TFARM: Transcription Factors Association Rules Miner

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

November 10, 2016

Contents

```
> library(TFARM)
```

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 synthetic, since they only provide a list of association rules. A novel method has been 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 binary matrix or data.frame in which columns identify transcription factors and rows represent genomic regions. The element (i,j) (with j >1) of this matrix is equal to 0 if transcription factor j is absent in region i, or to 1 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 unintresting regions. The first column of the dataset contains the chromosome of each region.
The dataset we consider here is obtained from the analysis of ENCODE ChIP-seq data: it concerns the localization of transcription factors binding sites and histon 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 25 transcription factors of the cell line MCF7 (i.e., all the transcription factors evaluated in ENCODE for this cell line), in the 2944 promotorial regions of chromosome 1:

```
> data("MCF7_chr1")
> dim(MCF7_chr1)
```

```
[1] 2944   26
```

```
> head(MCF7_chr1)
```

```
    chr PML SRF CTCF TCF7L2 FOSL2 SIN3AK20 HDAC2 EP300 GABPA EGR1 HA.E2F1 GATA3 REST
19 chr1   0   0    1      1     0        0     0     0     0    0       0     0    0
20 chr1   0   0    1      1     0        0     0     0     0    0       0     0    0
21 chr1   0   0    1      1     0        0     0     0     0    0       0     0    0
22 chr1   0   0    1      1     0        0     0     0     0    0       0     0    0
31 chr1   0   1    1      0     0        1     0     0     0    0       1     0    0
32 chr1   0   0    1      0     0        0     0     0     1    0       0     0    0
   FOXM1 MYC MAX TEAD4 CEBPB JUND RAD21 TAF1 TCF12 ELF1 ZNF217 NR2F2
19     0   0   0     0     0    0     0    0     0    0      0     0
20     0   0   0     0     0    0     0    0     0    0      0     0
21     0   0   0     0     0    0     0    0     0    0      0     0
22     0   0   0     0     0    0     0    0     0    0      0     0
31     0   1   1     0     0    0     0    1     0    1      0     0
32     0   1   0     0     0    0     0    0     0    0      0     0
```

3 The extraction of the most relevant associations

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

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

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 calling the `apriori` function of the *arules* package [4] [5] [6]. It takes in input:

- the indicator of presence matrix, without the chromosome column (first column of the Indicator of presence matrix);
- 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 right-hand-side of each extracted rule;
- in the second column the left-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 MCF7,
> # 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:
>
> m <- dim(MCF7_chr1)[2]
> r_TEAD4 <- rulesGen(MCF7_chr1[,2:m], "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     10  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 ...[24 item(s)] done [0.00s].
set transactions ...[24 item(s), 2944 transaction(s)] done [0.00s].
sorting and recoding items ... [24 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 done [0.22s].
writing ... [30 rule(s)] done [0.01s].
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 at least one region of the considered dataset;
- 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 <- colnames(MCF7_chr1)[2:m]
> 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"
```

```
> m <- 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[2:m], r_TEAD4, TRUE)$abs
> a

 [1] "SRF=1"    "CTCF=1"   "TCF7L2=1" "EP300=1"  "EGR1=1"   "REST=1"   "FOXM1=1"
 [8] "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 The 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 presence 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 evalute:

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 $\{\mathrm{R}^I\}$ (with J number of rules in $\{\mathrm{R}^I\}$, $J=|\{\mathrm{R}^I\}|$).
Each element of $\{\mathrm{R}^I{}_j\}_{j=1:J}$ 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 removing item I from 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

$\{TF2 = 1, TF3 = 1\} \rightarrow \{TFt = 1\}$ 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::

$$z_{s\,j}^I = \frac{s_j^I - \bar{s^I}}{S_s^I}, \quad z_{c\,j}^I = \frac{c_j^I - \bar{c^I}}{S_c^I}, \quad z_{l\,j}^I = \frac{l_j^I - \bar{l^I}}{S_l^I} \tag{5}$$

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_s^I, S_c^I, S_l^I 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: $\Delta z_{sj} = z_{s\,j}^I - z_s^{I^-}{}_j \qquad \Delta z_{cj} = z_{c\,j}^I - z_c^{I^-}{}_j \qquad \Delta z_{lj} = z_{l\,j}^I - z_l^{I^-}{}_j$

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:

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

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.

