Supplementary Information

**RTNsurvival** case studies: regulon activity as a predictor variable in univariate and multivariate survival analyses.

Clarice S. Groeneveld\(^1\), Vinícius S. Chagas\(^1\), Steven J. M. Jones\(^2\), A. Gordon Robertson\(^2\), Bruce A. J. Ponder\(^3\), Kerstin B. Meyer\(^3\)^\(^4\), Mauro A. A. Castro\(^1\)*.

\(^1\)Bioinformatics and Systems Biology Lab, Federal University of Paraná, Curitiba, 81520-260, Brazil.  
\(^2\)Canada’s Michael Smith Genome Sciences Center, BC Cancer Agency, Vancouver, V5Z 4S6, Canada.  
\(^3\)Department of Oncology and Cancer Research UK Cambridge Institute, University of Cambridge, Li Ka Shing Centre, Cambridge, United Kingdom.  
\(^4\)Wellcome Sanger Institute, Hinxton, CB10 1SA, United Kingdom.

*Corresponding author: mauro.castro@ufpr.br

Contents

1. **METABRIC breast cancer cohort** 1
   1.1 Context .................................................. 2
   1.2 Package installation and data sets .................................................. 2
   1.3 Data preprocessing .................................................. 3
   1.4 Regulon activity of individual samples .................................................. 3
   1.5 Regulon activity profiles .................................................. 3
   1.6 Univariate and multivariate survival analyses with RTNsurvival .................. 4
   1.7 Identification of proliferation-related regulons ....................................... 6
   1.8 Other metrics for assessing regulator activity ........................................... 7

2. **TCGA hepatocellular carcinoma cohort (TCGA-LIHC)** 9
   2.1 Context .................................................. 9
   2.2 Download pre-processed data .................................................. 9
   2.3 Inference of the regulatory network with RTN ........................................ 10
   2.4 Univariate and multivariate survival analyses with **RTNsurvival** ................ 10

3. Conclusions and perspectives 13

Session information 14

Supplementary References 15
1. METABRIC breast cancer cohort 1

1.1 Context

For the METABRIC breast cancer cohort, Castro et al. (2016) described a survival analysis that used regulon activity to sort samples in the cohort, which was then stratified and evaluated by Kaplan Meyer (KM) and Cox regression approaches. The authors also described 36 transcription factors (TFs) that were associated with genetic risk of breast cancer. For these 36 TFs, Fletcher et al. (2013) reconstructed regulons using the cohort’s microarray transcriptome data (Curtis et al., 2012). Our goals in this section are, for the METABRIC cohort 1 (n=997): (1) to estimate regulon activity for these 36 TFs in individual samples, (2) to use regulon activity to sort and stratify the samples, considering sorted covariates, and (3) to assess regulon activity as predictor variable in univariate and multivariate survival analyses.

1.2 Package installation and data sets

The RTNsurvival package is available from the R/Bioconductor repository, together with other required packages. Installing and then loading the Fletcher2013b data package will make available all data required for this case study.

```r
#-- Set the Bioconductor repository
#-- Please make sure to use bioc version >= 3.8 (R >= 3.5)
source("https://bioconductor.org/biocLite.R")
biocVersion()

#-- Install RTNsurvival and other required packages
#-- RTNsurvival (>=1.4.4); Fletcher2013b (>=1.16.0); RTN (>= 2.6.2)
biocLite(c("RTNsurvival","Fletcher2013b"))
install.packages("pheatmap")

#-- Call packages
library(RTNsurvival)
library(Fletcher2013b)
library(pheatmap)

#-- Load 'rtni1st' data object, which includes regulons and expression profiles
data("rtni1st")

The rtni1st data also provides clinical and molecular information for 997 samples from the METABRIC cohort 1 (Curtis et al., 2012). The following variables are included in the rtni1st data: time to disease-specific death (time), event death (event), age (Age), tumour grade (Grade, G1, G2 and G3), tumour size (Size), lymph nodes (LN), ER status from IHC (ER+ and ER-), PAM50 subtypes (LumA, LumB, Basal, Her2, and Normal), hormone therapy (HT) and ethnicity (Ethnicity).

#-- Check available attributes in 'colAnnotation'
colAnnotation <- tni.get(rtni1st, what="colAnnotation")
head(colAnnotation)

#-- A list of transcription factors of interest (here, 36 risk-associated TFs)
```
1.3 Data preprocessing

