博士論文

Doctoral Thesis

A new computational method to predict transcriptional activity of a DNA sequence from diverse datasets of massively parallel reporter assays

(多様な超並列レポーターアッセイデータセットを用いた DNA 配列の内在的転写活性予測に関する新規計算手法の開発)

劉　瑩

Contents

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Gene transcription regulatory code is surely encoded in the sequences of cis-regulatory elements and revealing functional elements from *cis*-regulatory elements is a key of exploring its regulatory mechanisms of transcription. Massively parallel reporter assay (MPRA) technology is a kind of reporter assays based on DNA barcoding and next generation sequencing. The applications of MPRAs for different purposes produced a lot of data which contain the primary activities of target sequences. To analyze the functions of sequences, it requires a computational model to estimate the relation between sequences and their transcriptional activities. However, a computational method which could be applied to diverse MPRA data sets is not existed yet. In this research, I designed a computational method to predict transcriptional activities using DNA sequences and the corresponding activities by TRANSFAC database and machine learning algorithms of regression tree and MARS. According to the analysis of predictive functions which were estimated by the proposed method, it could reveal the active transcription factor binding sites (TFBSs). The proposed method could be applied to diverse MPRA as well as conventional luciferase reporter assay data sets despite of different transfected cell types (human cell lines, mouse and yeast), different sequence lengths (several ten bp to more than 1k bp), different number of sequences (several hundred to more than several ten thousand) and different sequence types (promoters, enhancers, artificial sequences, ChIP-seq peak regions and genomic variants). The applications of the proposed method also suggest that the method could predict the transcriptional activities of unknown sequences by using the predictive functions for known data sets.

1 INTRODUCTION

In 1970s, the central dogma was proposed by Francis Crick (1) and the processes of transferring genetic information in cells were characterized as transcription and translation, that is, DNAs produce RNAs and RNAs produce proteins. And in the past decades, the biological progresses of gene expression were rapidly investigated and detected.

Gene expression levels are regulated by complex biological processes, such as histone modification, chromatin structure and functional sequences. With the technology of the high-throughput sequencing (also known as the next generation sequencing) improving, the molecular mechanism of transcription is a key point of studying gene expression by investigating the DNA levels and mRNA levels in cells.

A lot of experimental technologies which are based on the next generation sequencing were developed to massively detect the different features related to gene expression regulation, such as ENCODE Project (2) for detecting the histone modifications, MPRA (massively parallel reporter assays) or MPRA-like methods (3–8) for measuring the primary activities of target sequences, STARR-seq (self-transcribing active regulatory region sequencing, (9)) for assaying the enhancer activity and Hi-C technology (10) for characterizing chromatin 3D-structure.

Transcription regulation is an important step in the processes of protein or RNA production and performs great contributions to final gene expression levels. Transcriptional initiation, elongation and termination are controlled by different transcription factors (TFs). Transcription factors bind to special DNA sequences which are usually called cis-regulatory elements. Thus, cis-regulatory elements contain gene transcription regulatory codes and the analysis of cis-regulatory elements could provide vital information for investigating regulatory processes of gene transcription (6, 11).

1.1 C*is*-regulatory elements

C*is*-regulatory elements is a kind of DNA regions which provide transcription factor binding sites (TFBSs) and recruit transcription factors binding to the corresponding sequences. Cis-regulatory elements are usually marked by several genomic and epigenomic characters. For example, several studies suggested that CpG islands, where high frequency of CpG dinucleotides locate, are involved in transcription regulation (12–14).

C*is*-regulatory elements contain two types of DNA sequences: promoters and enhancers (Figure 1). C*is*-regulatory elements have been used frequently to explore TF binding affinity in transcription processes (3, 4, 7, 15, 16). In recent research, however, several evidences suggested that the specificities between promoters and enhancers are not distinct. A proportion of promoters having enhancer activities were identified by STARR-seq (17) and the further study found the promoters which have enhancer activities could regulate distal gene expression by interacting with the promoter of transcribed genes (18).

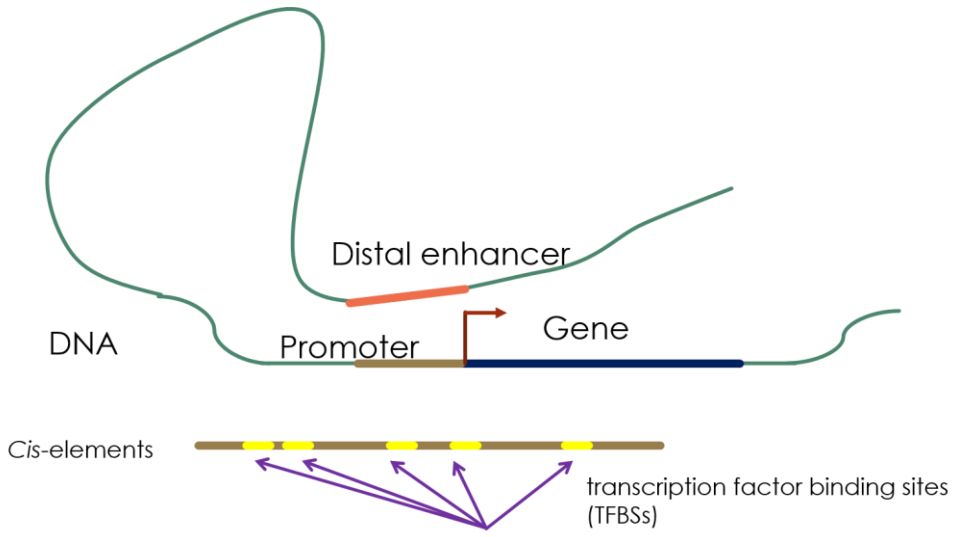


Figure 1: General structure of cis-regulatory elements.

1.1.1 Promoter

In eukaryotes, a promoter is a region in the proximal upstream of a gene and usually has the transcriptional start sites (TSS). The promoter regions contain different TFBSs and have the functions related to transcriptional initiation by transcription factors binding to TFBSs (Figure 1). During the gene transcription initiation of eukaryotes, the preinitiation complex, which is assembled by RNA polymerase and general transcription factors, binds to the promoter region and initiate the synthesis of transcripts (19).

1.1.2 Enhancer

In eukaryotic cells, an enhancer is a kind of chromatin regions which locate remotely from the coding genes and could be bound by transcription factors to regulate gene transcription (Figure 1). During the regulatory processes, the distal enhancers are considered as having the small spatial distances from the regulated or interacted regions. Enhancers are usually marked by different histone modifications such as H3K4me1 and H3K27ac (20).

1.2 Luciferase reporter assays

Luciferase reporter assay is a technology of quantitatively measuring gene expression levels and widely used for detecting the activity of transcription regulatory sequences. Luciferase reporter assays construct the plasmid vectors as the form of a target sequence inserting into upstream of the reporter gene. After transfection and cell culture, the levels of reporter gene expression could be identified by bio-luminesce of the reporter gene. However, the throughput of luciferase reporter assay is generally up to several thousand and difficult to satisfy the requirements of high- throughput measurements.

1.3 Massively Parallel Reporter Assay (MPRA)

Massively Parallel Reporter Assay (MPRA) is a kind of transient reporter assay of measuring the transcriptional activities based on next generation sequencing and barcoding technology. The same as conventional luciferase reporter assays, MPRA could measure the primary activities which are encoded as DNA sequences.

