group.mRNA=group.mRNA,group.miRNA=group.miRNA,
+ adjust="BH",gene.set.tests="all",verbose=TRUE)

[1] "Assuming that group names of miRNA samples are the same as of mRNA samples!"
[1] "Note: For compatibility reasons ROAST is not available in this version."
[1] " It will be added to the next version of miRtest."
[1] " To use miRtest with ROAST refer to older versions."
[1] "Gene Set 1 of 10 Gene Sets at 2014-03-25 12:17:14"
[1] "Gene Set 2 of 10 Gene Sets at 2014-03-25 12:17:14"
[1] "Gene Set 3 of 10 Gene Sets at 2014-03-25 12:17:17"
[1] "Gene Set 4 of 10 Gene Sets at 2014-03-25 12:17:19"
[1] "Gene Set 5 of 10 Gene Sets at 2014-03-25 12:17:20"
[1] "Gene Set 6 of 10 Gene Sets at 2014-03-25 12:17:21"
[1] "Gene Set 7 of 10 Gene Sets at 2014-03-25 12:17:21"
[1] "Gene Set 8 of 10 Gene Sets at 2014-03-25 12:17:24"
[1] "Gene Set 9 of 10 Gene Sets at 2014-03-25 12:17:28"
[1] "Gene Set 10 of 10 Gene Sets at 2014-03-25 12:17:29"
[1] "Starting romer procedure..."
[1] "Finished romer procedure..."
```

P now contains the Benjamini-Hochberg-adjusted p-values for the miRNAs we investigated. Note that for competitive testing the entire mRNA-data would be necessary, while only a subset of mRNAs was used here.

References

Artmann, S., Jung, K., Bleckmann, A., and Beißbarth, T. Detection of Simultaneous Group Effects in microRNA Expression and related functional Gene Sets. *PLoS ONE 7(6):e38365.* http://www.ncbi.nlm.nih.gov/pubmed/22723856/

Brunner, E. (2009). Repeated measures under non-sphericity. *Proceedings of the National Academy of Sciences of the United States of America*, pages 605–609.

Goeman, J.~J., van~de Geer, S.~A., de~Kort, F., and van Houwelingen, H.~C. (2004). A global test for groups of genes: testing association with a clinical outcome. *Bioinformatics*, **20**(1), 93 –99.

Hummel, M., Meister, R., and Mansmann, U. (2008). GlobalANCOVA: exploration and assessment of gene group effects. *Bioinformatics*, **24**(1), 78 –85.

Jung, K., Becker, B., Brunner, E., and Beißbarth, T. (2011). Comparison of Global Tests for Functional Gene Sets in Two-Group Designs and Selection of Potentially Effect-causing Genes. *Bioinformatics*, **27**, 1377–1383.

Majewski, I.~J., Ritchie, M.~E., Phipson, B., Corbin, J., Pakusch, M., Ebert, A., Busslinger, M., Koseki, H., Hu, Y., Smyth, G.~K., Alexander, W.~S., Hilton, D.~J., and Blewitt, M.~E. (2010). Opposing roles of polycomb repressive complexes in hematopoietic stem and progenitor cells. *Blood*, **116**(5), 731–739.

Mansmann, U. and Meister, R. (2005). Testing differential gene expression in functional groups. goeman's global test versus an ANCOVA approach. *Methods of Information in Medicine*, **44**(3).

Joseph A Nielsen and Pierre Lau and Dragan Maric and Jeffery L Barker and Lynn D Hudson Integrating microRNA and mRNA expression profiles of neuronal progenitors to identify regulatory networks underlying the onset of cortical neurogenesis. *BMC Neuroscience*, **2009**

Smyth, G.~K. (2004). Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, **3**(1).

Wu, D., Lim, E., Vaillant, F., Asselin-Labat, M., Visvader, J.~E., and Smyth, G.~K. (2010). ROAST: rotation gene set tests for complex microarray experiments. *Bioinformatics (Oxford, England)*, **26**(17).