The data preprocessing consists of a single step that creates a **TNS-class** object. This step uses the `tni2tnsPreprocess` function, which requires (1) a transcriptional regulatory network computed by the RTN package, and (2) a list of regulators.

```r
#-- Create TNS-class object from the 'rtni1st'
#-- ...for a faster (parallel) option, please see the 'tnsGSEA2' documentation
rtni1st <- tni2tnsPreprocess(tni = rtni1st, regulatoryElements = risk.tfs, 
                           time = "time", event = "event", endpoint = 120, 
                           keycovar = c("Age","Grade"))
```

1.4 Regulon activity of individual samples

The `tnsPlotGSEA2` function estimates a regulon activity score for a single sample in a cohort, using a two-tailed Gene Set Enrichment Analysis (GSEA-2T). In GSEA-2T, a regulon’s positive and negative targets are each considered separate as pos and neg gene sets. These gene sets are evaluated against a differential gene expression signature, which differs between samples, and is typically calculated in RTNsurvival as follows: For each gene in a sample, a differential gene expression is calculated from its expression in the sample relative to its average expression in the cohort; the genes are then ordered as a ranked list representing a differential gene expression signature, also called the sample’s phenotype. **Supplementary Figure 1a** shows the estimation of ESR1 regulon activity for a single tumour sample from the METABRIC breast cancer cohort. For each gene set (pos and neg) a walk down the ranked list is performed, stepwise. When a gene in the gene set is found, its position is marked in the rug plot, with the colour corresponding to the gene set. A running sum, shown as the pink and blue (pos and neg gene sets, respectively) lines, increases when the gene at that position belongs to the gene set and decreases when it doesn’t. The maximum distance of each running sum from the x-axis represents the enrichment score. GSEA-2T produces two per-sample enrichment scores (ES), whose difference (dES = ES_{pos} - ES_{neg}) represents the regulon activity. The goal is to assess, for each sample, whether the target genes are overrepresented among the genes that are more positively or negatively differentially expressed. For a sample within a cohort, a large positive dES indicates an induced (activated) regulon, while a large negative dES indicates a repressed regulon. Luminal A sample MB-5365 has an activated pattern for ESR1 (**Supplementary Figure 1a**), while basal-like sample MB-2742 has a repressed pattern (**Supplementary Figure 1b**). The regulon status is assigned as undetermined when ES_{pos} and ES_{neg} distributions are skewed to the same side of the ranked list of genes (**Supplementary Figure 1c**).

```r
#-- Two-tailed GSEA plots for individual samples
#-- ...for a faster (parallel) option, please see the 'tnsGSEA2' documentation
#-- Create TNS-class object from the 'rtni1st'
rtni1st <- tnsGSEA2(tni = rtni1st)
```

1.5 Regulon activity profiles

Regulon activity profiles (RAPs) seek to characterize regulatory program similarities and differences between samples in a cohort. In order to assess a large number of samples, we implemented a function that computes the two-tailed GSEA for the entire cohort. For each regulon, the `tnsGSEA2` function estimates a regulon activity score for each sample in the METABRIC cohort 1.

```r
#-- Compute regulon activity for individual samples (this may take a while)
tns1st <- tnsGSEA2(tns1st)
```

3
### Supplementary Figure 1: Example of using a two-tailed GSEA to calculate ESR1 regulon activity in individual tumour samples. The phenotype is the sample’s differential gene expression signature, which is obtained by comparing the expression of each gene in the current sample with its average expression across all samples in the cohort. The phenotype is used to generate the ranked list of genes on which the two-tailed GSEA is carried out for positive and negative targets (red and blue bars, respectively). For sample PAM50 LumA MB-5365 (a) the ESR1 regulon is activated (dES>0), while for sample PAM50 basal-like MB-2742 (b) the ESR1 regulon is repressed (dES<0). Sample MB-5027 (c) represents an inconclusive case, with positive and negative targets skewed to the same side of the ranked list of genes. These plots reproduce results from Castro et al. (2016).

