TFARM: Transcription Factor Association Rule Miner

Liuba Nausicaa Martino liuban.martino@gmail.com
Alice Parodi
Gaia Ceddia
Piercesare Secchi
Stefano Campaner
Marco Masseroli

June 28, 2017

Contents

1 Introduction 1

2 Dataset 1

3 Extraction of the most relevant associations 3

4 Importance Index of a transcription factor 6

4.1 Validation of the Importance Index formula . . . 12

5 Visualization tools 14

```
> library(TFARM)
> library(GenomicRanges)
>
```

1 Introduction

Looking for association rules between transcription factors in genomic regions of interest can be useful to find direct or indirect interactions among regulatory factors of DNA transcription. However, the results provided by the most recent algorithms for the search of association rules [1] [2] alone are often not enough intelligible and summarized, since they only provide a list of association rules. A novel method is proposed for a subsequent mining of these results to evaluate the contribution of the items in each association rule. The *TFARM* package allows to identify and extract the most relevant association rules with a given transcription factor target, and compute the *Importance Index* of a transcription factor (or a combination of some of them) in the extracted rules. Such index is useful to associate a numerical value to the contribution of one or more transcription factors to the co-regulation with a given transcription factor target.

2 Dataset

Association rules are extracted from a GRanges object in which metadata columns identify transcription factors and genomic coordinates are represented in the left-hand-side of the GRanges, therefore each rows is a different genomic region. The element (i,j) (with j > 4) of the metadata section is equal to 0 if a binding site of transcription factor j is absent in region i, or to 1 (or any other value) if it is present. This dataset, called *Indicator of presence matrix*, should not have rows with only 0 values, since we consider regions with no transcription factors as uninteresting regions. The first column of the genomic ranges contains the chromosome name of each region, the second column contains the genomic coordinates of each region (i.e., *left* coordinate and *right* coordinate are the ends of the region along the DNA, respectively the coordinates of the leftmost and rightmost bases of the region); the third column contains the region strand of each region (encoded as ”+”, ”-”, or ”*” if unknown). The GRanges we consider here is obtained from the analysis of ENCODE ChIP-seq data: it concerns the localization of transcription factors binding sites and histone modifications in DNA, as well as RefSeq data (https://www.ncbi.nlm.nih.gov/refseq/); specifically here we focus on promotorial regions, but further analysis are possible on distal regions (or any DNA region in general, for which an annotation exists). Such data have been processed and extracted with GMQL (GenoMetric Query Language [3], http://www.bioinformatics.deib.polimi.it/GMQL/) queries.

In this example, the dataset we consider is the Indicator of presence matrix of the binding sites of 25 transcription factors of the MCF-7 human breast adenocarcinoma cell line (i.e., all the transcription factors evaluated in ENCODE for this cell line), in the 2,944 promotorial regions of chromosome 1:

```
> # Load and visualize the dataset:
>
> data("MCF7_chr1")
> dim(as.data.frame(MCF7_chr1))

[1] 2944   30

> head(MCF7_chr1)

GRanges object with 6 ranges and 25 metadata columns:
     seqnames           ranges strand |       PML       SRF      CTCF    TCF7L2     FOSL2
        <Rle>        <IRanges>  <Rle> | <numeric> <numeric> <numeric> <numeric> <numeric>
  19     chr1 [565116, 568115]      - |         0         0         1         1         0
  20     chr1 [565156, 568155]      - |         0         0         1         1         0
  21     chr1 [565264, 568263]      - |         0         0         1         1         0
  22     chr1 [566844, 569843]      + |         0         0         1         1         0
  31     chr1 [713069, 716068]      - |         0         1         1         0         0
  32     chr1 [760971, 763970]      + |         0         0         1         0         0
      SIN3AK20     HDAC2     EP300     GABPA      EGR1   HA.E2F1     GATA3      REST
     <numeric> <numeric> <numeric> <numeric> <numeric> <numeric> <numeric> <numeric>
  19         0         0         0         0         0         0         0         0
  20         0         0         0         0         0         0         0         0
  21         0         0         0         0         0         0         0         0
  22         0         0         0         0         0         0         0         0
  31         1         0         0         0         0         1         0         0
  32         0         0         0         1         0         0         0         0
         FOXM1       MYC       MAX     TEAD4     CEBPB      JUND     RAD21      TAF1
     <numeric> <numeric> <numeric> <numeric> <numeric> <numeric> <numeric> <numeric>
  19         0         0         0         0         0         0         0         0
  20         0         0         0         0         0         0         0         0
  21         0         0         0         0         0         0         0         0
  22         0         0         0         0         0         0         0         0
  31         0         1         1         0         0         0         0         1
  32         0         1         0         0         0         0         0         0
         TCF12      ELF1    ZNF217     NR2F2
     <numeric> <numeric> <numeric> <numeric>
  19         0         0         0         0
  20         0         0         0         0
  21         0         0         0         0
  22         0         0         0         0
  31         0         1         0         0
  32           0           0           0           0
  -------
  seqinfo: 24 sequences from an unspecified genome; no seqlengths
>
```

3 Extraction of the most relevant associations

We define a relevant association for the prediction of the presence of transcription factor TFt in the considered genomic regions as an association rule of the type:

$$\{\text{TF1=1,TF2=1,TF3=1}\} \rightarrow \{\text{TFt=1}\}$$

which means that the presence of the transcription factors TF1, TF2 and TF3 implies the presence of transcription factor TFt. Every association rule is completely characterized by a set of three measures: support, confidence and lift:

- *support*:
  $$supp(X \rightarrow Y) = \frac{supp(X \cup Y)}{N} \tag{1}$$
  where N is the number of transactions, X ∪ Y is a set of items and Supp(X ∪ Y) is the support of the itemset {X,Y}, defined as
  $$supp(X) = |\{t_i | X \subseteq t_i, t_i \in T\}| \tag{2}$$
  that is the number of transactions containing the itemset X. The support of an association rule measures the frequency of a rule in the dataset and varies in the interval [0,1].
- *confidence*:
  $$conf(X \rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)} \tag{3}$$
  It gives an estimate of the conditioned probability P(Y|X), that is the probability to find the right-hand-side (RHS) of the rule (i.e., the itemset Y) in a set of transactions, given that these transactions also contain the left-hand-side (LHS) of the rule (i.e., the itemset X). Therefore, it measures the realiability of the inference made by the rule X $\rightarrow Y$. The higher is the confidence of the rule, the higher is the probability to find the itemset Y in a transaction containing the itemset X. It varies in the interval [0,1].
- *lift*:
  $$lift(X \rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)supp(Y)} \tag{4}$$
  It measures the strength of the rule, and varies in the interval [0,∞].

To extract a set of relevant associations the user has to specify:

1. the presence/absence of the transcription factor target to be predicted, TFt;
2. the minimal support threshold of the rules to be extracted;
3. the minimal confidence threshold of the rules to be extracted.

Points 2. and 3. strongly depend on the dimensions of the dataset (i.e., number of rows - regions - and numer of columns - transcription factors), the presence of the transcription factor target in the considered regions, the number of relevant associations that the user wants to find. Usually, the confidence threshold is set greater than 0.5, since it measures the posterior probability to have TFt given the presence of the pattern in the left-hand-side of the rule (e.g., {TF1=1,TF2=1,TF3=1}). The function `rulesGen` in the *TFARM* package extracts the association rules by calling the `apriori` function of the *arules* package [4] [5] [6]. It takes in input:

- the GRanges object in which the Indicator of presence matrix is represented;
- the transcription factor target;
- the minimum support threshold of the rules to be extracted;
- the minimum confidence threshold of the rules to be extracted;

- the logical parameter *type* that sets the type of left-hand-side of the rules to be extracted (i.e., containing only present transcription factors, or containing present and/or absent transcription factors).

The result of the rulesGen function is a data.frame containing:

- in the first column the left-hand-side of each extracted rule;
- in the second column the right-hand-side of each extracted rule (that is the presence/absence of the given transcription factor target);
- in the third column the support of each extracted rule;
- in the fourth column the confidence of each extracted rule;
- in the fifth column the lift of each extracted rule.

