Exploiting sequence-based features for predicting enhancer-promoter interactions

Yang Yang, Ruochi Zhang, Shashank Singh, and Jian Ma

Supplementary Materials

A Supplementary Methods

A.1 Training the Gradient Tree Boosting model

As we mentioned in the main text, Gradient Tree Boosting (GTB) approximates $F^*(\boldsymbol{x})$ with $F(\boldsymbol{x})$, an additive ensemble of base learner functions. The additive model is as follows:

$$F(\boldsymbol{x}) = \sum_{m=1}^{M} \beta_m h(\boldsymbol{x}; \boldsymbol{\theta}_m), \tag{1}$$

where $h(\boldsymbol{x}; \boldsymbol{\theta}_m)$ is a decision tree, which is a function of $\boldsymbol{x}$ with the parameter setting $\boldsymbol{\theta}_m$, and β_m is the expansion coefficient. M is the total number of decision trees in the ensemble. For $m = 1, \cdots, M$,

$$F_m(\boldsymbol{x}) = F_{m-1}(\boldsymbol{x}) + \beta_m h(\boldsymbol{x}; \boldsymbol{\theta}_m). \tag{2}$$

A new decision tree is generated at each stage to minimize the loss function given the current model. Suppose $L(y, F(x))$ is the loss function,

$$(\beta_m, \boldsymbol{\theta}_m) = \arg\min_{\beta, \boldsymbol{\theta}} \sum_{i=1}^{N} L(y_i, F_{m-1}(\boldsymbol{x_i}) + \beta h(\boldsymbol{x}_i; \boldsymbol{\theta})). \tag{3}$$

Gradient descent method [1] is used to estimate the parameters of the new decision tree. $\boldsymbol{\theta}_m$ and β_m are updated with a two-step strategy. Firstly,

$$\boldsymbol{\theta}_m = \arg\min_{\boldsymbol{\theta}, \rho} \sum_{i=1}^{N} [\tilde{y}_{im} - \rho h(\boldsymbol{x}_i; \boldsymbol{\theta})]^2, \tag{4}$$

where

$$\tilde{y}_{im} = -\frac{\partial L(y_i, F(\boldsymbol{x}_i))}{\partial F(\boldsymbol{x}_i)}|_{F(\boldsymbol{x}) = F_{m-1}(\boldsymbol{x})}. \tag{5}$$

Accordingly, β_m can be updated through a second-step optimization.

$$\beta_m = \arg\min_{\beta} \sum_{i=1}^{N} L(y_i, F_{m-1}(\boldsymbol{x}_i) + \beta h(\boldsymbol{x}_i; \boldsymbol{\theta}_m)). \tag{6}$$

We also tried to choose the most appropriate thresholds for the classifier. A sample is classified as positive if the probability of being from the positive class exceeds the threshold. It has been suggested that adjusted thresholds of the classifier be used in the case of imbalanced data [2, 3]. The threshold of the classifier was tuned as a parameter using only training data. In each fold of the 10-fold cross validation (outer round), we performed 5-fold cross validation (inner round) on the training set to select the threshold of the classifier.

Prior to model training on all the six cell lines, we tuned the parameters $n\_estimators$ (the number of decision trees in the ensemble) and max_depth (maximal depth of each tree) of the XGBClassifier. Ensembles with more estimators are more robust to the over-fitting problem, and deeper decision trees are able to learn higher order of interactions of the features [4]. We evaluated the performance of the trained classifier with respect to different choices of the parameters on the training data. The parameter tuning processes were mainly performed on GM12878 and K562, which have larger sample sizes than the other cell lines. The parameters of XGBClassifier adopted in both PEP-Motif and PEP-Word are $n\_estimators = 1000$ and $max\_depth = 10$, with the other parameters as default.

A.2 Motif clustering based on motif similarity estimation

We used TomTom [5] to compute the similarity between each pair of the 641 motifs from the HOCOMOCO Human v10 database used in PEP-Motif and constructed a connectivity graph showing the motif similarity relationships. Motifs with significantly similar PWMs were connected in the graph, using a threshold of E-value of 10^{-3} in measuring the similarity confidence with results from TomTom. There are 357 components in the graph, representing coarse-level clustering of motifs. Examples of the components are shown in Supplementary Figure S4. We further implemented the Highly Connected Subgraphs clustering algorithm [6] to group motifs in each component into clusters. Any pair of motifs in a resulted cluster are within distance of 2 to each other, a property guaranteed by the HSC algorithm.

A.3 Weighted pooling to generate features in PEP-Word

We used weighted pooling to generate feature representation for enhancer/promoter sequences from features of K-mers. Each K-mer is assigned a weight, based on two principles [7]. First, K-mers related to unwanted background features have lower weights. Second, K-mers with more occurrences have higher weights. Accordingly, we employed the approach of calculating Item Frequency-Inverse Document Frequency (TF-IDF) [8–10] in our weight assignment, with the goal of repressing background noise while retaining the influence of K-mers with frequent occurrences. TF-IDF has been effectively applied in information retrieval and text mining [11]. It reflects the importance of a word to a document in a corpus of documents. We used GenSim [12] to build a TF-IDF dictionary for all the K-mers involved in the word embedding model of the enhancers (or promoters). The TF-IDF of each K-mer is calculated as follows.

$$\text{tfidf}(w, d, D) = \text{tf}(w, d) \cdot \text{idf}(w, D), \tag{7}$$

where w is a K-mer, d is the sequence in which the K-mer appears, and D is the corpus of all the sequences of the enhancers (or promoters). $\text{tf}(w, d)$ is the number of occurrences of K-mer w in sequence d. Specifically,

$$\text{idf}(w, d) = \log \frac{|D|}{|\{d \in D : t \in d\}|}, \tag{8}$$

where $|D|$ is the total number of sequences and $|\{d \in D : t \in d\}|$ is the number of sequences in which the K-mer appears. Therefore, if a K-mer appears commonly across the sequences, it is assumed to involve background noise and receives a reduced TF-IDF value.

A.4 Choosing parameters of the word embedding model in PEP-Word

In PEP-Word we adjusted parameters such as K-mer size and the embedded feature vector size. We chose the parameters in seeking balance between computational efficiency and prediction performance. In PEP-Word the K-mer size is K=6 and the embedded feature vector size is n=300. As the vocabulary size of the word embedding model is approximately 4^K, larger K leads to exponential increase of vocabulary size and higher computational expense. We found that the performance improves as K increases from 4 to 6, which suggests the longer K-mers capture more discriminative patterns. Model with K=7 has similar or slightly better performance than model with K=6 in three tested cell lines (GM12878, K562 and HeLa-S3) (Supplementary Figure S5). However, there is no obvious improvement as K increases to 8. Since K=7 increases computation cost and training time, K=6 is preferred for practical use and selected for the model for all the performance comparison in this work. We also evaluate the performance by varying the embedded feature vector size from 100 to 600. Results showed that choosing n=300 has balanced benefit of computational efficiency and performance (Supplementary Figure S5).