### Supplementary Figure 2 shows a heatmap of regulon activity profiles across the METABRIC cohort, together with tumour ER+/− status and PAM50 subtypes. To a large extent, regulon activity segregates samples into meaningful tumour subtypes. These results are consistent with previous studies showing that regulon activity can be used to sort samples in a cohort (for details, examples and additional interpretations on using the dES metric, please refer to Campbell et al. (2016), Castro et al. (2016), Robertson et al. (2017) and Campbell et al. (2018)).

```r
#-- Get regulon activity and sample attributes
regact_gsea <- tnsGet(tns1st, "regulonActivity")$dif
sdata <- tnsGet(tns1st, "survivalData")
attribs <- c("ER+", "ER-", "LumA", "LumB", "Basal", "Her2", "Normal")

#-- Plot regulon activity profiles
pheatmap(t(regact_gsea), annotation_col = sdata[,attribs], show_colnames = FALSE,
  annotation_legend = FALSE, clustering_method = "ward.D2",
  clustering_distance_rows = "correlation",
  clustering_distance_cols = "correlation")
```

### 1.6 Univariate and multivariate survival analyses with RTNsurvival

The RTNsurvival package uses regulon activity as a predictor variable to study associations between regulons and survival. The tnsKM function can be used to generate Kaplan-Meier curves for one covariate (i.e. regulon) at a time. Supplementary Figure 3a separates the METABRIC cohort (n=997 samples) into three strata according to ESR1 regulon activity (dES<0, undetermined, and dES>0), and Supplementary Figure 3b shows the corresponding Kaplan-Meier curves. High ESR1 regulon activity is strongly associated with better survival (log-rank P = 1.96e-08), reproducing results from Castro et al. (2016). Supplementary Figures 3c-d illustrate an inverse case, with high PPARD regulon activity associated with poorer survival (log-rank P = 1.03e-07). This representation is very convenient for describing the predictor variable along with sample attributes (covariates) and survival curves.
#-- Run KM analysis for regulons

tns1st <- tnsKM(tns1st)

tnsPlotKM(tns1st, rega = "ESR1", attribs = attribs, panelWidths=c(3,1,4), width = 6)

tnsPlotKM(tns1st, rega = "PPARD", attribs = attribs, panelWidths=c(3,1,4), width = 6)

Additionally, in order to study the main effects of survival predictors in a multivariate analysis we use the tnsCox function, which can adjust the analysis by including confounding factors or other covariates. This function relates the activity of one regulon to times-to-events in a multivariate, additive Cox proportional hazards model, and generates a graphic showing the calculated hazard ratios (HR). **Supplementary Figure 3e** shows that within the 36 regulons there are two subsets with statistically significant hazard ratios (HR < 1 or HR > 1, 95% CI). The regulons associated with with higher risk have higher activity values in ER-tumours, particularly basal-like tumors; conversely, regulons associated with lower risk have higher activity in ER+ tumours (**Supplementary Figure 2**).

#-- Run Cox analysis for regulons

tns1st <- tnsCox(tns1st)

tnsPlotCox(tns1st, height = 7)

**Supplementary Figure 2**: Unsupervised hierarchical clustering of regulon activity profiles across the 997 samples of METABRIC cohort 1 for the set of 36 TFs associated with genetic risk of breast cancer described in Castro *et al.* (2016).
1.7 Identification of proliferation-related regulons

Previous literature has indicated challenges in gene set-based survival analysis. Shimoni (2018) described a “random bias” that was attributed to a large proliferation signature that affects a substantial proportion of the genes in the genome. The author implemented a method that removes the bias by adjusting the gene expression data. The method is largely based on the meta-PCNA signature described by Venet et al. (2011), which consists of 131 genes that are associated with proliferation in breast cancer. Shimoni (2018) used the meta-PCNA signature to adjust gene expression for a large number of other cancer types. We used the meta-PCNA signature in our original study (Castro et al., 2016) to identify regulons associated with proliferation in breast cancer, following a method that we described in Fletcher et al. (2013). The method consists of an enrichment analysis where we test which regulons are enriched with the meta-PCNA genes. Since the meta-PCNA signature was inferred in breast cancer, we can apply it to the METABRIC cohort.