See *arulesViz* package for visualization tools of association rules.

```
> # Coming back to the example on the transcription factors of cell line MCF-7,
> # in the promotorial regions of chromosome 1.
> # Suppose that the user wants to find the most relevant association rules for the
> # prediction of the presence of the transcription factor TEAD4 and such that the
> # left-hand-side of the rules contains only present transcription factors.
> # This means extracting all the association rules with right-hand-side equal to
> # {TEAD4=1} setting the parameter type = TRUE; the minimun support and minimum
> # confidence thresholds are set, as an example, to 0.005 and 0.62, respectively:
>
> r_TEAD4 <- rulesGen(MCF7_chr1, "TEAD4=1", 0.005, 0.62, TRUE)
```

```
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target
       0.62    0.1    1 none FALSE            TRUE       5   0.005      1     20  rules
   ext
 FALSE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 14

set item appearances ...[25 item(s)] done [0.00s].
set transactions ...[25 item(s), 2944 transaction(s)] done [0.00s].
sorting and recoding items ... [25 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 done [0.12s].
writing ... [30 rule(s)] done [0.02s].
creating S4 object  ... done [0.00s].
```

```
> dim(r_TEAD4)
```

```
[1] 30  5
```

```
> head(r_TEAD4)
```

```
                                       lhs       rhs     support confidence     lift
1       {GABPA=1,TCF12=1,ZNF217=1,NR2F2=1} {TEAD4=1} 0.005095109  0.6521739 27.42857
2         {FOSL2=1,GABPA=1,MYC=1,ZNF217=1} {TEAD4=1} 0.007812500  0.6571429 27.63755
3 {GABPA=1,MAX=1,TCF12=1,ZNF217=1,NR2F2=1} {TEAD4=1} 0.005095109  0.6521739 27.42857
4 {FOSL2=1,GABPA=1,MYC=1,ZNF217=1,NR2F2=1} {TEAD4=1} 0.006453804  0.6333333 26.63619
```

```
5 {FOSL2=1,HDAC2=1,GABPA=1,MYC=1,ZNF217=1} {TEAD4=1} 0.006114130  0.6428571 27.03673
6   {FOSL2=1,GABPA=1,MYC=1,MAX=1,ZNF217=1} {TEAD4=1} 0.007133152  0.6363636 26.76364

>
```

Once the set of the most relevant association rules (i.e., with support and confidence higher than the thresholds specified as parameters) is extracted, the user can look for *candidate co-regulator transcription factors* with the transcription factor target (in the example TEAD4), which are the transcription factors present in the LHS of the extracted rules. This is provided by the function `presAbs` of the *TFARM* package.
The function `presAbs` takes in input:

- a string vector containing the names of all the transcription factors present in the Indicator of presence matrix;
- the set of the most relevant association rules previously extracted with `rulesGen`;
- a logical parameter, *type*, which refers to the type of rules extracted with the `rulesGen` function. If *type* = *TRUE*, the LHS of the rules can contain only items of the type TF=1, otherwise, if *type* = *FALSE*, the LHS of the rules can contain both items of the type TF=1 and TF=0.

The `presAbs` function has two outputs:

- *pres*, which is a string vector containing all the items present in the LHSs of the considered set of rules;
- *abs*, which is a string vector containing all the items absent in the LHSs of the considered set of rules.

```
> # Transcription factors present in at least one of the regions in the considered dataset:
>
> c <- names(elementMetadata(MCF7_chr1))
> c

 [1] "PML"      "SRF"      "CTCF"     "TCF7L2"   "FOSL2"    "SIN3AK20" "HDAC2"
 [8] "EP300"    "GABPA"    "EGR1"     "HA.E2F1"  "GATA3"    "REST"     "FOXM1"
[15] "MYC"      "MAX"      "TEAD4"    "CEBPB"    "JUND"     "RAD21"    "TAF1"
[22] "TCF12"    "ELF1"     "ZNF217"   "NR2F2"

> lc <- length(c)
> names(presAbs(c, r_TEAD4, TRUE))

[1] "pres" "abs"

> # Transcription factors present in at least one of the association rules:
>
> p <- presAbs(c, r_TEAD4, TRUE)$pres
> p

 [1] "FOSL2=1"    "SIN3AK20=1" "HDAC2=1"    "GABPA=1"    "HA.E2F1=1"  "GATA3=1"
 [7] "MYC=1"      "MAX=1"      "TCF12=1"    "ELF1=1"     "ZNF217=1"   "NR2F2=1"

> # Transcription factors absent in all the association rules:
>
> a <- presAbs(c[1:lc], r_TEAD4, TRUE)$abs
> a

 [1] "PML=1"    "SRF=1"    "CTCF=1"   "TCF7L2=1" "EP300=1"  "EGR1=1"   "REST=1"
 [8] "FOXM1=1"  "TEAD4=1"  "CEBPB=1"  "JUND=1"   "RAD21=1"  "TAF1=1"

>
```

All the transcription factors in p are said to be *candidate co-regulator transcription factors* with the TFt in the most relevant associations extracted with `rulesGen`.

4 Importance Index of a transcription factor

The extraction of candidate co-regulator transcription factors with a given transcription factor target TFt can be useful to provide a global vision of the possible associations of the transcription factor target TFt. However, since the number of association rules and candidate co-regulators can be very high, this list does not provide an intelligible result, giving the lack of a measure of how much each transcription factor contributes to the existence of a certain complex of transcription factors.
Let us consider for example the rule

$$\{\text{TF1=1,TF2=1,TF3=1}\} \rightarrow \{\text{TFt=1}\}$$

Just looking at it, the user could not tell if the presences of TF1, TF2 and TF3 equally contribute to the prediction of the presence of TFt. A solution to this problem can be given by removing, alternatively, TF1, TF2 and TF3 from the rule and evaluate:

1) if the rule keeps on existing and being relevant
2) how the three quality measures of support, confidence and lift of the rule change.

If a rule is not found as relevant after removing a transcription factor from its LHS, then the presence of that transcription factor in the pattern {TF1=1,TF2=1,TF3=1} is fundamental for the existence of the association rule {TF1=1,TF2=1,TF3=1} → {TFt=1}. Otherwise, if the rule keeps on existing as relevant, and its quality measures are similar to the ones of the rule initially considered, then the presence of that transcription factor in the pattern {TF1=1,TF2=1,TF3=1} is not fundamental for the existence of the association rule {TF1=1,TF2=1,TF3=1} → {TFt=1}.
Let us fix an item I (i.e., a candidate co-regulator transcription factor with the transcription factor target) and extract the subset of all the most relevant associations containing I, named {R^I} (with J number of rules in {R^I}, J=|{R^I}|). Each element of {R^I$_j$}$_{j=1:J}$ is described by a set of quality measures of support, confidence and lift: $\{s_j^I,\ c_j^I,\ l_j^I\}_{j=1:J}$.

rule	support	confidence	lift
R_1^I	s_1^I	c_1^I	l_1^I
...	...	...	...
R_J^I	s_J^I	c_J^I	l_J^I

Table 1: Rules containing item I, and corrispondent measures of support, confidence and lift.

Let then be $\{R_j^{I-}\}_{j=1:J}$ the set of rules obtained substituing the presence of the item I with its absence in each element of $\{R_j^I\}_{j=1:J}$. For example, if I is TF1 and R_j^I is the rule {TF1=1,TF2=1,TF3=1} → {TFt=1}, with measures $\{s_j^I, c_j^I, l_j^I\}$, then R_j^{I-} will be the rule $\{TF1 = 0, TF2 = 1, TF3 = 1\} \rightarrow \{TFt = 1\}$ with measures $\{s_j^{I-}, c_j^{I-}, l_j^{I-}\}$.
If a rule in R_j^{I-} is not in the rules that imply the presence of the transcription factor target, then its support, confidence and lift are set to zero.
So now $\{R_j^{I-}\}_{j=1:J}$ is still described by the set $\{s_j^{I-}, c_j^{I-}, l_j^{I-}\}_{j=1:J}$ but where $s_j^{I-} = 0, c_j^{I-} = 0, l_j^{I-} = 0$ for each j such that LHS $\{R_j^{I-}\} \not\rightarrow \{TFt = 1\}$, where TFt is the transcription factor target chosen in the analysis.

rule	support	confidence	lift
R_1^{I-}	s_1^{I-}	c_1^{I-}	l_1^{I-}
...	...	...	...
R_J^{I-}	s_J^{I-}	c_J^{I-}	l_J^{I-}

Table 2: Rules originally containing item I obtained by removing I, and corrispondent support, confidence and lift measures.

