

# **miRNAAtap** example use

Maciej Pajak, T. Ian Simpson

October 13, 2014

## **Contents**

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Installation</b>	<b>2</b>
<b>3</b>	<b>Workflow</b>	<b>2</b>
<b>4</b>	<b>Session Information</b>	<b>5</b>
	<b>References</b>	<b>5</b>

# 1 Introduction

`miRNAtap` package is designed to facilitate implementation of workflows requiring miRNA prediction. Aggregation of commonly used prediction algorithm outputs in a way that improves on performance of every single one of them on their own when compared against experimentally derived targets. microRNA (miRNA) is a 18-22nt long single strand that binds with RISC (RNA induced silencing complex) and targets mRNAs effectively reducing their translation rates.

Targets are aggregated from 4 most commonly cited prediction algorithms: DIANA (Maragkakis et al., 2011), Miranda (Enright et al., 2003), PicTar (Lall et al., 2006) and TargetScan (Friedman et al., 2009).

Programmatic access to sources of data is crucial when streamlining the workflow of our analysis, this way we can run similar analysis for multiple input miRNAs or any other parameters. Not only does it allow us to obtain predictions from multiple sources straight into R but also through aggregation of sources it improves the quality of predictions.

Finally, although direct predictions from all sources are only available for *Homo sapiens* and *Mus musculus*, this package includes an algorithm that allows to translate target genes to other speices (currently only *Rattus norvegicus*) using homology information where direct targets are not available.

# 2 Installation

This section briefly describes the necessary steps to get `miRNAtap` running on your system. We assume that the user has the R program (see the R project at <http://www.r-project.org>) already installed and is familiar with it. You will need to have R 3.0.0 or later to be able to install and run `miRNAtap`. The `miRNAtap` package is available from the Bioconductor repository at <http://www.bioconductor.org> To be able to install the package one needs first to install the core Bioconductor packages. If you have already installed Bioconductor packages on your system then you can skip the two lines below.

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

Once the core Bioconductor packages are installed, we can install the `miRNAtap` and accompanying database `miRNAtap.db` package by

```
> source("http://bioconductor.org/biocLite.R")
> biocLite("miRNAtap")
> biocLite("miRNAtap.db")
```

# 3 Workflow

This section explains how `miRNAtap` package can be integrated in the workflow aimed at predicting which processes can be regulated by a given microRNA.

In this example workflow we'll use `miRNA` as well as another Bioconductor package `topGO` together with Gene Ontology (GO) annotations. In case we don't have `topGO` or GO annotations on our machine we need to install them first:

```
> source("http://bioconductor.org/biocLite.R")
> biocLite("topGO")
> biocLite("org.Hs.eg.db")
```

Then, let's load the required libraries

```
> library(miRNA)
> library(topGO)
> library(org.Hs.eg.db)
```

Now we can start the analysis. First, we will obtain predicted targets for human miRNA *miR-10b*

```
> mir = 'miR-10b'
> predictions = getPredictedTargets(mir, species = 'hsa',
+                                 method = 'geom', min_src = 2)
```

Let's inspect the top of the prediction list.

```
> head(predictions)
```

	source_1	source_2	source_3	source_4	rank_product	rank_final
133923	NA	22.5	NA	1	2.371708	1
7707	1	98.5	NA	NA	4.962358	2
6095	28	9.0	16	NA	5.305410	3
627	210	20.5	1	NA	5.422545	4
27253	60	2.0	NA	NA	5.477226	5
253559	162	3.0	11	NA	5.828477	6

We are using *geometric mean* aggregation method as it proves to perform best when tested against experimental data from MirBase (Griffiths-Jones et al., 2008).

We can compare it to the top of the list of the output of *minimum* method:

```
> predictions_min = getPredictedTargets(mir, species = 'hsa',
+                                     method = 'min', min_src = 2)
> head(predictions_min)
```

	source_1	source_2	source_3	source_4	rank_product	rank_final
627	210	20.5	1	NA	1	2.5
7707	1	98.5	NA	NA	1	2.5
8013	109	1.0	86	NA	1	2.5
133923	NA	22.5	NA	1	1	2.5
7022	183	375.0	2	198	2	6.0
10152	2	367.5	NA	71	2	6.0

Where predictions for rat genes are not available we can obtain predictions for mouse genes and translate them into rat genes through homology. The operation happens automatically if we specify species as `rno` (for *Rattus norvegicus*)

```
> predictions_rat = getPredictedTargets(mir, species = 'rno',
+                                     method = 'geom', min_src = 2)
```

Now we can use the ranked results as input to GO enrichment analysis. For that we will use our initial prediction for human *miR-10b*

```
> rankedGenes = predictions[, 'rank_product']
> selection = function(x) TRUE
> # we do not want to impose a cut off, instead we are using rank information
> allGO2genes = annFUN.org(whichOnto='BP', feasibleGenes = NULL,
+                          mapping="org.Hs.eg.db", ID = "entrez")
> GOdata = new('topGOdata', ontology = 'BP', allGenes = rankedGenes,
+             annot = annFUN.GO2genes, GO2genes = allGO2genes,
+             geneSel = selection, nodeSize=10)
```