In MPRA, firstly, each target sequence is inserted in the upstream of a reporter gene and a random barcode is attached to the 3' site of the reporter gene to label the sequence. After the plasmid construction, the plasmid libraries which contain several thousand to several ten thousand (or more) constructs are transfected to *in vivo* or *ex vivo* cells. Then cell culture would allow the transcription processing of the reporter gene, and the mRNAs with attached random barcodes are extracted from the cells and reverse transcribed into cDNAs. Accordingly, the number of cDNAs could be counted by sequencing their random barcodes. On the other hand, the counts of DNA plasmids are also detected by sequencing the corresponding barcodes. In MPRA, the transcriptional activities are generally identified by the ratios of barcode counts of mRNA to the template DNAs (Figure 2).

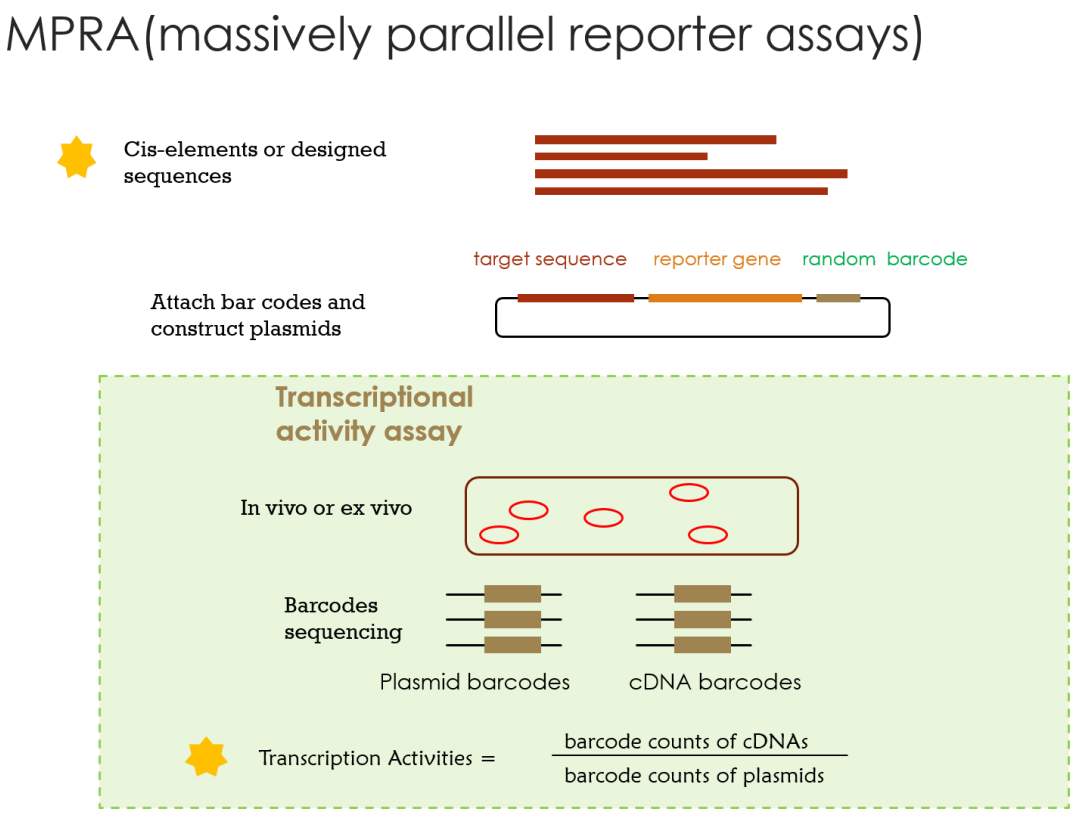


Figure 2. The scheme of Massively Parallel Reporter Assays. In MPRA, firstly, each target sequence is inserted in the upstream of a reporter gene and a random barcode is attached to the 3' site of the reporter gene to label the individual sequences. Then the plasmid libraries which contain several thousand to several ten thousand (or more) constructs are transfected to in vivo or ex vivo cells. After cell culture, the mRNAs with attached random barcodes are extracted from the cells and reverse transcribed into cDNAs. Accordingly, the number of cDNAs could be counted by sequencing the random barcodes. On the other hand, the counts of DNA plasmids are also detected by sequencing the corresponding barcodes. In MPRA, the transcriptional activities are generally identified by the ratios of barcode counts of mRNA to the template DNAs.

1.4 Machine learning algorithms in this study

In this research, machine learning algorithms were considered for training data sets and constructing predictive functions of transcriptional activities. I introduced several machine learning algorithms to construct the computational method of predicting transcriptional activities and several other algorithms for method comparisons.

1.4.1 Regression Tree

A regression tree (21) is a decision tree learning algorithm which could be applied for both of classification and regression. The result of the regression tree is showed via a tree structure. In a regression tree, a leaf indicates a cluster (or a class) and a branch of two leaves in a higher level of the tree indicates that the samples represented by the branch are separated into two clusters.

There is an example of regression tree with the TFBS enrichment scores as explanatory variables and the transcription activities as response variables. In the tree, the logical conditions in the nodes indicate that the samples in left branches satisfy the corresponding conditions and the right branches do not. The values and percentages in all the nodes show the average values of the subpopulations and the proportions of samples.

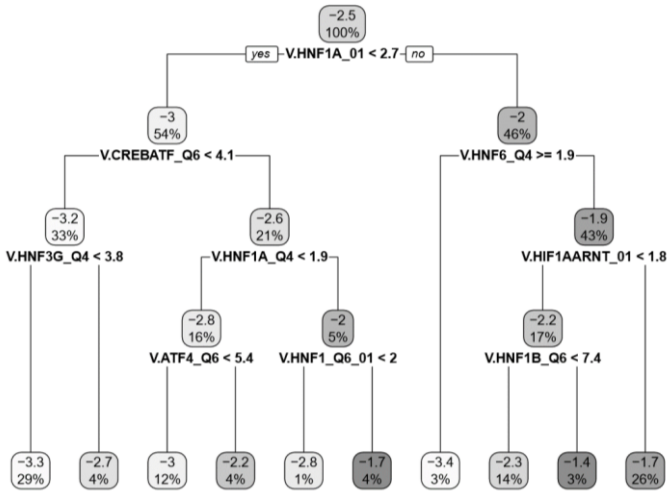


Figure 3. An example of regression tree with TFBS enrichment scores as explanatory variables. In the tree, the logical conditions in the nodes indicate that the samples in left branches satisfy the corresponding conditions and the right branches do not. The values and percentages in all the nodes show the average values of the subpopulations and the proportions of samples.

However, it is known that a limitation of regression tree is its vulnerability to over-fit. To avoid over-fitting, the depths of regression tree are usually modified by several parameters of controlling the tree structure. In this research, for diverse data sets, it is unreasonable to customize different tree structures for different properties of data sets. Here, I designed a feature redundancy-dependent formula to automatically control the tree structures for being applied to different data sets.

1.4.2 Multivariate Adaptive Regression Splines (MARS)

MARS (22) is a well-known algorithm of regression that builds its response functions in the form of splines. It employs hinge functions and/or productions of different hinge functions to construct the predictors of response variables. A hinge function has the form of *max(0,x-c)* or *max(0,c-x)*

which could capture the features of switch-on and switch-off. Here, *c* is a constant estimated by MARS and *x* is selected from explanatory variables.

There is an example of regression result of MARS with the TFBS enrichment scores as explanatory variables and the transcriptional activities as response variables. The functions of h(-) indicate hinge functions that construct the predictors. The coefficients indicate the relative effects of the corresponding predictors on the response variables.

Table 1. An example of output of MARS with the TFBS enrichment scores as explanatory variables and the transcriptional activities as response variables. The functions of h(-) indicate hinge functions and construct the predictors. The coefficients indicate the relative effects of the corresponding predictors on the response variables.