To analyze the importance of a transcription factor, for example TF1, we can compare the two distributions $\{s_j^I,\ c_j^I,\ l_j^I\}_{j=1:J}$ and $\{s_j^{I^-},\ c_j^{I^-},\ l_j^{I^-}\}_{j=1:J}$ for each j in {1,...,J}.
Since support, confidence and lift distributions have different means and standard deviations, and since support and confidence vary in [0,1] while lift in [0, ∞], for a coherent comparison they have to be standardized.
In particular, the standardized measures $\{z_s, z_c, z_l\}$ are obtained as::

$$z^I_{s\,j} = \frac{s^I_j - \bar{s^I}}{S^I_s}, \quad z^I_{c\,j} = \frac{c^I_j - \bar{c^I}}{S^I_c}, \quad z^I_{l\,j} = \frac{l^I_j - \bar{l^I}}{S^I_l} \tag{5}$$

where $\bar{s^I}$, $\bar{c^I}$, $\bar{l^I}$ are the mean values of the three distributions s^I, c^I, l^I and S^I_s, S^I_c, S^I_l are the standard deviations of the three distributions s^I, c^I, l^I.

rule	support	confidence	lift
R$^{I}{}_{1}$	z$_s{}^{I}{}_{1}$	z$_c{}^{I}{}_{1}$	z$_l{}^{I}{}_{1}$
...	...	...	...
R$^{I}{}_{J}$	z$_s{}^{I}{}_{J}$	z$_c{}^{I}{}_{J}$	z$_l{}^{I}{}_{J}$

Table 3: Standardized support, confidence and lift distributions of the set of rules containing I, before removing I.

We can define an index of importance of the item I in the rule R$^{I}{}_{j}$ for j in {1,...,J} as:

$$imp(I)_j = \Delta z_{s\,j} + \Delta z_{c\,j} + \Delta z_{l\,j} \tag{6}$$

with: $\Delta z_{s\,j} = z^I_{s\,j} - z^{I^-}_{s\ \ j}$ $\quad \Delta z_{c\,j} = z^I_{c\,j} - z^{I^-}_{c\ \ j}$ $\quad \Delta z_{l\,j} = z^I_{l\,j} - z^{I^-}_{l\ \ j}$

The importance of I in its set of rules {R^{I}} is obtained evaluating the mean of all its importances imp(I)$_j$ in the set of rules:

$$imp(I) = \frac{\sum_{j=1}^{J} imp(I)_j}{J} \tag{7}$$

Then, evaluating the index imp(I) for each item I in the relevant association rules extracted can be useful to rank the transcription factors by their importance in the association with the transcription factor target, TFt. The presence of the transcription factors with highest mean Importance Index is assumed to be fundamental for the existence of some regulatory complexes (i.e., association rules assumed to be relevant); the transcription factors with lower mean importances, instead, do not significantly influence the pattern of transcription factors associated with the transcription factor target.

The definition of the Importance Index can be extended to couples of items, triplettes and so on. This can be easily done substituting the item I with a set of items (for example for a couple of items I becomes, for instance, I={TF1=1,TF2=1}), and applying the rest of the procedure in a completely analogous way. Thus, we identify as R^I the set of rules containing both TF1 and TF2 and R^{I-} as the set of correspondent rules without the two transcription factors. This kind of approach allows the identification of interactions between transcription factors that would be unrevealed just looking at a list of association rules.

The `rulesTF` function in *TFARM* package provides the subset of input rules containing a given transcription factor TFi. It takes in input:

- a set of rules
- the transcription factor TFi that the user wants to find in the LHSs of a subset of the considered rules
- a logical parameter, *verbose*: if *verbose* = *TRUE*, a console message is returned if the searched subset of rules is empty.

rule	support	confidence	lift
R$^{I^-}{}_{1}$	z$_s{}^{I^-}{}_{1}$	z$_c{}^{I^-}{}_{1}$	z$_l{}^{I^-}{}_{1}$
...	...	...	...
R$^{I^-}{}_{J}$	z$_s{}^{I^-}{}_{J}$	z$_c{}^{I^-}{}_{J}$	z$_l{}^{I^-}{}_{J}$

Table 4: Standardized support, confidence and lift distributions of the set of rules originally containing I, after removing I.

The output of the function is a data.frame containing the subset of rules whose LHSs contain TFi, and the corresponding quality measures. Using the introduced notation, the output of the `rulesTF` function is $\{\mathrm{R}^{I}{}_{j}\}_{j=1:J}$ with the quality measures $\{\mathrm{s}^{I}{}_{j},\ \mathrm{c}^{I}{}_{j},\ \mathrm{l}^{I}{}_{j}\}_{j=1:J}$. The data.frame has J rows and five columns: the first colum contains the LHS of the selected rules, the second one contains the RHS of the rules and the last three columns contain $\mathrm{s}^{I}{}_{j}$, $\mathrm{c}^{I}{}_{j}$, $\mathrm{l}^{I}{}_{j}$ (that is a data.frame like the one in Table 1).

```
> # To find the subset of rules containing the transcription factor FOSL2:
>
> r_FOSL2 <- rulesTF(TFi  = 'FOSL2=1', rules =  r_TEAD4, verbose = TRUE)
> head(r_FOSL2)

                                                 lhs       rhs     support confidence
1                   {FOSL2=1,GABPA=1,MYC=1,ZNF217=1} {TEAD4=1} 0.007812500  0.6571429
2           {FOSL2=1,GABPA=1,MYC=1,ZNF217=1,NR2F2=1} {TEAD4=1} 0.006453804  0.6333333
3           {FOSL2=1,HDAC2=1,GABPA=1,MYC=1,ZNF217=1} {TEAD4=1} 0.006114130  0.6428571
4             {FOSL2=1,GABPA=1,MYC=1,MAX=1,ZNF217=1} {TEAD4=1} 0.007133152  0.6363636
5   {FOSL2=1,HDAC2=1,GABPA=1,GATA3=1,MYC=1,ZNF217=1} {TEAD4=1} 0.005434783  0.6400000
6 {FOSL2=1,GABPA=1,HA.E2F1=1,GATA3=1,MYC=1,ZNF217=1} {TEAD4=1} 0.005095109  0.6250000
      lift
1 27.63755
2 26.63619
3 27.03673
4 26.76364
5 26.91657
6 26.28571

> dim(r_FOSL2)[1]

[1] 28

>
```

```
> # If none of the rules in input to rulesTF contains the given item TFi,
> # and verbose = TRUE, a console message warns for an error:
>
> r_CTCF <- rulesTF(TFi = 'CTCF=1', rules = r_TEAD4, verbose = TRUE)

[1] "None of the rules contains CTCF=1"

>
```

If the user wants to evaluate the importance of an item I in a set of rules R^I, the user needs to substitute the presence of I in all the left-hand-side patterns of R^I with its absence: this is done using the function `rulesTF0` in *TFARM* package. This function takes in input:

- the transcription factor TFi to be removed
- a set of rules containing TFi
- the total set of rules
- the GRanges object containing the Indicator of presence matrix
- the transcription factor target.