(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 to 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 unreveald 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 the considered rules
- a logical parameter, *verbose*: if *verbose* = *TRUE* a console message is returned if the searched subset of rules is empty.

The output of the function is a data.frame containing the subset of rules whose LHSs contain TFi, and the correspondent quality measures. Using the introduced notation, the output of the `rulesTF` function is $\{R^I{}_j\}_{j=1:J}$ with the quality measures $\{s^I{}_j,\ c^I{}_j,\ l^I{}_j\}_{j=1:J}$. 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 $s^I{}_j$, $c^I{}_j$, $l^I{}_j$ (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 remove I from all the left-hand-side patterns of R^I: this is done using the function `rulesNTF` 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;

it returns a data.frame with the rules obtained removing TFi and the corrispondent measures. Using the introduced notation, the output of the `rulesNTF` 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 <- rulesNTF('FOSL2=1', r_FOSL2, r_TEAD4)
> head(r_noFOSL2)

                                      lhs       rhs support confidence lift
1                    {GABPA=1,MYC=1,ZNF217=1} {TEAD4=1}       0          0    0
2            {GABPA=1,MYC=1,ZNF217=1,NR2F2=1} {TEAD4=1}       0          0    0
3            {HDAC2=1,GABPA=1,MYC=1,ZNF217=1} {TEAD4=1}       0          0    0
4              {GABPA=1,MYC=1,MAX=1,ZNF217=1} {TEAD4=1}       0          0    0
5    {HDAC2=1,GABPA=1,GATA3=1,MYC=1,ZNF217=1} {TEAD4=1}       0          0    0
6 {GABPA=1,HA.E2F1=1,GATA3=1,MYC=1,ZNF217=1} {TEAD4=1}       0          0    0

>
> # Since none of the rules in r_FOSL2 has been found in the set of rules r_TEAD4
> # once removed FOSL2, the three measures of all the obtained rules are set to zero.
```

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 `rulesNTF`)
- 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 three outputs:

- imp, wich is the set of importances index of TFi in the given set of rules (rules_TF)
- delta, wich is the matrix of variations of standardidez support, confidence and lift obtained removing TFi from rules_TF.
- the plot 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*.

```
> imp_FOSL2 <- IComp('FOSL2=1', r_FOSL2, r_noFOSL2, figures=TRUE)
> names(imp_FOSL2)

[1] "imp"   "delta"

> imp_FOSL2$imp
```

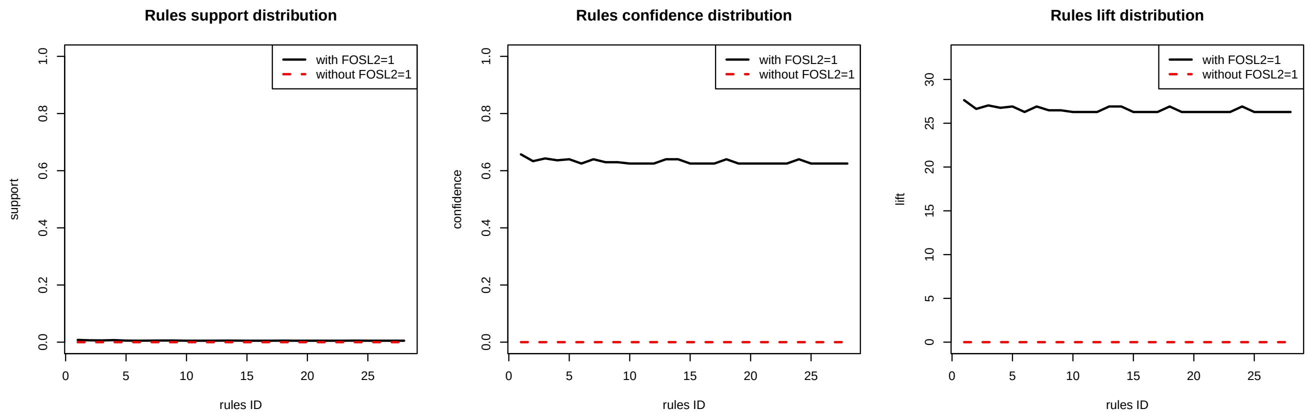

Figure 1: **Support, Confidence and Lift for the extracted rules before and after the removal of item I.** Left panel: Support distribution $\{s_j^I\}_{j=1:J}$, black thick line and $\{s_j^{I-}\}_{j=1:J}$,red dotted line. Middle panel: Confidence distribution $\{c_j^I\}_{j=1:J}$, black thick line and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