Predictors	Coefficients
(Intercept)	-0.46
h(V$REST_Q5-1.704)	2.93
h(1.704-V$REST_Q5)	0.23
h(V$TATA_01-10.445)	-0.02
h(10.445-V$TATA_01)	0.07
h(V$HNF3B_Q6-2.731)*h(10.445-V$TATA_01)	-0.01
h(2.731-V$HNF3B_Q6)*h(10.445-V$TATA_01)	-0.01
h(V$MZF1_Q5-0.967)*h(1.704-V$REST_Q5)	0.24
h(V$AP2ALPHA_03-3.514)*h(1.704-V$REST_Q5)	-0.11
h(3.514-V$AP2ALPHA_03)*h(1.704-V$REST_Q5)	-0.02
h(V$DBP_Q6-8.957)*h(1.704-V$REST_Q5)	0.11
h(8.957-V$DBP_Q6)*h(1.704-V$REST_Q5)	-0.01

MARS generate its predictors by two steps: the step of adding predictors based on reducing the sum-of-squares residual (RSS) and step of removing predictors according the generalized cross validation (GCV). In the training process of MARS, MARS recursively adds new predictors and then removes the less effective predictors to avoid over-fitting (22).

Furthermore, because MARS is a non-linear regression algorithm, the degree of model, that is the number of hinge functions whose production constructs one predictor, usually is used for controlling the complexity of being trained models.

1.4.3 Multiple linear regression (MLR)

MLR (23) is a supervised machine learning algorithm which builds a relation between multiple explanatory variables and response variables in a linear way. MLR is a very simple model and widely used in informatics and bioinformatics field because it has the interpretable model structure and costs low time. MLR basically calculates the solution of response functions by minimum least square.

If given a matrix of explanatory variable $X^T = (X_1, X_2, \ldots, X_p)$ to predict Y via multiple linear regression, the model builds the predictive function as:

$$Y = \beta_0 + \sum_{j=1}^{p} X_p \beta_p$$

Where X is the matrix of explanatory variables and Y is the vector of response variables and β is the vector of coefficients which are estimated by multiple linear regression.

1.4.4 Lasso regression

Lasso regression (24) is a kind of regression analysis with feature selection and the number of selected features is user-dependent by setting the corresponding parameters. The same as MLR, Lasso regression builds a linear relation between explanatory variables and response variables.

If given a matrix of explanatory variable $X^T = (X_1, X_2, \ldots, X_p)$ to predict Y via Lasso regression, Lasso regression builds its response functions by minimizing:

$$\frac{1}{2}\sum_{j=1}^{p}(y_p - X_p\mathrm{B})^2 + \lambda\sum_{j=1}^{p}|\beta_p|$$

Where X is the matrix of explanatory variables and Y is the vector of response variables and β is the vector of coefficients which are estimated by Lasso regression. λ is the parameter that controls the number of selected variables in the final model.

1.4.5 Bayesian quantile regression (BQR)

Bayesian quantile regression (BQR, (25)) is a kind of quantile regression and widely used for different bioinformatics related problems. Different from MLR, it estimates the quantiles of response variables rather than the means and gives the solution in a form of probability distribution.

1.5 TRANScription FACtor (TRANSFAC) database

TRANSFAC ((26, 27), TRANScription FACtor) is a well-known eukaryotic TFBS profile database which is frequently used for searching known TFBSs from DNA sequences. In this study, I

introduced the TRANSFAC database for encoding DNA sequences into TFBS enrichment scores.

By searching TRANSFAC, the candidate TFBSs and their positions, stands, matrix scores and core scores are provided for a DNA sequence. The Position Weight Matrix (PWM) scores could be calculated by different TFBS profiles such as vertebrate, fungi and tissue specific for different sequence types.

1.6 Quantitative Sequence-Activity Model (QSAM)

A Quantitative Sequence-Activity Model (QSAM) (28) is a computational model for predicting transcriptional activities at a single-nucleotide resolution. That is, a nucleotide at a position is encoded into a binary code (0-1) and one position has three variables because of four types of nucleotides. The number of variables of QSAM is the three times of the sequence length. The QSAM could calculate the scores of individual positions along the sequences and independent of other databases. The property of database-free gives QSAM the ability to investigate unknown functional elements. However, from the variable encoding processes of QSAMs, we could find that the QSAMs are only adaptive to the sequence library with an equal length.

In the former study of (8), they employed QSAMs to predict transcriptional activities for their MPRA data.

1.7 Purposes of this study

In recent years, the applications of MPRA technology were dramatically increasing for diverse purposes. However, a computational method which could analyze diverse MPRA data is not existed yet. In this research, I want to construct a new computational method to inquire into the

behavior of transcription factors binding to cis-regulatory elements from different MPRA data sets.

Estimating the active TFBSs of given sequences, it could provide the clues of transcriptional processes and could give information of predicting the transcriptional activities of new sequences. Furthermore, if we know the active TFBSs and their effects, it also could provide the messages of designing cis-regulatory elements.

2 METHODS

The proposed method consists of four steps: 1. Because the data sets were derived from different studies for different purposes, the formats of different data sets were not unified. Here, data pre-processing was performed for making a unique format of different MPRA data sets; 2. For data training, it is required to encode the target DNA sequences into explanatory variables. I used the TRANSFAC database to encode individual sequences into different variables and construct the explanatory variable matrix; 3. For some data sets, which usually have relative large data sizes and/or sparse distributions of features, it is difficult to build a unique predictive function to characterize the relation between DNA sequences and the corresponding transcriptional activities because of the diverse sequence patterns. Hence, the process of clustering variables was introduced to assemble variables into more compact subpopulation by regression tree; 4. After clustering, I performed MARS in individual clusters to construct predictive functions (Figure 4).

In the details of the TRANSFAC database searching step, sequences were characterized into TFBS enrichments scores which indicate the copy number of individual TFBSs in the corresponding sequences. In the variable clustering step, conventional regression tree was not suitable for diverse data sets and I designed a formula for regression tree to be automatically adapted to different data patterns.

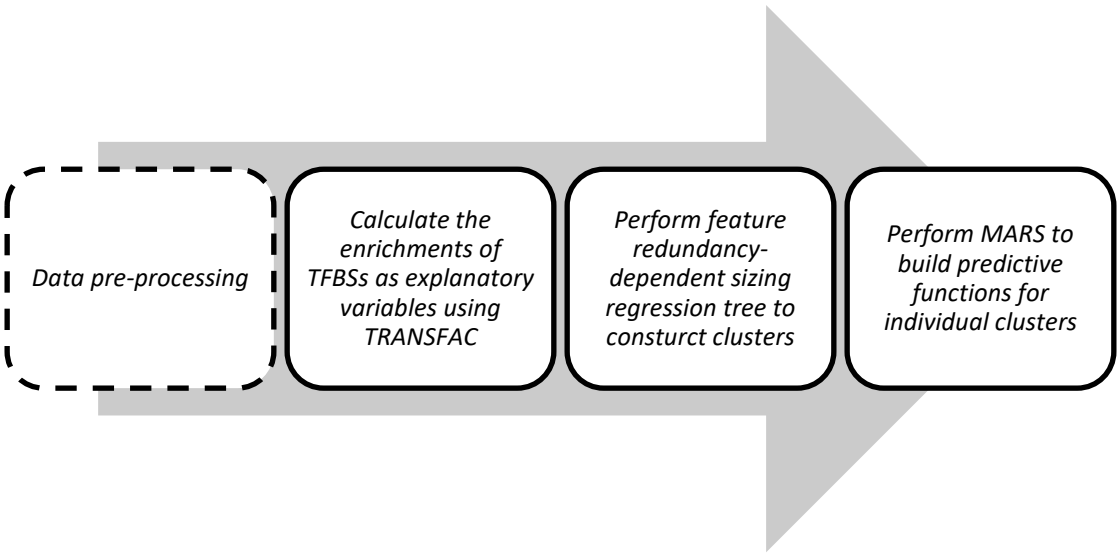


Figure 4. Workflow of proposed method.

