TEPIC 2 - An extended framework for transcription factor binding prediction and integrative epigenomic analysis

TEPIC is a versatile framework for the analysis of transcription factor (TF) binding and offers several machine learning approaches for integrative analysis of predicted transcription factor binding sites (TFBS) and gene-expression data. Briefly, TEPIC offers:

- Annotation of user defined regions with TF affinities using TRAP and a variety of provided TF-motifs,
- Aggregation of TF affinities to TF-gene scores,
- Computation of statistical scores such as peak-length, peak-count or peak-signal per gene,
- Inclusion of long range chromatin contacts,
- Discretization of continuous TF affinities using a background distribution into a binary measure for TF-binding,
- Linear regression analysis to infer key transcriptional regulators within one sample,
- Logistic regression classifier to suggest key transcriptional regulators between samples,
- Generate input for DREM to infer important TFs from temporal epigenomic and gene expression data.

These points are illustrated in Figure 1. This document provides a brief introduction into the functionality of TEPIC and the machine learning approaches.

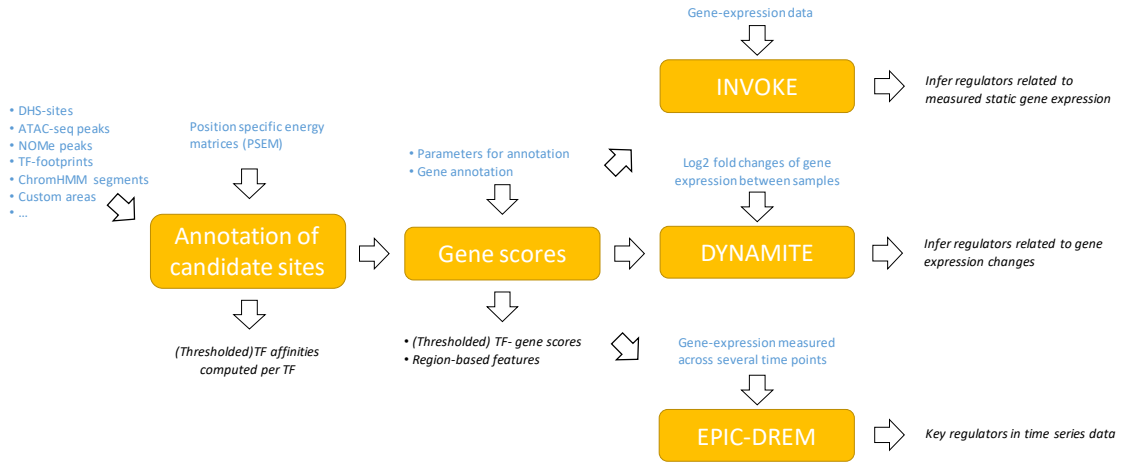

Figure 1: Overview on the workflows supported by TEPIC. Input to the submodules is shown in blue while the output is shown in black.

1 Introduction to TEPIC

1.1 Motivation

TF are essential players in transcriptional regulation. To understand their function, it is essential to know their binding sites genome-wide. Although TFBS can be inferred from ChIP-seq experiments, several in-silico approaches have been developed as well to overcome the burden and complexity of wet-lab experiments. Especially computational methods considering epigenetics data in the prediction have been used succesfully to predict TFBS. The main advantage of considering epigenetics data for the task of TF binding prediction is that the number of false positive predictions can be reduced [15]. One way of incorporating epigenetics data is to reduce the genomic search space to a few candidate regions of TF binding. As shown before, genome-wide candidate sites for TF binding can be determined by open-chromatin experiments [19, 6, 3, 10], e.g. peaks or footprints in DNase1-seq data, and/or by considering Histone marks [2, 6], e.g. H3K4me3.

Here, we compute TF affinities for currated sets of *Position Specific Energy Matrices (PSEMs)* using *TRAP* [16] which is based on a biophysical model of TF binding [18]. A major advantage of affinity based predictions compared to hit-based methods like Fimo [5] is that low-affinity binding sites can be included [17, 16]. Using the *TEPIC* method, we compute TF gene scores by aggregating TF predictions calculated for a user defined set of candidate regions. The scores, either per peak/region or gene, can be interpreted as a quantitative measurement of TF binding.

1.2 Collection of TF-motifs

We obtained *Position Count Matrices (PCMs)* from JASPAR [11], which is also including data from Uniprobe [7], HOCOMOCO [9] and the Kellis Lab ENCODE Motif database [8].

There are three folder containing Position specific energy matrices (PSEMs): Our current collection of PSEMs *PWMs/2.1*. The previously used motifs are provided in the folders *PWMs/2.0* and *PWMs/1.0*. TF motifs used in the original TEPIC manuscript are stored in the file *PWMs/1.0/pwm_vertebrates_jaspar_uniprobe_original.PSEM*.

In detail, the current collection contains from the *JASPAR 2018 Core* database:

- 579 PSEMs for vertebrates
- 176 PSEMs for fungi
- 26 PSEMs for nematodes
- 489 PSEMs for plants
- 1 PSEM for urochordates
- 133 PSEMs for insects

Additionally, we provide species specific collections of JASPAR matricies:

- 3 PSEMs for Antirrhinum majus
- 5 PSEMs for Arabidopsis lyrata
- 440 PSEMs for Arabidopsis thaliana
- 22 PSEMs for Caenorhabditis elegans
- 132 PSEMs for Drosophila melanogaster
- 1 PSEMs for Fragaria x ananassa
- 7 PSEMs for Gallus gallus
- 6 PSEMs for Glycine max
- 1 PSEM for Halocynthia roretzi
- 459 PSEMs for Homo sapiens