It returns a data.frame with the rules obtained substituing the presence of TFi with its absence and the corrispondent measures. Using the introduced notation, the output of the `rulesTF0` function is $\{\mathrm{R}^{I-}{}_{j}\}_{j=1:J}$ with the quality measures $\{\mathrm{s}^{I-}{}_{j},\ \mathrm{c}^{I-}{}_{j},\ \mathrm{l}^{I-}{}_{j}\}_{j=1:J}$. The data.frame has J rows and five columns: the first colum contains the LHS of the rules in R^I without TFi, the second one contains the RHS of the rules and the last three columns contain $\mathrm{s}^{I-}{}_{j}$, $\mathrm{c}^{I-}{}_{j}$, $\mathrm{l}^{I-}{}_{j}$ (that is a data.frame like the one in Table 2).

```
> # For example to evaluate FOSL2 importance in the set of rules r_FOSL2:
>
> r_noFOSL2 <- rulesTF0('FOSL2=1', r_FOSL2, r_TEAD4, MCF7_chr1, "TEAD4=1")
>
> head(r_noFOSL2)

                                                 lhs       rhs      support confidence
1                   {FOSL2=0,GABPA=1,MYC=1,ZNF217=1} {TEAD4=1} 0.0027173913 0.09876543
2           {FOSL2=0,GABPA=1,MYC=1,ZNF217=1,NR2F2=1} {TEAD4=1} 0.0020380435 0.13043478
3           {FOSL2=0,HDAC2=1,GABPA=1,MYC=1,ZNF217=1} {TEAD4=1} 0.0010190217 0.06521739
4             {FOSL2=0,GABPA=1,MYC=1,MAX=1,ZNF217=1} {TEAD4=1} 0.0027173913 0.10126582
5   {FOSL2=0,HDAC2=1,GABPA=1,GATA3=1,MYC=1,ZNF217=1} {TEAD4=1} 0.0006793478 0.05714286
6 {FOSL2=0,GABPA=1,HA.E2F1=1,GATA3=1,MYC=1,ZNF217=1} {TEAD4=1} 0.0006793478 0.04166667
      lift
1 4.153792
2 5.485714
3 2.742857
4 4.258951
5 2.403265
6 1.752381

>
```

Now that the two sets of rules $\{\mathrm{R}^{I}{}_{j}\}_{j=1:J}$ and $\{\mathrm{R}^{I-}{}_{j}\}_{j=1:J}$ and the two sets of measures $\{\mathrm{s}^{I}{}_{j}$, $\mathrm{c}^{I}{}_{j}$, $\mathrm{l}^{I}{}_{j}\}_{j=1:J}$ and $\{\mathrm{s}^{I-}{}_{j}$, $\mathrm{c}^{I-}{}_{j}$, $\mathrm{l}^{I-}{}_{j}\}_{j=1:J}$ are obtained, the user can compute the Importance Index distribution for the chosen transcription factor TFi.
This can be done with the function `IComp` in the *TFARM* package which takes in input:

- the transcription factor TFi
- the subset of rules rules_TF containing TFi (provided by the function `rulesTF`) with their quality measures of support, confidence and lift
- the subset of rules rules_noTF obtained from rules_TF removing TFi (provided by the function `rulesTF0`)
- a logical parameter (figures) to graphically rapresent $\{\mathrm{s}^{I}{}_{j}$, $\mathrm{c}^{I}{}_{j}$, $\mathrm{l}^{I}{}_{j}\}_{j=1:J}$ and $\{\mathrm{s}^{I-}{}_{j}$, $\mathrm{c}^{I-}{}_{j}$, $\mathrm{l}^{I-}{}_{j}\}_{j=1:J}$; set *figures* = *TRUE* to get it as an output.

The function has five outputs:

- imp, which is the set of importance index values of TFi in the given set of rules (rules_TF), one value for each rule.
- delta, which is the matrix of variations of standarded support, confidence and lift obtained removing TFi from rules_TF.
- rwi, which is a data.frame that contains rules from `rulesTF` associated with transcription factors having Importance Index greater than one.
- rwo, which is a data.frame with rules in rwi obtained removing each transcription factor TFi.
- the plots of $\{\mathrm{s}^{I}{}_{j}$, $\mathrm{c}^{I}{}_{j}$, $\mathrm{l}^{I}{}_{j}\}_{j=1:J}$ and $\{\mathrm{s}^{I-}{}_{j}$, $\mathrm{c}^{I-}{}_{j}$, $\mathrm{l}^{I-}{}_{j}\}_{j=1:J}$ obtained if the user sets *figures* = *TRUE*.

```
> # Perform the IComp function to compute the Importance Index distribution:
>
> imp_FOSL2 <- IComp('FOSL2=1', r_FOSL2, r_noFOSL2, figures=TRUE)
> names(imp_FOSL2)

[1] "imp"   "delta" "rwi"   "rwo"

> imp_FOSL2$imp

 [1] 4.6295239 3.8002131 3.2324984 0.3728276 3.7140239 0.1584065 1.1195624 3.2324984
 [9] 3.2324984 0.2321155 0.2321155 1.1195624 1.1195624 3.2324984 0.1570690 1.1195624
```

```
> head(imp_FOSL2$delta)

  diff_supp_Z diff_conf_Z diff_lift_Z
1  1.02954330   1.7999903   1.7999903
3  0.97910426   1.4105544   1.4105544
5  0.46114595   1.3856762   1.3856762
6 -0.04672455   0.2097761   0.2097761
8  1.46679915   1.1236124   1.1236124
9  0.47123376  -0.1564136  -0.1564136

> head(imp_FOSL2$rwi)

                                                 lhs       rhs     support confidence
1                   {FOSL2=1,GABPA=1,MYC=1,ZNF217=1} {TEAD4=1} 0.007812500  0.6571429
3           {FOSL2=1,HDAC2=1,GABPA=1,MYC=1,ZNF217=1} {TEAD4=1} 0.006114130  0.6428571
5   {FOSL2=1,HDAC2=1,GABPA=1,GATA3=1,MYC=1,ZNF217=1} {TEAD4=1} 0.005434783  0.6400000
6 {FOSL2=1,GABPA=1,HA.E2F1=1,GATA3=1,MYC=1,ZNF217=1} {TEAD4=1} 0.005095109  0.6250000
8   {FOSL2=1,HDAC2=1,GABPA=1,MYC=1,ZNF217=1,NR2F2=1} {TEAD4=1} 0.005774457  0.6296296
9     {FOSL2=1,HDAC2=1,GABPA=1,MYC=1,MAX=1,ZNF217=1} {TEAD4=1} 0.005774457  0.6296296
      lift
1 27.63755
3 27.03673
5 26.91657
6 26.28571
8 26.48042
9 26.48042

> head(imp_FOSL2$rwo)

                                                 lhs       rhs      support confidence
1                   {FOSL2=0,GABPA=1,MYC=1,ZNF217=1} {TEAD4=1} 0.0027173913 0.09876543
3           {FOSL2=0,HDAC2=1,GABPA=1,MYC=1,ZNF217=1} {TEAD4=1} 0.0010190217 0.06521739
5   {FOSL2=0,HDAC2=1,GABPA=1,GATA3=1,MYC=1,ZNF217=1} {TEAD4=1} 0.0006793478 0.05714286
6 {FOSL2=0,GABPA=1,HA.E2F1=1,GATA3=1,MYC=1,ZNF217=1} {TEAD4=1} 0.0006793478 0.04166667
8   {FOSL2=0,HDAC2=1,GABPA=1,MYC=1,ZNF217=1,NR2F2=1} {TEAD4=1} 0.0003396739 0.03225806
9     {FOSL2=0,HDAC2=1,GABPA=1,MYC=1,MAX=1,ZNF217=1} {TEAD4=1} 0.0010190217 0.06521739
      lift
1 4.153792
3 2.742857
5 2.403265
6 1.752381
8 1.356682
9 2.742857

>
```

The most useful application of the function `IComp` is the ranking of candidate co-regulator transcription factors through their importances.
As previously seen, the candidate co-regulators are returned by the function `presAbs`. The evaluation of the mean importance of each co-regulator can be computed cycling the three functions `rulesTF`, `rulesTF0` and `IComp` over a string vector with all the transcription factors present in the set of relevant association rules extracted, as returned by the function presAbs.

```
> # For the considered example the user could run:
>
> library(plyr)
```

