How to use MiRaGE Package

Y-h. Taguchi

October 1, 2024

Contents

1 Introduction **1**

2 Background **2**

3 Quick start **4**

4 Data Structure **4**
4.1 Input: target gene expression 4
4.2 Output: P-values 6

5 Example **6**
5.1 Example1: non-differentiated vs differentiated ES cell 6
5.2 Example 2: Universal Human Reference RNA vs brain 7

6 Rapid use & Off line use **8**
6.1 Suppressing downloading 8
6.2 Save & load tables 8
6.3 miRNATarget package 9
6.4 Generation of tables from scratch 9

7 Multiple comparison correction **10**

1 Introduction

This document describes briefly how to use `MiRaGE` package in order to infer the target gene regulation by miRNA, based upon target gene expression.

MiRaGE is based upon the algorithm proposed in [1, 2, 3]. Basically, its function is the same as the MiRaGE Server. In order to infer the target gene regulation by miRNAs, we made use of target gene (mRNA) expression. Suppose x_{gs} is the expression of the gth gene in the sth sample. Then P-value to measure the amount of target gene

regulation by the mth miRNAs is computed by several statistical test. More detailed and comprehensive explanations can be found in [4].

2 Background

miRNA is short non-coding RNA (ncRNA) which is believed to degrade target genes. Target genes are believed to be decided by seed match between 7-mer at 5' untranslated region (UTR) of miRNA and 3' UTR of target mRNAs. However because of huge number of miRNAs (c.a. 1000) and the huge number of target genes (c.a. hundreds) of each miRNA, it is not easy to experimentally decide which miRNA regulates target genes.

MiRaGE infers target gene regulation from target gene expression and computationally predicted target gene table. It gives the rejection probability to reject null hypothesis that target genes of a specific miRNA are equally regulated as other genes.

When t-test is employed, P-value is

$$P(S_m^{ss'} > S_m'^{ss'})$$

or

$$P(S_m^{ss'} < S_m'^{ss'})$$

where P is the rejection probability of null hypothesis $S_m = S'_m$ when the alternative hypothesis is either $S_m > S'_m$ or $S_m < S'm$. S_m and S'_m are the test variable to measure the target gene regulation by the mth miRNA.

When P-values are computed via t-test, S_m is the mean gene expression logarithmic ratio of the mth miRNA's target genes, i.e.,

$$S_m^{ss'} = \frac{1}{N(G_m)} \sum_{g \in G_m} \log \frac{x_{gs}}{x_{gs'}}$$

where G_m is the set of the mth miRNA's target genes and $N(G_m)$ is the total number of genes in G_m. S'_m is the mean expression logarithmic ratio of genes not targeted by the mth miRNA but any other miRNAs and is defined as

$$S_m'^{ss'} = \frac{1}{N(G'_m)} \sum_{g \in G'_m} \log \frac{x_{gs}}{x_{gs'}}$$

where G'_m is the set of the mth miRNA's target genes and $N(G'_m)$ is the total number of genes in G'_m.

On the other hands, if P-values are computed by Wilcoxon rank sum test, they are

$$P\left(U_m^{ss'} > \frac{N(G_m)N(G'_m)}{2}\right)$$

or

$$P\left(U_m^{ss'} < \frac{N(G_m)N(G'_m)}{2}\right)$$

which are the rejection probabilities of null hypothesis $U = \frac{N(G_m)N(G'_m)}{2}$ when the alternative hypothesis is either $U > \frac{N(G_m)N(G'_m)}{2}$ or $U < \frac{N(G_m)N(G'_m)}{2}$. Here $U^{ss'}$ is the test variable,

$$U_m^{ss'} = R_m^{ss'} - \frac{N(G_m)(N(G_m)+1)}{2}$$

Here

$$R_m^{ss'} = \sum_{g \in G_m} R^{ss'}\left(\log \frac{x_{gs}}{x_{gs'}}\right)$$

where $R^{ss'}(\ldots)$ is the rank order of the mth genes logarithmic ratio among all of considered genes.