```
 [1] 28.30251 27.27598 27.68571 27.40713 27.56201 26.91581 27.56201 27.11583 27.11583
[10] 26.91581 26.91581 26.91581 27.56201 27.56201 26.91581 26.91581 26.91581 27.56201
[19] 26.91581 26.91581 26.91581 26.91581 26.91581 27.56201 26.91581 26.91581 26.91581
[28] 26.91581
```

```
> head(imp_FOSL2$delta)
```

```
  diff_supp_Z diff_conf_Z diff_lift_Z
1 0.007812500   0.6571429    27.63755
2 0.006453804   0.6333333    26.63619
3 0.006114130   0.6428571    27.03673
4 0.007133152   0.6363636    26.76364
5 0.005434783   0.6400000    26.91657
6 0.005095109   0.6250000    26.28571
```

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`, `rulesNTF` and `IComp` over a string vector with all the transcription factors present in the set of relevant association rules extracted.

```
> # For the considered example the user could run:
>
> library(plyr)
> A <- list()
> B <- list()
> IMP <- matrix(0, length(p), 4)
> IMP <- data.frame(IMP)
> IMP[,1] <- paste(p)
> colnames(IMP) <- c('TF', 'imp', 'sd', 'nrules')
> IMP_Z <- list()
> for (i in 1:length(p))  {
+         A[[i]] <- rulesTF(p[i], r_TEAD4, FALSE)
+         B[[i]] <- rulesNTF(p[i], A[[i]], r_TEAD4)
+         IMP_Z[[i]] <- IComp(p[i], A[[i]], B[[i]], figures=FALSE)$imp
+         IMP[i,2] <- mean(IMP_Z[[i]])
+         IMP[i,3] <- sqrt(var(IMP_Z[[i]]))
+         IMP[i,4] <- length(IMP_Z[[i]])
+ }
> IMP.ord <- arrange(IMP, desc(imp))
> IMP.ord

            TF        imp        sd nrules
1      TCF12=1 28.0858405 0.0000000      2
2    HA.E2F1=1 28.0177651 1.7275712      4
3       ELF1=1 27.7182593 2.5047172      8
4      GABPA=1 27.2366549 0.4202592     30
5     ZNF217=1 27.2366549 0.4202592     30
6      FOSL2=1 27.1759988 0.3639631     28
7        MYC=1 27.1759988 0.3639631     28
8      NR2F2=1  3.8864175 2.6407567     10
9      GATA3=1  3.1317165 2.5657938     22
10 SIN3AK20=1  2.5764108 1.8894671     12
11     HDAC2=1  2.2442937 1.6051963     15
12       MAX=1  0.7599249 0.9095241     14
```

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 and the number of rules in which each item 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

Importance Index of an item in an association rule has been defined as a linear combination of variations of the standardized support, confidence and lift of the rule, obtained removing the item from the left-hand-side of the association rule (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.

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, with variations of standardized distributions of support, confidence and lift
- a matrix with the mean importance of every canidate co-regulator transcription factor and the number of rules in which each of them appears.

It returns:

- 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 used 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

```
> DELTA <- list()
> for (i in 1:length(p)){
+ DELTA[[i]] <- IComp(p[i], A[[i]], B[[i]], figures=FALSE)$delta
+ }
> 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     18.1489458 2.97036423 1.6458589
Proportion of Variance  0.9661737 0.02588045 0.0079458
Cumulative Proportion   0.9661737 0.99205420 1.0000000

> i.pc$loadings

Loadings:
          Comp.1 Comp.2 Comp.3
delta z_s -0.563 -0.646  0.516
delta z_c -0.581 -0.134 -0.802
delta z_l -0.587  0.752  0.300