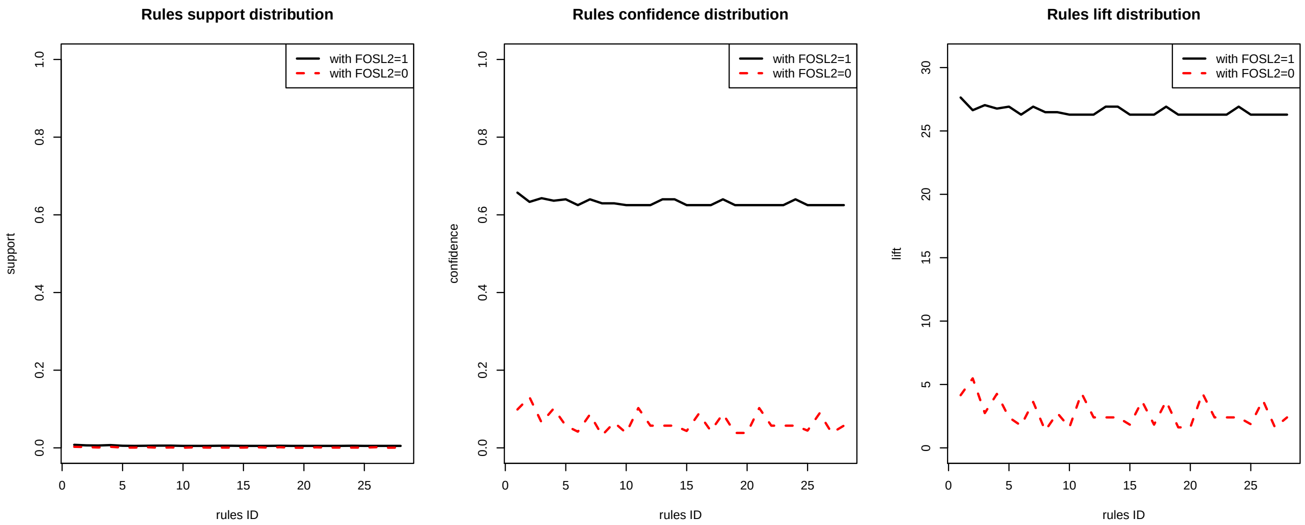

Figure 1: **Support, Confidence and Lift for the extracted rules before and after the removal of item I.** Left panel: Support distribution $\{s_j^I\}_{j=1:J}$, black thick line, and $\{s_j^{I-}\}_{j=1:J}$,red dotted line. Middle panel: Confidence distribution $\{c_j^I\}_{j=1:J}$, black thick line, and $\{c_j^{I-}\}_{j=1:J}$, red dotted line. Right panel: Lift distribution $\{l_j^I\}_{j=1:J}$, black thick line and $\{l_j^{I-}\}_{j=1:J}$, red dotted line.

```
> A <- vector("list", length(p))
> B <- vector("list", length(p))
> IMP <- matrix(0, length(p), 4)
> IMP <- data.frame(IMP)
> IMP[,1] <- paste(p)
> colnames(IMP) <- c('TF', 'imp', 'sd', 'nrules')
> IMP_Z <- vector("list", length(p))
> for (i in 1:length(p))  {
+         A[[i]] <- rulesTF(p[i], r_TEAD4, FALSE)
+         B[[i]] <- rulesTF0(p[i], A[[i]], r_TEAD4, MCF7_chr1, "TEAD4=1")
+         IMP_Z[[i]] <- IComp(p[i], A[[i]], B[[i]], figures=FALSE)$imp
+ }
> IMP[, 2] <- sapply(IMP_Z, mean)
> IMP[, 3] <- sapply(IMP_Z, sd)
> IMP[, 4] <- sapply(IMP_Z, length)
>

> # Sort by imp column of IMP
>
> IMP.ord <- arrange(IMP, desc(imp))
> IMP.ord
```

```
           TF        imp        sd nrules
1     TCF12=1 28.0858405 3.0000000      2
2   HA.E2F1=1 26.9158094 3.0000000      4
3      ELF1=1 26.9158094 3.0000000      8
4     NR2F2=1  6.0654609 4.8049564      3
5       MYC=1  3.3151961 0.5123463      7
6     GATA3=1  3.1294004 2.6995539     11
7    ZNF217=1  2.9563087 2.2906579     12
```

```
8  SIN3AK20=1  2.2083521 1.5544120      7
9     FOSL2=1  1.9190336 1.5885303     16
10    HDAC2=1  1.6198970 0.9829202     10
11    GABPA=1  1.4608760 1.8299519     20
12      MAX=1  0.4240596 0.5357242     12

>
```

In this way we get, besides the mean Importance Index of each candidate co-regulator of TFt (TFt = TEAD4 in the example), the standard deviation of the distribution of the Importance Index of each candidate co-regulator of TFt, and the number of rules in which each candidate co-regulator of TFt is present. The function `IComp` can be easily generalized

for the computation of the mean Importance Index of combinations of transcription factors (see the example used for the `heatI` function in the following section).

4.1 Validation of the Importance Index formula

We defined the Importance Index of an item in an association rule as a linear combination of the variations of the standardized support, confidence and lift of the rule, obtained substituting the presence of the item in the left-hand-side of the association rule, with its absence (as in Formula 6). In this way we assume that each of the three variations equally contributes to the evaluation of the contribution of the item to the prediction of the presence of another item in the right-hand-side of the considered association rule.
Neverthless, one of the three quality measures might be more or less sensitive than the others to the removal of the item from the rule, leading to a greater or smaller variation of one or more of the standardized values of support, confidence and lift.
We observe that for each item I, the variations of support, confidence and lift obtained removing I from a set of rules in which I is involved, are placed in a 3D space defined by the terns (Δz_s, Δz_c, Δz_l).

TF	Δz_s	Δz_c	Δz_l
TF_1	$\Delta z_{s,1}$	$\Delta z_{c,1}$	$\Delta z_{l,1}$
...	...	...	...
TF_1	$\Delta z_{s,n_1}$	$\Delta z_{c,n_1}$	$\Delta z_{l,n_1}$
...	...	...	...
TF_M	$\Delta z_{s,K-n_M+1}$	$\Delta z_{c,K-n_M+1}$	$\Delta z_{l,K-n_M+1}$
...	...	...	...
TF_M	$\Delta z_{s,K}$	$\Delta z_{c,K}$	$\Delta z_{l,K}$

Table 5: Matrix with the variations of the standardized support, confidence and lift, obtained removing each transcription factor from the subset of rules in which it is present. M is the total number of transcription factors, K is the total number of rules and n_i is the number of rules for transcription factor TFi.

Thanks to the Principal Components Analysis [7] [8], computed by the function `IPCA` in the *TFARM* package, we can evaluate if it is possible to find a subspace of $\mathbb{R}^3$ in which the most variability of the dataset containing the variations of the standardized measures (Table 5) is captured. This can be easily done by extracting the delta variations of support, confidence and lift, using the function `IComp`, simply getting its *delta* output, as well as a matrix containing the candidate co-regulators found, and the number of rules in which each of them appears.

A principal component is a combination of the original variables after a linear transformation; the set of principal components defines a new reference system. The new coordinates of data rapresented in the reference system defined by principal components are called *scores*, and the coefficients of the linear combination that define each principal component are called *loadings* (so, loadings give a measure of the contribution of every observation to each principal component).

The `IPCA` function takes in input:

- the list of variations of standardized distributions of support, confidence and lift measures, obtained from the `IComp` function

- a matrix with the mean importance of every candidate co-regulator transcription factor and the number of rules in which each of them appears.

