# Computing transcription factor scores using TEPIC

## Motivation

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, 5, 3, 9], e.g. peaks or footprints in DNase1-seq data, and/or by considering Histone marks [2, 5], e.g. H3K4me3.

Here, we compute TF affinities for a species specific set of *Position Specific Energy Matrices (PSEM)* using *TRAP* [16] which is based on a biophysical model of TF binding [18]. The main advantage of affinity based predictions compared to hit-based methods like Fimo [4] 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.

## Preprocessing of Position Count Matrices (PCM)

We obtained *Position Count Matrices (PCMs)* from JASPAR [10], which is also including data from Uniprobe [6], HOCOMOCO [8], the Kellis Lab ENCODE Motif database [7], and TRANSFAC [11] for five species: *homo sapiens, mus musculus, rattus norvegicus, drosophila melanogaster,* and *caenorhabditis elegans*.

Briefly, we downloaded the JASPAR CORE Vertebrata data set to cover *homo sapiens, mus musculus,* and *rattus norvegicus*, for *drosophila melanogaster* we use the JASPAR CORE Insecta data set and for *caenorhabditis elegans* we obtained the JASPAR CORE Nematoda PCMs. From HOCOMOCO we use the provided data for *homo sapiens* and *mus musculus*. The latter is also used for *rattus norvegicus*. The Kellis Lab ENCODE Motifs are based on human ChIP-seq data. Thus, we consider these *PCMs* for *homo sapiens, mus musculus*, and *rattus norvegicus*. From TRANSFAC, we obtained species specific sets for all considered organisms.

From this initial set, we removed all TFs that could not be mapped to an Ensembl gene ID, using the Ensemble Genes 87 database, and the current versions of the reference genomes: GRCh38, GRCm38, Rnor_6, BDGP6, and WBcel235. Thereby, we generate species specific sets of *PCMs* assuming a motif conservation among vertebrates for the JASPAR CORE Vertebrata *PCMs*, the HOCOMOCO mouse *PCMs*, and Kellis Lab ENCODE Motif *PCMs*. Neither HOCOMOCO nor Kellis Lab ENCODE Motifs are considered for *drosophila melanogaster* and *caenorhabditis elegans*.

Next, for each species set, we computed the information content $IC$ of each

*PCM* $M$ normalized per motif length $|M|$ as

$$P(i,j) = \frac{M(i,j) + pc}{4 * pc + \sum_i M(i,j)}, \tag{1}$$

$$IC = -\frac{\sum_{ij} log(P(i,j) * P(i,j)}{|M|}, \tag{2}$$

with $i \in \{A, C, G, T\}$, $j \in \{1, ...|M|\}$, and a pseudo count $pc = 1$.

Note that the smaller the $IC$ value of a matrix, the more informative the matrix is.

In Figure 1, a violin plot shows the distribution of the normalized information content for the *homo sapiens* data set. Across all species collections, we find that the JASPAR matrices have a small variance and that the poorest JASPAR *PCM* is still more informative than several *PCMs* from other databases. Therefore, we decided to consider all JASPAR *PCMs* and use the normalized information content value of the poorest JASPAR *PCM* as a species specific cut-off value for the remaining databases. The cut-offs are:

- *homo sapiens:* 1.55928
- *mus musculus:* 1.55928
- *rattus norvegicus:* 1.55928
- *drosophila melanogaster:* 1.67157
- *caenorhabditis elegans:* 1.33879

In case that multiple motifs exists for one distinct TF in one database, we consider only the *PCM* with the best $IC$ value. If the motifs are marked specifically as a secondary or tertiary binding motif, as in JASPAR, we do not remove them.

In order to merge the different databases per species, we execute the following merging procedure on the filtered sets of the individual databases:

1. Consider all JASPAR matrices.
2. Add all HOCOMOCO matrices that are not included in the set of step (1).
3. Add all Kellis Lab ENCODE Motif database matrices that are not included in the set of step (2).
4. Add all TRANSFAC matrices that are not included in the set of step (3).