- 1 PSEM for Hordeum vulgare
- 1 PSEM for Medicago truncatula
- 1 PSEM for Meleagris gallopavo
- 157 PSEMs for Mus musculus
- 1 PSEM for Neurospora crassa
- 1 PSEM for Nicotiana
- 4 PSEMs for Orcytolagus
- 7 PSEMs for Oryza sativa
- 1 PSEM for Petunia x hybrida
- 1 PSEM for Phaeodactylum tricornutum
- 9 PSEMs for Physcomitrella patens
- 3 PSEMs for Pisum sativum
- 1 PSEM for Populus trichocarpa
- 2 PSEMs for Rattus norvegicus
- 2 PSEMs for Rattus rattus
- 176 PSEMs for Saccaromyces cerevisiae
- 2 PSEMs for Solanum lycopersicum
- 1 PSEM for Triticum aestivum
- 4 PSEMs for Xenopus laevis
- 8 PSEMs for Zea mays

All JASPAR matrices can be found in *PWMs/2.1/JASPAR_PSEMs*

From HOCOMOCO we provide 402 motifs for homo sapiens and 358 for mus musculus, available in *PWMs/2.1/HOCOMOCO_PSEMs*

The Kellis set contains 58 motifs, stored in *PWMs/2.1/Kellis_PSEMs*.

Additionally we provide non-redundant collections for homo sapiens and mus musculus considering motifs from all three sources:

- 561 PSEMS for homo sapiens
- 380 PSEMs for mus musculus

The matrices are stored in the folder *PWMs/2.1/Merged_PSEMs*

Furthermore, we used a motif clustering approach (7), to merge similar motifs of the files containing matrices from all three sources. This lead to

- 483 PSEMs for homo sapiens
- 306 PSEMs for mus musculus

The matrices are stored in the folder *PWMs/2.1/Clustered_PSEMs*

Files holding the length of the PSEMs are provided too.

1.3 Converting position count matrices to position specific energy matrices

As mentioned above, *TRAP* computes TF affinities that are based on a biophysical model of TF binding. Therefore *PCMs* have to be converted to *Position Specific Energy Matrices (PSEMs)* such that they can be used in *TRAP*. Intuitively, *PSEMs* represent the mismatch energy of a given motif. For a detailed explanation and motivation of the energy based score, please check [16]. A *PCM* M is converted to a *PSEM* E according to:

$$E_{i,j} = \frac{1}{\lambda} log(\frac{M_{max,j}}{M_{i,j}} b_{i,j}), \tag{1}$$

$$M_{max,j} = \max_{i \in \{A,C,G,T\}} (M_{i,j}). \tag{2}$$

The parameter λ is used for scaling the mismatch energies and $b_{i,j}$ denotes the background frequency of the nucleotide i with respect to the most frequent nucleotide at position j. This conversion formula is part of the mismatch energy postulated in formula (4) in [16]. By definition, if $j = max$, than $E_{i,j} = 0$, as there should be no mismatch energy for the best possible sequence match. Note that, during conversion, a pseudo count $pc = 1$ is added to each $M_{i,j}$.

The conversion is done by a C++ tool provided by the authors of *TRAP*. This is also included in the *TEPIC* repository. As suggested in [16], we use the following parameters for the conversion:

- $\lambda = 0.7$
- $m = 0.584$
- $n = -5.66$

The parameters *slope* m and *intercept* n are used to compute a matrix specific parameter R_0 that combines the concentration of the corresponding TF and the equilibrium constant of the binding reaction with its optimal binding site as defined in [16]. The authors of *TRAP* found a linear approximation for R_0 with:

$$ln(R_0) = m * |M| + n, \tag{3}$$

where $|M|$ denotes the length of the *PCM* as above.

Further, we exploit species specific GC-content values:

- *homo sapiens* $= 0.41$
- *mus musculus* $= 0.42$
- *rattus norvegicus* $= 0.42$
- *drosophila melanogaster* $= 0.43$
- *caenorhabditis elegans* $= 0.36$

In all other cases, a default GC-content of 0.42 is used.

1.4 Computing TF gene scores

Using our collections of *PSEMs*, *TRAP* computes TF binding affinities in all user provided regions that could be found in the reference genomes of the respective species and overlap with a window of user defined size w that is centered at the most $5'$ TSS of all annotated genes in the considered organism. Then, TF-gene scores are computed by incorporating all candidate binding sites within the window centered around the $5'$ TSS of genes in the final score. The contribution of the individual sites is weighted by their distance to the selected TSS with an exponential decay function [14]. Formally, the TF gene score $a_{g,i}$ for gene g and TF i is computed as

$$a^w_{g,i} = \sum_{p \in P_{g,w}} a_{p,i} e^{-\frac{d_{p,g}}{d_0}}, \tag{4}$$

where $a_{p,i}$ is the affinity of TF i in peak p, the set $P_{g,x}$ contains all open-chromatin peaks in a window of size w around gene g, $d_{p,g}$ is the distance from the center of peak p to the TSS of gene g, and d_0 is a constant fixed at 5000bp [14]. Additionally, affinities can be normalized by peak(and motif)-length during the computation of gene-TF scores:

$$a_{g,i}^{w} = \sum_{p \in P_{g,w}} \frac{a_{p,i}}{|p| - |m|} e^{-\frac{d_{p,g}}{d_0}}, \tag{5}$$

where $|p|$ is the length of peak p, $|m_i|$ is the length of the motif of TF i, with an extra count of 1. If the signal within a peak should be directly considered in the gene-TF score, we compute:

$$a_{g,i}^{w} = \sum_{p \in P_{g,w}} \frac{a_{p,i}}{|p| - |m|} s_p e^{-\frac{d_{p,g}}{d_0}}, \tag{6}$$

where s_p is the per base signal in peak p. This computation can be done with and without length normalisation of the affinities. The workflow of TEPIC is depicted in Figure 2.