A.5 Integration of features from PEP-Motif and PEP-Word

We combined the feature vector from PEP-Word with a selected subset of putative important features from PEP-Motif to form integrated feature representation (PEP-Integrate) of each enhancer-promoter pair, in attempt to exploit potential complementary characteristics of the two kind of features. We assume motif-based features to be more accurate in the capturing specific TFBS patterns as a curated motif database is used for motif scanning. On the other hand, features generated by PEP-Word are abstract distributed representations and not limited to TFBS patterns, with the possibility of capturing patterns in the potential whole feature space which are not modeled by existing TFBS motifs. The integrated feature vector for the i-th sample pair is as follows:

$$f^{(i)} = (f_W^{(i)}, f_{M,s}^{(i)}) = (f_{W,1}^{(i)}, \cdots, f_{W,n}^{(i)}, f_{M,s_1}^{(i)}, \cdots, f_{M,s_k}^{(i)}). \tag{9}$$

$f_W^{(i)}$ and $f_M^{(i)}$ are respectively the feature vectors extracted by PEP-Word and PEP-Motif. Let s be the set of indices of selected motif features in $f_M^{(i)}$. Suppose $s = (s_1, ..., s_k)$, i.e., there are k selected motif features to be combined with $f_W^{(i)}$. We performed similar feature importance ranking and recursive feature selection as described in the section of PEP-Motif to choose features contained in s. We ranked motif features based on their importance estimated in PEP-Motif, and sequentially increased the number of top ranking motif-based features selected for PEP-Integrate, refitting, and evaluating the predictor trained with GTB accordingly. Performance evaluation with respect to different sizes of s is shown in Supplementary Figure S2. We chose the top 300 important motif features for PEP-Integrate in each cell line, which gains performance improvement over individual modules and does not induce high feature dimensionality.

A.6 Choice of flanking region of enhancers

The enhancers are mostly only a few hundred base-pairs in length. In forming the feature representation of the enhancers, we involve the flanking regions of each enhancer in attempt to utilize the information encoded in the context for more effective feature extraction. Flanking region of 4kb on each side of the originally annotated enhancer is included as extension of the enhancer. We changed the length of the flanking region L and evaluated the performance of the re-trained model with respect to different choices of L on cell lines GM12878, K562, and HeLa-S3. The performance comparison on AUPR is shown in Supplementary Figure S5. Accordingly, we selected $L = 4$kb to keep the balance between performance and computational efficiency.

A.7 Analysis of potential predictive feature interactions

We took a look at the interacting enhancer-associated motif features and promoter-associated motif features that are important based on our GTB model. For each cell type, we performed 10-fold cross validation and used XGBFIR [13] to extract a number of most predictive E-P feature interactions from the training data. The top N important E-P feature interactions from each fold were first merged and then we selected the ones that were within top N in more than two training folds (denoted as feature set $S_1^{(N)}$). We then sorted all the merged features by their relative rank within E-P feature interactions in the respective training fold, and selected the top N feature interactions (denoted as feature set $S_2^{(N)}$). We obtained the union of $S_1^{(N)}$ and $S_2^{(N)}$ (denoted as $S^{(N)}$) while retaining the order of the interactions within each set, with $S_1^{(N)}$ assigned priority. Thus $S^{(N)}$ contains important features from the corresponding cell line with respect to the choice of N. We repeated the procedure for all the six cell lines. For $N = 30$, the E-P feature interactions selected to $S^{(N)}$ in GM12878, K562, and HeLa-S3 are shown in Supplementary Figure S8, S9 and S10. For $N = 200$, we chose the feature interactions that are selected to $S^{(N)}$ in at least two cell lines. We obtained 40 feature interactions (shown in Supplementary Table S10).

A.8 Evaluation metrics used in model performance assessment

Different evaluation metrics are used in our study to assess the performance of the model. The evaluation metrics include AUROC (Area Under the Receiver Operating Characteristic curve), AUPR (Area Under the Precision-Recall curve), Precision, Recall, F_1 score, and MCC (Matthews Correlation Coefficient). Receiver Operator Characteristic (ROC) curves are widely used in evaluation of binary decision problems [14]. However, it has been shown that ROC curves can overly optimistically evaluate an algorithms performance if the class distributions are very imbalanced [15]. Precision-Recall (PR) curves have often been used as an alternative to ROC curves for predictive tasks on heavily imbalanced data.

Precision (Positive Predictive Value, PPV) is the fraction of predictions that are true positive. Recall (Sensitivity or True Positive Rate, TPR) is the fraction of true positive among all the predictions. F_1 score is the harmonic mean of precision and recall. MCC (Matthews Correlation Coefficient) takes into account true and false positives and negatives, and is generally regarded as a balanced measure even if the classes are very imbalanced. The definitions are shown below.

$$\text{Precision} = PPV = \frac{TP}{TP + FP}, \tag{10}$$

$$\text{Recall} = TPR = \frac{TP}{TP + FN}, \tag{11}$$

$$F_1 = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision+Recall}}, \tag{12}$$

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}. \tag{13}$$

A.9 Performance evaluation in comparison with TargetFinder and RIPPLE

To obtain AUPR and AUROC in performance comparison, probabilistic predictions of TargetFinder (E/P/W) and TargetFinder (EE/P) on the E/P datasets and EE/P datasets in six cell lines as described in the Methods section were obtained using the source code of TargetFinder through 10-fold cross validation, in the same way as PEP modules were evaluated. The calculated performance of Precision, Recall, F_1 score, and MCC are also consistent with the results reported in [16]. Probabilistic predictions of RIPPLE on the EE/P datasets were obtained through 10-fold cross validation by using the source code of RIPPLE and employing the same set of functional genomic features collected for TargetFinder (EE/P) in the annotated extended enhancer and promoter regions. As RIPPLE only outputs probabilistic predictions and does not make binary classification, to compute Precision, Recall, F_1 score, and MCC, we drew the Precision-Recall curve of RIPPLE and chose the threshold corresponding to the best F_1 based on the PR curve. Samples were classified by the selected threshold and Precision, Recall, F_1 score, and MCC were calculated accordingly, which are highly likely to be better than real performance if a threshold is estimated from cross validation or chosen with prior knowledge. RIPPLE uses both continuous features or binary features. We performed evaluation with both modes. To obtain binary features, the original features were converted to 1 if the feature values are larger than zero, and to 0 otherwise. We compared performance from both modes and chose continuous features, which performed better.

B Supplementary Tables

Cell line	Enhancers (total number)	Promoters (total number)	Positive sample size	Negative sample size	Total sample size
GM12878	100036	8453	2113	42200	44313
K562	82806	8196	1977	39500	41477
IMR90	108996	5253	1254	25000	26254
HeLa-S3	103460	7794	1740	34800	36540
HUVEC	65358	8180	1524	30400	31924
NHEK	144302	5254	1291	25600	26891

Table S1: Summary of Enhancer/Promoter (E/P) data. E/P data were used in [16] to evaluate the performance of TargetFinder (E/P/W), TargetFinder (E/P) and the baseline model, of which TargetFinder (E/P/W) achieved the strongest performance. The negative sample size is about 20 times of the positive sample size in E/P data.

Cell line	Enhancers (total number)	Promoters (total number)	Positive sample size	Negative sample size	Total sample size
GM12878	100036	8453	3559	71000	74559
K562	82806	8196	2750	55000	57750
IMR90	108996	5253	1897	37800	39697
HeLa-S3	103460	7794	2146	42900	45046
HUVEC	65358	8180	1932	38600	40532
NHEK	144302	5254	1559	31000	32559

Table S2: Summary of Extended Enhancer/Promoter (EE/P) data.EE/P data were used in [16] to evaluate the performance of TargetFinder (EE/P). The enhancer-promoter interactions were identified on basis of the extended enhancers and the promoters, resulting in more positive samples. The negative sample size is also about 20 times of the positive sample size in EE/P data. The sample size of EE/P data is therefore larger than that of E/P data.