It returns:

- a summary, containing: the standard deviation on each principal component, the proportion of variance explained by each principal component, and the cumulative proportion of variance described by each principal component;
- the scores of each principal component
- the loadings of each principal component
- a plot with the variability and the cumulate percentage of variance explained by each principal component
- a plot with the loadings of the principal components

```
> # Extract the delta variations of support, confidence and lift:
>
> DELTA <- vector("list", length(p))
> for (i in 1:length(p)){
+ DELTA[[i]] <- IComp(p[i], A[[i]], B[[i]], figures=FALSE)$delta
+ }
> # Select the candidate co-regulators and the number of rules associated with them, then
> # perform the Principal Component Analysis:
>
> colnames(IMP)

[1] "TF"     "imp"    "sd"     "nrules"

> I <- data.frame(IMP$TF, IMP$imp, IMP$nrules)
> i.pc <- IPCA(DELTA, I)
> names(i.pc)

[1] "summary"  "scores"   "loadings"

> i.pc$summary

Importance of components:
                          Comp.1     Comp.2     Comp.3
Standard deviation     8.4420918 2.78122053 1.57276171
Proportion of Variance 0.8747047 0.09493628 0.03035898
Cumulative Proportion  0.8747047 0.96964102 1.00000000

> head(i.pc$loadings)

[1] -0.40810581 -0.61522453 -0.67449865 -0.87806008  0.06225877  0.47448323

> head(i.pc$scores)

        Comp.1      Comp.2      Comp.3
[1,] 1.7070022 -0.39838625 -0.34511780
[2,] 1.8393873  0.58179136 -0.99166754
[3,] 2.2617460  0.07731102 -0.08263486
[4,] 2.7841527  0.47798824  0.18638357
[5,] 2.6043040 -0.16366502 -0.36989264
[6,] 0.7976814  0.61114422 -0.59782177

>
```

As we can see looking at the value of the variance associated with the first principal component in Figure 2, this value explains the 87.47% of the variability of the DELTA dataset. Moreover, from the plot of the loadings in Figure 2, it is easy to note that the first principal component is a linear combination of the variations of standardized support, confidence and lift, that equally contribute to the combination. So, it is reasonable to define the Importance Index as in Formula 6.

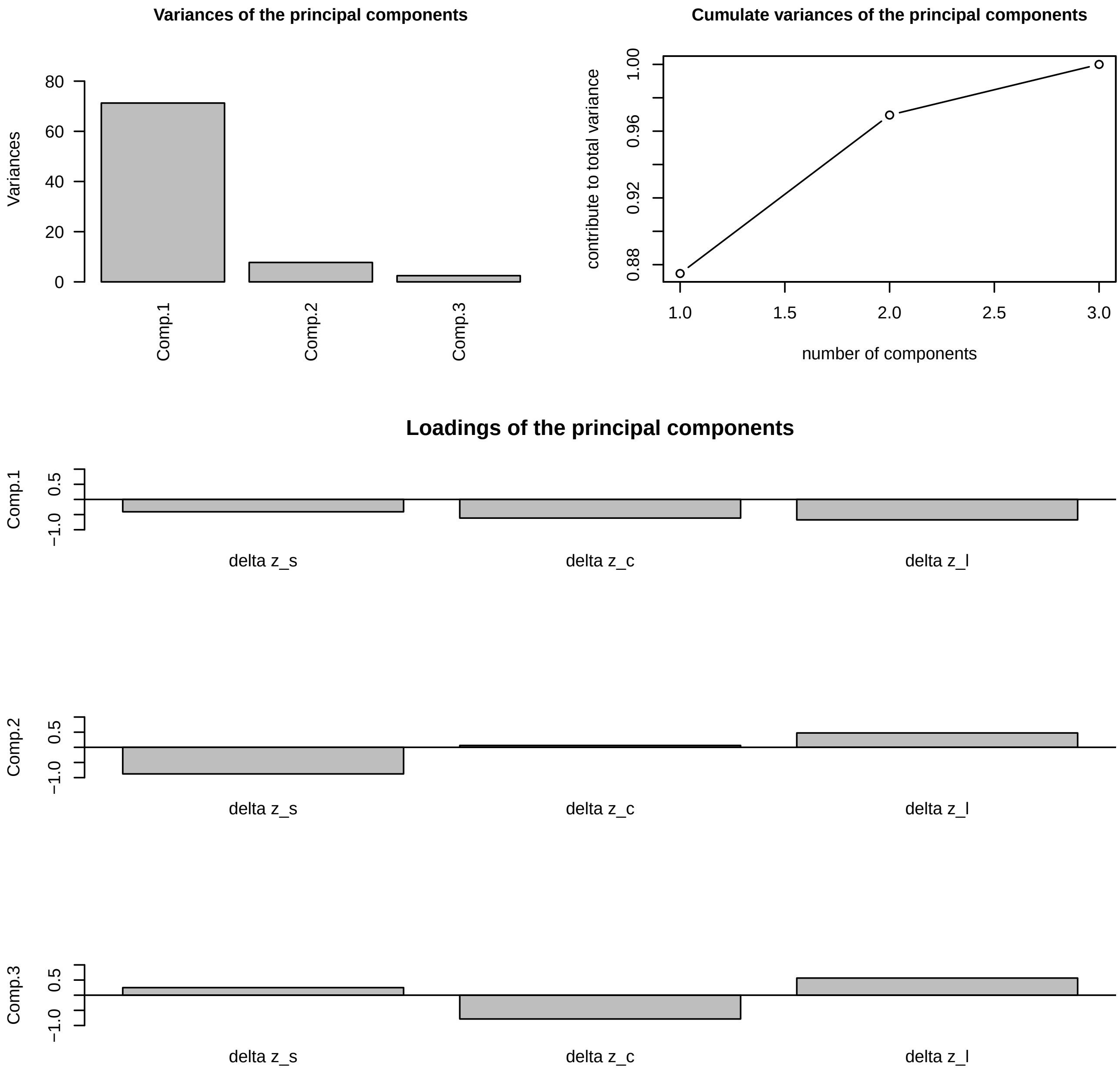

Figure 2: **Principal Component Analysis of Importance Index** Variances of each of the three principal components (on the top left), cumulate proportion of variance explained by each principal component (on the top right), and loadings of the three principal components.

5 Visualization tools

The function `distribViz` in the *TFARM* package provides a boxplot visualization of the Importance Index distributions of a set of transcription factors (or of combinations of transcription factors).

```
> # Considering for example the candidate co-regulator transcription factors
> # found in the set of rules r_TEAD4:
>
> distribViz(IMP_Z, p)

$stats
         [,1]      [,2]      [,3]      [,4]     [,5]      [,6]       [,7]     [,8]
[1,] 25.29562 0.1570690 0.6518149 0.9649109 24.31773 0.5467096 0.07131959 3.508845
[2,] 25.29562 0.3024715 0.6518149 0.9649109 24.31773 1.0505777 0.07131959 3.508845
[3,] 25.29562 1.1195624 0.8236844 0.9649109 26.91581 1.0505777 0.07131959 3.508845
[4,] 28.53599 3.2324984 1.1810186 4.5175059 29.51389 2.3291325 1.01023868 3.508845
[5,] 31.77636 4.6295239 1.5328197 7.5461342 29.51389 3.2435816 1.48744255 3.508845
          [,9]    [,10]    [,11]    [,12]
[1,] 0.5171752 0.983859 25.96452 2.320627
[2,] 4.6783894 0.983859 25.96452 2.320627
[3,] 8.8396037 0.983859 28.08584 2.320627
[4,] 8.8396037 3.696249 30.20716 3.231715
[5,] 8.8396037 4.130531 30.20716 4.178313

$n
 [1]  8 16 20 11  4 10 12  7  3  7  2 12

$conf
         [,1]        [,2]      [,3]       [,4]     [,5]      [,6]       [,7]     [,8]
[1,] 23.48551 -0.03779824 0.6367173 -0.7275025 22.81085 0.4117607 -0.3569277 3.508845
[2,] 27.10574  2.27692302 1.0106514  2.6573243 31.02077 1.6893947  0.4995669 3.508845
          [,9]      [,10]    [,11]    [,12]
[1,]  5.043688 -0.6359366 23.34584 1.905074
[2,] 12.635519  2.6036545 32.82584 2.736180

$out
[1] 8.6328733 2.9652559 3.0387048 2.1533040 9.4810373 0.7145803 0.7145803

$group
[1]  3  3  3  8 12 12 12

$names
 [1] "ELF1=1"     "FOSL2=1"    "GABPA=1"    "GATA3=1"    "HA.E2F1=1"  "HDAC2=1"
 [7] "MAX=1"      "MYC=1"      "NR2F2=1"    "SIN3AK20=1" "TCF12=1"    "ZNF217=1"

>
```

The shape of a boxplot changes as follows:

- The higher the number of rules containing the candidate co-regulator I, the larger the boxplot for I is
- The higher the variability of the Importance Index of I, the longer the boxplot for I is
- The higher the median of the Importance Index distribution of I, the higher the boxplot for I is aligned with respect to the y axis.