In addition to the TF gene scores, TEPIC can compute features for peak length (pl_g), peak count (pc_g), and peak signal(ps_g) following the same scoring formulation as for TF affinities:

$$pl_g = \sum_{p \in P_{g,w}} |p| e^{-\frac{d_{p,g}}{d_0}}, \tag{7}$$

$$pc_g = \sum_{p \in P_{g,w}} e^{-\frac{d_{p,g}}{d_0}}, \tag{8}$$

$$ps_g = \sum_{p \in P_{g,w}} s_p e^{-\frac{d_{p,g}}{d_0}}, \tag{9}$$

where $|p|$ is the length of p. These features can be used for example to assess the influence of chromatin accessibility on gene expression without considering TF binding predictions.

Furthermore, TEPIC can compute a TF-specific affinity cut-off derived from either user-defined, or randomly generated sequences, to distinguish likely bound sites from unbound sites. These scores can be used to come-up with a binary TF-gene assignment. Further details on this mode are provided in Section 5.

With version 2.2 of TEPIC, we introduced support for the inclusion of long range chromatin conformation capture data. In addition to the promoter centric windows used before, we calculate TF affinities $a_{g,i}*$ and peak scores pl_g*, pc_g*, ps_g* for all DHSs residing in genomic loci looping into the promoter region of a gene, summarizied in P_{g,V_g}, where V_g is the set of all regions looped into the promoter region of gene g:

$$a_{g,i} = \sum_{p \in P_{g,V_g}} a_{p,i}, \tag{10}$$

$$pl_g* = \sum_{p \in P_{g,V_g}} |p|, \tag{11}$$

$$pc_g* = \sum_{p \in P_{g,V_g}} , \tag{12}$$

$$ps_g* = \sum_{p \in P_{g,V_g}} s_p. \tag{13}$$

Note that scores computed for $p \in P_{g,V_g}g$ are never considering the exponential decay as a direct interaction of the respective sites with the promoter region of gene g has been determined by chromatin conformation capture experiments.

1.5 Required input

To compute TF gene scores, a user needs to specify:

- a reference genome (-g option),

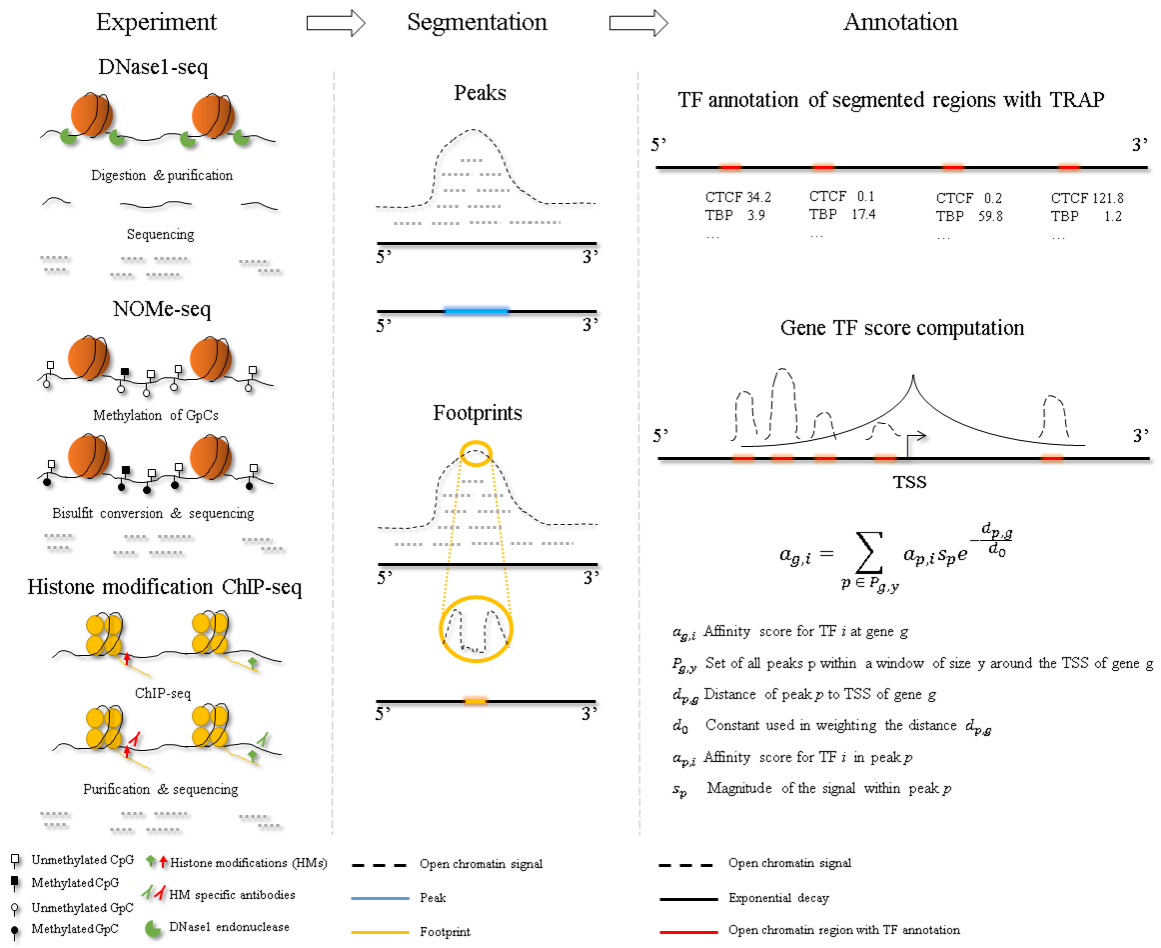

Figure 2: The general workflow of *TEPIC* is as follows: Data of an open-chromatin or Histone modification ChIP-seq experiment needs to be preprocessed to generate a genome segmentation, either by peak for footprint calling. Using the segmentation, *TEPIC* applies *TRAP* in all regions of interest, and computes TF gene scores using exponential decay to reweigh TF binding predictions in open-chromatin regions based on their distance to a genes TSS.

