Vignette for *RTN*: reconstruction of transcriptional networks and analysis of master regulators.

Mauro AA Castro, Xin Wang, Michael NC Fletcher,
Florian Markowetz and Kerstin B Meyer *
mauro.a.castro@gmail.com

July 15, 2017

Contents

1 Overview

The package *RTN* is designed for reconstruction and analysis of transcriptional networks (TN) using mutual information [1]. It is implemented by S4 classes in *R* [2] and extends several methods previously validated for assessing transcriptional regulatory units, or regulons (*e.g.* MRA [3], GSEA [4], synergy and shadow [5]). The package computes mutual information (MI) between annotated transcription factors (TFs) and all potential targets using gene expression data. It is tuned to deal with large gene expression datasets in order to build genome-wide transcriptional networks centered on TFs and regulons. Using a robust statistical pipeline, *RTN* allows user to set the stringency of the analysis in a stepwise process, including a boostrep routine designed to remove unstable associations. Parallel computing is available for critical steps demanding high-performance.

2 Quick start

2.1 Transcriptional network inference

- 1 - Load a sample dataset
  The `dt4rtn` dataset consists of a list with 6 objects used for demonstration purposes only. It was extracted, pre-processed and size-reduced from [6] and [7] and contains a named gene expression matrix (`gexp`), a data frame with `gexp` annotation (`gexpIDs`), a named numeric vector with differential gene expression data (`pheno`), a data frame with `pheno` annotation (`phenoIDs`), a character vector with genes differentially expressed (`hits`), and a named vector with transcriptions factors (`tfs`).

*Cancer Research UK - Cambridge Institute, Robinson Way Cambridge, CB2 0RE, UK.

```
> library(RTN)
> data(dt4rtn)
```

- 2 - Create a new TNI object and run pre-processing
  Objects of class `TNI` provide a series of methods to do transcriptional network inference from high-throughput gene expression data. In this 1st step, the generic function `tni.preprocess` is used to run several checks on the input data.

```
> #Input 1: 'gexp', a named gene expression matrix (samples on cols)
> #Input 2: 'transcriptionFactors', a named vector with TF ids (3 TFs for quick demonstration!)
> #Input 3: 'gexpIDs', an optional data frame with gene annotation (it can be used to remove duplicated genes)
> rtni <- new("TNI", gexp=dt4rtn$gexp,
+             transcriptionFactors=dt4rtn$tfs[c("PTTG1","E2F2","FOXM1","E2F3","RUNX2")]
+             )
> rtni <- tni.preprocess(rtni, gexpIDs=dt4rtn$gexpIDs)
```

- 3 - Run permutation analysis
  The `tni.permutation` function takes the pre-processed `TNI` object and returns a transcriptional network inferred by mutual information (with multiple hypothesis testing corrections).

```
> rtni <- tni.permutation(rtni)
```

- 4 - Run bootstrap analysis
  In an additional step, unstable interactions can be removed by bootstrap analysis using the `tni.bootstrap` function, which creates a consensus bootstrap network (referred here as *refnet*).

```
> rtni <- tni.bootstrap(rtni)
```

- 5 - Run DPI filter
  In the TN each target can be linked to multiple TFs and regulation can occur as a result of both direct (TF-target) and indirect interactions (TF-TF-target). The Data Processing Inequality (DPI) algorithm [8] is used to remove the weakest interaction in any triangle of two TFs and a common target gene, thus preserving the dominant TF-target pairs, resulting in the filtered transcriptional network (referred here as *tnet*). The filtered TN has less complexity and highlights the most significant interactions.

```
> rtni <- tni.dpi.filter(rtni)
```

- 6 - Get results
  All results available in the `TNI` object can be retrieved using the `tni.get` function:

```
> tni.get(rtni, what="summary")
> refnet <- tni.get(rtni, what="refnet")
> tnet <- tni.get(rtni, what="tnet")
```

- 7 - Build a graph
  The inferred transcriptional network can also be retrieved as an `igraph` [9] object using the `tni.graph` function. The graph object includes some basic network attributes pre-formatted for visualization in the R package *RedeR* [10].

```
> g <- tni.graph(rtni)
```