Alternatively, Kolmogorov-Smirnov test can be employed. In this case, test variable is

$$D_m^{ss'} = \sup_g \left(F_m(x_{gs}) - F'_m(x_{gs})\right)$$

or

$$D'^{ss'}_m = \sup_g \left(F'_m(x_{gs}) - F_m(x_{gs})\right)$$

where

$$F_m^{ss'}(x_{gs}) = \frac{1}{N(G_m)} \sum_{g' \in G_m} \Theta\left(\log \frac{x_{gs}}{x_{gs'}} - \log \frac{x_{g's}}{x_{g's'}}\right)$$

and

$$F'^{ss'}_m(x_{gs}) = \frac{1}{N(G'_m)} \sum_{g' \in G'_m} \Theta\left(\log \frac{x_{gs}}{x_{gs'}} - \log \frac{x_{g's}}{x_{g's'}}\right)$$

where $\Theta(x)$ is the step function,

$$\Theta(x) = \begin{cases} 1 & x > 0 \\ 0 & x < 0 \end{cases}$$

Then P-value is computed via $P(D_m^{ss'} > 0)$ or $P(D'^{ss'}_m > 0)$ under the null hypothesis $D_m^{ss'} = 0$ or $D'^{ss'}_m = 0$.

Since the target genes table is generated by the simple seed match, MiRaGE does not need any other external programs to obtain target gene table. Another advantage is the exclusion of mRNA targeted by no miRNAs. This enables us more accurate prediction of target gene regulation by miRNAs.

3 Quick start

```
> library(MiRaGE)
> data(gene_exp)
> library(Biobase)
> result <- MiRaGE(gene_exp,species="HS")
```

Then `result$P0` and `result$P1` include P-values for upregulation and downregulation of target genes by miRNAs, respectively. The definition "up" or "down" depends upon the order of columns of expression data in `gene_exp` (see below).

Caution I strongly recommend user to use `location`="web" option, since it will be most frequently updated. Default setting requires experimental package *miRNATarget* (see Sec. 6.3). Data set on the web can be stored for the later usage, too (see below).

```
> result <- MiRaGE(gene_exp,location="web",species="HS")
```

4 Data Structure

4.1 Input: target gene expression

In order to execute analysis, you need ExpressionSet objects which stores target gene expression in it. In order to see this, it is easier to see sample data `gene_exp` as follows.

```
> data(gene_exp)
> gene_exp
```

```
ExpressionSet (storageMode: lockedEnvironment)
assayData: 45015 features, 4 samples
  element names: exprs
protocolData: none
phenoData
  sampleNames: neg.1 neg.2 day1.1 day1.2
  varLabels: sample_name
  varMetadata: labelDescription
featureData
  featureNames: 1 2 ... 45015 (45015 total)
  fvarLabels: gene_id
  fvarMetadata: labelDescription
experimentData: use 'experimentData(object)'
Annotation:
```

The above example displays the ExpressionSet object, `gene_exp`. As you can see, each row corresponds to each gene and each column corresponds to each sample (experiment).

gene_id in featureData must includes gene id. sample_name in phenoData must include sample names. Since MiRaGE package tries to compare two distinct states, you need at least a set of gene expression corresponding to each of them. In gene_exp data, we have two biological replicates of negative control and the results one day after treatment. Thus, the 1st and 2nd columns of expression data are named as neg.1 and neg.2, respectively (This means "negative control 1" and "negative control 2", respectively). The 3rd and 4th columns of expression data corresponds to the two biological replicates one day after the treatment. Thus, they are names as day1.1 and day1.2, respectively.

These column names which express distinct samples keep some flexibilities but must have the form *group.n*, where *group* corresponds to either of sample groups and n must be integer starting from 1. This means, it you have N biological replicates for the first group (typically it includes un-treated or negative control samples) names as *groupA* and M for the second group (typically it includes treated samples) names as *groupB*, data structure of ExpressioSet which stores target gene expression is,

```
sample_name:  groupA.1 groupA.1 ... groupA.N groupB.1 groupB.2 ... groupB.M
gene_id : gene1,gene2,gene3....
```

gene_id is much easier. It can includes any of gene id which can be treated by MiRaGE. They can be a mixture of the different types of gene ids. In this case, only gene expression having gene id specified when MiRaGE is called are treated as target genes.