- a set of *PSEMs* (-p option),
- a set of genomic regions in BED format (-b option).
- a gtf file containing the genome annotation (-a option).

Note that the chromosome identifiers in the BED file must match the identifiers used in the reference genomes. Otherwise they can not be considered. Special care should be taken for *caenorhabditis elegans*, as Roman digits are used for enumeration of chromosomes.

1.6 Output

TEPIC outputs:

1. TF affinities for all selected *PSEMs* in the regions provided by the user that passed the filtering step (*_Affinity.txt*).
2. (Length normalized) TF gene scores for all selected *PSEMs* calculated as described above (optionally including peak features) (*_Affinity_Gene_View.txt*).
3. A meta data file listing all used parameters (*amd.tsv*).

4. Optionally a seperate file containing the signal information in peaks (*_Peak_Coverage.txt*).
5. TF affinities with all values below an inferred threshold set to zero (*_Thresholded_Affinity.txt*)
6. A sparse representation linking TF to genes (*_Sparse_Affinity_Gene_View.txt*)

2 Identification of key transcriptional regulators using epigenetics data (INVOKE)

Epigenetics data contains a wealth of information on gene regulation. It was shown that especially data on open-chromatin is well suited to build predictive models of gene-expression [17, 13, 1, 12]. Interpreting these models allows the inference of regulators that may play a key role in gene-expression regulation.

Hear, we offer an integrated analysis of epigenetics data, e.g. open-chromatin data (DNase1-seq, ATAC-seq, NOMe-seq) and gene-expression data to suggest key transcriptional regulators in the analysed sample.

Note that, although incorporating epigenetic data greatly improved the performance of TF binding predictions, both computing TF binding predictions and linking TFs to genes are still unsolved problems and all predictions should be seen as suggestions and not as the absolute truth.

The *INVOKE* analysis is split up into two main steps.

1. Computing TF gene scores on the basis of epigenetic data using *TEPIC* (see above).
2. Learning a linear regression model to predict gene expression from TF gene scores computed in (1).

2.1 Linear regression to predict gene expression

2.1.1 Motivation

In order to learn about potentially important regulators, we build a linear, interpretable regression model, comparable to methods proposed in [17, 13, 1, 12]. Here, we use TF gene scores computed with *TEPIC* as features in a linear regression setup to predict gene expression. In such a *per sample* approach, we stick to the simplifying assumption that all genes are regulated similarly. Features with a high regression coefficient can be suggested to be key regulators in the analysed sample, as they seem to affect the expression of a large portion of the genes under consideration. However, the results of this method should be seen as suggestions for possible regulators and not as the absolute truth.

Details on the learning setup and on the available regularization methods are provided in the next section.

2.1.2 Available regularization methods

We offer three different regularization techniques:

- Lasso:

$$\hat{\beta} = \arg\min_{\beta} ||y - X\beta||^2 + ||\beta||, \tag{14}$$

- Ridge:

$$\hat{\beta} = \arg\min_{\beta} ||y - X\beta||^2 + ||\beta||^2, \tag{15}$$

- Elastic net:

$$\hat{\beta} = \arg\min_{\beta} ||y - X\beta||^2 + \alpha||\beta||^2 + (1-\alpha)||\beta||, \tag{16}$$

where, β represents the regression coefficient vector, $\hat{\beta}$ represents the estimated coefficients, X is the feature matrix, y is the response vector, and the parameter α controls the distribution between Ridge and Lasso penalty in the elastic net.

Using Lasso regularization, models are sparse and can be learned very fast. But, Lasso cannot

properly deal with correlated features, e.g. instead of distributing the coefficients among them, only one is selected. Also, Lasso solutions are not stable and therefore should be interpreted with caution. Nevertheless, Lasso regularization is good to get a first impression of model performance.

The disadvantage of Ridge regression is that it cannot produce sparse models (many coefficients being exactly 0), which may hinder interpretability.

Elastic net regularization was designed to overcome the limitations of both regularization techniques mentioned above. It resolves the correlation between features by distributing the feature weights among them, and simultaneously leads to sparse and stable models [20]. However, learning a model using elastic net penalty is slower than using either only Lasso or Ridge regularization.

2.1.3 Details on the learning setup

The data matrix X, containing TF gene scores, and the response vector y, containing gene expression values, are log-transformed, with a pseudo-count of 1, centered and scaled to fit them as. Regression coefficients are computed in an inner cross validation, the α parameter of elastic net regularization is optimized with a default step size of 0.1.

We offer two ways to use our learning pipeline:

1. Learn a model for feature interpretation without computing performance measures: In order to provide a time efficient way of obtaining an interpretable model and to prevent a potential loss of information by considering only a portion of the full data set for model training, the regression coefficients are determined on the entire data set.

2. Learn a model for feature interpretation and compute model performance: Nested cross-validation is used to learn the models and to assess their performance. Per default, 20% of the data are used as test data and 80% are used as training data. Model performance is assessed in an outer cross validation. We report the mean pearson correlation, the mean spearman correlation, and the mean squared error over the outer folds as measures of model performance. Additionally, a model is learned on the entire data set as described in (1) for interpretation of the coefficients.

All parameters mentioned in this section can be changed by the user. The learning process is sketched in Figure 3.

2.1.4 Required input

In addition to the input required for the computation of TF gene scores in TEPIC, a file containing gene expression data must be provided. This file should be structured such that column 1 contains the gene identifiers and column 2 holds expression values.

2.1.5 Output and hints for interpretation

The user is always provided with the following files:

- a list of regression coefficients computed on the entire data set,
- a bar plot showing the regression coefficients with an absolute value > 0.025.