2.2 Transcriptional network analysis

- 1 - Create a new TNA object (and run TNI-to-TNA pre-processing)
  Objects of class `TNA` provide a series of methods to do enrichment analysis on transcriptional networks. In this 1st step, the generic function `tni2tna.preprocess` is used to convert the pre-processed `TNI` object to `TNA`, also running several checks on the input data.

```
> #Input 1: 'object', a TNI object with a pre-processed transcripional network
> #Input 2: 'phenotype', a named numeric vector, usually with log2 differential expression values
> #Input 3: 'hits', a character vector of gene ids considered as hits
> #Input 4: 'phenoIDs', an optional data frame with anottation used to aggregate genes in the phenotype
> rtna <- tni2tna.preprocess(object=rtni,
+                            phenotype=dt4rtn$pheno,
+                            hits=dt4rtn$hits,
+                            phenoIDs=dt4rtn$phenoIDs
+                            )
```

- 2 - Run MRA analysis pipeline
  The `tna.mra` function takes the `TNA` object and returns the results of the Master Regulator Analysis (RMA) [3] over a list of regulons from a transcriptional network (with multiple hypothesis testing corrections). The MRA computes the overlap between the transcriptional regulatory unities (regulons) and the input signature genes using the hypergeometric distribution (with multiple hypothesis testing corrections).
  ```
  > rtna <- tna.mra(rtna)
  ```

- 3 - Run overlap analysis pipeline
  A simple overlap among all regulons can also be tested using the `tna.overlap` function:
  ```
  > rtna <- tna.overlap(rtna)
  ```

- 4 - Run GSEA analysis pipeline
  Alternatively, the gene set enrichment analysis (GSEA) can be used to assess if a given transcriptional regulatory unit is enriched for genes that are differentially expressed among 2 classes of microarrays (*i.e.* a differentially expressed phenotype). The GSEA uses a rank-based scoring metric in order to test the association between gene sets and the ranked phenotypic difference. Here regulons are treated as gene sets, an extension of the GSEA statistics as previously described [4].
  ```
  > rtna <- tna.gsea1(rtna, stepFilter=FALSE, nPermutations=100)
  > # ps. default 'nPermutations' is 1000.
  ```

- 5 - Run two-tailed GSEA analysis pipeline
  The two-tailed GSEA tests whether positive or negative targets for a TF are enriched at each extreme of a particular response (*e.g.* differentially expressed genes). The pipeline splits the regulon into a group of activated and a group of repressed genes, based on the Pearson's correlation, and then asks how the two sets are distributed in the ranked list of genes (please refer to [11] and [12] for more details).
  ```
  > rtna <- tna.gsea2(rtna, tfs="PTTG1", nPermutations=100)
  > # ps. default 'nPermutations' is 1000.
  ```

- 6 - Get results
  All results available in the `TNA` object can be retrieved using the `tna.get` function:
  ```
  > tna.get(rtna, what="summary")
  > tna.get(rtna, what="mra")
  > tna.get(rtna, what="overlap")
  > tna.get(rtna, what="gsea1")
  > tna.get(rtna, what="gsea2")
  ```

- 7 - Plot GSEA
  To visualize the GSEA distributions, the user can apply the `tna.plot.gsea1` and `tna.plot.gsea2` functions that plot the one-tailed and two-tailed GSEA results, respectively:
  ```
  > tna.plot.gsea1(rtna, file="tna_gsea1", labPheno="abs(log2) diff. expression")
  > tna.plot.gsea2(rtna, file="tna_gsea2", labPheno="log2 diff. expression")
  ```