Cell line	Method	AUROC	AUPR	Precision	Recall	F_1	MCC
GM12878	TargetFinder(E/P/W)	0.9618	0.8322	0.8778	0.7549	0.8117	0.8055
GM12878	PEP-Motif	0.9503	**0.8532**	**0.9039**	**0.7610**	**0.8263**	**0.8217**
GM12878	PEP-Word	0.9489	**0.8449**	**0.8962**	**0.7601**	**0.8225**	**0.8174**
GM12878	PEP-Integrate	0.9524	**0.8607**	**0.9189**	**0.7719**	**0.8390**	**0.8351**
K562	TargetFinder(E/P/W)	0.9640	0.8782	0.8872	0.8113	0.8476	0.8412
K562	PEP-Motif	0.9472	0.8454	**0.8929**	0.7416	0.8129	0.8080
K562	PEP-Word	0.9461	0.8420	0.8849	0.7425	0.8075	0.8021
K562	PEP-Integrate	0.9514	0.8498	**0.8938**	0.7532	0.8175	0.8124
IMR90	TargetFinder(E/P/W)	0.9621	0.8197	0.8501	0.7281	0.7844	0.7770
IMR90	PEP-Motif	0.9388	0.8161	0.8500	**0.7504**	**0.7971**	**0.7893**
IMR90	PEP-Word	0.9331	**0.8376**	**0.8994**	**0.7488**	**0.8172**	**0.8126**
IMR90	PEP-Integrate	0.9416	**0.8470**	**0.9255**	**0.7632**	**0.8365**	**0.8334**
HeLa-S3	TargetFinder(E/P/W)	0.9758	0.9089	0.8971	0.8466	0.8711	0.8652
HeLa-S3	PEP-Motif	0.9613	0.8823	**0.9217**	0.7845	0.8476	0.8436
HeLa-S3	PEP-Word	0.9613	0.8741	**0.9070**	0.7621	0.8282	0.8238
HeLa-S3	PEP-Integrate	0.9645	0.8865	**0.9300**	0.7793	0.8480	0.8447
HUVEC	TargetFinder(E/P/W)	0.9563	0.8044	0.8674	0.6909	0.7692	0.7643
HUVEC	PEP-Motif	0.9318	0.7649	0.8013	0.6640	0.7262	0.7173
HUVEC	PEP-Word	0.9298	0.7787	0.8062	0.6877	0.7422	0.7329
HUVEC	PEP-Integrate	0.9401	0.7896	0.8259	0.6850	0.7489	0.7411
NHEK	TargetFinder(E/P/W)	0.9831	0.9272	0.9234	0.8683	0.8950	0.8903
NHEK	PEP-Motif	0.9624	0.8815	0.8985	0.7885	0.8399	0.8344
NHEK	PEP-Word	0.9726	0.9003	**0.9249**	0.8110	0.8642	0.8599
NHEK	PEP-Integrate	0.9778	0.9073	**0.9536**	0.8118	0.8770	0.8744

Table S3: Performance evaluation of TargetFinder (E/P/W), PEP-Motif, PEP-Word, and PEP-Integrate on E/P data of six cell lines. K=6 is used for K-mer by PEP-Word for training the word embedding model. PEP-Integrate features are PEP-Word features combined with on E/P data 300 top ranked important motif features selected from PEP-Motif. The performance of PEP-Motif, PEP-Word or PEP-Integrate that shows improvement over the corresponding performance of TargetFinder (E/P/W) is in bold font.

Cell line	Method	AUROC	AUPR	Precision	Recall	F_1	MCC
GM12878	TargetFinder(EE/P)	0.9704	0.8804	0.8888	0.8039	0.8442	0.8380
GM12878	RIPPLE	0.9583	0.8416	0.8880	0.7378	0.8061	0.8011
GM12878	PEP-Motif	0.9634	**0.8806**	**0.9011**	0.7963	**0.8455**	**0.8400**
GM12878	PEP-Word	0.9661	0.8784	**0.8983**	0.7839	0.8372	0.8317
GM12878	PEP-Integrate	0.9665	**0.8888**	**0.9139**	0.7963	**0.8511**	**0.8463**
K562	TargetFinder(EE/P)	0.9640	0.8782	0.8796	0.7542	0.8121	0.8060
K562	RIPPLE	0.9448	0.7935	0.8615	0.6807	0.7605	0.7557
K562	PEP-Motif	0.9557	0.8409	0.8758	0.7309	0.7968	0.7912
K562	PEP-Word	0.9579	0.8385	0.8463	0.7232	0.7799	0.7724
K562	PEP-Integrate	0.9603	0.8480	0.8779	0.7268	0.7952	0.7898
IMR90	TargetFinder(EE/P)	0.9650	0.8633	0.9056	0.7739	0.8346	0.8297
IMR90	RIPPLE	0.9533	0.8235	0.8940	0.7116	0.7925	0.7889
IMR90	PEP-Motif	0.9509	0.8419	0.9032	0.7333	0.8094	0.7770
IMR90	PEP-Word	0.9583	0.8627	0.8856	0.7628	0.8196	0.8137
IMR90	PEP-Integrate	0.9609	**0.8731**	**0.9232**	0.7607	0.8341	**0.8309**
HeLa-S3	TargetFinder(EE/P)	0.9671	0.8669	0.8785	0.7852	0.8292	0.8226
HeLa-S3	RIPPLE	0.9569	0.8265	0.8646	0.7260	0.7893	0.7829
HeLa-S3	PEP-Motif	0.9668	**0.8725**	**0.8962**	0.7563	0.8203	0.8153
HeLa-S3	PEP-Word	0.9662	**0.8684**	0.8723	0.7548	0.8093	0.8028
HeLa-S3	PEP-Integrate	**0.9715**	**0.8792**	**0.8935**	0.7590	0.8208	0.8155
HUVEC	TargetFinder(EE/P)	0.9376	0.7559	0.8025	0.6351	0.7090	0.7015
HUVEC	RIPPLE	0.9310	0.7006	0.7340	0.6113	0.6670	0.6550
HUVEC	PEP-Motif	**0.9410**	**0.7799**	**0.8094**	**0.6615**	**0.7280**	**0.7198**
HUVEC	PEP-Word	**0.9447**	**0.7793**	0.7702	**0.6801**	**0.7224**	**0.7109**
HUVEC	PEP-Integrate	**0.9473**	**0.7953**	**0.8082**	**0.6869**	**0.7426**	**0.7334**
NHEK	TargetFinder(EE/P)	0.9721	0.8759	0.8738	0.7864	0.8278	0.8209
NHEK	RIPPLE	0.9640	0.8535	0.9033	0.7492	0.8191	0.8147
NHEK	PEP-Motif	0.9656	0.8750	**0.9134**	0.7716	**0.8366**	**0.8323**
NHEK	PEP-Word	**0.9727**	**0.8927**	**0.8883**	0.7858	**0.8339**	**0.8278**
NHEK	PEP-Integrate	**0.9752**	**0.9019**	**0.9133**	**0.7903**	**0.8473**	**0.8426**

Table S4: Performance evaluation of TargetFinder(EE/P), RIPPLE, PEP-Motif, PEP-Word, and PEP-Integrate on EE/P data of six cell lines. K=6 is used for K-mer by PEP-Word for training the word embedding model. PEP-Integrate features are PEP-Word features combined with the top 300 ranked important motif features selected from PEP-Motif. The performance of PEP-Motif, PEP-Word or PEP-Integrate that shows improvement over the corresponding performance of TargetFinder (EE/P) is in bold font.