The larger a regression coefficient, the stronger is the inferred effect of the corresponding TF on gene expression. Positive coefficients suggest an activating influence of TFs, negative coefficients suggest an inhibiting effect.

If model performance was assessed, the following is available in addition:

- a summary on model performance containing the aforementioned measures (pearson correlation, spearman correlation, mean squared error),
- a list of regression coefficients determined in the outer cross validation,
- a heatmap visualizing the regression coefficients determined in the outer cross validation for at most the top 10 positive and negative features, sorted according to their median.
- an image showing a box plot for pearson and spearman correlation respectively.

- scatter plots showing the predicted vs the measured gene expression for each outer cross validation fold.

The heatmap can be easily used to judge model performance, as it shows the regression coefficients of all outer-cross validation runs. The box plots provide further insights into model performance and stability across the outer folds of the cross validation.

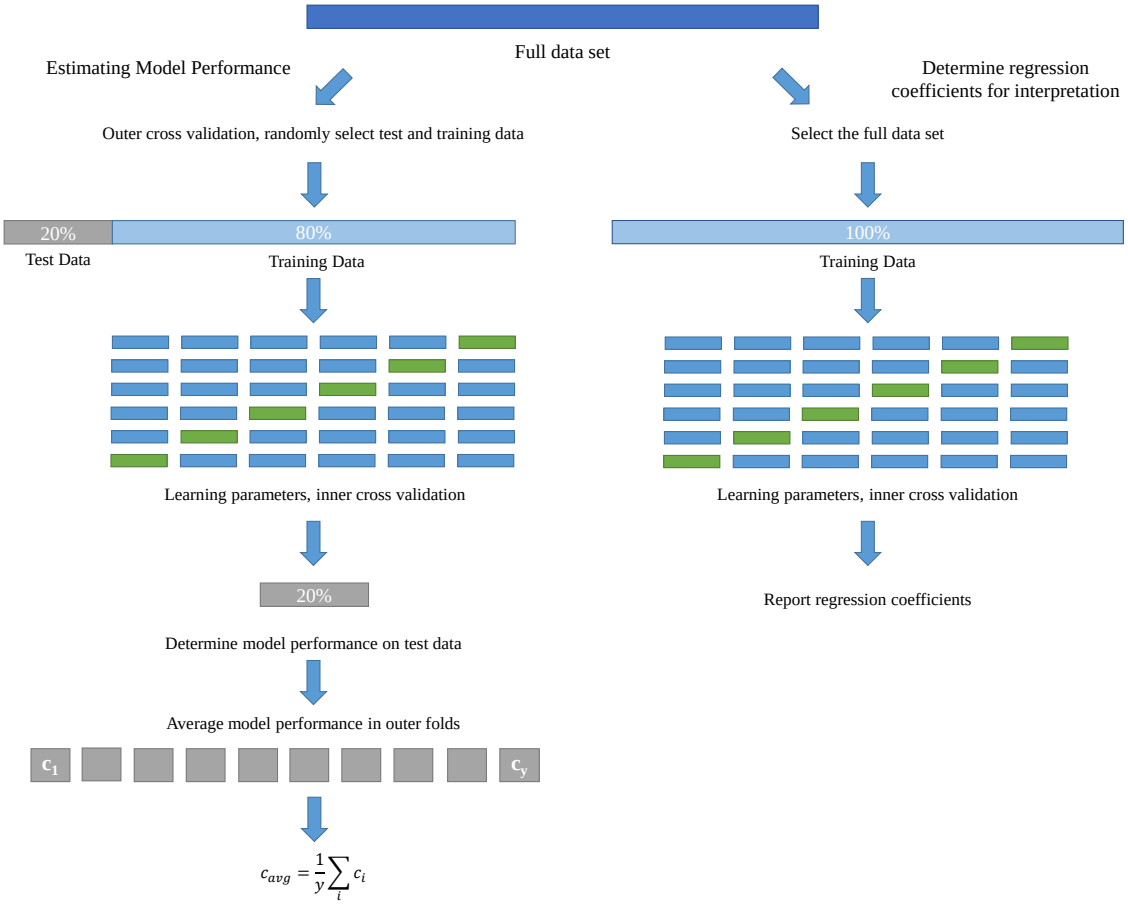

Figure 3: Overview of the learning process. The left part of the Figure describes the assessment of model performance in a y-fold outer cross validation and 6-fold inner cross validation. The right hand site illustrates model training on the entire data set, again using a 6-fold inner cross validation for parameter learning.

3 Differential analysis to identify novel transcriptional regulators for differentially expressed genes (DYNAMITE)

Although a variety of methods have been proposed to generate genome-wide TF binding predictions [[17, 3, 10, 6]] and to establish *TF to tissue* associations [[13, 14, 17]], systematic, feasible, and easy to use ways of linking TFs to distinct genes are rare.

In addition to the *INVOKE* analysis, we propose a method to infer the most likely transcriptional regulators for a set of differentially expressed genes. We use TF scores, computed using *TEPIC*, and logistic regression to identify TFs that have explanatory power to distinguish between up- and down-regulated genes.

3.1 Input

To run *DYNAMITE*, a user most provide candidate regions of TF binding for two groups of samples, A and B, e.g. control and disease. These can be derived, for example, by open chromatin experiments such as DNase-seq. It is essential that the candidate regions reflect the characteristics of chromatin organization in the analysed tissues. In addition, a list of differential expressed genes between two groups as well as log2 fold changes of the expression are needed.

Method

Our method consists of two parts: (1) gene-TF score computation, and (2) identification of key TFs.