2.1 Data sets

To exam the performance of the proposed method, I selected 10 public data sets from 8 previous works (Table 2) which contain 8 MPRA data sets and 2 data sets of conventional luciferase reporter assays. MPRA data sets generally have relative larger data sizes and smaller sequence lengths than the data sets of conventional luciferase reporter assays. And the sequence types of MPRA data sets are variety such as promoters, enhancers and ENCODE segments. The details of individual data sets were described as follows.

"CREInducedInHEK293": 10% unbiased-random mutations were introduced into the 87-nt CRE (cAMP response element) enhancer and obtained about 27,000 constructs. The MPRA was applied in HEK293.

"DHSInMouseRetina": 3,500 DNase I hypersensitive sites with unequal sequence lengths were assayed in mouse retina;

"TFBS75InYeast": 6,016 artificial designed sequences with length of 103 bp were assayed in yeast;

"TFBS12InHepG2Mouse": 12 liver-specific TFBSs with equal length of 168bp were assayed in HepG2 and Mouse;

"RBCVariantsGATA1InK562": 2,756 SNPs were assayed in normal K562 cells and GATA1 overexpression K562;

"PromoterLucInHEK293": 734 promoter sequences were assayed in HEK293 by luciferase reporter assays;

"CREBBPInMouseNeuron": 253 distal enhancers and 234 promoters were assayed by MPRA and STARR-seq;

"PromoterLuc8celltypes": Promoters were assayed in 8 cell types by luciferase reporter assays.

In addition, data sets of "*CREBBPInMouseNeuron*" and "*PromoterLuc8celltypes*" contain reporter assays under multiple (>2) experimental conditions, and I constructed integrated predictive functions across different conditions.

Table 2.The basic information of data sets.

Data sets	Description	Construct lengths	Cell types	Assayed loci	# of Constructs	Reference
CREInducedInHEK293	CRE enhancer with 10% random mutations	87bp	HEK293	*ex vivo*	27000	(8)
DHSInMouseRetina	3500 DNase I hypersensitive sites	181-703bp (median 466bp)	mouse retina	*ex vivo*	27161	(6)
TFBS75InYeast	Designed 75 yeast TFBSs	103bp	yeast	*in vivo*	6016	(16)
TFBS12InHepG2Mouse	12 liver-specific TFBSs assayed in HepG2 and Mouse	168bp	mouse, HepG2	*in vivo, ex vivo*	4742	(7)
RBCVariantsGATA1InK562	2,756 SNPs assayed in GATA1 overexpression+/- K562	145bp	K562	*ex vivo*	15733	(5)
PromoterLucInHEK293	Promoters	755-1201bp (median 1081bp)	HEK293	*ex vivo*	734	(29)
CREBBPInMouseNeuron	253 distal enhancers and 234 promoters assayed by MPRA and STARR-seq	139bp	mouse cortical neurons	*ex vivo*	3409	(17)
PromoterLuc8celltypes	Promoters assayed in 8 cell types	614-1301bp (median 983bp)	Ags G402 HCT116 Hela Hepg2 HT1080 T98G U87mg	*ex vivo*	4575	(30)

2.2 Data pre-processing

Data pre-processing aims to format the data sets from different studies. The transcriptional activities of MPRAs were calculated by the log2 ratios of mRNA tag counts to template DNA tag counts of identical barcodes (except for "*TFBS75InYeast*"). And for the data sets of luciferase reporter assays, the transcriptional activities were identified by log2 of reporter gene expression. On the other hand, I also removed the samples with 0 DNA tag counts and added further process into the samples with 0 mRNA tag counts. The data pre-processing for individual data sets are showed as the following description.

"CREInducedInHEK293": The samples with 0 DNA tag counts were removed and the transcriptional activities were calculated by the log2 ratios of mRNA tag counts to DNA tag counts;

"DHSInMouseRetina": In the data set, the transcriptional activities were calculated by log2 ratios of mRNA tag counts to DNA tag counts. If the samples have 0 mRNA tag counts, the pseudo value of 0.001 was added before performing the logarithm. There are three experimental replicates of MPRAs in the data set, and I removed the samples that the standard deviations of the activities were smaller than 3;

"TFBS75InYeast": The transcriptional activities were measured by YFP or mCherry expression and log2 was performed;

"TFBS12InHepG2" and "*TFBS12InMouse"*: In the study, the three MPRA replicates were applied, and I took the average values of the three replicates as the corresponding activities;

"RBCVariantsCtrlInK562" and "*RBCVariantsGATA1InK562"*: In the data sets, there are three replicates MPRAs. The raw counts of RNA and template DNA were provided, and I calculated the transcriptional activities by taking ratios of RNA counts to DNA counts. Log2 ratios

were calculated for the median values of the three replicates (described by “http://www.bloodgenes.org/RBC_MPRA/RBC_MPRA_code.html”);

“CREBBPInMouseNeuron”: There are also two replicates of MPRAs and STARR-seq, and the average of log2 experimental values were used as the corresponding transcriptional activities.

2.3 TRANSFAC searching

For different purposes, the different studies usually had different sequence patterns. There is a problem with directly handling the diverse sequence patterns from the independent studies.

To encode diverse sequence patterns into a uniform format, I used the TRANSFAC database to search the candidate TFBSs occurring in individual sequences. And then the enrichment scores were calculated for individual TFBSs in the corresponding sequences.

The TRANSFAC database could characterize the candidate TFBSs in a sequence by their positions, orientations, PWM scores, core scores and TFBS consensus sequences (Figure 5). A well-known limitation for TRANSFAC database is the high false positive rates. To reduce the false positive rates which result from TRANSFAC database searching, I wanted to use the relative highly significant features as explanatory variables. Here, I ignored the relative trivial features which are provided by TRANSFAC results but have limited effects on transcriptional activities such as positions and strands.

TFBS labels	position (strand)	matrix score	core score	consensus sequence
V$CREBP1_01	12 (+)	0.897	0.869	TGACGtca
V$CREBP1_01	12 (-)	0.897	0.869	tgaCGTCA
V$CREBP1_01	35 (-)	0.897	0.598	ggcCGTCA
V$CREBP1_01	43 (-)	0.717	0.590	tacTGTGA
V$CREBP1_01	48 (+)	0.897	0.743	TGACGtct
V$CREBP1_01	48 (-)	0.663	0.718	tgaCGTCT
V$CREBP1_01	70 (+)	0.897	0.869	TGACGtca
V$CREBP1_01	70 (-)	0.897	0.869	tgaCGTCA
V$NF1_Q6	12 (-)	0.921	0.899	tgacgtcagctGCCAGat
V$NF1_Q6	31 (+)	0.911	0.888	ccATGGCcgtcatactgt
V$RREB1_01	29 (+)	0.873	0.615	tCCCATggccgtca
V$RREB1_01	64 (+)	0.873	0.652	cCCCATtgacgtca
V$RFX1_01	18 (-)	0.738	0.697	caGCTGCcagatcccat
V$RFX1_01	37 (+)	0.802	0.737	ccgtcatactGTGACgt
V$RFX1_01	50 (+)	0.786	0.714	acgtctttcaGACACcc
V$GEN_INI_B	2 (+)	0.823	0.795	cacCAGAC

Figure 5. Examples of TRANSFAC searching results.