Cell line	Estimated cell-type specific top 5% important predictive motif features in enhancer region
GM12878	(RORG,RORA,NR1D1), ETV5, EOMES, (PBX1,PKNOX1,PBX2), BRCA1, (TCF3,TAL1(S)), (NKX23,NKX22), SRF, (PBX3,NFYB,FOXI1,NFYA), TLX1(S), (HNF1B,HNF1A), TEAD1, TGIF1, (ZNF589,SPZ1), EBF1, KLF13, HSFY1
K562	(TAL1,GATA1,GATA1(S)), E2F8, FOXC2, BATF, RARB, SMAD1, BPTF, NR6A1, PKNOX2, HOXD4
IMR90	(MEIS2,TGIF2LX,TGIF2), (ESR1,ESR1(S),ESR2), ZNF384, (NFIA,NFIC,TLX1), TCF7L2, PRD14, (SCRT2,SCRT1), RREB1, (MEF2D,MEF2A,MEF2C), TBX2, FOXH1, (THRB,THRB(S),THRA) (HES7,HES5,HEY1), FOXO6, MSC, (GLIS1,GLIS2,GLIS3), ZNF219, HOXA10, HOXA11
HeLa-S3	(GLI2,GLI3,GLI1), ZBTB6, (FOXA1,FOXA2,FOXF2), FOXO3, MYOD1, HMGA2, IRX3, CENPB
HUVEC	(TP63,TP53,TP73), YBX1, IRX2, ZSCA4, (RFX5,RFX1), AIRE, (SREBF1,SREBF2), ZBTB18, VDR, ZNF410, MAFB, RBPJ, MYF6, HOXC13, POU2F3, BSX
NHEK	TBX20, NR1H4, ZEB1, PITX3, FIGLA, NKX21, KLF8, (FOS,FOSL1,JUND,FOSB), SMRC1

Table S5: Enhancer-associated motif representatives with top 5% feature importance in a single cell line. A motif representative is either a single motif or a motif cluster. If it is a motif cluster, all the members are shown in combination and by the order of their estimated feature importance. A motif is denoted by its corresponding TF.

Cell line	Estimated cell-type specific top 5% important predictive motif features in promoter region
K562	PAX5(S), (RARG,NR2C1,RARA,RARG(S)), UBIP1, INSM1, TBX19, POU6F2
HeLa-S3	MYOG, IRF5, IRF9
HUVEC	(NRF1,ZNF639), KLF15, CLOCK, FOXO4, HIF1A
NHEK	EGR4, MZF1, ETV7

Table S6: Promoter-associated motif features with top 5% importance in a single cell line.

Cell line	Features	Total	Comparable	Top 50%	Top 50%	Top 30%	Top 30%	Top 25%	Top 25%
	(E/W)	TFs	TFs	(T)	(T-P)	(T)	(T-P)	(T)	(T-P)
GM12878	100	85	60	53	48	38	25	34	21
K562	136	120	59	58	55	48	40	42	27
IMR90	56	23	7	7	7	4	4	4	3
HeLa-S3	73	58	29	26	25	20	15	16	12
HUVEC	23	8	6	6	6	4	2	4	2
NHEK	20	5	1	1	1	1	1	1	1

Table S7: Comparison of the predictive features discovered by TargetFinder(E/P/W) in enhancer/window region with PEP-Motif. "(T)" in the header line represents TargetFinder (E/P/W). "(T-P)" represents TargetFinder (E/P/W) and PEP-Motif. Top 50% (T-P) represents features estimated both by TargetFinder (E/P/W) and PEP-Motif to be at top 50% feature importance level.

Cell line	Common TFs of top 25% feature importance in enhancer/window region
GM12878	ZNF384, CTCF, RUNX3, SPI1, EBF1, SP1, IRF3, ELF1, NFKB1, PAX5, MAFK, NFIC, BCL11A, EGR1, SRF, IRF4, NFYB, PBX3, STAT3, NFATC1, TBP
K562	CTCF, SRF, ZNF384, YY1, MAZ, SPI1, MAFF, MEF2A, CEBPD, REST, NR2F2, EGR1, ELF1, TEAD4, GATA2, TAL1, USF1, STAT5A, ZBTB7A, CEBPB, NFYB, BACH1, MAFK, GABPA, SP1, NFE2, CTCFL
IMR90	CTCF, MAFK, MAZ
HeLa-S3	CTCF, JUND, CEBPB, JUN, STAT1, MAFK, STAT3, PRDM1, MAZ, USF2, NFYB, FOS
HUVEC	CTCF, GATA2
NHEK	CTCF

Cell line	Common TFs of top 40% feature importance in promoter region
GM12878	EGR1, RUNX3, PAX5, ELF1, MAZ, NRF1
K562	ELF1, MAZ, EGR1, ZNF384, YY1, CTCF, TBP, E2F6, GABPA
IMR90	MAZ, CTCF
HeLa-S3	CTCF

Table S8: TFs with top 25% feature importance discovered both by TargetFinder (E/P/W) and PEP-Motif in enhancer/window region in six cell lines (the upper part) and TFs with top 40% feature importance discovered both by TargetFinder (E/P/W) and PEP-Motif in promoter region in four cell lines (the lower part). There are no TFs with top 40% feature importance estimated by TargetFinder (E/P/W) in HUVEC and NHEK. The displayed TFs are ordered by their feature importance estimated by TargetFinder (E/P/W).

Cell line	Features	Total	Comparable	Top 50%	Top 50%	Top 40%	Top 40%	Top 30%	Top 30%
	(P)	TFs	TFs	(T)	(T-P)	(T)	(T-P)	(T)	(T-P)
GM12878	100	85	60	22	14	10	6	2	2
K562	136	120	59	24	16	12	9	3	2
IMR90	56	23	7	2	2	2	2	0	0
HeLa-S3	73	58	29	12	6	4	1	0	0
HUVEC	23	8	6	1	1	0	0	0	0
NHEK	20	5	1	1	1	0	0	0	0

Table S9: Comparison of the predictive features discovered by TargetFinder(E/P/W) in promoter region with PEP-Motif. “(T)” in the header line represents TargetFinder (E/P/W). “(T-P)” represents TargetFiner (E/P/W) and PEP-Motif.