Step 1: Computing Gene-TF Scores

Using TEPIC, we compute gene-TF scores g_{ij} for all differentially expressed genes i and distinct TFs j considering the provided candidate regions for all replicates a of group A and for all replicates b of group B. As a result, gene-TF matrices M_k for all replicates of both groups are obtained. To account for biological variation among the replicates, we compute two matrices M_A, M_B holding the mean gene-TF scores among all replicates of a group, where

$$M_{A_{ij}} = \frac{\sum_{a \in A} M_{a_{ij}}}{|A|}, \tag{17}$$

$$M_{B_{ij}} = \frac{\sum_{b \in B} M_{b_{ij}}}{|B|}. \tag{18}$$

Using matrices M_A and M_B we compute a matrix R_{AB} that holds the ratios of gene-TF scores for all genes and all TFs:

$$R_{AB_{ij}} = \frac{M_{A_{ij}}}{M_{B_{ij}}}. \tag{19}$$

Thus, R_{AB} represents the changes in TF binding between groups A and B on a gene level. The feature computation is sketched in Figure 4.

3.2 Step 2: Identification of Key Transcription Factors

To identify those TFs that can explain the differential expression state of as many genes as possible, we build a logistic regression classifier. We use matrix R_{AB} computed in Step 1 as the feature matrix X, and a binary vector of gene expression changes as response y. An example is shown in Figure 5. We perform logistic regression with elastic net regularisation (11). As above, we tune the parameter α that distributes the weight between lasso and ridge penalty in a grid search with user defined step-size between 0 and 1.

Model parameters are learned in an inner cross validation, while the accuracy of our classifier can be assessed through an outer cross validation. This is the same learning paradigm that is described for the *INVOKE* analysis (Figure 3). We use the entire dataset for model training and to interpret the regression coefficients. TFs that correspond to features with a non-zero regression coefficient can be seen as being essential to explain the observed expression differences and should be further investigated.

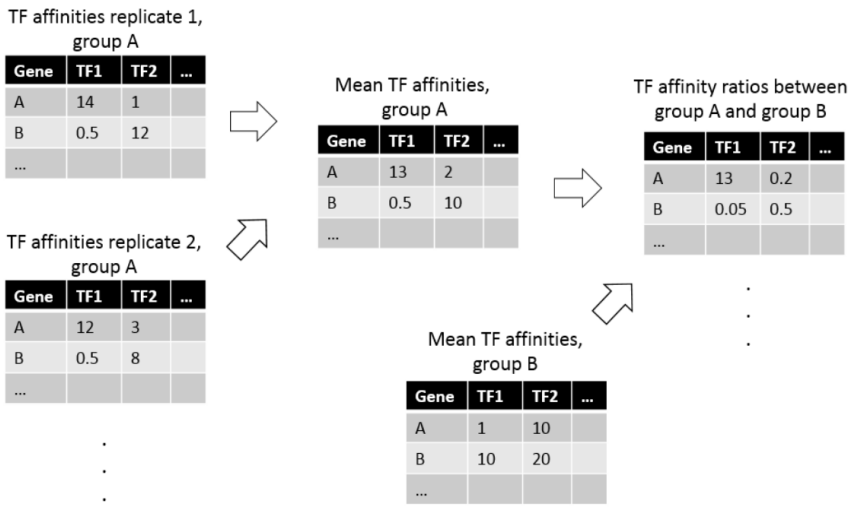

Figure 4: Computation of differential TF features between two groups.

Gene	Expression Changes	TF1	TF2	...
A	Up	1.2	3.9	
B	Down	4.2	0.7	
C	Down	0.8	1.7	
D	Up	0.4	1.6	
E	Up	1.0	1.2	
...				

Figure 5: Example for a matrix used as input to the logistic regression. The column *Expression Changes* is used as response, while the affinity ratios *TFx* are used as features.

4 Output

Model performance is reported in a *txt* file and visually in a bar plot using mean test and training accuracy as well as the F1 measure. A heatmap shows the regression coefficients in the outer cross validation folds. Additionally, we report confusion matrices for the outer cross validation folds. We generate a bar-plot with the regression coefficients of all TFs selected in the final model. A positive coefficient is used by the model to predict genes as upregulated, a negative coefficient is related to genes that are predicted as downregulated. The interpretation of the model can be simplified if the user makes sure that both TF ratios and gene expression fold changes are computed in the same order.

We provide an additional script to generate further plots per feature that can help to understand the model. As shown in Figure 6, density plots, and scatter plots are generated to help elucidating why a particular feature was selected by the model.