This procedure ensures that we generate the largest possible set of open-source *PCMs* from our collection of TF binding motifs. In addition to the unified set, we also provide the user with the option to work with all *PCMs* from a single database.

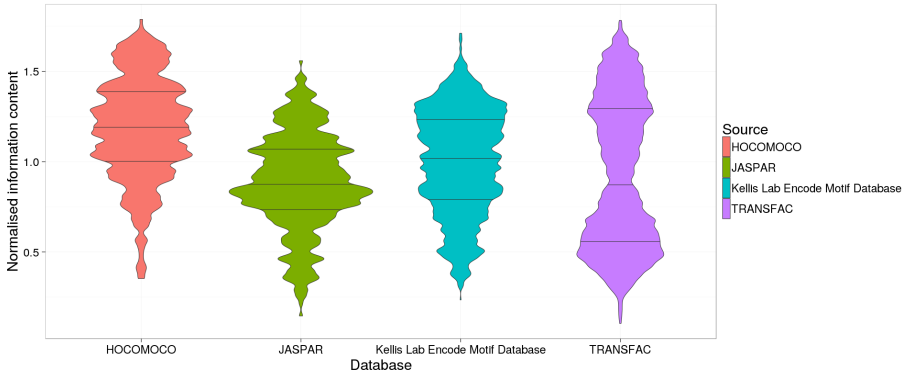


Figure 1: Normalized information content for *PCMs* extracted from JASPAR Core Vertebrata, HOCOMOCO human, the Kellis ENCODE Motif database, and TRANSFAC Human. Small values indicate high quality matrices. Clearly, the poorest *PCM* in JASPAR is still better than several *PCMs* out of the other databases.

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:

$$E_{i,j} = \frac{1}{\lambda} log(\frac{M_{max,j}}{M_{i,j}} b_{i,j}), \tag{3}$$

$$M_{max,j} = \max_{i \in \{A,C,G,T\}} (M_{i,j}). \tag{4}$$

The parameter $\lambda$ 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:

$$ln(R_0) = m * |M| + n, \tag{5}$$

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

Table 1 shows the number of *PCMs* that based the quality filtering and the within database redundancy check. Table 2 provides an overview on the final counts of the unified set of PCMs. The entire processing workflow of *PCMs* to *PSEMs* is shown in Figure 2.

| | Jaspar | Hocomoco | Kellis Lab Encode Motif Database | Transfac |
|---|---|---|---|---|
| Homo sapiens | 515 | 390 | 559 | 1747 |
| Mus musculus | 499 | 261 | 523 | 921 |
| Rattus norvegicus | 489 | 253 | 513 | 389 |
| Drosophila melanogaster | 129 | 0 | 0 | 290 |
| Caenorhabditis elegans | 26 | 0 | 0 | 42 |

Table 1: Overview on *PCM* counts per species and database after quality filtering and a within database redundancy check.

| | Jaspar | Hocomoco | Kellis Lab Encode Motif Database | Transfac | Total |
|---|---|---|---|---|---|
| Homo sapiens | 515 | 81 | 130 | 584 | 1310 |
| Mus musculus | 499 | 67 | 124 | 194 | 884 |
| Rattus norvegicus | 489 | 67 | 121 | 50 | 727 |
| Drosophila melanogaster | 129 | 0 | 0 | 92 | 221 |
| Caenorhabditis elegans | 26 | 0 | 0 | 14 | 40 |

Table 2: Overview on *PCM* counts per species and database after quality and redundancy filtering. We considered all Jaspar matrices and added additional non redundant *PCMs* from Hocomoco, the Kellis Lab Encode Motif Database, and Transfac.

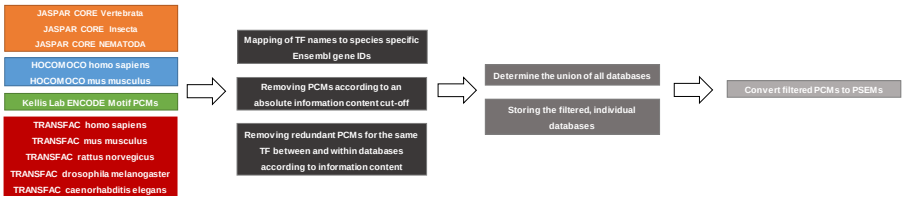


Figure 2: Visualization of the PCM preprocessing workflow.