Feature in enhancer region	Feature in promoter region
CTCF, CTCFL	HOXD8, POU2F2, POU3F3, POU5F1B
VSX2	GSC2
CTCF, CTCFL	GLI3, GLI1, GLI2
CTCF, CTCFL	CTCF, CTCFL
NKX25	KLF16, MAZ, SP1, SP2
CTCF, CTCFL	ETV6, ELK1, ERG, ELK4
CTCF, CTCFL	AP2D
MEF2B	ETV6, ELK1, ERG, ELK4
CTCF, CTCFL	TF2LX, MEIS2, TGIF2
CTCF, CTCFL	TEF
BACH1, NF2L2, NFE2, MAFK(S)	CTCF, CTCFL
CTCF, CTCFL	KLF4, KLF1, KLF3
NR2C1, RARA, RARG, RARG(S)	CTCF, CTCFL
CTCF, CTCFL	STAT4, STAT1, STAT1(S)
NR2F1, NR2F1(S), NR1H2, NR2F2(S)	BACH1, NFE2, NF2L2, MAFK(S)
CTCF, CTCFL	FOXD3
PITX2	CTCF, CTCFL
CTCF, CTCFL	RUNX1, PEBB, RUNX3
ZNF713	HES5, HES7, HEY1
CTCF, CTCFL	HOXC8
CTCF, CTCFL	NRF1, ZNF639
PRDM1	CTCF, CTCFL
CTCF, CTCFL	GFI1B, GFI1
CTCF, CTCFL	SOX18
EHF(S)	MBD2
CTCF, CTCFL	MNT, SPIC
ZNF652	E2F1, TFDP1(S), E2F4
CTCF, CTCFL	PKNX1, PBX2, PBX1
CTCF, CTCFL	FOXF2, FOXA1, FOXA2
CTCF, CTCFL	KLF14
RARB	CTCF, CTCFL
PLAL1	RFX2, ZBT7B, RFX3, RFX4
ZNF219	MYC, MAX, MYCN
MEF2D, MEF2A, MEF2C	ETV6, ELK1, ERG, ELK4
STAT6	SP1(S)
CTCF, CTCFL	TBP
FOXO1	TEAD3
SOX3	RARG, NR2C1, RARA, RARG(S)
STAT1, STAT4, STAT1(S)	FOSB, FOSL1, FOS, JUND
HESX1, HEY2	TWST1, SNAI1

Table S10: Highly predictive interactions between the motif features in enhancers and the motif features in promoters (E-P feature interactions) shared by at least two cell types. The interactions are selected from the union of around top 300 important E-P feature interactions in each cell type. The E-P feature interactions appear in the order of their highest rank in the respective cell types. If a motif is from a motif cluster, all the members of this motif cluster are displayed in combination. For example, (CTCF, CTCFL) represents either CTCF or CTCFL since the two motifs are very similar and clustered.

C Supplementary Figures

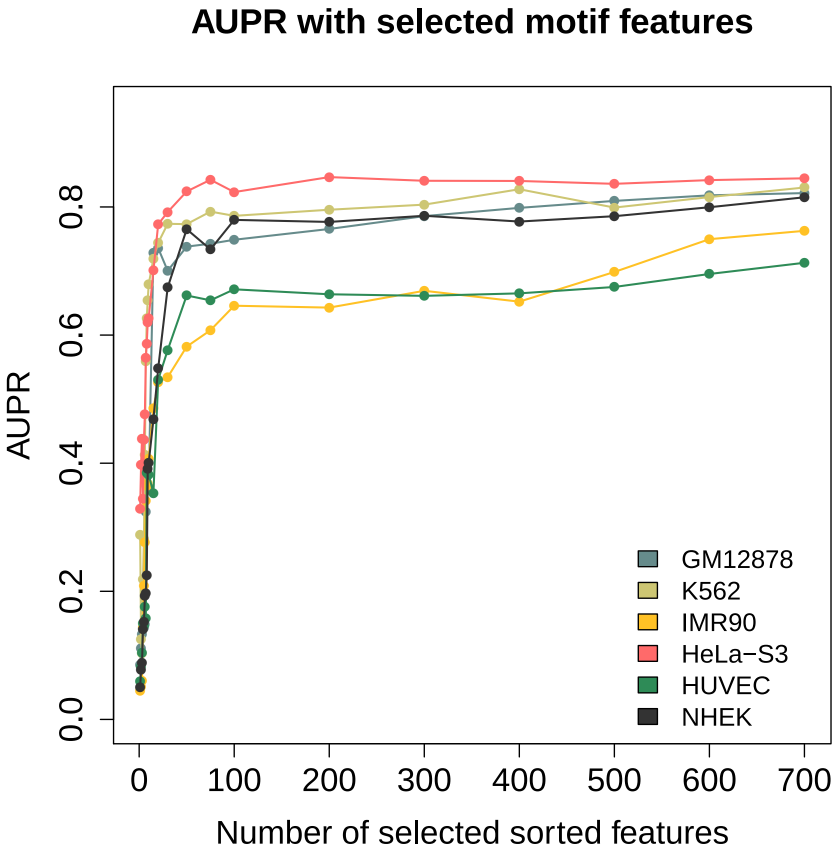

Figure S1: AUPR (Y-axis) of PEP-Motif with increasing number of selected TF motif features on E/P data from six cell lines. There are about 1280 dimensional motif features (enhancer-associated features and promoter-associated features concatenated together) for each cell line. The motif features were selected by their estimated feature importance in descending order. The sorted feature index (X-axis) represents the number of selected features.

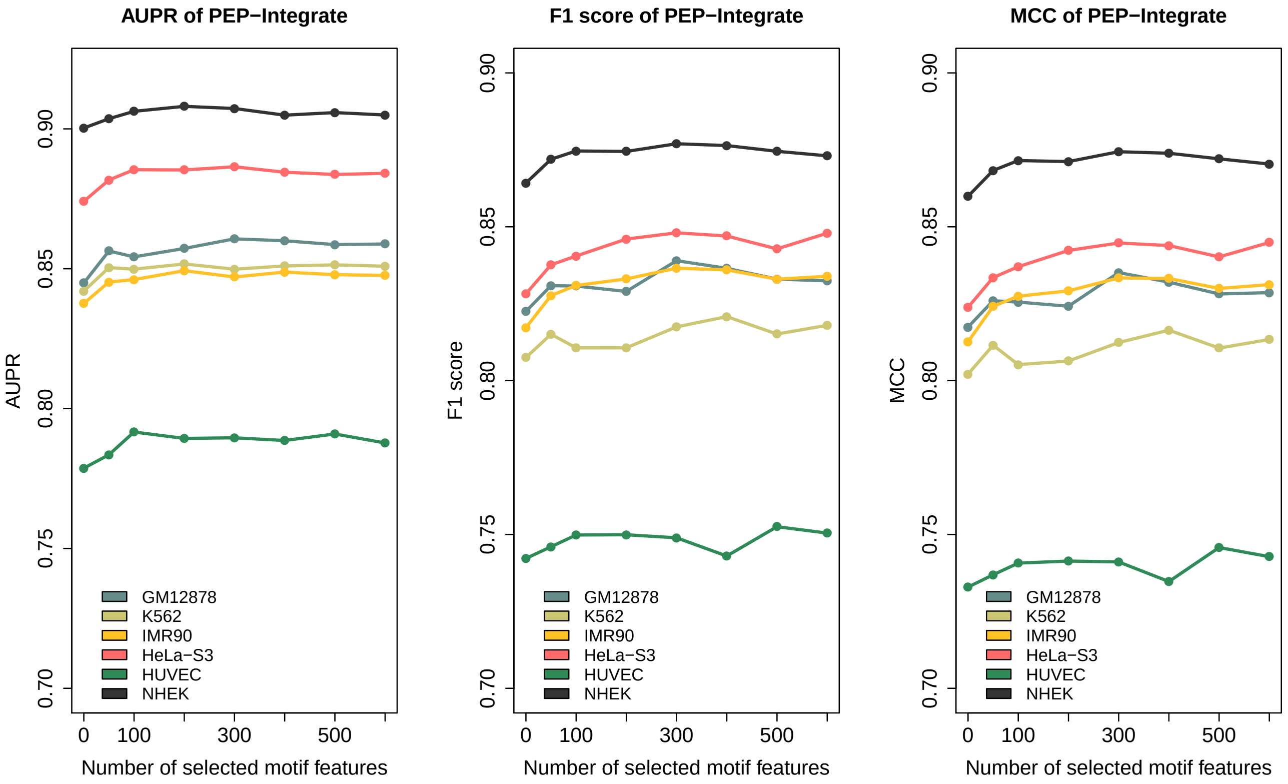

Figure S2: Performance evaluation of PEP-Integrate on E/P data of six cell lines with respect to different sets of selected motif features. The motif features were selected in the order of their estimated feature importance. The selected motif features were concatenated with features from PEP-Word for joint feature representation. AUPR, F_1 score, and MCC were used for evaluation.