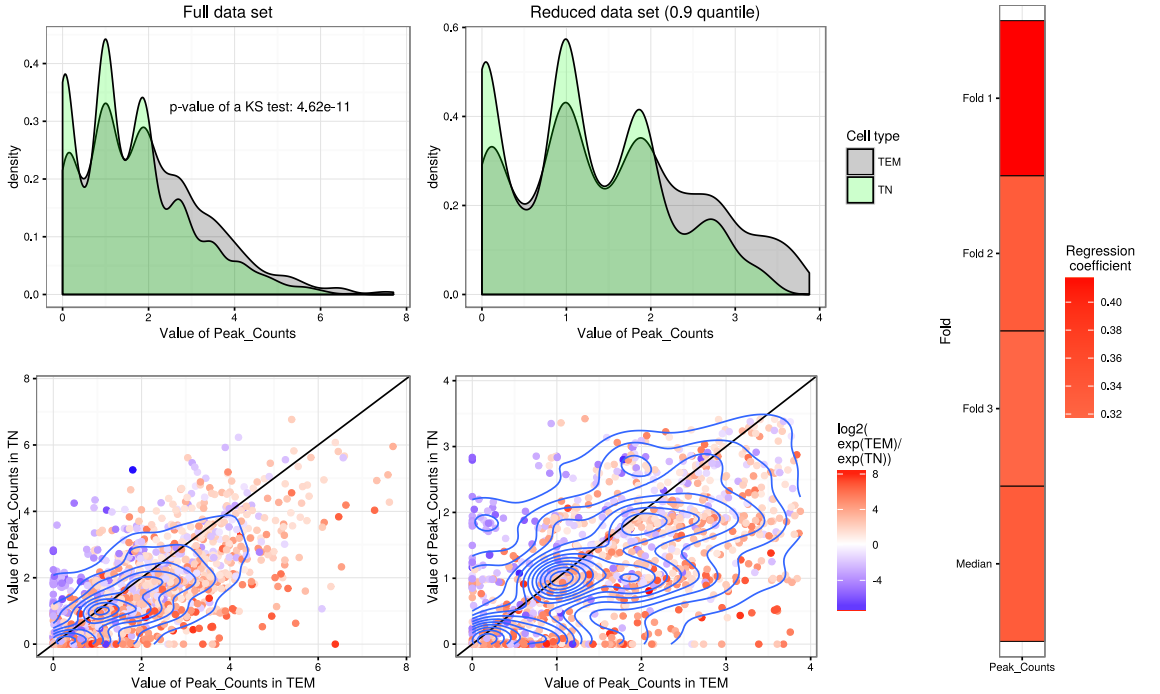

Figure 6: Example for an automatically created feature analysis Figure generated on the example data provided in the repository. The density plots show the distribution of TF affinities, the scatter plot relates the TF affinities to the observed expression changes. The miniature heatmap shows the regression coefficients determined during the outer cross validation.

5 Determine important transcriptional regulators from time serires data (EPIC-DREM)

EPIC-DREM is a combination of *TEPIC* and the *Dynamic Regulatory Events Miner (DREM)* [4]. Instead of using static ChIP-seq data, which is provided in *DREM 2.0*, we suggest to use time-point specific TF binding predictions based on time-dependent epigenomic profiles. Thereby, *DREM* can infer regulators that can be linked to expression changes at distinct points in time. We have shown that using the predicted, dynamic TF binding events is superior to the static data included in *DREM*.

In order generate a sparse input matrix for *DREM*, we devised a strategy to threshold TF affinities based on a set of background sequences. These can be either chosen automatically or be provided by the user. Please check the README file for detailed options.

In Figure 7, we illustrate how TF affinities can be discretized and illustrate their usage in *DREM*. Note that *DREM* is not included in the *TEPIC* repository. It is available online.

5.1 Thresholded TF affinities

In some applications it is required to make a binary decision whether a factor is binding or not. To infer this information from TF affinities, *TEPIC* allows the computation of a TF specific affinity threshold by calculating TF affinities on a randomly selected set of genomic regions. When selected by TEPIC, these regions show similar characteristics compared to the provided regions (GC content and length). Alternatively, a set of background regions can be provided by the user. By applying a user defined p-value on the distribution of affinities computed on the random regions, a threshold is chosen. Per TF, all affinities that are smaller than the selected threshold, are set to zero, thus a sparse matrix with TF-gene interactions can be generated.

5.1.1 Required input

In addition to the input mentioned above, a reference genome in 2bit format is required. Optionally, the user can provide a bed file containing background regions. These replace the automated generation

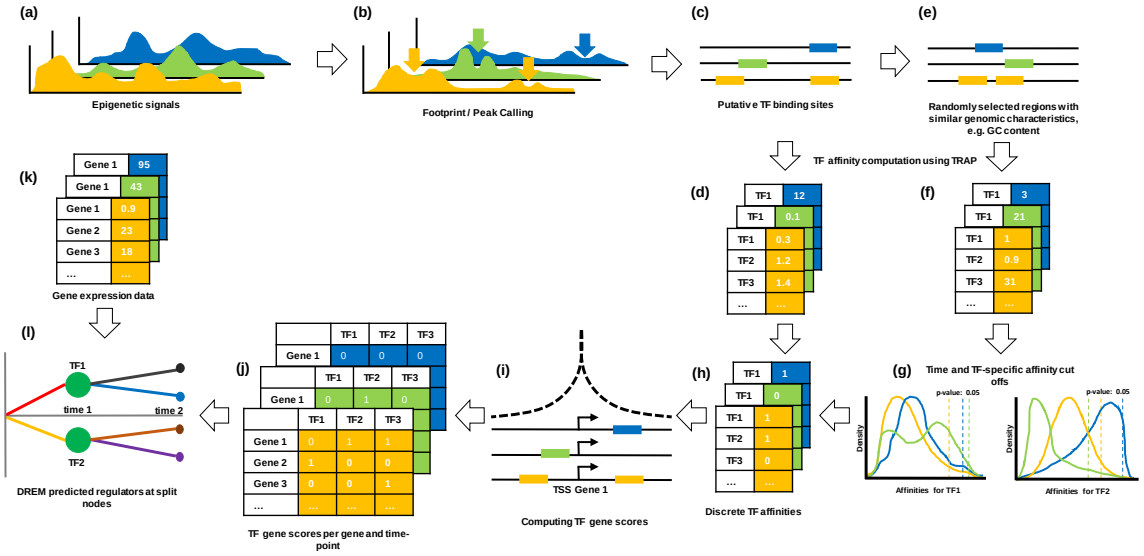

Figure 7: Overview on the EPIC-DREM approach: Note that in this Figure, different time points are indicated by different colors. First, epigenetic data, e.g. DNase1 experiments, are conducted for different points (a). Next, putative TF binding sites are identified by peak and/or footprint calling (b,c) and annotated with TF affinities (d). From the putative binding sites, a random set of genomic regions is chosen (e) and annotated with TF affinities as well (f). By applying a p-value cut-off on the distribution of TF affinities calculated on the random regions (g), a suitable, TF specific affinity threshold is chosen to discretize the original TF affinities (h). Using the default TEPIC TF-gene score formulation (i), a TF-gene interaction matrix (j) is computed. Together with gene expression data (k), the sparse matrix (j) can be used as input for DREM (l) to identify potential key regulators of expression changes in time series data.

of background sequences.