The easiest way to generate ExpressionSet which include target gene expression may be importing files including gene expression using standard R function,

```
> x_gene <- read.csv(system.file("extdata/x_all_7a.csv",package="MiRaGE"),sep="\t")
> x_gene[101:103,]

            gene   neg.1   neg.2 day1.1  day1.2
101     BG167701    7.09    7.63    7.3    8.42
102 NM_001029863  995.00 2090.00  669.0 1370.00
103 NM_001014445 2540.00 6570.00 2070.0 3270.00
```

As can be seen, the first column includes gene id, which is "refseq" here, and the second to the fifth columns include gene expression. Data frame x_gene can be transformed to ExpressionSet objects gene_exp as

```
> gene_exp <- new("ExpressionSet",expr=data.matrix(x_gene[,-1]))
> fData(gene_exp)[["gene_id"]] <- x_gene[,1]
> pData(gene_exp)[["sample_name"]] <- colnames(x_gene)[-1]
```

For users' convenience, we have places a file x_all_7a.csv under csv directory. Please refer to this file for the preparation of files including target gene expression.

4.2 Output: P-values

As mentioned in the above, output of `MiRaGE` is a list which includes two dataframes named as `P0` and `P1`, respectively. `P0` includes the rejection probabilities that the target gene expression in the first sample group is less than that in the second group. This means, smaller P-values indicate the target gene expression in the first sample group is more likely less than the second sample groups. Inversely, `P1` includes the rejection probabilities that the target gene expression in the second sample is less than that in the first group. Thus, smaller P-values indicate target gene expression in the second group is more likely less than the first groups.

```
> result$P1[1:3,]
```

```
         Refseq mixed
1 hsa-let-7a-5p     0
2 hsa-let-7b-5p     0
3 hsa-let-7c-5p     0
```

In the above, we have shown the first three lines in the dataframe `result$P1`. Since these are small, target genes of these three miRNAs is possibly expressive in the second group. In the first column of `result$P0` and `result$P1`, names of considered miRNAs are listed. The number of miRNAs considered varys dependent upon the argument `conv` of `MiRaGE`. The second column includes P-values attributed to each miRNA. Dependent upon argument `method`, the number of columns which store P-values may change (see below).

5 Example

5.1 Example1: non-differentiated vs differentiated ES cell

In this section, we demonstrate how to infer target gene regulation via `MiRaGE`. First we import data set from experimental package *humanStemCell*

```
> require(humanStemCell)
> data(fhesc)
```

In this data set, human stem cells were assayed using Affymetrix 133plus 2 arrays. There were six arrays, three were biological replicates for undifferentiated cells, the other three were biological replicates for differentiated cells. In order to analyze this set, we modify ExpressionSet `fhesc` as

```
> pData(fhesc)[["sample_name"]] <- c("neg.1","neg.2","neg.3",
+ "pos.1","pos.2","pos.3")
> fData(fhesc)[["gene_id"]] <-rownames(exprs(fhesc))
```

Then, first three are designated as non-differentiated ES cell and the later three are differentiated ES cell. Obtaining P-values is easy,

```
> require(MiRaGE)
> result <- MiRaGE(fhesc,species="HS",ID="affy_hg_u133a_2")
```

Using the results, we can list miRNAs whose target genes are upregulated in the later (i.e., differentiated ES cell) group with P-values.

```
> result$P0[order(result$P0[,2])[1:5],]
```

```
           Refseq mixed
7  hsa-miR-15a-5p     0
8   hsa-miR-16-5p     0
9   hsa-miR-17-5p     0
13 hsa-miR-20a-5p     0
28  hsa-miR-93-5p     0
```

Since miRNAs are believed to suppress target genes, these miRNAs are supposed to be upregulated in the former (i.e., non-differentiated ES cell) group.

5.2 Example 2: Universal Human Reference RNA vs brain

In this section, we demonstrate how to infer target gene regulation via `MiRaGE` using another example.