In this example we show how to identify regulons enriched with the meta-PCNA signature. From our 36 risk TFs, only 3 regulons (E2F2, E2F3 and ENO1) are enriched with the signature. All three are linked to poor outcomes, consistent with their enrichment with proliferation markers. Please refer to Castro et al. (2016) and Fletcher et al. (2013) for additional details.

```r
#-- Load meta-PCNA signature available from Fletcher2013b data package
data("miscellaneous")
```
```r
#-- Run MRA analysis pipeline
rtna1st <- tni2tna.preprocess(rtni1st, hits=metaPCNA)
rtna1st <- tna.mra(rtna1st)

#-- Check regulons enriched with meta-PCNA genes
metaPCNA_enriched <- tna.get(rtna1st, what="mra")

Table 1: Top 10 regulons enriched with meta-PCNA signature

<table>
<thead>
<tr>
<th>Regulon</th>
<th>Pvalue</th>
<th>Adjusted.Pvalue</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTTG1</td>
<td>1.0e-49</td>
<td>5.7e-47</td>
</tr>
<tr>
<td>FOXM1</td>
<td>1.9e-34</td>
<td>5.5e-32</td>
</tr>
<tr>
<td>E2F2</td>
<td>6.1e-24</td>
<td>1.1e-21</td>
</tr>
<tr>
<td>E2F8</td>
<td>2.1e-23</td>
<td>2.9e-21</td>
</tr>
<tr>
<td>HMGB2</td>
<td>1.7e-16</td>
<td>1.9e-14</td>
</tr>
<tr>
<td>ILF2</td>
<td>5.3e-13</td>
<td>5.0e-11</td>
</tr>
<tr>
<td>VENTX</td>
<td>5.3e-11</td>
<td>4.3e-09</td>
</tr>
<tr>
<td>ZNF395</td>
<td>9.1e-11</td>
<td>6.5e-09</td>
</tr>
<tr>
<td>TGIF2</td>
<td>1.1e-10</td>
<td>6.6e-09</td>
</tr>
<tr>
<td>PURA</td>
<td>7.0e-10</td>
<td>4.0e-08</td>
</tr>
</tbody>
</table>

intersect(metaPCNA_enriched$Regulon, risk.tfs)
```

```r
## [1] "E2F2" "E2F3" "ENO1"
```

### 1.8 Other metrics for assessing regulator activity

There are other tools that provide computational infrastructure to explore regulatory networks. Lefebvre *et al.* (2010) and Tarca *et al.* (2009) developed competing methods to infer sample-specific activities of curated pathways, called *PARADIGM* (PAthway Recognition Algorithm using Data Integration on Genomic Models) and *SPIA* (Signaling Pathway Impact Analysis), respectively. Both approaches predict pathway activities in a sample using gene expression and/or other genomic data (*e.g.* copy number alterations). One essential aspect of these approaches is that they have been designed to assess activity of curated pathways, usually represented by sets of genes annotated in a peer-reviewed process dedicated to provide understanding on, *e.g.* cells, organisms and ecosystems. Currently a large number of resources provide reference pathway annotation, for example, *KEGG* (Kanehisa *et al.*, 2016), *Reactome* (Fabregat *et al.*, 2018), *PID* (Schaefer *et al.*, 2009), *Gene Ontology* (The Gene Ontology Consortium, 2017) and *MSigDB* (Liberzon *et al.*, 2015), the latter representing gene set collections that encompass various other curated pathway resources. However, neither of these approaches is designed to reconstruct TF-centric regulons for a tissue of interest, and neither calculates regulon activity on an individual sample basis. To our knowledge, only *RTN* (Castro *et al.*, 2016; Fletcher *et al.*, 2013) and *VIPER* (Alvarez *et al.*, 2016) provide computational infrastructure for that purpose, both tools using the same principles as the *MARINa* algorithm (Lefebvre *et al.*, 2010), which is inspired by the two-tailed GSEA (Lamb *et al.*, 2006). Alvarez *et al.* (2016) compared 12 regulon activity metrics and concluded that the three-tailed analytic Rank-based Enrichment Analysis (aREA-3T) algorithm provides better accuracy and specificity in detecting changes in protein activity after genetic perturbations, closely followed by GSEA-2T. Both GSEA-2T and aREA-3T algorithms are available in *RTNsurvival* for sorting samples in a cohort. **Supplementary Figures 3a,b** show GSEA-2T results for the ESR1 regulon. To calculate similar results using aREA-3T:

```r
#-- Compute regulon activity for individual samples using aREA-3T algorithm
tns1st_area <- tnsAREA3(tns1st)
```
Supplementary Figure 4a shows that aREA-3T and GSEA-2T algorithms are highly concordant in sorting samples by ESR1 regulon activity. Supplementary Figures 4b,c show a KM analysis run by RTNsurvival using aREA-3T (compare to Supplementary Figures 3a,b). As the regulon activity scores from the current aREA-3T implementation follow a more continuous distribution than those from GSEA-2T, aREA-3T provides clearer boundaries to stratify the cohort into pos vs. neg groups, but less-clear boundaries to assign the undetermined group; therefore the cohort is simply divided into two groups with positive and negative aREA scores.
2. TCGA hepatocellular carcinoma cohort (TCGA-LIHC)