5.1.2 Output

The following output files are generated in addition:

1. TF affinities for all selected *PSEMs* in the regions provided by the user that passed the filtering step, where all affinities below the TF specific thresholds are set to 0.
2. (Length normalized) TF gene scores for all selected *PSEMs* calculated as described above (optionally including peak features) using the thresholded affinities.
3. A sparse representation of TF gene interactions.

Either (2) or (3) can be combined with RNA-seq data and used as input for *DREM*.

References

[1] D. M. Budden, D. G. Hurley, and E. J. Crampin. Predictive modelling of gene expression from transcriptional regulatory elements. *Brief. Bioinformatics*, 16(4):616–628, Jul 2015.

[2] D. M. Budden, D. G. Hurley, J. Cursons, J. F. Markham, M. J. Davis, and E. J. Crampin. Predicting expression: the complementary power of histone modification and transcription factor binding data. *Epigenetics Chromatin*, 7(1):36, 2014.

[3] G. Cuellar-Partida, F. A. Buske, R. C. McLeay, T. Whitington, W. S. Noble, and T. L. Bailey. Epigenetic priors for identifying active transcription factor binding sites. *Bioinformatics*, 28(1):56–62, Jan 2012.

[4] J. Ernst, O. Vainas, C. T. Harbison, I. Simon, and Z. Bar-Joseph. Reconstructing dynamic regulatory maps. *Mol. Syst. Biol.*, 3:74, 2007.

[5] Charles E. Grant, Timothy L. Bailey, and William Stafford Noble. Fimo: Scanning for occurrences of a given motif. *Bioinformatics*, 27(7):1017–1018, 2011.

[6] E. G. Gusmao, C. Dieterich, M. Zenke, and I. G. Costa. Detection of active transcription factor binding sites with the combination of DNase hypersensitivity and histone modifications. *Bioinformatics*, 30(22):3143–3151, Nov 2014.

[7] M. A. Hume, L. A. Barrera, S. S. Gisselbrecht, and M. L. Bulyk. UniPROBE, update 2015: new tools and content for the online database of protein-binding microarray data on protein-DNA interactions. *Nucleic Acids Res.*, 43(Database issue):D117–122, Jan 2015.

[8] P. Kheradpour and M. Kellis. Systematic discovery and characterization of regulatory motifs in ENCODE TF binding experiments. *Nucleic Acids Res.*, 42(5):2976–2987, Mar 2014.

[9] I. V. Kulakovskiy, Y. A. Medvedeva, U. Schaefer, A. S. Kasianov, I. E. Vorontsov, V. B. Bajic, and V. J. Makeev. HOCOMOCO: a comprehensive collection of human transcription factor binding sites models. *Nucleic Acids Res.*, 41(Database issue):195–202, Jan 2013.

[10] K. Luo and A. J. Hartemink. Using DNase digestion data to accurately identify transcription factor binding sites. *Pac Symp Biocomput*, pages 80–91, 2013.

[11] A. Mathelier, O. Fornes, D. J. Arenillas, C. Y. Chen, G. Denay, J. Lee, W. Shi, C. Shyr, G. Tan, R. Worsley-Hunt, et al. JASPAR 2016: a major expansion and update of the open-access database of transcription factor binding profiles. *Nucleic Acids Res.*, 44(D1):D110–115, Jan 2016.

[12] R. C. McLeay, T. Lesluyes, G. Cuellar Partida, and T. L. Bailey. Genome-wide in silico prediction of gene expression. *Bioinformatics*, 28(21):2789–2796, Nov 2012.

[13] A. Natarajan, G. G. Yardimci, N. C. Sheffield, G. E. Crawford, and U. Ohler. Predicting cell-type-specific gene expression from regions of open chromatin. *Genome Res.*, 22(9):1711–1722, Sep 2012.

[14] Z. Ouyang, Q. Zhou, and W. H. Wong. ChIP-Seq of transcription factors predicts absolute and differential gene expression in embryonic stem cells. *Proc. Natl. Acad. Sci. U.S.A.*, 106(51):21521–21526, Dec 2009.

[15] R. Pique-Regi, J. F. Degner, A. A. Pai, D. J. Gaffney, Y. Gilad, and J. K. Pritchard. Accurate inference of transcription factor binding from DNA sequence and chromatin accessibility data. *Genome Res.*, 21(3):447–455, Mar 2011.

[16] H. G. Roider, A. Kanhere, T. Manke, and M. Vingron. Predicting trancription factor affinities to DNA from a biophysical model. *Bioinformatics*, 23(2):134–141, Jan 2007.

[17] F. Schmidt, N. Gasparoni, G. Gasparoni, K. Gianmoena, C. Cadenas, J. K. Polansky, P. Ebert, K. Nordstrom, M. Barann, A. Sinha, S. Frohler, J. Xiong, A. Dehghani Amirabad, F. Behjati Ardakani, B. Hutter, G. Zipprich, B. Felder, J. Eils, B. Brors, W. Chen, J. G. Hengstler, A. Hamann, T. Lengauer, P. Rosenstiel, J. Walter, and M. H. Schulz. Combining transcription factor binding affinities with open-chromatin data for accurate gene expression prediction. *Nucleic Acids Res.*, Nov 2016.

[18] Peter H Von Hippel and Otto G Berg. On the specificity of dna-protein interactions. *Proceedings of the National Academy of Sciences*, 83(6):1608–1612, 1986.

[19] G. G. Yardımcı, C. L. Frank, G. E. Crawford, and U. Ohler. Explicit DNase sequence bias modeling enables high-resolution transcription factor footprint detection. *Nucleic Acids Res.*, 42(19):11865–11878, Oct 2014.

[20] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005.