               Comp.1 Comp.2 Comp.3
SS loadings     1.000  1.000  1.000
```

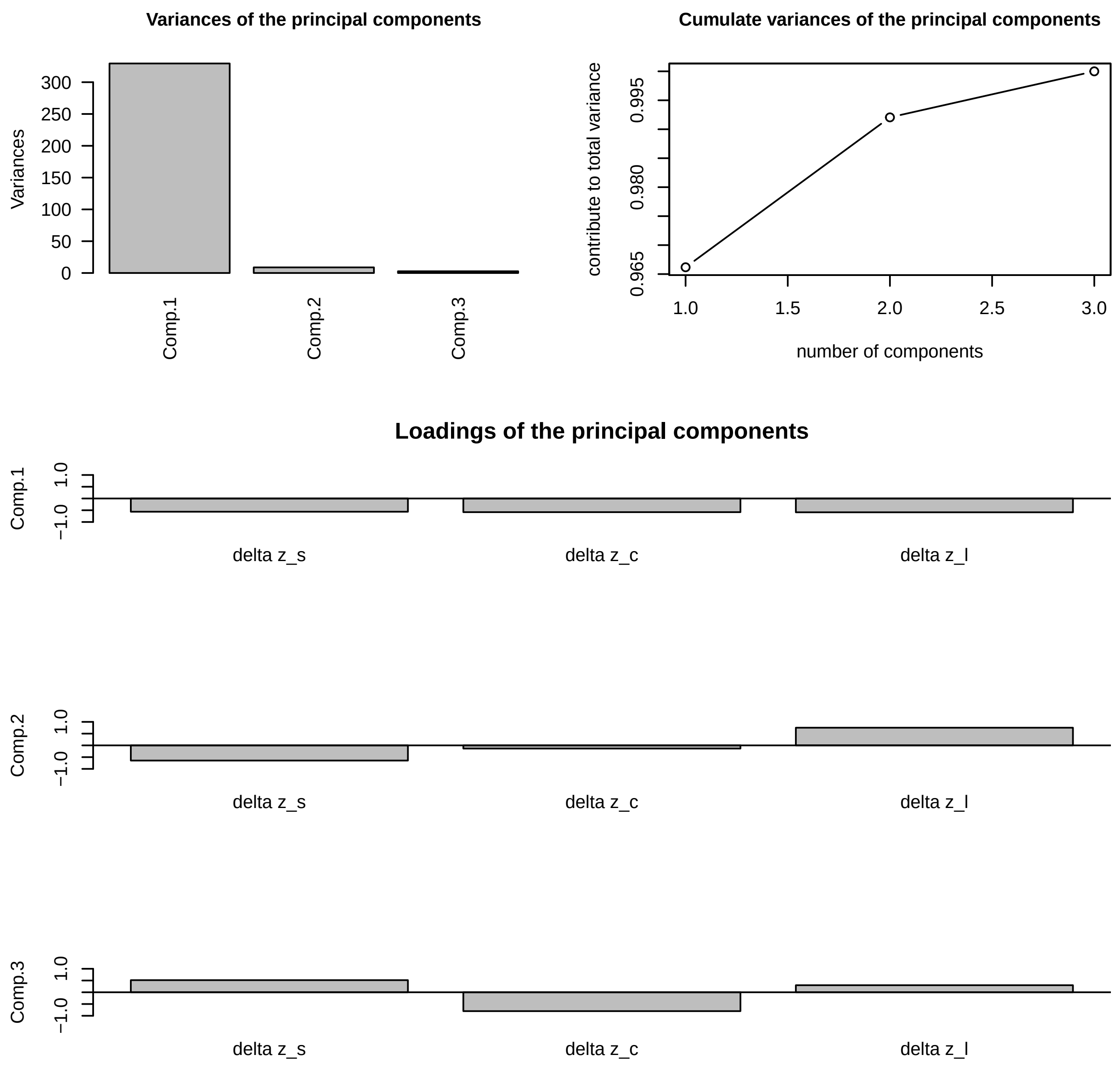

Figure 2: **Principal Component Analysis of Importance Index** Variances of each of the three principal components (on the left) and cumulate proportion of variance explained by each principal component (on the right). Loadings of the three principal components.