2.1 Context

In section 1, we used a precalculated transcriptional network for the METABRIC breast cancer cohort, which we made available as the Fletcher2013b data package. In section 2, we work with a TCGA cohort. We walk through how to use RTN and RTNsurvival with harmonized GRCh38/hg38 RNA-seq data, which we download from the Genomic Data Commons (GDC, https://gdc.cancer.gov) with the TCGAbiolinks package (Colaprico et al., 2016). We combine the gene expression data with the cohort’s molecular and clinical data, which we download from the The Cancer Genome Atlas Research Network (2017) supplements. We use outcomes data that we download from the Cell web site for the Pan-Cancer Atlas clinical data publication (Liu et al., 2018). We show how to calculate the network from this data with RTN, then how to perform outcome analysis with RTNsurvival. Our goals are similar to those in section 1.

2.2 Download pre-processed data

To run RTNsurvival for a new cohort, we need a gene expression matrix for the cohort, a list of transcriptional factors, and patient metadata from the cohort. The patient metadata may consist solely of some outcome — e.g. overall survival (OS), progression-free interval (PFI), disease-free interval (DFI). While the patient information must be include at least two variables, time and event, it may also contain more information that can be used as attributes and covariates in RTNsurvival functions.

First, we’ll download the pre-processed SummarizedExperiment object. All the preprocessing steps, from the initial GDC download to the final object, are available on the csgroen/RTN_example_TCGA_LIHC repository on Github. The downloaded object consists of three main components: a gene expression matrix, a patient metadata data frame and a gene metadata data frame. We will also get a separate object that contains a list of transcription factors with the necessary annotation.

First, we’ll download the pre-processed SummarizedExperiment object. All the preprocessing steps, from the initial GDC download to the final object, are available on the csgroen/RTN_example_TCGA_LIHC repository on Github. The downloaded object consists of three main components: a gene expression matrix, a patient metadata data frame and a gene metadata data frame. We will also get a separate object that contains a list of transcription factors with the necessary annotation.

```
#-- Repository link and file names
repo_link <- "https://github.com/csgroen/RTN_example_TCGA_LIHC/raw/master/"
fname_exp <- "tcgaLIHCdata_preprocessed.RData"
fname_tfs <- "tfEnsembles.RData"

#-- Download TCGA LIHC data
download.file(paste0(repo_link, fname_exp), fname_exp)
load(fname_exp)

#-- Download transcription factor list and pre-process
download.file(paste0(repo_link, fname_tfs), fname_tfs)
load(fname_tfs)

#-- Call libraries
library(RTNsurvival)
library(SummarizedExperiment)
```
2.3 Inference of the regulatory network with RTN

The RTN pipeline starts with the construction of a TNI-class object, using the `tni.constructor` method. This method takes in a matrix of gene expression and metadata on the samples and genes, as well as a vector of the regulators to be evaluated. Here, the expression matrix and metadata are available as a `SummarizedExpression` object.

```r
#-- TNI constructor
lihcTNI <- tni.constructor(tcgaLIHCdata, regulatoryElements = tfEnsembls)
```

This method also performs pre-processing to check the consistency of all the given arguments and to maximize algorithm performance. It returns a TNI (Transcriptional Network - Inference) object. The next steps run the RTN pipeline to generate the regulons (please refer to Fletcher et al. (2013), Castro et al. (2016) and Robertson et al. (2017) for additional details). To run in multithreaded mode, we suggest looking at the `tni.permutation` and `tni.bootstrap` documentation.

```r
#-- RTN pipeline
#-- Note: this may take some time; for multithreaded mode, please see
#-- 'tni.permutation' or 'tni.bootstrap' documentation
lihcTNI <- tni.permutation(lihcTNI, pValueCutoff = 10^-5, estimator = "spearman")
lihcTNI <- tni.bootstrap(lihcTNI, nBootstraps = 200)
lihcTNI <- tni.dpi.filter(lihcTNI)
```