First we import data set from experimental package *beadarrayExampleData*

```
> require(beadarrayExampleData)
> data(exampleBLData)
> data(exampleSummaryData)
```

The data in this package are a subset of the MAQC bead-level data available in the beadarrayUseCases package. Bead-level refers to the availability of intensity and location information for each bead on each BeadArray in an experiment. In this dataset, BeadArrays were hybridized with either Universal Human Reference RNA (UHRR, Stratagene) or Brain Reference RNA (Ambion) as used inthe MAQC project. This object is a representation of the bead-level data for 2 arrays and was created by the beadarray package.

Since this is two color array, and the number of columns of expression must be the number of columns of expression data MUST be the length of `sample_name`, we omit later half of samples and employ only the first twelve samples, for simplicity.

```
> vv <- exampleSummaryData[,1:12]
>  fData(vv)[["gene_id"]] <- fData(exampleSummaryData)[["IlluminaID"]]
>  pData(vv)[["sample_name"]] <- c("neg.1","neg.2","neg.3","neg.4",
+ "neg.5","neg.6","brain.1","brain.2","brain.3","brain.4","brain.5","brain.6")
> result <- MiRaGE(vv,species="HS",ID="illumina_humanwg_6_v3")
```

Then we can list miRNAs whose target genes are upregulated in negative control, i.e., miRNAs which are expected to be upregulated in brain as follows.

```
> result$P1[order(result$P1[,2])[1:5],]
```

```
            Refseq        mixed
82  hsa-miR-124-3p 0.0001196863
157 hsa-miR-506-3p 0.0001196863
143   hsa-miR-451a 0.0128220812
99  hsa-miR-191-5p 0.1640405989
102 hsa-miR-126-3p 0.4025617836
```

6 Rapid use & Off line use

Although the default value of `location` is "local", when `location`="web", `MiRaGE` every time tries to access `MiRaGE Server`[1] to download target gene tables, gene id conversion table, and miRNA conservation table. It is a time consuming process. Especially, since the target gene table is huge, it may take a few minutes. It may not be often to use `MiRaGE` iteratively many times, we offer the method to avoid "every time download".

6.1 Suppressing downloading

In `MiRaGE`, we offer the option to suppress downloading. If you repeatedly use `MiRaGE` with keeping either `species`, `ID`, or `conv` unchanged, you can suppress time consuming download process by specifying either `species_force`, `ID_force`, or `conv_force` as `FALSE` (Defaults for these are `TRUE`).

Caution Do not omit the arguments either `species`, `ID`, or `conv` if they differ from defaults, even if they are not modified during iterative usage and either `species_force`, `ID_force`, or `conv_force` is `FALSE`. They are used for other purposes than specifying what should be downloaded.

6.2 Save & load tables

More advanced and convenient way is to save the objects storing target gene tables, gene id conversion table, and miRNA conservation table. The names of objects are,

- `TBL2` : Target gene tables
- `id_conv` : Gene id conversion table
- `conv_id` : MiRNA conservation table

[1] http://www.granular.com/DATA2/

Thus, for example TBL2 is saved as

```
> save(file="TBL2",TBL2)
```

you can use it later by loading as

```
> load("TBL2")
```

Then you can execute MiRaGE with specifying species_force=F as

```
> result <- MiRaGE(...,species_force=F)
```

Now, you can skip time consuming download processes for the target gene table. Similar procedures are possible for id_conv and conv_id, too. Execute MiRaGE, save downloaded tables, and use the tables later by loading them when these arguments take same values.

6.3 miRNATarget package

One can also install experimental package *miRNATarget* instead of the usage of web. Once you install experimental package *miRNATarget*, you will never be required to access to internet.

```
> library(MiRaGE)
> data(gene_exp)
> library(Biobase)
> result <- MiRaGE(gene_exp,species="HS")
```

6.4 Generation of tables from scratch

I have also prepared functions which generate TBL2, id_conv and conv_id from scratch. Usually, user do not need them since prepared tables can be obtained from the web or as experimental package as mentioned above.