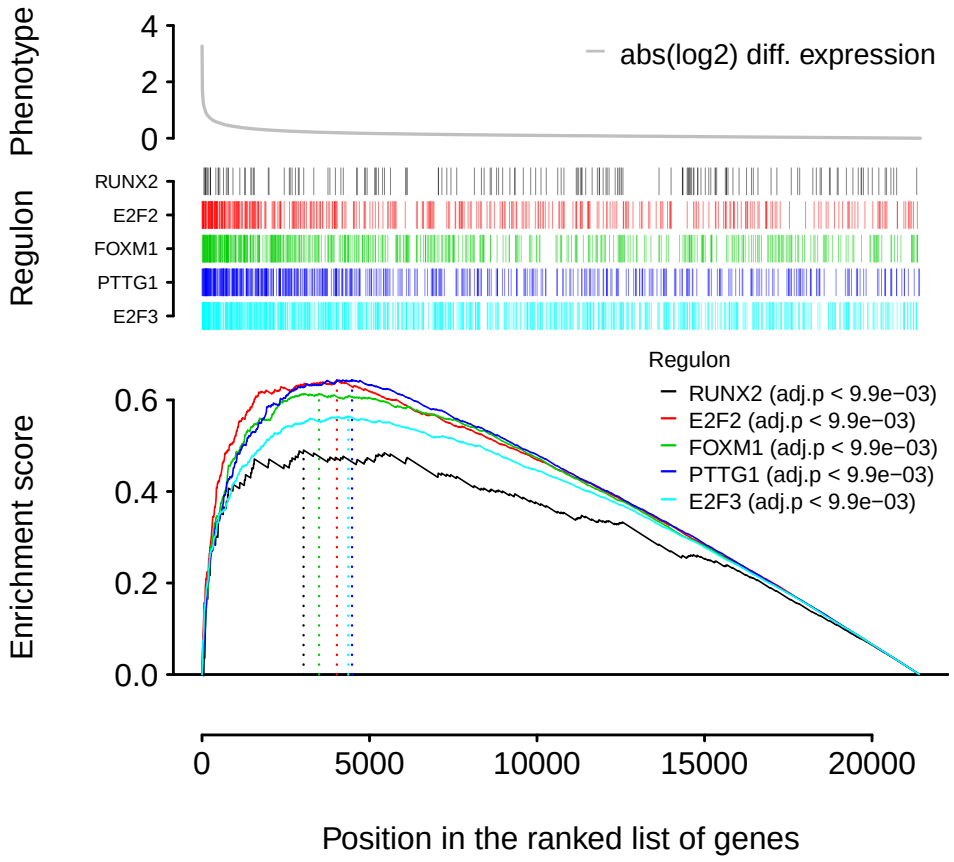

Figure 1: GSEA analysis showing genes in each regulon (as hits) ranked by their differential expression (as phenotype). This toy example illustrates the output from the *TNA* pipeline evaluated by the *tna.gsea1* method.

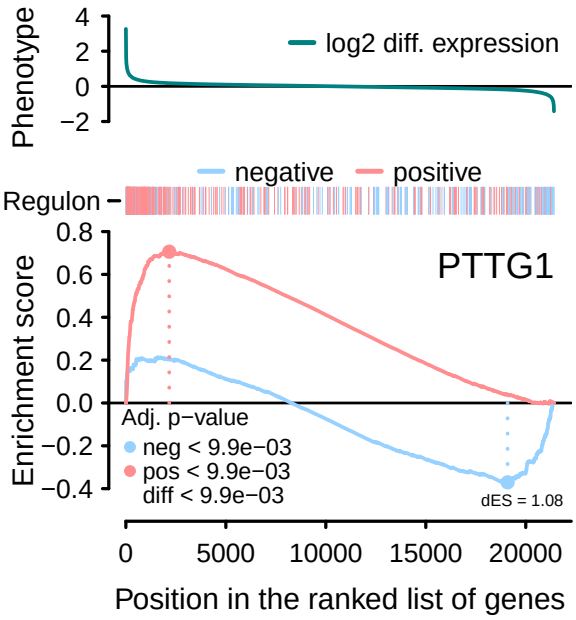

Figure 2: Two-tailed GSEA analysis showing positive or negative targets for a TF (as hits) ranked by their differential expression (as phenotype). This toy example illustrates the output from the *TNA* pipeline evaluated by the *tna.gsea2* method (for detailed interpretation of results from this method, please refer to [11] and [12]).

3 Session information

```
R version 3.4.1 (2017-06-30)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows Server 2012 R2 x64 (build 9600)

Matrix products: default

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] RTN_1.14.1

loaded via a namespace (and not attached):
 [1] Rcpp_0.12.11         compiler_3.4.1       nloptr_1.0.4         tools_3.4.1
 [5] minet_3.34.0         digest_0.6.12        lme4_1.1-13          evaluate_0.10.1
 [9] nlme_3.1-131         lattice_0.20-35      mgcv_1.8-17          Matrix_1.2-10
[13] igraph_1.0.1         yaml_2.1.14          parallel_3.4.1       SparseM_1.77
[17] stringr_1.2.0        knitr_1.16           MatrixModels_0.4-1   S4Vectors_0.14.3
[21] IRanges_2.10.2       stats4_3.4.1         rprojroot_1.2        nnet_7.3-12
[25] grid_3.4.1           data.table_1.10.4    snow_0.4-2           rmarkdown_1.6
[29] limma_3.32.3         minqa_1.2.4          RedeR_1.24.1         car_2.1-5
[33] magrittr_1.5         backports_1.1.0      htmltools_0.3.6      BiocGenerics_0.22.0
[37] MASS_7.3-47          splines_3.4.1        pbkrtest_0.4-7       BiocStyle_2.4.0
[41] quantreg_5.33        stringi_1.1.5
```