The `tni.regulon.summary` method lets us get information about the regulons reconstructed by our network. For most calculations, we’ll use the DPI-filtered network, which is enriched with direct regulation relationships. From the summary below, we see that the median regulon size is 30 targets and the mean size is about 49, and, while most regulons in the network will be small, some regulons have over 400 targets.

```r
tni.regulon.summary(lihcTNI)
```

```r
## This regulatory network comprised of 807 regulons.
## -- DPI-filtered network:
## regulatoryElements Targets Edges
## 807 17709 39425
## Min. 1st Qu. Median Mean 3rd Qu. Max.
## 0.0 12.0 30.0 48.9 64.0 434.0
## -- Reference network:
## regulatoryElements Targets Edges
## 807 17709 1646659
## Min. 1st Qu. Median Mean 3rd Qu. Max.
## 0 137 1376 2040 3622 7807
```  

2.4 Univariate and multivariate survival analyses with RTNsurvival

For the survival analysis, we’ll define Age and Tumour Stage as covariates for the Cox regression and evaluate 5-year (60 months) overall survival (OS).

```r
#-- RTNsurvival pipeline
lihcTNS <- tni2tnsPreprocess(lihcTNI, 
                               time = "OS.time.months", event = "OS", 
                               endpoint = 60, keycovar = c("Age", "Tumor_Stage"))
lihcTNS <- tnsGSEA2(lihcTNS)
```
We can explore the Kaplan-Meier and Cox model results compactly in tables.

##-- Explore results

```r
head(tnsGet(lihcTNS, "kmTable"), 10)
```

**Table 2: Top 10 regulons in survival curve differences (G-rho test).**

<table>
<thead>
<tr>
<th>Regulons</th>
<th>ChiSquare</th>
<th>Pvalue</th>
<th>Adjusted.Pvalue</th>
</tr>
</thead>
<tbody>
<tr>
<td>FUBP1</td>
<td>35.91304</td>
<td>0.0e+00</td>
<td>0.0000012</td>
</tr>
<tr>
<td>TAL1</td>
<td>34.36671</td>
<td>0.0e+00</td>
<td>0.0000013</td>
</tr>
<tr>
<td>YBX1</td>
<td>30.82942</td>
<td>0.0e+00</td>
<td>0.0000053</td>
</tr>
<tr>
<td>E2F6</td>
<td>29.10570</td>
<td>1.0e-07</td>
<td>0.0000096</td>
</tr>
<tr>
<td>HMGA1</td>
<td>32.48896</td>
<td>1.0e-07</td>
<td>0.0000099</td>
</tr>
<tr>
<td>ENO1</td>
<td>31.71557</td>
<td>1.0e-07</td>
<td>0.0000107</td>
</tr>
<tr>
<td>GMEB1</td>
<td>27.80588</td>
<td>1.0e-07</td>
<td>0.0000107</td>
</tr>
<tr>
<td>ETV5</td>
<td>25.72268</td>
<td>4.0e-07</td>
<td>0.0000276</td>
</tr>
<tr>
<td>TBX19</td>
<td>23.25694</td>
<td>1.4e-06</td>
<td>0.0000883</td>
</tr>
<tr>
<td>TSC22D4</td>
<td>22.56147</td>
<td>2.0e-06</td>
<td>0.0001142</td>
</tr>
</tbody>
</table>

```r
head(tnsGet(lihcTNS, "coxTable"), 10)
```