Furthermore, for predicting the transcriptional activities, I also intended to construct simple computational models with a small number of predictors which have relative high contributions to transcriptional activities. Thus, only the TFBS enrichments were selected as the explanatory variables for the computational model training. The TFBS enrichment score was calculated as formula (1), the summation of PWM scores for identical TFBS were calculated as one explanatory variable. All the TFBS enrichment scores aligned by different sequences and constructed an explanatory variable matrix for next processing step.

$$\text{TFBS enrichment score}_{ij} = \sum_{\text{k}} \text{PWM matrix scores of } k\text{-th TFBS } i \text{ in sequence } j \qquad (1)$$

For the data sets having more than two experimental conditions, additional variables (e.g. cell types) were introduced into the explanatory variable matrix in the binary (0-1) format (Figure 6).

An example of explanatory variable matrix for experimental conditions < =2

V$AIRE_01	V$AML1_Q5	V$AP1_Q6_02	V$AP2ALPHA_03	V$ARID5A_03	V$BBX_03	V$BBX_04	V$BCL6_Q3_01	V$BEN_01	V$BLIMP1_Q6_01	
6.318	2.795	4.307	2.611	2.516	2.987	8.102	3.657	7.088	0	
4.475	3.614	5.871	7.457	0	0	1.427	0.852	7.26	0	
3.861	2	9.587	10.33	0.815	5.428	4.603	6.362	15.28	0.889	
5.563	0.884	2.65	11.656	0	0	1.447	1.81	26.05	0.854	.....
5.292	1.908	5.467	8.498	0	4.727	9.244	0	18.605	0	
10.289	0	4.348	11.534	0.803	1.49	1.457	5.422	22.584	0	
6.385	2.724	4.318	6.092	1.716	1.513	5.08	5.412	13.694	0.873	
4.289	0	7.541	5.924	0.805	4.78	6.795	1.711	18.493	0.89	

An example of explanatory variable matrix for experimental conditions >2

ht1080	g402	t98g	hct116	hela	hepg2	ags	u87mg	V$AHR_Q6	V$AIRE_01	V$AML1_Q5	V$AP1_Q6_02	V$AP2ALPHA_03	
1	0	0	0	0	0	0	0	0	15.121	1.911	17.466	7.398	
1	0	0	0	0	0	0	0	0	9.418	2.822	10.626	2.66	.....
1	0	0	0	0	0	0	0	3.994	18.722	3.703	19.589	9.757	
1	0	0	0	0	0	0	0	4.992	20.584	4.535	12.808	30.13	
1	0	0	0	0	0	0	0	0	17.517	4.587	20.509	13.698	
1	0	0	0	0	0	0	0	1.998	13.3	5.538	13.903	28.619	
1	0	0	0	0	0	0	0	3.994	8.717	5.535	8.617	25.493	
1	0	0	0	0	0	0	0	0.998	13.932	3.597	9.613	27.166	
1	0	0	0	0	0	0	0	1.996	14.827	0.908	2.546	16.279	

Figure 6. Examples of explanatory variable matrices. (Upper) An example of explanatory variable matrix for data sets with experimental conditions <=2; (Lower) An example of explanatory variable matrix for data sets with experimental conditions>2 and additional variables (e.g. cell types) were introduced into the explanatory variable matrix in the binary (0-1) format.

2.4 Variable clustering

From the PCA projection of explanatory variables of different data sets, we could find that their feature patterns are diverse (Figure 7). Considering the diversity of the different data sets, the samples with different sequence patterns were encoded into relative sparse variables in some data sets. For these data sets, it is difficult to use a single predictive function to characterize sparse variable patterns. Here, assembling samples into different clusters is a reasonable solution to separate samples into several subpopulations with more compact features.

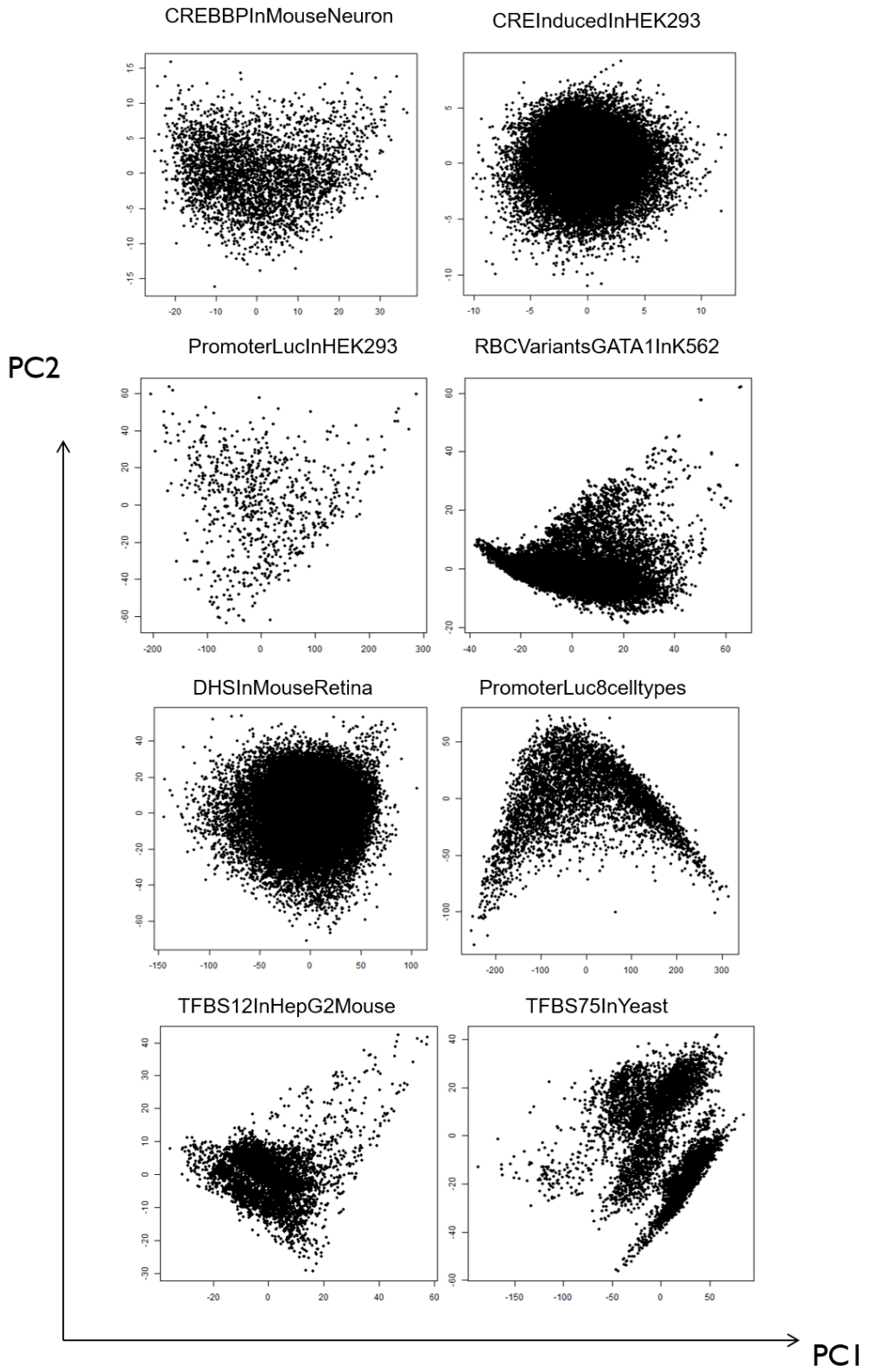


Figure 7. The PCA Projections of TFBS enrichment scores onto PC1 and PC2 of different data sets.