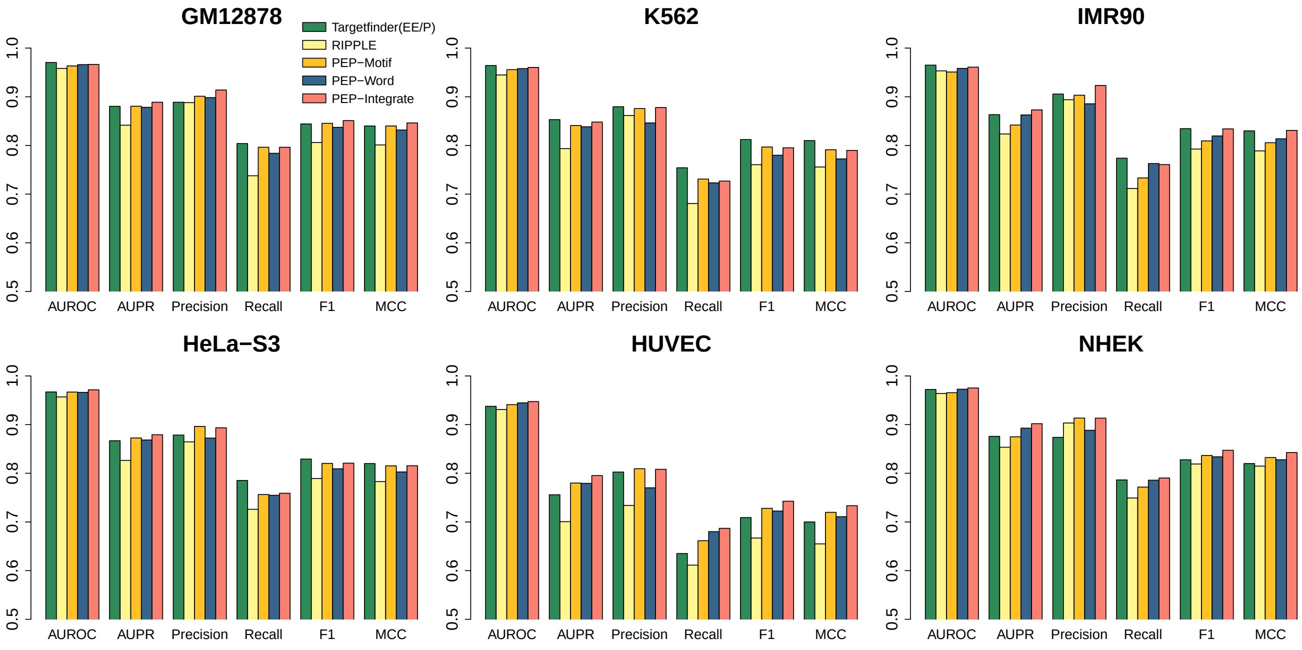

Figure S3: Performance evaluation of PEP-Motif, PEP-Word, and PEP-Integrate (K=6 for K-mer) on EE/P data of six cell lines in comparison with TargetFinder (EE/P) and RIPPLE.

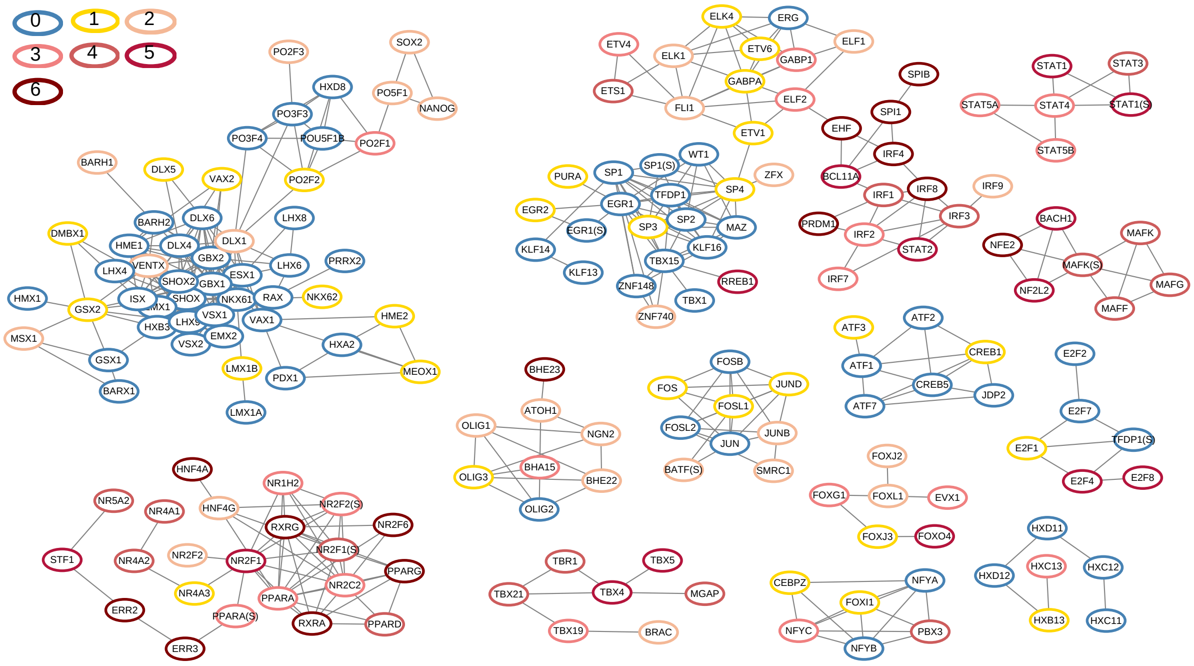

Figure S4: Examples of constructed motif clusters in the motif similarity connectivity graph. Each vertex represents a motif. The border color of a vertex indicates the number of cell lines in which the corresponding motif feature is top 25% important, as annotated in the graph, e.g., blue represents the motif is not top 25% predictive in any cell line and yellow represents the motif is top 25% predictive in one cell line. Components of the constructed graph that have more than 5 vertices and at least two vertices (motifs) possessing top 25% feature importance in at least one cell line are shown. Any two vertices (motifs) in a component are connected to each other by paths and vertices of different components are not connected. There are 357 components found in the graph, representing coarse-level motif clusters based on motif similarity. The largest three components consist of 51, 46, and 22 motifs, respectively, which are shown in the figure with another 10 smaller-scale components.