**Table 3: Top 10 regulons in Cox Proportional Hazards model.**

<table>
<thead>
<tr>
<th>Regulons</th>
<th>HR</th>
<th>Lower95</th>
<th>Upper95</th>
<th>Pvalue</th>
<th>Adjusted.Pvalue</th>
</tr>
</thead>
<tbody>
<tr>
<td>FUBP1</td>
<td>2.1408242</td>
<td>1.4948719</td>
<td>3.0659003</td>
<td>4.00e-07</td>
<td>0.0002328</td>
</tr>
<tr>
<td>YBX1</td>
<td>2.0069439</td>
<td>1.4249945</td>
<td>2.8265538</td>
<td>1.20e-06</td>
<td>0.0003319</td>
</tr>
<tr>
<td>HMGA1</td>
<td>1.4307089</td>
<td>1.1916198</td>
<td>1.7177693</td>
<td>2.90e-06</td>
<td>0.0004085</td>
</tr>
<tr>
<td>E2F6</td>
<td>2.0915900</td>
<td>1.4349031</td>
<td>3.0488112</td>
<td>2.90e-06</td>
<td>0.0004085</td>
</tr>
<tr>
<td>TAL1</td>
<td>0.4488133</td>
<td>0.2940245</td>
<td>0.6850903</td>
<td>6.00e-06</td>
<td>0.0006811</td>
</tr>
<tr>
<td>GMEB1</td>
<td>1.9245884</td>
<td>1.3521668</td>
<td>2.7393368</td>
<td>9.40e-06</td>
<td>0.0007730</td>
</tr>
<tr>
<td>ZNF408</td>
<td>1.7870618</td>
<td>1.3063348</td>
<td>2.4446946</td>
<td>9.60e-06</td>
<td>0.0007730</td>
</tr>
<tr>
<td>Tumor_Stage</td>
<td>1.6178076</td>
<td>1.2370032</td>
<td>2.1158406</td>
<td>1.85e-05</td>
<td>0.0012039</td>
</tr>
<tr>
<td>KLF9</td>
<td>0.7333000</td>
<td>0.6161108</td>
<td>0.8727795</td>
<td>2.09e-05</td>
<td>0.0012039</td>
</tr>
<tr>
<td>E2F5</td>
<td>1.8708093</td>
<td>1.3156337</td>
<td>2.6602598</td>
<td>2.14e-05</td>
<td>0.0012039</td>
</tr>
</tbody>
</table>

The `tnsPlotKM` method can provide a more complete picture, showing the dynamic range of the activity of a regulon, and how other variables (e.g. Stage, mRNA subtypes) are distributed when the cohort is ordered by activity. In this example, we use Tumor Stage and mRNA-cluster membership (only available for the 196 core tumour samples, see TCGA, 2017) to get an idea of how samples with low and high HMGA1 activity differ.

```r
#-- Kaplan-Meier panel

```r
tnsPlotKM(lihcTNS, "HMGA1",
          attribs = list(c("Stage_I", "Stage_II", "Stage_III", "Stage_IV"),
                         c("mRNA1", "mRNA2", "mRNA3", "mRNA4", "mRNA5")),
          panelWidths = c(2,1,3))
```

**Supplementary Figure 5a** shows the distribution of HMGA1 regulon activity in the cohort tumours, with low activity at the bottom and high activity at the top. The same order is used for
the covariate tracks in the center panel, showing tumour stage and mRNA cluster. Given the distribution of
the tumours, Stage is an interesting covariate for the Cox model. From Supplementary Figure 5b, we see
that even when evaluated with Age and Stage, HMGA1 is still informative of survival and linked to increased
hazard. In this model, each unit increase in HMGA1’s regulon activity corresponds to a 43% higher hazard.

High mobility group A proteins are chromatin remodelers (Sgarra et al., 2018). HMGA1 overexpression
induces oncogenesis and metastasis in cultured cell lines of many phenotypes (Sumter et al., 2016). Indeed, its
overexpression is also linked to poorer prognostic is several cancer types, including hepatocarcinoma (Chang
et al., 2005) (Andreozzi et al., 2016).

For the regulon activity metric, we don’t consider the expression of the gene itself, only of its inferred targets;
hence, it’s a measure of how active a regulator is in a given tumour, not of the regulator’s expression in
that tumour. Here, we show that in addition to HMGA1’s expression being a prognostic marker (see above
publications), its regulon activity is also associated with poorer outcomes.
3. Conclusions and perspectives

*RTNsurvival* extends the functionality of the *RTN* package by finding regulons that are associated with outcomes like survival or progression. The regulon survival analysis uses information about the state of the regulon (i.e. the targets of a regulator) to find these associations.

In these examples, we have used transcription factors as examples of regulators. Transcription factors are particularly well-suited for transcriptional networks, but any regulators whose effect can be reliably measured at the transcriptional level can be used by *RTN* and *RTNsurvival*.

While the multivariate analysis provided by the package considers covariates of the user’s choice, its default analysis it considers only one regulon at the time with these covariates. (e.g. Supplementary Figure 5b) For a multivariate survival analysis that considers covariates and more than one regulon at a time, the regulon activity and all relevant covariates can be recovered from the TNS-class object, as follows.