Here, regression tree was introduced into this study because of its interpretable and visible results and thus, the biological meaning of different clusters could be simply understood. However, conventional regression tree could not be adapted to diverse data sets and suffered into over-fit for some data sets. To overcome this problem, I considered to use the properties of the input variables to avoid over-fit. A formula which could automatically control regression tree structure according the properties of variables was proposed to modify the regression tree.

The formula, that is called "*minbucket*" (In R package "rpart", the formula is used to specify the "*minbucket*" parameter), indicates the minimum number of samples of all the clusters. The rule of calculating "*minbucket*" is defined as formula (2) which is based on the feature redundancy of explanatory variables.

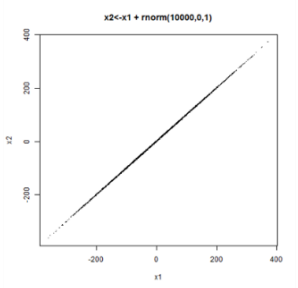
$$minbucket = \frac{2^{-\text{Proportion of variance of the first principal component}*1e+07}}{number\ of\ observations} \quad (2)$$

In the formula (2), the feature redundancy of an explanatory variable matrix was presented by the proportion of variance of the first principal component (PC1) in a minus exponential form. The proportion of variance of PC1, which is also calculated by the proportion of the first eigenvalue of covariance of the variables, has the information of variable redundancy.

For example, supposing there are two variable vectors X1 and X2 and setting X2=X1 + random values with the normal distribution (which has the mean value of 0 and standard deviation of 1), we could see that X1 and X2 have very high redundancy (almost equal) and the proportion of variance of the first component is approximately 1 (The upper panel of Figure 8).

For another example, supposing there are two variable vectors X1 and X2 too. And setting X2=X1 + random values with the normal distribution (which has the mean value of 0 and standard deviation of 100). We could see that the redundancy of this case is smaller than the first case because of the relative larger standard deviation of the added random values (or noise). For this example, the proportion of variance of the first component is approximately 0.87, which is also smaller than the first case (The middle panel of Figure 8).

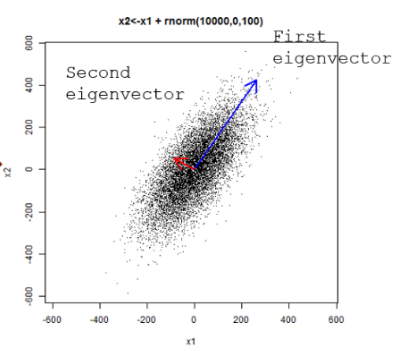
Furthermore, if there are two variable vectors X1 and X2 and they are independent. That is, I set different random values with the normal distribution (which has the mean value of 0 and standard deviation of 100) to X1 and X2, respectively. We could see that the two variable vectors almost have no redundancy and the proportion of variance of PC1 is approximately 0.5 (The lower panel of Figure 8).

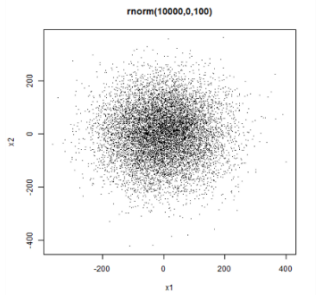


X2=X1+ random values with normal distribution(mean=0 ;sd=1)
Proportion of variance of first component=0.9999746

X2=X1+ random values with normal distribution(mean=0 ;sd=100)
Proportion of variance of first component= 0.8702066

```
Covariance of (x1,x2)   Eigenvalues:
   x1         x2        25735.133 3838.456
x1 10022.994  9857.602  First eigenvalues/summation
x2  9857.602 19550.594  of all the eigenvalues=
                        0.8702066
```





X1= random values with normal distribution(mean=0 ;sd=100)
X2= random values with normal distribution(mean=0 ;sd=100)
Proportion of variance of first component= 0.5084375

Figure 8. Three cases of simulations of two variable vectors. (Upper) supposing there are two variable vectors X1 and X2 and setting X2=X1 + random values with normal distribution (which has the mean value of 0 and standard deviation of 1), X1 and X2 have very high redundancy (almost equal) and the proportion of variance of the first component is approximately 1. (Middle) Setting X2=X1 + random values with normal distribution (which has the mean value of 0 and standard deviation of 100). The redundancy of this case is smaller than the first case (Upper one) because of the relative larger standard deviation of random values (or noise). For this example, the proportion of variance of the first component is approximately 0.87. (Lower) If there are two variable vectors X1 and X2 and X1 and X2 are independent. That is, I set the different random values (which has the mean value of 0 and standard deviation of 100) to X1 and X2, respectively. The two variable vectors almost have no redundancy and the proportion of variance of the first component is approximately 0.5

In the formula (2), the feature redundancy is divided by the number of observations which means that the relatively larger data sets should build more number of clusters. Other constants in the formula of "*minbucket*" are introduced to allow the values calculated by the formula (2) for different data sets loading within a desired scale.

The values of proportion of variance of PC1 and "*minbucket*" were calculated for individual data sets (Table 3). We could find that the data set "*CREInducedInHEK293*" has the lowest level of proportion of variance of PC1, it is probably because the sequences of "*CREInducedInHEK293*" are the mutant CRE enhancers with 10% unbiased-random mutations. We also could see that both promoter data sets of "*PromoterLuc8celltypes*" and "*PromoterLucInHEK293*" have relative high feature redundancies.

Table 3. Values of the proportion of variance of the first component and "minbucket" of different data sets.

Data set	Proportion of variance of PC1	*"minbucket"*
CREBBPInMouseNeuron	0.37	566.06
CREInducedInHEK293	0.10	354.31
DHSInMouseRetina	0.47	266.57
PromoterLuc8celltypes	0.78	159.29
PromoterLucInHEK293	0.71	8318.43
RBCVariantsCtrlInK562, RBCVariantsGATA1InK562	0.48	456.04
TFBS12InHepG2, TFBS12InMouse	0.35	1657.98
TFBS75InYeast	0.50	1177.21

The improvements of predictive precisions by introducing the formula (2) into a conventional regression tree are shown in the results (Figure 9). All the predictive precisions in this study are estimated by the Pearson's R (correlation coefficients).

The proposed formula of "*minbucket*" which aims to balance the over-fit and under-fit of predictive function training, was added to the conventional processes so that regression tree analysis could be adapted to different data types. For the data sets of "*DHSInMouseRetina*" and "*TFBS75InYeast*" we could see that the conventional regression tree analysis fell into over-fit because the open tests (100-fold cross-validation) are dramatically decreased. The proposed formula could eliminate the over-fit errors for the two data sets. For the data sets of "*CREBBPInMouseNeuron*", "*PromoterLucInHEK293*", "*RBCVariantsGATA1InK562*" and "*TFBS12InHepG2*" the formula (2) also increase performances of open tests by approximately 10 - 26% (Figure 9).