```
Proportion Var  0.333  0.333  0.333
Cumulative Var  0.333  0.667  1.000
```

As we can see looking at the plot in Figure 2, the first principal components explains the 96.61% 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.

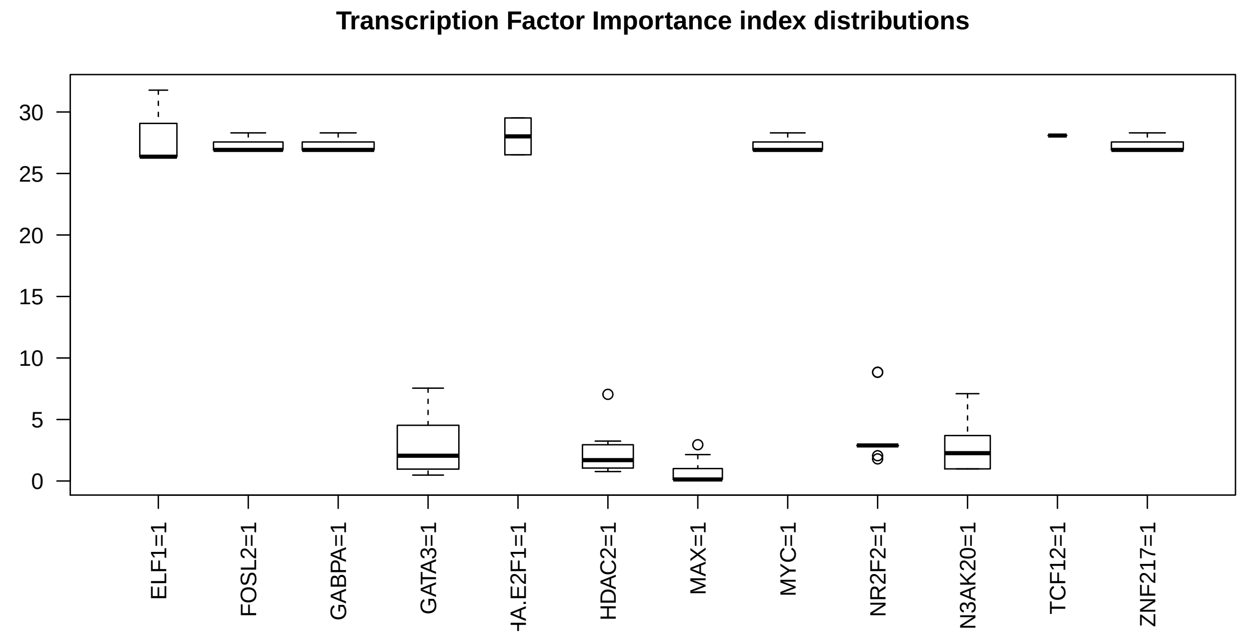

Figure 3: **Importance Index distribtion.** Importance Index distribution of candidate co-activators of TEAD4 in the set of 30 most relevant associations for the prediction of the presence of TEAD4 in promotorial regions of chromosome 1 in cell line MCF7.

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 a set 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)
```

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 is the median of the Importance Index distribution 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.

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

1. FOSL2, GABPBA, MYC, TCF12 and ZNF217 have the highest median and lowest variability of their Importance Index distribution; moreover FOSL2, GABPBA, MYC and ZNF217 appear in a similar number of relevant rules, while TCF12 appears in less relevant rules;
2. ELF1 and HA.E2F1 have high median, but higher variability of their Importance Index distribution than FOSL2, GABPBA, MYC, TCF12 and ZNF217;
3. GATA3, HDAC2, NR2F2 and SIN3AK20 have lower median and different variabilities of their Importance Index distribution;
4. MAX has the lowest median and low variability of its Importance Index distribution.

It can also be noticed that for the transcription factors HADAC2, MAX and NR2F2 there are some outlier rules, that are rules in which the Importance Index of the candidate co-regulator transcription factor is a lot higher than in the rest of the distribution.
These outliers can be extracted as reported in the following texttt:

```
> # Select the index of the list of importances IMP_Z
> # containing importance distributions of transcription factor HDAC2
> HDAC2_index <- which(p == 'HDAC2=1')
> # select outlier rules where HDAC2 has importance greater than 5
> o <- which(IMP_Z[[HDAC2_index]] > 5)
> rule_o <- A[[HDAC2_index]][o,]
> rule_o
                                                    lhs       rhs     support
6 {FOSL2=1,HDAC2=1,GABPA=1,GATA3=1,MYC=1,ELF1=1,ZNF217=1} {TEAD4=1} 0.005095109
  confidence     lift
6      0.625 26.28571

> # So, HDAC2 is very relevant in the pattern of transcription factors
> # {FOSL2=1,HDAC2=1,GABPA=1,GATA3=1,MYC=1,ELF1=1,ZNF217=1}
> # for the prediction of the presence of TEAD4.
>
> # To extract support, confidence and lift of the correspondent rule without HDAC2:
> B[[HDAC2_index]][o,]
                                            lhs       rhs   support confidence     lift
6 {FOSL2=1,GABPA=1,GATA3=1,MYC=1,ELF1=1,ZNF217=1} {TEAD4=1} 0.0078125  0.6571429 27.63755

>
> # Since none of the three measures of the rule obtained removing HDAC2 are equal to zero,
> # the rule {FOSL2=1,GABPA=1,GATA3=1,MYC=1,ELF1=1,ZNF217=1} -> {TEAD4=1} is
> # obtained removing HDAC2, 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*. 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*.