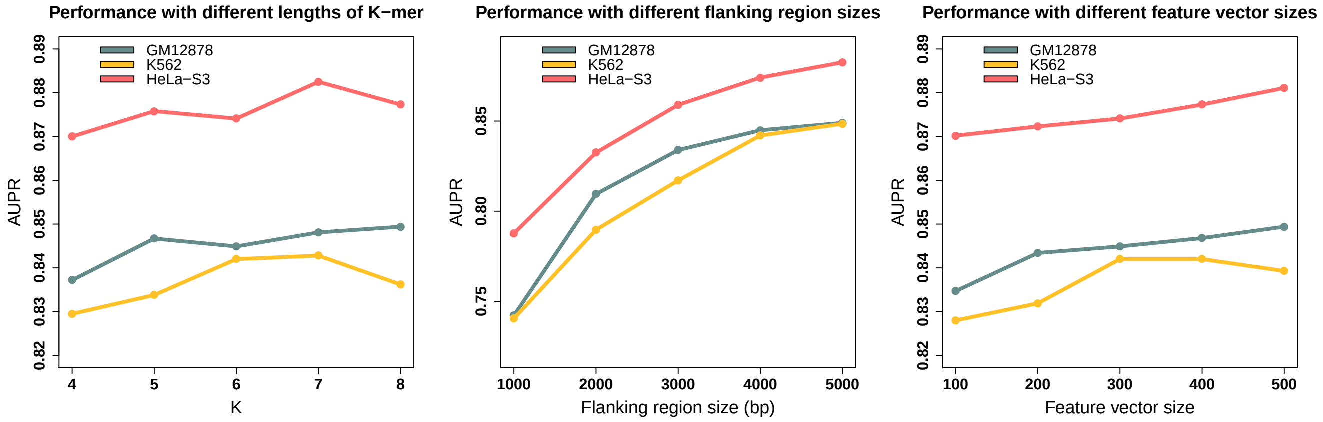

Figure S5: Performance evaluation of PEP-Word on E/P data on three cell lines (GM12878, K562, HeLa-S3) with respect to different choices of length of K-mer, length of flanking region and embedded feature vector size.

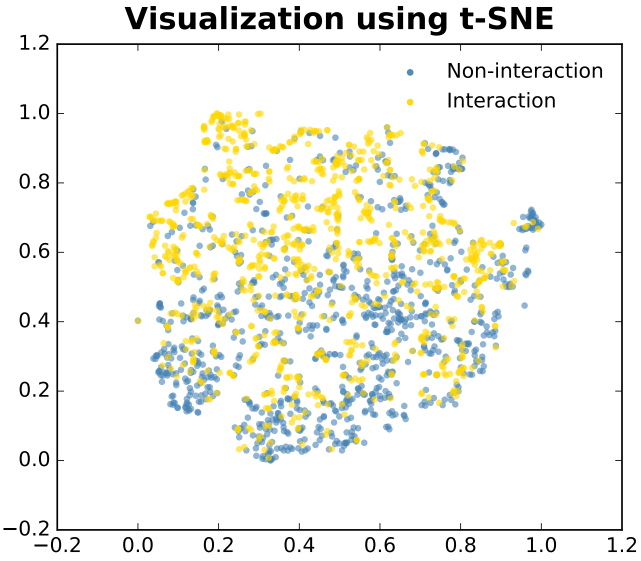

Figure S6: t-SNE visualization of a randomly selected set of 1000 positive samples and 1000 negative samples in GM12878. The 600 dimensional feature vector was first reduced to 64-dimensions using autoencoder and then reduced to two-dimensions for visualization. An autoencoder neural network was used in the first dimension reduction stage and output of the middle layer with 64 neurons was extracted for further dimension reduction and visualization.

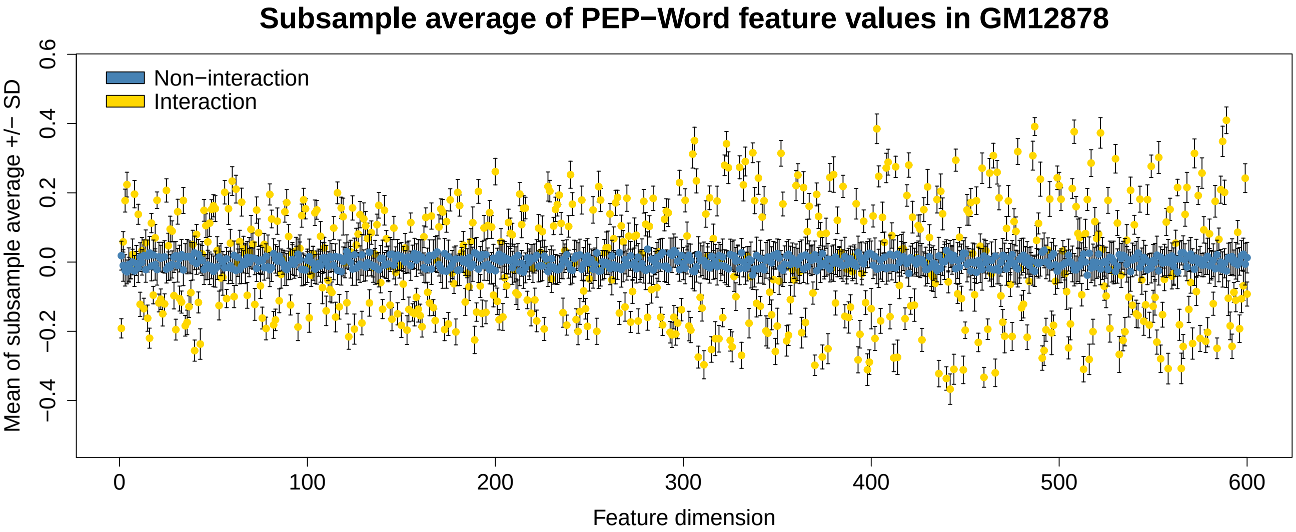

Figure S7: Mean value of subsample features (from PEP-Word) in cell line GM12878. 500 positive samples (EPI) and 500 negative samples (non-EPI) were randomly selected at the ith sampling step, named respectively as positive subsample S_{P_i} and negative subsample S_{N_i}. Mean of feature vectors of the subsample was calculated within S_{P_i} and S_{N_i}, respectively, noted as $\mathbf{x}_{P_i}$ and $\mathbf{x}_{N_i}$, both of which are n-dimensional vectors. We repeated the sampling 20 times and obtained subsample average feature vectors $\{\mathbf{x}_{P_i}\}_{i=1}^{20}$ and $\{\mathbf{x}_{N_i}\}_{i=1}^{20}$. On each feature dimension, we computed mean and standard deviation of $\{\mathbf{x}_{P_i}^{(k)}\}_{i=1}^{20}$ and $\{\mathbf{x}_{N_i}^{(k)}\}_{i=1}^{20}$ respectively, where $\mathbf{x}^{(k)}$ stands for the kth dimension of a feature vector, $k = 1, \cdots, n$. The mean and standard deviation of subsample averages on the each of n dimensions were shown in the figure. The sampling process was performed to compare the mean feature values with balanced positive samples and negative samples, in order to address comparison bias resulted from high imbalance of the datasets. We found that the positive samples and negative samples are distributed differently in the feature space constructed by PEP-Word.