## Computing TF gene scores

Currently, we offer the annotation of five different species, including the most common model organisms: *homo sapiens, mus musculus, rattus norvegicus, drosophila melanogaster,* and *caenorhabditis elegans*. Using our collections of species specific *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

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

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 normalised by peak(and motif)-length during the computation of gene-TF scores:

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

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

$$a_{g,i}^{w} = \sum_{p \in P_{g,w}} \frac{a_{p,i}}{|p| - |m|} s_p e^{-\frac{d_{p,g}}{d_0}}, \tag{8}$$

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 3.

In addition to the TF gene scores, TEPIC can compute features for peak length, peak count, and peak signal following the same scoring formulation as for TF affinities. These features can be used for example to assess the influence of chromatin accessiblity on gene expression without considering TF binding predictions.

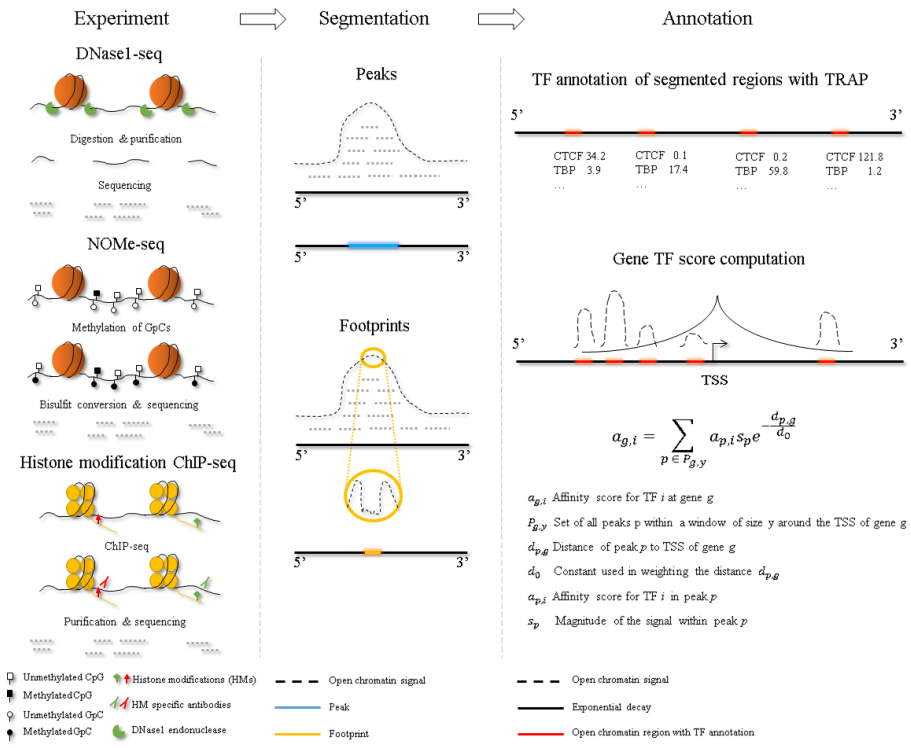


Figure 3: The general workflow of *TEPIC* is as follows: Data of an open-chromatin or Histone modification ChIP-seq experiment needs to be preprocessed to generate a genome segmentation, either by peak for footprint calling. Using the segmentation, *TEPIC* applies *TRAP* in all regions of interest, and computes TF gene scores using exponential decay to reweigh TF binding predictions in open-chromatin regions based on their distance to a genes TSS.

## Required input

To compute TF gene scores a user needs to specify:

- a reference genome,
- a set of *PSEMs*,
- a set of genomic regions in BED format.

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

### Output

This step generates the following output:

1. TF affinities for all selected *PSEMs* in the regions provided by the user that passed the filtering step.
2. (Length normalised) TF gene scores for all selected *PSEMs* calculated as described above (optionally including peak features).
3. A meta data file listing all used parameters.
4. Optionally a seperate file containing the signal information in peaks.