References

[1] Adam Margolin, Ilya Nemenman, Katia Basso, Chris Wiggins, Gustavo Stolovitzky, Riccardo Favera, and Andrea Califano. ARACNE: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, 7(Suppl 1):S7, 2006. doi:10.1186/1471-2105-7-S1-S7.

[2] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2012. ISBN 3-900051-07-0. URL: http://www.R-project.org/.

[3] Maria S Carro, Wei K Lim, Mariano J Alvarez, Robert J Bollo, Xudong Zhao, Evan Y Snyder, Erik P Sulman, Sandrine L Anne, Fiona Doetsch, Howard Colman, Anna Lasorella, Ken Aldape, Andrea Califano, and Antonio Iavarone. The transcriptional network for mesenchymal transformation of brain tumours. *Nature*, 463(7279):318–325, 01 2010. doi:10.1038/nature08712.

[4] Aravind Subramanian, Pablo Tamayo, Vamsi K. Mootha, Sayan Mukherjee, Benjamin L. Ebert, Michael A. Gillette, Amanda Paulovich, Scott L. Pomeroy, Todd R. Golub, Eric S. Lander, and Jill P. Mesirov. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America*, 102(43):15545–15550, 2005. doi:10.1073/pnas.0506580102.

[5] Celine Lefebvre, Presha Rajbhandari, Mariano J Alvarez, Pradeep Bandaru, Wei Keat Lim, Mai Sato, Kai Wang, Pavel Sumazin, Manjunath Kustagi, Brygida C Bisikirska, Katia Basso, Pedro Beltrao, Nevan Krogan, Jean Gautier, Riccardo Dalla-Favera, and Andrea Califano. A human b-cell interactome identifies myb and foxm1 as master regulators of proliferation in germinal centers. *Mol Syst Biol*, 6, 06 2010. doi:10.1038/msb.2010.31.

[6] Michael NC Fletcher, Mauro AA Castro, Suet-Feung Chin, Oscar Rueda, Xin Wang, Carlos Caldas, Bruce AJ Ponder, Florian Markowetz, and Kerstin B Meyer. Master regulators of FGFR2 signalling and breast cancer risk. *Nature Communications*, 4:2464, 2013. doi:10.1038/ncomms3464.

[7] Christina Curtis et al. The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature*, 486:346–352, 2012. doi:10.1038/nature10983.

[8] Patrick Meyer, Frederic Lafitte, and Gianluca Bontempi. minet: A R/bioconductor package for inferring large transcriptional networks using mutual information. *BMC Bioinformatics*, 9(1):461, 2008. doi:10.1186/1471-2105-9-461.

[9] Gabor Csardi and Tamas Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006. URL: http://igraph.sf.net.

[10] Mauro AA Castro, Xin Wang, Michael NC Fletcher, Kerstin B Meyer, and Florian Markowetz. RedeR: R/bioconductor package for representing modular structures, nested networks and multiple levels of hierarchical associations. *Genome Biology*, 13(4):R29, 2012. doi:10.1186/gb-2012-13-4-r29.

[11] Thomas M Campbell, Mauro AA Castro, Ines de Santiago, Michael NC Fletcher, Silvia Halim, Radhika Prathalingam, Bruce AJ Ponder, and Kerstin B Meyer. Fgfr2 risk snps confer breast cancer risk by augmenting oestrogen responsiveness. *Carcinogenesis*, 37(8):741, 2016. doi:10.1093/carcin/bgw065.

[12] Mauro Castro, Ines de Santiago, Thomas Campbell, Courtney Vaughn, Theresa Hickey, Edith Ross, Wayne Tilley, Florian Markowetz, Bruce Ponder, and Kerstin Meyer. Regulators of genetic risk of breast cancer identified by integrative network analysis. *Nature Genetics*, 48:12–21, 2016. doi:10.1038/ng.3458.