Moreover, named q_1 and q_3 the first and third quartiles of the Importance Index distribution for a given item I, all the rules where I has importance $x \leq q_1 - 1.5 * (q_3 - q_1)$ or $x \geq q_1 + 1.5 * (q_3 - q_1)$ are considered outlier rules, and represented with a circle outside the boxplot.

For example, in the boxplots in Figure 3 it is easy to notice that:

1. ELF1, HA.E2F1 and TCF12 have the highest median Importance Index, and they are present in an intermediate number of relevant association rules

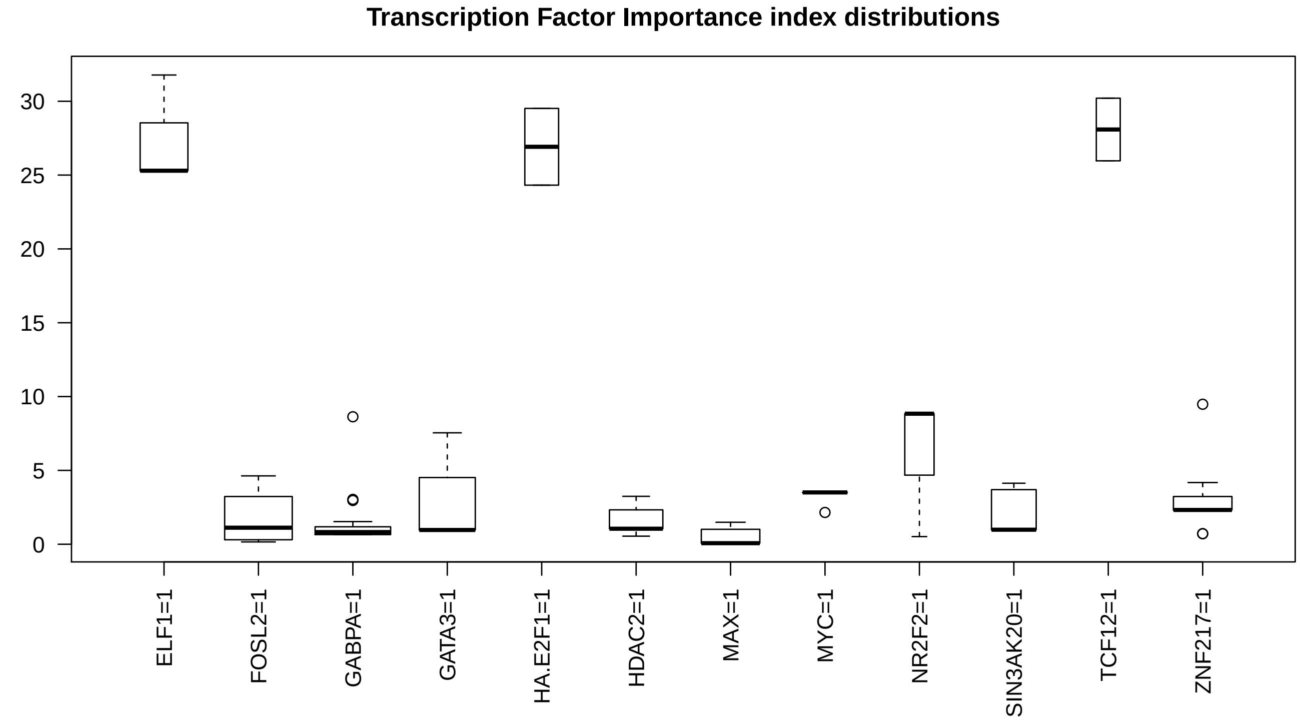

Figure 3: **Importance Index distribution.** Importance Index distributions of candidate co-regulators of TEAD4 in the set of the 30 most relevant associations for the prediction of the presence of TEAD4 in promotorial regions of chromosome 1 in cell line MCF7.

2. GABPA, HDAC2, MAX, MYC and ZNF217 have low median Importance Index and the lowest variability Importance Index distribution
3. FOSL2 and GATA3 are present in a high number of relevant association rules, but they have low median Importance Index and high variability of the Importance Index distribution.

It can also be noticed that for the transcription factors GABPA, MYC and ZNF217 there are some outlier rules, that are rules in which the Importance Index of the candidate co-regulator transcription factor is a lot higher or lower than in the rest of the distribution.
These outliers can be extracted as reported in the following text:

```
> # Select the index of the list of importances IMP_Z
> # containing importance distributions of transcription factor ZNF217
> ZNF217_index <- which(p == 'ZNF217=1')
> # select outlier rules where ZNF217 has importance greater than 5
> o <- IMP_Z[[ZNF217_index]] > 5
> rule_o <- A[[ZNF217_index]][o,]
> rule_o

                                                              lhs         rhs     support
1                              {GABPA=1,TCF12=1,ZNF217=1,NR2F2=1} {TEAD4=1} 0.005095109
13   {FOSL2=1,SIN3AK20=1,GABPA=1,GATA3=1,MYC=1,ZNF217=1,NR2F2=1} {TEAD4=1} 0.005095109
25 {FOSL2=1,HDAC2=1,GABPA=1,GATA3=1,MYC=1,MAX=1,ELF1=1,ZNF217=1} {TEAD4=1} 0.005095109
   confidence     lift
1   0.6521739 27.42857
```

```
13  0.6250000 26.28571
25  0.6250000 26.28571
```

```
> # So, ZNF217 is very relevant in the pattern of transcription factors
> # {GABPA=1,TCF12=1,ZNF217=1,NR2F2=1}
> # for the prediction of the presence of TEAD4.
>
> # To extract support, confidence and lift of the corresponding rule without ZNF217:
> B[[ZNF217_index]][o,]

                                                             lhs       rhs      support
1                              {GABPA=1,TCF12=1,ZNF217=0,NR2F2=1} {TEAD4=1} 0.0003396739
13   {FOSL2=1,SIN3AK20=1,GABPA=1,GATA3=1,MYC=1,ZNF217=0,NR2F2=1} {TEAD4=1} 0.0000000000
25 {FOSL2=1,HDAC2=1,GABPA=1,GATA3=1,MYC=1,MAX=1,ELF1=1,ZNF217=0} {TEAD4=1} 0.0000000000
   confidence     lift
1      0.0625 2.628571
13     0.0000 0.000000
25     0.0000 0.000000

>
> # Since none of the three measures of the rule obtained removing ZNF217 are equal to zero,
> # the rule {GABPA=1,TCF12=1,ZNF217=0,NR2F2=1} -> {TEAD4=1},
> # obtained removing ZNF217, is found in the relevant rules for the prediction
> # of the presence of TEAD4.
>
```

The function `heatI` is another useful visualization tool of the package *TFARM*; it takes in input:

- a string vector with names of transcription factors
- a vector of mean importances of pairs of transcription factors in the previous input.

It returns a heatmap visualization of the mean importances of transcription factors in the considered string vector.

Evaluating importances of combinations of transcription factors, the number of Importance Index distribution grows combinatorially. This makes more difficult to see which are the most important combinations (even sorting them by their mean importances).
For the pairs of transcription factors, the function `heatI` gives an heatmap visualization of a square matrix whose elements are as follows (Table 6): called M the number of candidate co-regulators transcription factors, the element (i,j) of such matrix is the mean importance of the couple of transcription factors (TF_i, TF_j). This matrix is symmetric with respect to the main diagonal.