## 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).

### Linear regression to predict gene expression

#### 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 effect 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.

### Available regularization methods

We offer three different regularization techniques:

- Lasso:

$$\hat{\beta} = \arg\min_{\beta} ||y - X\beta||^2 + ||\beta||, \tag{9}$$

- Ridge:

$$\hat{\beta} = \arg\min_{\beta} ||y - X\beta||^2 + ||\beta||^2, \tag{10}$$

- Elastic net:

$$\hat{\beta} = \arg\min_{\beta} ||y - X\beta||^2 + \alpha||\beta||^2 + (1-\alpha)||\beta||, \tag{11}$$

where, $\beta$ represents the regression coefficient vector, $\hat{\beta}$ represents the estimated coefficients, $X$ is the feature matrix, $y$ is the response vector, and the parameter $\alpha$ 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 can not 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 can not 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.

### 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 a

inner cross validation, the $\alpha$ 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 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. Besides, we support the upload of a matrix containing gene expression data for several samples. In that case, the user has to select the column/sample that should be used for model construction.

**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 mean.
- 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.

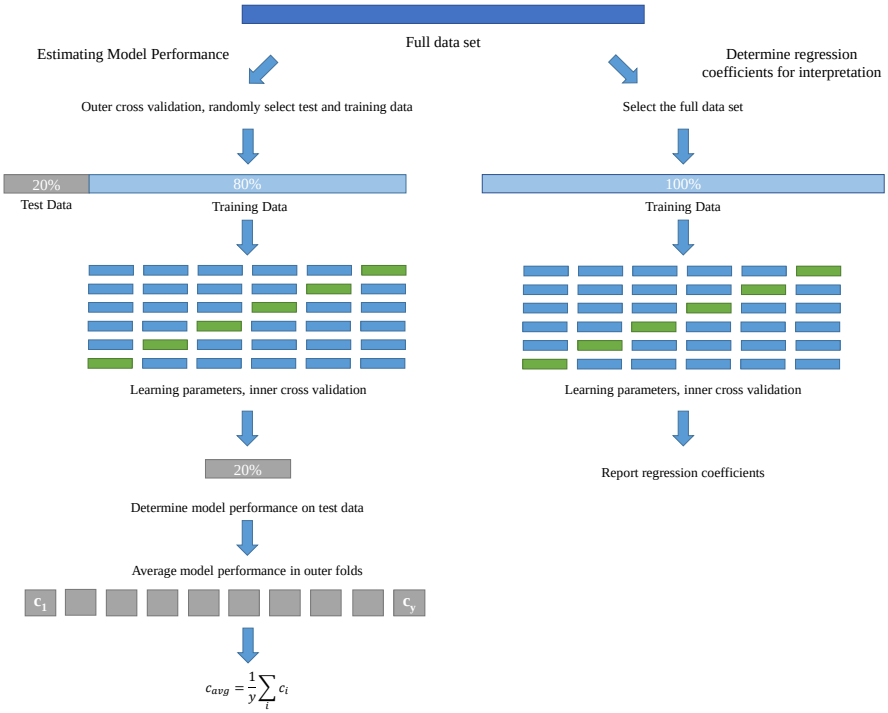


Figure 4: Overview of the learning process. The left part of the Figure describes the assessment of model performance in a $y$-fold outer cross validation and 6-fold inner cross validation. The right hand site illustrates model training on the entire data set, again using a 6-fold inner cross validation for parameter learning.

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

[10] 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.

[11] V. Matys, O. V. Kel-Margoulis, E. Fricke, I. Liebich, S. Land, A. Barre-Dirrie, I. Reuter, D. Chekmenev, M. Krull, K. Hornischer, N. Voss,

P. Stegmaier, B. Lewicki-Potapov, H. Saxel, A. E. Kel, and E. Wingender. TRANSFAC and its module TRANSCompel: transcriptional gene regulation in eukaryotes. *Nucleic Acids Res.*, 34(Database issue):D108–110, Jan 2006.

[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 transcription 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.