```
> couples_0 <- combn(p, 2)
> couples <- apply(couples_0, 2, function(x){
+         paste(x[1], x[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 of each couple is then computed as previously done
> # for single transcription factors:
>
> A_c <- list()
> B_c <- list()
> 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 <- list()
> for (i in 1:length(couples))  {
+         A_c[[i]] <- rulesTF(couples[i], r_TEAD4, FALSE)
+         B_c[[i]] <- rulesNTF(couples[i], A_c[[i]], r_TEAD4)
+         IMP_c[[i]] <- IComp(couples[i], A_c[[i]], B_c[[i]], figures=FALSE)$imp
+         I_c[i,2] <- mean(IMP_c[[i]])
+ }
> I_c <- I_c[which(!is.na(I_c[,2])),]
> I_c_ord <- arrange(I_c, desc(imp))
> head(I_c_ord)
```

```
                 TF      imp
1     GATA3=1,ELF1=1 28.20011
2  SIN3AK20=1,ELF1=1 28.17839
3 SIN3AK20=1,NR2F2=1 28.17839
4     HDAC2=1,ELF1=1 28.17839
5    GABPA=1,TCF12=1 28.08584
6      MAX=1,TCF12=1 28.08584
```

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

```
> I_c_2 <- arrange(rbind(IMP[,1:2], I_c_ord), desc(imp))
> heatI(p, I_c_2)
```

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

This rapresentation is useful to notice that, for example:

- GABPA=1 and ZNF217=1 have the higher mean importance taken as single items, and in couple with all the other transcription factors;
- MYC=1 has low mean importance with TCF12=1, but high mean importance with all the other transcription factors.

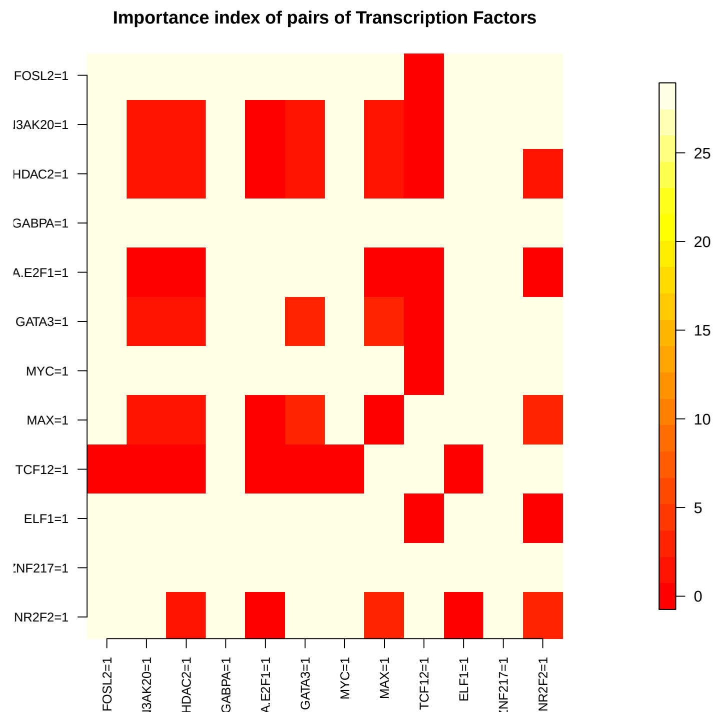

Figure 4: **Heatmap.** Mean importance of couples of candidate co-regulator transcription factors in the set of 30 most relevant rules for the prediction of the presence of TEAD4 in promotorial regions of chromosome 1 in cell line MCF7. The mean importances of single transcrption factors are rapresented in the main diagonal as in Table 6.

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, Dean W Wichern, et al. *Applied multivariate statistical analysis*, volume 5. Prentice hall Upper Saddle River, NJ, 2002.

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