```r
#-- Get data and bind
full_survData <- tnsGet(lihcTNS, "survivalData")
regulon_activity <- tnsGet(lihcTNS, "regulonActivity")$dif
lihc_data <- cbind(full_survData, regulon_activity)

#-- Example Cox with multiple regulons (FUBP1 and HMGA1)
library(survival)
coxph(Surv(time, event) ~ Tumor_Stage + HMGA1 + FUBP1, data = lihc_data)
```

This approach can also be used for more complex survival models, such as LASSO, Adaptive LASSO, Elastic net and others. A LASSO approach was used by Robertson *et al.* (2017) to identify regulons and other covariates linked to outcome in bladder cancer. R packages *hdnom* (Xiao *et al.*, 2016) and *caret* (Kuhn, 2008) provide frameworks for these models.

The current implementation of *RTNsurvival* accepts only regulons identified by *RTN*; for a new cohort we recommend computing regulons with *RTN* (see section 2).

Given an *RTN* transcriptional network for a cohort, *RTNsurvival* allows a user to 1) estimate the regulon activity of individual samples, 2) generate regulon activity profiles across a cohort, 3) do univariate and multivariate analyses to associate regulon activity with time-to-event (i.e. outcomes) data. Current applications include: 1) assessing covariates across a cohort that has been sorted by regulon activity (Robertson *et al.*, 2017), 2) segregating a cohort for outcomes analysis (Robertson *et al.*, 2017) (Castro *et al.*, 2016), 3) assessing differences between subtypes (Kamoun *et al.*, 2018), and 4) assessing homogeneity/heterogeneity within a subtype (Robertson *et al.*, 2017).

The methods implemented in *RTNsurvival* can also be used with large-scale epigenomic data. For example, recently we showed that regulon activity profiles were consistent with ATAC-seq chromatin accessibility of distal enhancers in breast cancer (Corces *et al.*, 2018). This result provides additional support for regulon activities being a functional readout.
Session information

## R version 3.5.2 (2018-12-20)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 18.04.1 LTS
## Matrix products: default
## BLAS: /usr/lib/x86_64-linux-gnu/openblas/libblas.so.3
## LAPACK: /usr/lib/x86_64-linux-gnu/libopenblas-r0.2.20.so
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] survival_2.43-1     pheatmap_1.0.10    RTNsurvival_1.6.0
## [4] RTNduals_1.6.0       Fletcher2013b_1.18.0 igraph_1.2.2
## [7] RedeR_1.30.0         RTN_2.7.2        Fletcher2013a_1.18.0
## [10] limma_3.38.3
##
## loaded via a namespace (and not attached):
## [1] Rcpp_1.0.0        lattice_0.20-38
## [3] viper_1.16.0      class_7.3-15
## [5] snow_0.4-3        gtools_3.8.1
## [7] digest_0.6.18     GenomeInfoDb_1.18.1
## [9] futile.options_1.0.1 stats4_3.5.2
## [11] evaluate_0.12     e1071_1.7-0
## [13] highr_0.7         gplots_3.0.1
## [15] zlibbioc_1.28.0   VennDiagram_1.6.20
## [17] data.table_1.11.8 gdata_2.18.0
## [19] S4Vectors_0.20.1  Matrix_1.2-15
## [21] markdown_1.11     splines_3.5.2
## [23] BiocParallel_1.16.2 stringr_1.3.1
## [25] mixtools_1.1.0    RColorBrewer_1.1-2
## [27] munsell_0.5.0     DelayedArray_0.8.0
## [29] compiler_3.5.2    xfun_0.4
## [31] pkgconfig_2.0.2   BiocGenerics_0.28.0
## [33] segmented_0.5-3.0 htmltools_0.3.6
## [35] SummarizedExperiment_1.12.0 GenomeInfoDbData_1.2.0
## [37] IRanges_2.16.0    matrixStats_0.54.0
## [39] MASS_7.3-51.1     bitops_1.0-6
## [41] grid_3.5.2        gtable_0.2.0
## [43] magrittr_1.5      formatR_1.5
## [45] scales_1.0.0      minet_3.40.0
## [47] KernSmooth_2.23-15 stringi_1.2.4
## [49] XVector_0.22.0    futile.logger_1.4.3
## [51] lambda.r_1.2.3    RColorBrewer_1.1-2
## [53] tools_3.5.2       Biobase_2.42.0
## [55] parallel_3.5.2    yaml_2.2.0
## [57] colorspace_1.3-2  GenomicRanges_1.34.0
## [59] caTools_1.17.1.1  knitr_1.21
Supplementary References


Sumter, T. et al. (2016) The high mobility group a1 (HMGA1) transcriptome in cancer and development.
Current Molecular Medicine, 16, 353–393.