TBL2_HS can be saved in the current directly by executing

```
> TBL2_HS_gen()
```

and TBL2_MM can saved in the current directly by executing

```
> TBL2_MM_gen()
```

id_conv for mouse can saved in the current directly by executing

```
> id_conv_gen(SP="MM")
```

and id_conv for human can saved in the current directly by executing

```
> id_conv_gen(SP="HS")
```

`HS_conv_id` can saved in the current directly by executing

```
> HS_conv_id()
```

and `MM_conv_id` can saved in the current directly by executing

```
> MM_conv_id()
```

However, basically, execution of some of them are very time consuming. It is highly discouraged to build tables from scratch. It is much better to use prepared tables.

7 Multiple comparison correction

Obtained P-values are definitely underestimated, i.e., even if $P < 0.05$, this does not mean the rejection probability is less than 0.05. If one prefers to use adjusted P-values, we recommend to use `p.adjust` with parameter of `BH`, as

```
> p.adjust(result$P1[,2],method="BH")
```

```
  [1] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
  [6] 0.000000e+00 7.420103e-01 7.420103e-01 9.953000e-01 9.953000e-01
 [11] 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01
 [16] 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01
 [21] 9.953000e-01 9.953000e-01 9.953000e-01 3.114959e-01 9.953000e-01
 [26] 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01 0.000000e+00
 [31] 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01
 [36] 9.953000e-01 9.953000e-01 9.953000e-01 4.124657e-09 9.953000e-01
 [41] 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01
 [46] 1.875348e-01 7.037504e-01 9.364901e-01 9.364901e-01 9.953000e-01
 [51] 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01 9.364901e-01
 [56] 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01
 [61] 9.953000e-01 9.953000e-01 9.953000e-01 9.364901e-01 9.953000e-01
 [66] 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01
 [71] 9.953000e-01 9.953000e-01 9.953000e-01 0.000000e+00 0.000000e+00
 [76] 9.953000e-01 7.420103e-01 9.953000e-01 9.953000e-01 9.953000e-01
 [81] 9.953000e-01 7.591141e-01 9.953000e-01 9.953000e-01 9.953000e-01
 [86] 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01
 [91] 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01
 [96] 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01
[101] 9.953000e-01 9.953000e-01 9.953000e-01 6.556531e-01 8.396576e-01
[106] 9.953000e-01 9.953000e-01 5.944977e-01 7.420103e-01 9.953000e-01
[111] 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01
[116] 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01
[121] 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01
```

```
[126] 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01
[131] 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01
[136] 4.124657e-09 7.420103e-01 9.953000e-01 9.953000e-01 9.953000e-01
[141] 9.953000e-01 9.953000e-01 9.953000e-01 4.766116e-02 9.953000e-01
[146] 9.953000e-01 7.420103e-01 9.953000e-01 9.953000e-01 9.953000e-01
[151] 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01
[156] 9.953000e-01 7.591141e-01 7.420103e-01 5.857503e-01 9.953000e-01
[161] 5.857503e-01 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01
[166] 9.953000e-01 9.953000e-01 9.364901e-01 9.953000e-01 9.953000e-01
[171] 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01 9.953000e-01
[176] 9.953000e-01 9.953000e-01 9.364901e-01 9.953000e-01 9.953000e-01
[181] 0.000000e+00 9.953000e-01 0.000000e+00 9.953000e-01 9.953000e-01
[186] 9.953000e-01
```

Then we can see which P-values are really significant, e.g., less than 0.05. In addition to this, it will allow us to evaluate which miRNAs really regulate target genes, e.g.,

```
> result$P1[,1][p.adjust(result$P1[,2],method="BH")<0.05]

 [1] "hsa-let-7a-5p"   "hsa-let-7b-5p"   "hsa-let-7c-5p"   "hsa-let-7d-5p"
 [5] "hsa-let-7e-5p"   "hsa-let-7f-5p"   "hsa-miR-98-5p"   "hsa-miR-196a-5p"
 [9] "hsa-let-7g-5p"   "hsa-let-7i-5p"   "hsa-miR-196b-5p" "hsa-miR-490-3p"
[13] "hsa-miR-4458"    "hsa-miR-4500"
```

`x_gene` is the transfection expreiments of let-7a, it is reasnable that only a few miRNAs including let-7a have significant P-values.