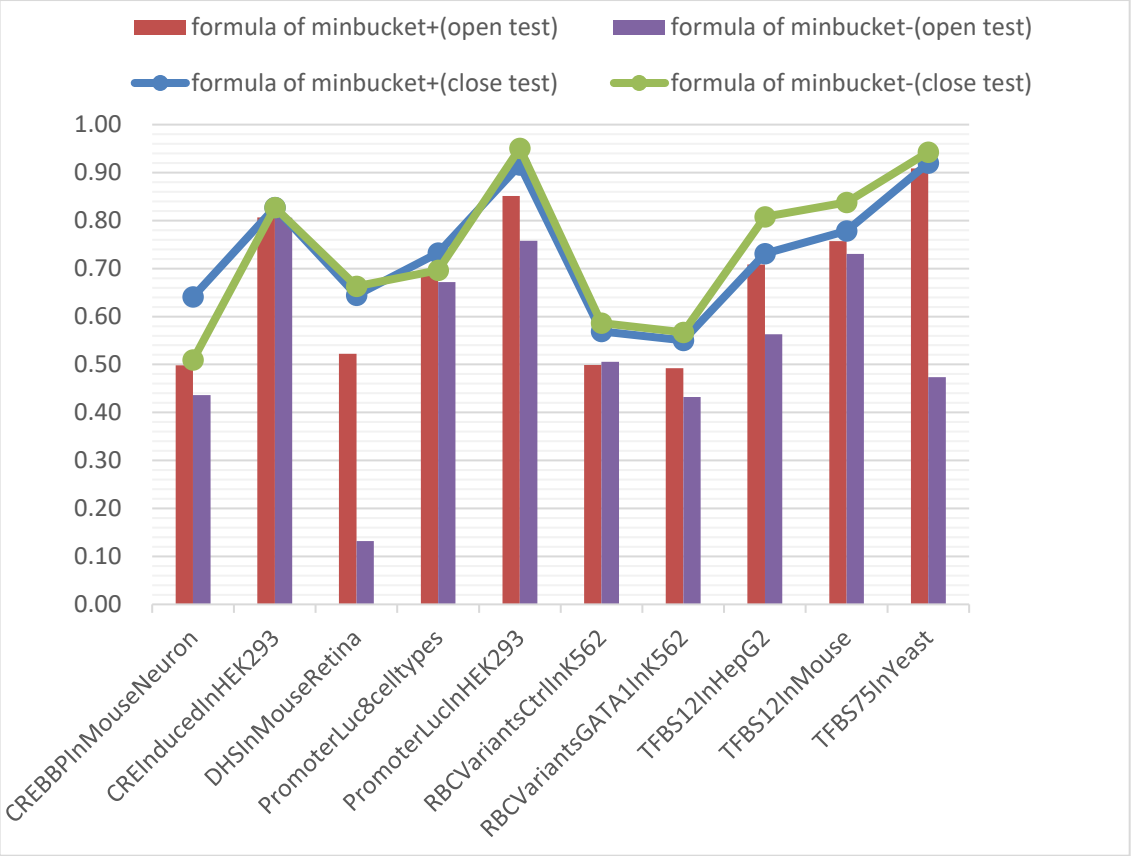


Figure 9. The performances of the proposed method with introducing the "minbucket" formula and without "minbucket" formula. The closed tests and open tests of 100-fold cross-validations were shown. The y-axis shows the correlation coefficients (Pearson's R) between predicted values and experimental values.

2.5 Performing MARS

In the variable clustering step, all the samples were separate into different clusters according to the feature redundancy of corresponding explanatory variable matrix. Next, I tried to construct predictive functions for individual clusters such that the samples in a same cluster having the same predictive function to predict transcriptional activities. Here, I chose the algorithm of MARS to estimate the relation between TFBS enrichment scores and transcriptional activities.

According the form of the response function of MARS, the estimated predictive functions take the form of formula (3). Here, the degree of MARS was set to 2 so that the forms of

predictors in predictive functions are only two types: a single hinge function or a production of two hinge functions. The relative simple predictors were used to avoid over-fit.

$$\text{transcriptional activity}_{\text{j}} = \sum \text{c}' * \text{h}(\text{TFBS enrichment score}_{i'j}) * \text{h}\left(\text{TFBS enrichment score}_{i''j}\right) + \sum \text{c}'' * \text{h}\left(\text{TFBS enrichment score}_{i'''j}\right) + \text{c} \quad (3)$$

In the formula (3), the TFBS enrichment scores are the explanatory variables which are calculated in the TRANSFAC searching step; the h(-) indicates the hinge function and the parameters are estimated by MARS; the coefficients *c* are also estimated by MARS.

2.6 Performances of the proposed method

The proposed method was applied to the 10 data sets (Table 2) and obtained the predictive precisions (Pearson's R between predicted values and experimental values) were approximately 0.5 to 0.9 (Table 3). The number of predictors of individual clusters are also shown for different data sets.

For data sets of "*CREInducedInHEK293*", "*PromoterLucInHEK293*" and "*TFBS75InYeast*", I got the relative high predictive precisions that the closed tests and open tests were both >0.8. It is probably because their sequences patterns and/or transcriptional processes have lower complexity than other data sets.

Table 4. The number of predictors of individual clusters in the estimated predictive functions and the correlation coefficients between predicted values and experimental values of closed tests and open tests (100-fold cross-validation).

Data set	Closed test	Open test	# of predictors
RBCVariantsGATA1InK562	0.55	0.49	16 - 30
RBCVariantsCtrlInK562	0.57	0.50	21 - 26
CREBBPInMouseNeuron	0.64	0.50	21 - 36
DHSInMouseRetina	0.64	0.52	20 - 48
TFBS12InHepG2	0.73	0.71	28 - 28
PromoterLuc8celltypes	0.73	0.70	21 - 50
TFBS12InMouse	0.78	0.76	35 - 35
CREInducedInHEK293	0.83	0.81	25 - 47
PromoterLucInHEK293	0.92	0.85	28 - 28
TFBS75InYeast	0.92	0.91	16 - 30

3 METHOD COMPARISONS

To evaluate the performances of the proposed method from different aspects, I also made the method comparisons. There were two kinds of method comparisons in this study: the proposed method compared with other machine learning algorithms and the proposed method compared with QSAMs. I examined the predictive precisions and the number of predictors for the proposed method and other methods.

Three machine learning algorithms (MLR, Lasso regression and BQR) were applied to the method comparisons. QSAMs is a computational model which was applied to predict transcriptional activities of MPRA data in the former study of (8). QSAMs construct their predicting models at a single nucleotide resolution by encoding all positions into the explanatory variables. Thus, it requires the process of pairwise alignment and only could be applied to data sets which have equal length sequences.

3.1 Method comparisons with the machine learning algorithms

Regard to method comparisons, three machine learning algorithms that are widely used in bioinformatics were applied to these data sets with the same explanatory variable matrices and response variables. The machine learning algorithms of MLR, Lasso regression and BQR were introduced into this study for method comparisons.

The three machine learning algorithms were applied to these data sets (Table 2) and the predictive precisions of closed tests and open tests were evaluated. We could find that for both of closed tests and open tests, the proposed method performed better predictive precisions than MLR, Lasso regression and BQR for all the data sets (Figure 10 and Figure 11).

For the data sets of "*TFBS75InYeast*", "*PromoterLucInHEK293*" and "*CREInducedInHEK293*", all the mentioned methods obtained relative high predictive precisions.

It is because their sequence patterns and/or transcriptional complexities are relative simple. And for data sets of "*DHSInMouseRetina*", "*CREBBPInMouseNeuron*", "*RBCVariantsCtrlInK562*" and "*RBCVariantsGATA1InK562*", the three machine learning algorithms performed relative low predictive precisions (open tests <0.45). The proposed method could improve the predictive precisions for these data sets. The average increased predictive precisions are approximately 22%, 26% and 51% for closed tests and 14%,16% and 43% for open tests comparing with MLR, Lasso regression and BQR, respectively.