	TF_1	TF_2	...	TF_{M-1}	TF_M
TF_1	imp(TF_1)	imp($\{TF_1,TF_2\}$)	...	imp($\{TF_1,TF_{M-1}\}$)	imp($\{TF_1,TF_M\}$)
TF_2	imp($\{TF_2,TF_1\}$)	imp(TF_2)	...	imp($\{TF_2,TF_{M-1}\}$)	imp($\{TF_2,TF_M\}$)
...					
TF_{M-1}	imp($\{TF_{M-1},TF_1\}$)	imp($\{TF_{M-1},TF_2\}$)	...	imp(TF_{M-1})	imp($\{TF_{M-1},TF_M\}$)
TF_M	imp($\{TF_M,TF_1\}$)	imp($\{TF_M,TF_2\}$)	...	imp($\{TF_M,TF_{M-1}\}$)	imp(TF_M)

Table 6: Mean importance matrix of couples of transcription factors

To get this matrix, all the possible combinations of two candidate co-regulator transcription factors need to be built. It can be easily computed with the function `combn` in the package *combinat*. This function takes as input a vector (which is a string vector of transcription factors) and the number of elements in the required combinations. Using combn(p, 2), it generates all combinations of the elements of p taken two at a time. The elements of each combination are then combined in the form *TF1,TF2*.

```
> # Construct couples as a vector in which all possible combinations of
> # transcription factors (present in at least one association rules)
> # are included:
>
> couples_0 <- combn(p, 2)
> couples <- paste(couples_0[1,], couples_0[2,], sep=',')
> head(couples)

[1] "FOSL2=1,SIN3AK20=1" "FOSL2=1,HDAC2=1"    "FOSL2=1,GABPA=1"    "FOSL2=1,HA.E2F1=1"
[5] "FOSL2=1,GATA3=1"    "FOSL2=1,MYC=1"

>

> # The evaluation of the mean Importance Index of each pair is
> # then computed similarly as previously done for single transcription factors:
>
> A_c <- vector("list", length(couples))
> B_c <- vector("list", length(couples))
> I_c <- matrix(0, length(couples), 2)
> I_c <- data.frame(I_c)
> I_c[,1] <- paste(couples)
> colnames(I_c) <- c('TF', 'imp')
> IMP_c <- vector("list", length(couples))
> # Compute rulesTF, rulesTF0 and IComp for each pair, avoiding pairs not
> # found in the r_TEAD4 set of rules
>
> for (i in 1:length(couples))  {
+         A_c[[i]] <- rulesTF(couples[i], r_TEAD4, FALSE)
+         if (all(!is.na(A_c[[i]][1]))){
+         B_c[[i]] <- rulesTF0(couples[i], A_c[[i]], r_TEAD4, MCF7_chr1, "TEAD4=1")
+           IMP_c[[i]] <- IComp(couples[i], A_c[[i]], B_c[[i]], figures=FALSE)$imp
+         }
+ }
> # Delete all NULL elements and compute the mean Importance Index of each pair
>
> null.indexes <- sapply(IMP_c, is.null)
> IMP_c <- IMP_c[!null.indexes]
> I_c <- I_c[!null.indexes,]
> I_c[, 2] <- sapply(IMP_c, mean)
>

> # Select rows with mean Importance Index different from NaN, then order I_c:
>
> I_c <- I_c[!is.na(I_c[,2]),]
> I_c_ord <- arrange(I_c, desc(imp))
> head(I_c_ord)

                TF      imp
1  GABPA=1,TCF12=1 28.08584
2 TCF12=1,ZNF217=1 28.08516
3  GABPA=1,NR2F2=1 27.20583
4 ZNF217=1,NR2F2=1 27.20583
5    GABPA=1,MAX=1 27.18724
6   MAX=1,ZNF217=1 27.18724

>

> # Construction of a vector in which mean Importance Index values of pairs
> # of transcription factors are represented.
> # These transcription factors are taken from the output of presAbs as
```

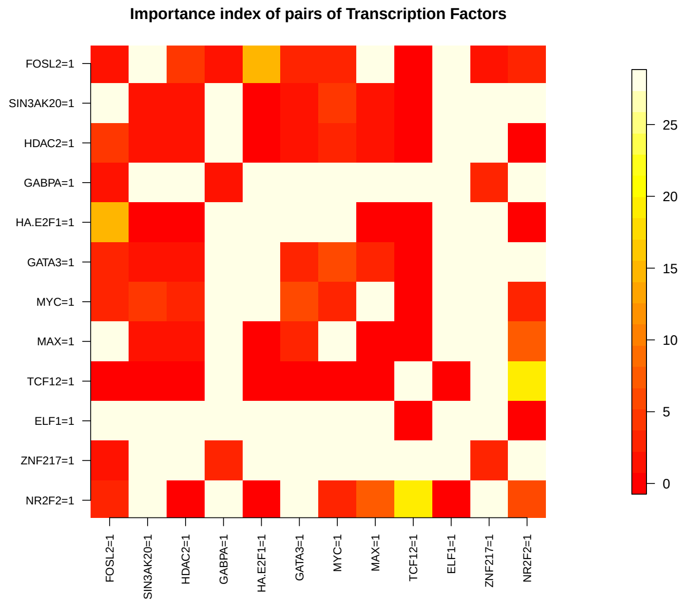

Figure 4: **Heatmap.** Mean importance of couples of candidate co-regulator transcription factors in the set of the 30 most relevant rules for the prediction of the presence of TEAD4 in promotorial regions of chromosome 1 in cell line MCF-7. The mean importances of single transcrption factors are rapresented in the main diagonal as in Table 6.

```
> # present in at least one association rules.
>
> # The function rbind is used to combine IMP columns and I_c_ord columns and
> # then the function arrange orders the data frame by the imp column.
>
> I_c_2 <- arrange(rbind(IMP[,1:2], I_c_ord), desc(imp))
> i.heat <- heatI(p, I_c_2)
>
```

To build the heatmap the user must also consider the single transcription factor mean importances (since the heatmap diagonal elements are the mean importances of single transcritpion factors).

The obtained heatmap is represented in Figure 4. The colour scale indicates that the lowest mean importances are represented in dark red, whereas the highest ones are represented in light white.

This rapresentation is useful to notice that, for example:

- ELF1 has high mean Importance Index alone and in couple with all the others candidate co-regulator transcription factors, except with NR2F2 and TCF12;
- HDAC2 has low mean Importance Index alone and in couple with all the others candidate co-regulator transcription factors, except with ELF, ZNF217 and GABPA.

References

[1] Christian Borgelt and Rudolf Kruse. Induction of association rules: Apriori implementation. In *Compstat*, pages 395–400. Springer, 2002.

[2] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *ACM sigmod record*, volume 22, pages 207–216. ACM, 1993.

[3] Marco Masseroli, Pietro Pinoli, Francesco Venco, Abdulrahman Kaitoua, Vahid Jalili, Fernando Palluzzi, Heiko Muller, and Stefano Ceri. Genometric query language: a novel approach to large-scale genomic data management. *Bioinformatics*, 31(12):1881–1888, 2015.

[4] Michael Hahsler, Christian Buchta, Bettina Gruen, and Kurt Hornik. *arules: Mining Association Rules and Frequent Itemsets*, 2016. R package version 1.5-0. URL: https://CRAN.R-project.org/package=arules.

[5] Michael Hahsler, Bettina Gruen, and Kurt Hornik. arules – A computational environment for mining association rules and frequent item sets. *Journal of Statistical Software*, 14(15):1–25, October 2005. URL: http://dx.doi.org/10.18637/jss.v014.i15.

[6] Michael Hahsler, Sudheer Chelluboina, Kurt Hornik, and Christian Buchta. The arules r-package ecosystem: Analyzing interesting patterns from large transaction datasets. *Journal of Machine Learning Research*, 12:1977–1981, 2011. URL: http://jmlr.csail.mit.edu/papers/v12/hahsler11a.html.

[7] Richard Arnold Johnson and Dean W Wichern. *Applied multivariate statistical analysis*. Number 8. Prentice hall Upper Saddle River, NJ, 2007.

[8] Rasmus Bro and Age K Smilde. Principal component analysis. *Analytical Methods*, 6(9):2812–2831, 2014.