Appendix:Arguments

Functionality of `MiRaGE` changes dependent upon the values of arguments. In this section, we will try to explain how the functionality of `MiRaGE` changes.

species

This specifies target species. Considered miRNAs are based upon miRBase[2]. Rel. 20. At the moment, supported species are human ("HS") and mouse ("MM"). `MiRaGE` downloads corresponding target gene table (named `TBL2`) from MiRaGE Server. Default is "MM".

[2]http://www.mirbase.org

ID

This specifies gene ID. Default is “refseq". If ID is not “refseq", `MiRaGE` downloads corresponding gene id conversion table (called `ID`) from RefSeq to specified gene ID from MiRaGE Server. Supported gene IDs are,

	common	
	`ID`	description
1	ensembl_gene_id	Ensembl Gene ID
2	ensembl_transcript_id	Ensembl Transcript ID
3	ensembl_peptide_id	Ensembl Protein ID
4	ensembl_exon_id	Ensembl Exon ID
5	ccds	CCDS ID
6	embl	EMBL (Genbank) ID
7	entrezgene	EntrezGene ID
8	merops	MEROPS ID
9	pdb	PDB ID
10	protein_id	Protein (Genbank) ID
11	refseq_peptide	RefSeq Protein ID [e.g. NP_001005353]
12	rfam	Rfam ID
13	rfam_transcript_name	Rfam transcript name
14	ucsc	UCSC ID
15	unigene	Unigene ID
16	uniprot_sptrembl	UniProt/TrEMBL Accession
17	uniprot_swissprot	UniProt/SwissProt ID
18	uniprot_swissprot_accession	UniProt/SwissProt Accession
19	uniprot_genename	UniProt Gene Name
20	uniprot_genename_transcript_name	Uniprot Genename Transcript Name
21	wikigene_name	WikiGene Name
22	wikigene_id	WikiGene ID
23	efg_agilent_sureprint_g3_ge_8x60k	Agilent SurePrint G3 GE 8x60k probe
24	efg_agilent_wholegenome_4x44k_v1	Agilent WholeGenome 4x44k v1 probe
25	efg_agilent_wholegenome_4x44k_v2	Agilent WholeGenome 4x44k v2 probe
26	codelink	Codelink probe
27	phalanx_onearray	Phalanx OneArray probe
28	smart	SMART ID
29	pfam	PFAM ID
30	tigrfam	TIGRFam ID
31	interpro	Interpro ID

	human	
	ID	description
1	hgnc_id	HGNC ID(s)
2	hgnc_symbol	HGNC symbol
3	hgnc_transcript_name	HGNC transcript name
4	affy_hc_g110	Affy HC G110 probeset
5	affy_hg_focus	Affy HG FOCUS probeset
6	affy_hg_u133_plus_2	Affy HG U133-PLUS-2 probeset
7	affy_hg_u133a_2	Affy HG U133A_2 probeset
8	affy_hg_u133a	Affy HG U133A probeset
9	affy_hg_u133b	Affy HG U133B probeset
10	affy_hg_u95av2	Affy HG U95AV2 probeset
11	affy_hg_u95b	Affy HG U95B probeset
12	affy_hg_u95c	Affy HG U95C probeset
13	affy_hg_u95d	Affy HG U95D probeset
14	affy_hg_u95e	Affy HG U95E probeset
15	affy_hg_u95a	Affy HG U95A probeset
16	affy_hugenefl	Affy HuGene FL probeset
17	affy_huex_1_0_st_v2	Affy HuEx 1_0 st v2 probeset
18	affy_hugene_1_0_st_v1	Affy HuGene 1_0 st v1 probeset
19	affy_u133_x3p	Affy U133 X3P probeset
20	agilent_cgh_44b	Agilent CGH 44b probe
21	illumina_humanwg_6_v1	Illumina HumanWG 6 v1 probe
22	illumina_humanwg_6_v2	Illumina HumanWG 6 v2 probe
23	illumina_humanwg_6_v3	Illumina HumanWG 6 v3 probe
24	illumina_humanht_12	Illumina Human HT 12 probe