In order to make use of the rank information we will use Kolomonogorov Smirnov (K-S) test instead of Fisher exact test which is based only on counts.

```
> results.ks = runTest(GOdata, algorithm = "classic", statistic = "ks")
```

```
-- Classic Algorithm --
```

```
the algorithm is scoring 776 nontrivial nodes
```

```
parameters:
```

```
test statistic: ks
```

```
score order: increasing
```

```
> results.ks
```

```
Description:
```

```
Ontology: BP
```

```
'classic' algorithm with the 'ks' test
```

```
776 GO terms scored: 15 terms with p < 0.01
```

```
Annotation data:
```

```
  Annotated genes: 446
```

```
  Significant genes: 446
```

```
  Min. no. of genes annotated to a GO: 10
```

```
  Nontrivial nodes: 776
```

We can view the most enriched GO terms (and potentially feed them to further steps in our workflow)

```
> allRes = GenTable(GOdata, KS = results.ks, orderBy = "KS", topNodes = 20)
> allRes[,c('GO.ID', 'Term', 'KS')]
```

	GO.ID	Term	KS
1	GO:0006366	transcription from RNA polymerase II pro...	0.0012
2	GO:0044087	regulation of cellular component biogene...	0.0018
3	GO:0006974	cellular response to DNA damage stimulus	0.0018
4	GO:0006357	regulation of transcription from RNA pol...	0.0020
5	GO:0032774	RNA biosynthetic process	0.0042
6	GO:2001141	regulation of RNA biosynthetic process	0.0042
7	GO:0018130	heterocycle biosynthetic process	0.0047
8	GO:0006351	transcription, DNA-templated	0.0058
9	GO:0006355	regulation of transcription, DNA-templat...	0.0058
10	GO:0034654	nucleobase-containing compound biosynthe...	0.0064
11	GO:0019438	aromatic compound biosynthetic process	0.0065
12	GO:0044271	cellular nitrogen compound biosynthetic ...	0.0065
13	GO:0016070	RNA metabolic process	0.0074
14	GO:0006950	response to stress	0.0083
15	GO:1901362	organic cyclic compound biosynthetic pro...	0.0090
16	GO:0090304	nucleic acid metabolic process	0.0116
17	GO:0045944	positive regulation of transcription fro...	0.0119
18	GO:0051252	regulation of RNA metabolic process	0.0144
19	GO:0006281	DNA repair	0.0212
20	GO:0008406	gonad development	0.0322

For more details about GO analysis refer to `topGO` package vignette (Alexa and Rahnenfuhrer, 2010).

Finally, we can use our predictions in a similar way for pathway enrichment analysis based on KEGG (Kanehisa and Goto, 2000), for example using Bioconductor's `KEGGprofile` (Zhao, 2012).

## 4 Session Information

- R version 3.1.1 Patched (2014-09-25 r66681),  
x86\_64-apple-darwin13.1.0
- Locale: C/en\_US.UTF-8/en\_US.UTF-8/C/en\_US.UTF-8/en\_US.UTF-8
- Base packages: base, datasets, grDevices, graphics, methods, parallel, stats, stats4, utils
- Other packages: AnnotationDbi 1.28.0, Biobase 2.26.0, BiocGenerics 0.12.0, DBI 0.3.1, GO.db 3.0.0, GenomeInfoDb 1.2.0, IRanges 2.0.0, RSQLite 0.11.4, S4Vectors 0.4.0, SparseM 1.05, graph 1.44.0, miRNAAtap 1.0.0, miRNAAtap.db 0.99.6, org.Hs.eg.db 3.0.0, topGO 2.18.0
- Loaded via a namespace (and not attached): RSQLite.extfuns 0.0.1, Rcpp 0.11.3, chron 2.3-45, grid 3.1.1, gsubfn 0.6-6, lattice 0.20-29, plyr 1.8.1, proto 0.3-10, sqldf 0.4-7.1, stringr 0.6.2, tools 3.1.1

## References

- Alexa, A. and Rahnenfuhrer, J. (2010). *topGO: topGO: Enrichment analysis for Gene Ontology*. R package version 2.16.0.
- Enright, A. J., John, B., Gaul, U., Tuschl, T., Sander, C., and Marks, D. S. (2003). MicroRNA targets in *Drosophila*. *Genome biology*, 5(1):R1.
- Friedman, R. C., Farh, K. K.-H., Burge, C. B., and Bartel, D. P. (2009). Most mammalian mRNAs are conserved targets of microRNAs. *Genome research*, 19(1):92–105.
- Griffiths-Jones, S., Saini, H. K., van Dongen, S., and Enright, A. J. (2008). miRBase: tools for microRNA genomics. *Nucleic acids research*, 36(Database issue):D154–8.
- Kanehisa, M. and Goto, S. (2000). KEGG: kyoto encyclopedia of genes and genomes. *Nucleic acids research*, 28(1):27–30.
- Lall, S., Grün, D., Krek, A., Chen, K., Wang, Y.-L., Dewey, C. N., Sood, P., Colombo, T., Bray, N., Macmenamin, P., Kao, H.-L., Gunsalus, K. C., Pachter, L., Piano, F., and Rajewsky, N. (2006). A genome-wide map of conserved microRNA targets in *C. elegans*. *Current biology : CB*, 16(5):460–71.
- Maragkakis, M., Vergoulis, T., Alexiou, P., Reczko, M., Plomaritou, K., Gousis, M., Kourtis, K., Koziris, N., Dalamagas, T., and Hatzigeorgiou, A. G. (2011). DIANA-microT Web server upgrade supports Fly and Worm miRNA target prediction and bibliographic miRNA to disease association. *Nucleic acids research*, 39(Web Server issue):W145–8.
- Zhao, S. (2012). *KEGGprofile: An annotation and visualization package for multi-types and multi-groups expression data in KEGG pathway*. R package version 1.6.1.