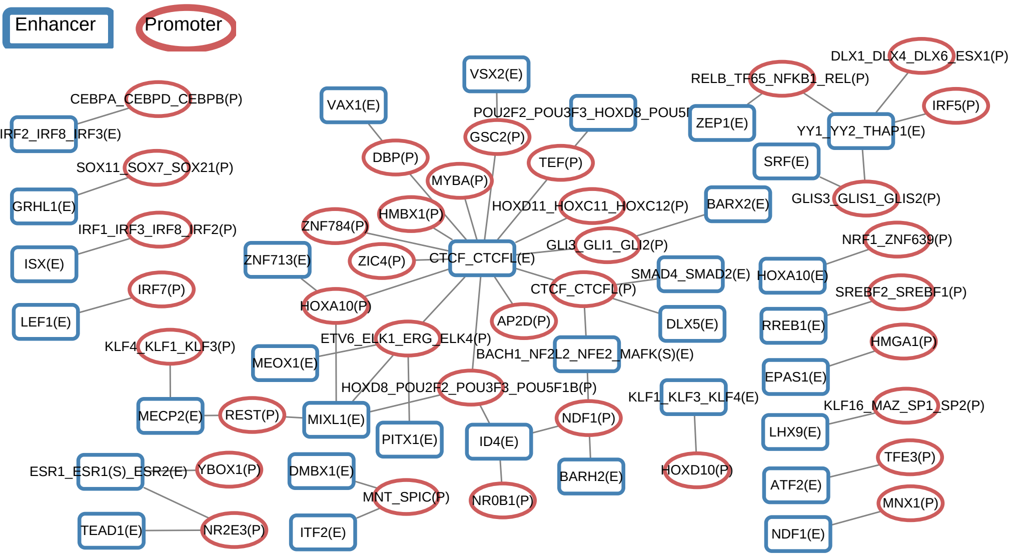

Figure S8: Examples of important interacting TFBS motif features in GM12878. The interactions are between enhancer-associated motif features and promoter-associated motif features.

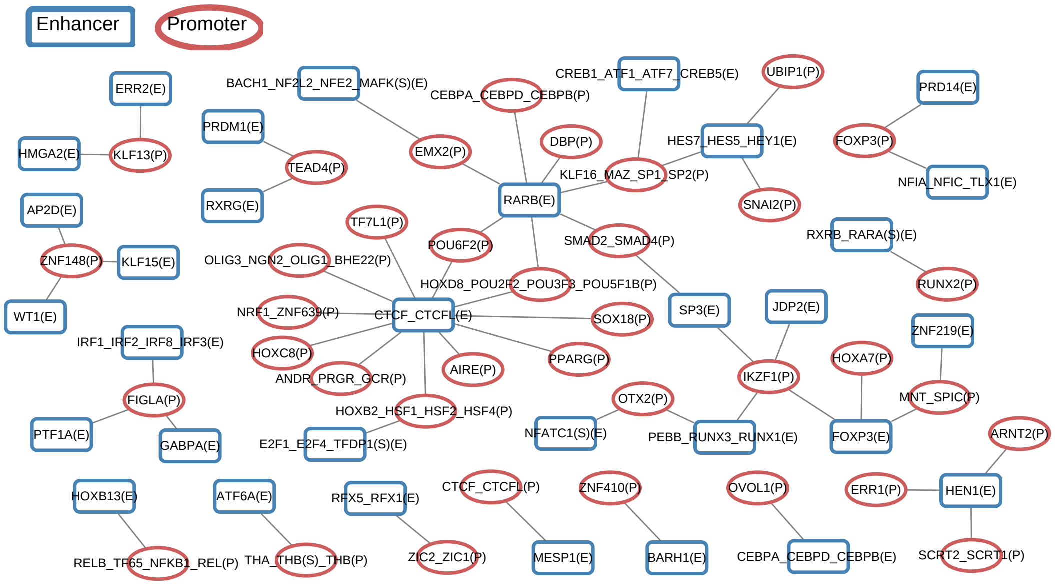

Figure S9: Examples of important interacting TFBS motif features in K562.

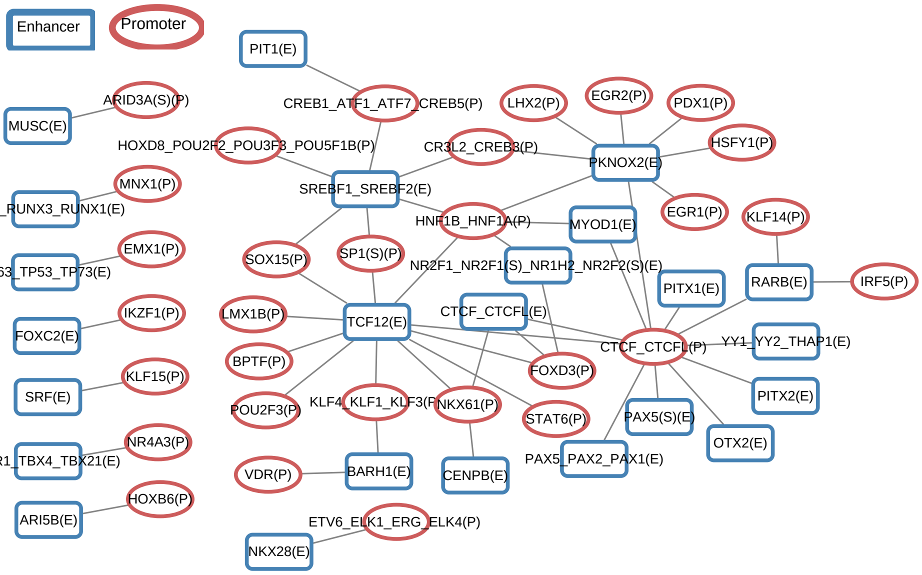

Figure S10: Examples of important interacting TFBS motif features in HeLa-S3.

References

[1] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001.

[2] A. Ali, S. M. Shamsuddin, and A. L. Ralescu. Classification with class imbalance problem. *Int. J. Advance Soft Compu. Appl*, 5(3), 2013.

[3] C. Lemnaru and R. Potolea. Imbalanced classification problems: systematic study, issues and best practices. In *International Conference on Enterprise Information Systems*, pages 35–50. Springer, 2011.

[4] J. Friedman, T. Hastie, R. Tibshirani, et al. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The Annals of Statistics*, 28(2):337–407, 2000.

[5] S. Gupta, J. A. Stamatoyannopoulos, T. L. Bailey, and W. S. Noble. Quantifying similarity between motifs. *Genome Biology*, 8(2):1, 2007.

[6] E. Hartuv and R. Shamir. A clustering algorithm based on graph connectivity. *Information Processing Letters*, 76(4):175–181, 2000.

[7] X. Wang and C. Qi. Action recognition using edge trajectories and motion acceleration descriptor. *Machine Vision and Applications*, pages 1–15, 2016.

[8] J. Leskovec, A. Rajaraman, and J. D. Ullman. *Mining of massive datasets*. Cambridge University Press, 2014.

[9] H. P. Luhn. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development*, 1(4):309–317, 1957.

[10] S. Robertson. Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of Documentation*, 60(5):503–520, 2004.

[11] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

[12] R. Řehůřek and P. Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010.

[13] B. Kostenko. XGBoost Feature Interactions Reshaped. https://github.com/limexp/xgbfir, 2016.

[14] A. P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

[15] J. Davis and M. Goadrich. The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 233–240. ACM, 2006.

[16] S. Whalen, R. M. Truty, and K. S. Pollard. Enhancer-promoter interactions are encoded by complex genomic signatures on looping chromatin. *Nature Genetics*, 48(5):488–496, 2016.