mouse		
ID	description	
1	fantom	Fantom ID
2	ipi	IPI ID
3	mgi_id	MGI ID
4	mgi_symbol	MGI symbol
5	mgi_transcript_name	MGI transcript name
6	affy_mg_u74a	Affy mg u74a probeset
7	affy_mg_u74av2	Affy mg u74av2 probeset
8	affy_mg_u74b	Affy mg u74b probeset
9	affy_mg_u74bv2	Affy mg u74bv2 probeset
10	affy_mg_u74c	Affy mg u74c probeset
11	affy_mg_u74cv2	Affy mg u74cv2 probeset
12	affy_moe430a	Affy moe430a probeset
13	affy_moe430b	Affy moe430b probeset
14	affy_moex_1_0_st_v1	Affy MoEx probeset
15	affy_mogene_1_0_st_v1	Affy MoGene probeset
16	affy_mouse430_2	Affy mouse430 2 probeset
17	affy_mouse430a_2	Affy mouse430a 2 probeset
18	affy_mu11ksuba	Affy mu11ksuba probeset
19	affy_mu11ksubb	Affy mu11ksubb probeset
20	illumina_mousewg_6_v1	Illumina MouseWG 6 v1 probe
21	illumina_mousewg_6_v2	Illumina MouseWG 6 v2 probe

Requirements for supporting any other gene IDs are welcomed.

method

This specifies how to treat replicates. if `method` is “mean", then averaged gene expression is attributed to each gene. If it is “mixed", they are used for statistical test as it is. This means, the number of target genes attributed to each miRNAs is as many as the number of replicates. If "one_by_one" is specified, all of combinations between the two groups, i.e.,

$$\text{groupA.1} \times \text{groupB.1}, \text{groupA.1} \times \text{groupB.2}, \ldots, \text{groupA.N} \times \text{groupB.M}.$$

are condiered. Thus, in this case, both `P0` and `P1` have $1 + N \times M$ columns, the later $N \times M$ includes P-value for each of combinations. Default is “mean".

test

This specifies the statititical methods to evaluate siginificance of reglation of target genes. Supported are “ks" (Kolmogorov-Smirnov test), “t" (t-test), and "wilcox" (Wilcoxon

test). These are performed by standard R functions, `ks.test`, `t.test`, and `wilcox.test`, respectively. Default is "ks".

conv

This specifies how well considered miRNAs must be conserved. Supported are “conserved", “weak_conserv" and “all". Baed upon TargetScan 7.2 [3], they correspond to broadly conserved, conserved, and others. For more detail, plese colusult with TargetScan. Default is “conserved".

Force download or not

`species_force`, `ID_force`, and `conv_force` spefify if target gene table, gene id conversion table, and miRNA conservation table are forced to be donwloaded. Dafult is `T`. If some of them have already been downloded and one would like to use it as it is, please specify they are `F`.

References

[1] Y-h Taguchi, Jun Yasuda, 2010, Inference of Gene Expression Regulation via microRNA Transfection, ICIC2010, Proceedings, Springer, 6215, 672-679.

[2] Y-h Taguchi, Jun Yasuda, 2012, MiRaGE: Inference of Gene Expression Regulation via MicroRNA Transfection II, ICIC2011, Proceedings, Springer, 6840,192-135

[3] M. Yoshizawa, Y-h. Taguchi, Jun Yasuda, 2011, Inference of Gene Regulation via miRNAs During ES Cell Differentiation Using MiRaGE Method, Int. J. Mol. Sci., 12[12]:9265-9276

[4] Taguchi, Y-h. (2013). Inference of Target Gene Regulation by miRNA via MiRaGE Server. Introduction to Genetics - DNA Methylation, Histone Modification and Gene Regulation. ISBN: 978-1477554-94-4. iConcept Press. Retrieved from http://www.iconceptpress.com/books/ IntroductionToGeneticsDNAMethylationHistoneModificationAndGeneRegulation/

[3] http://www.targetscan.org