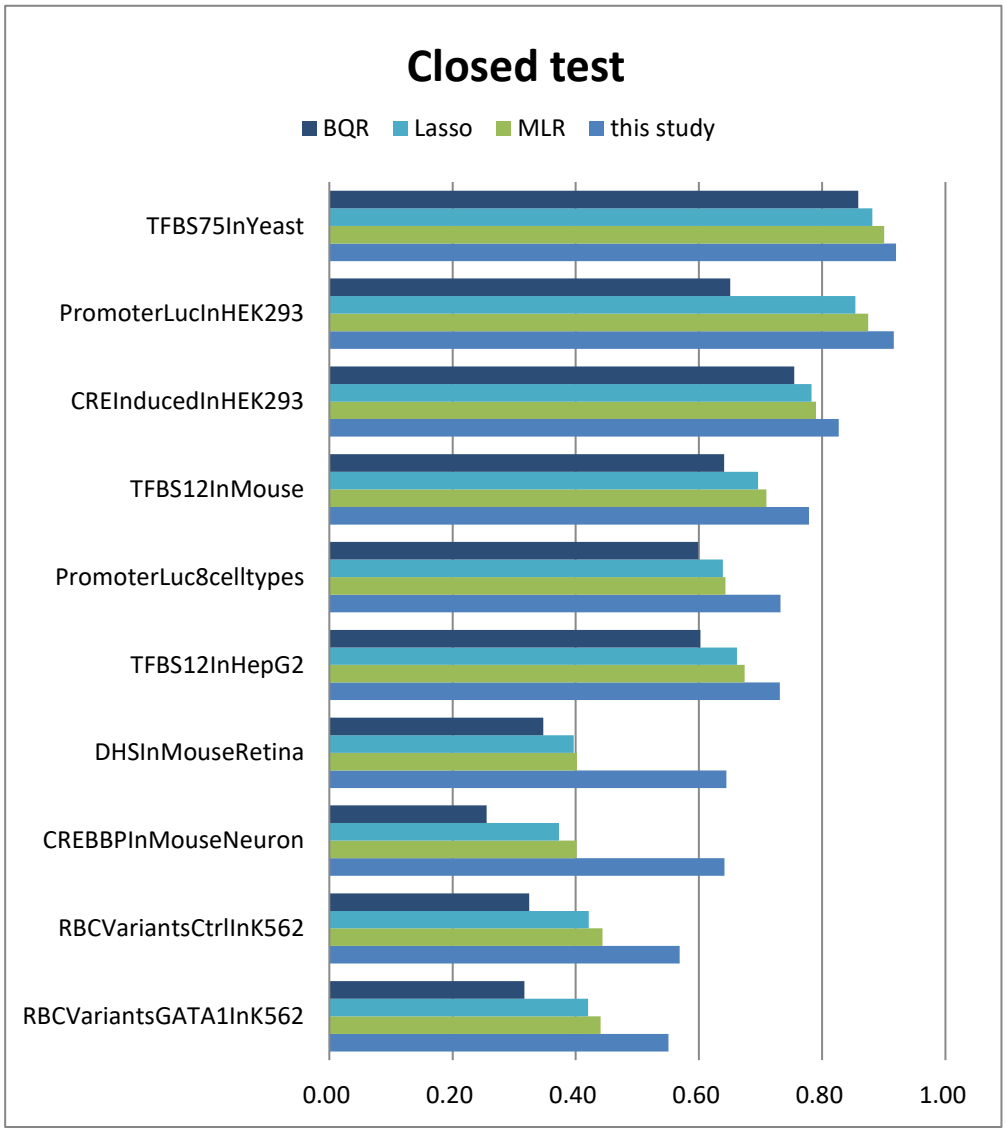


Figure 10. The closed test performances of the proposed method and other machine learning algorithms of MLR, Lasso regression and BQR, respectively. The x-axis indicates the correlation coefficients (Pearson's R) between predicted values and experimental values.

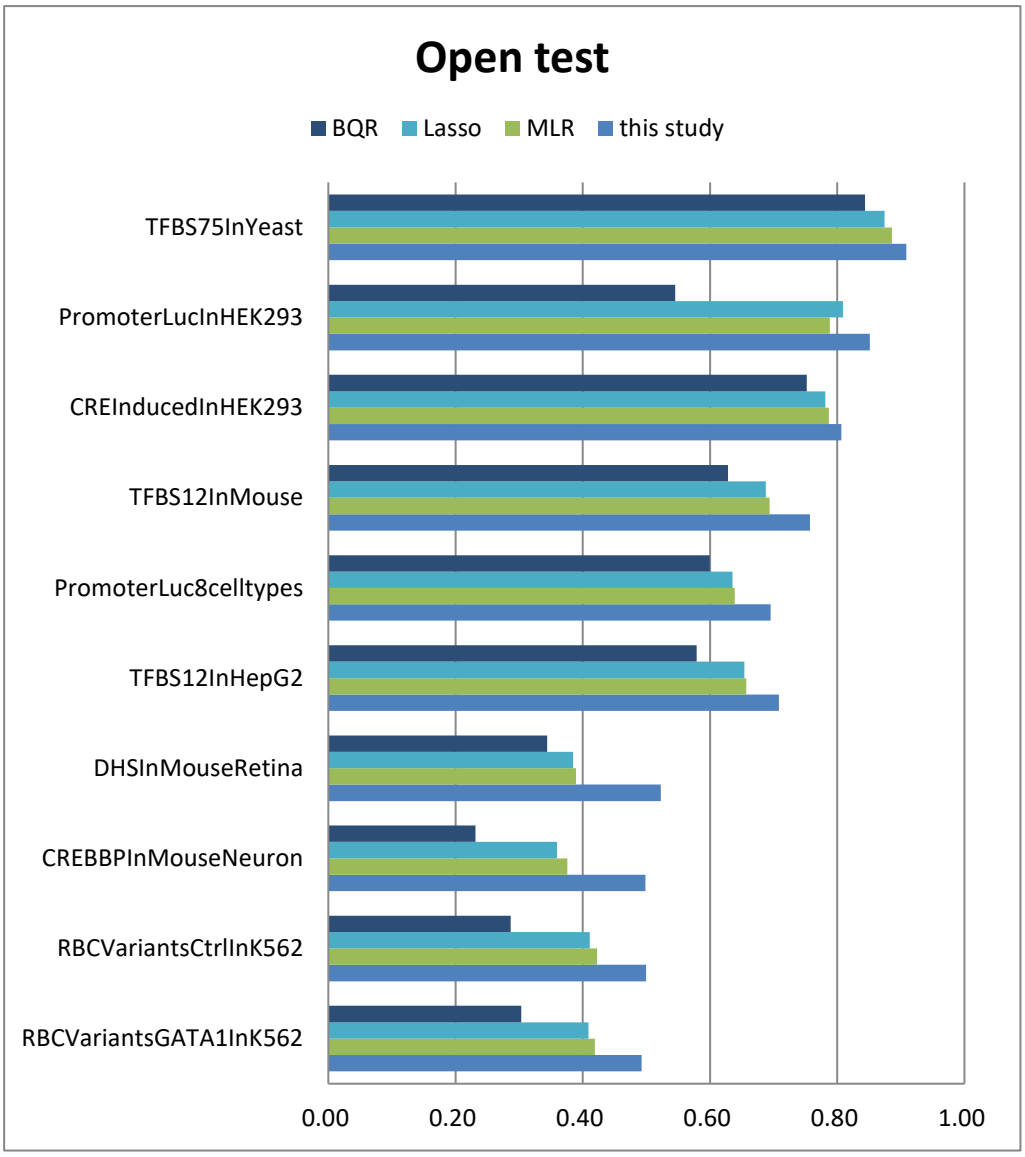


Figure 11. The open test performances of the proposed method and other machine learning algorithms of MLR, Lasso regression and BQR, respectively. The x-axis indicates the correlation coefficients (Pearson's R) between predicted values and experimental values.

Regarding the number of predictors estimated by the proposed method for individual clusters and other algorithms, we found that the proposed method had much smaller number (average 2.3-4.7 times smaller) of predictors in each cluster of the estimated predictive functions than MLR, Lasso and BQR (Figure 12). It suggests that the proposed method could obtain higher predictive precisions by using smaller number of predictors in contrast to MLR, Lasso and BQR.

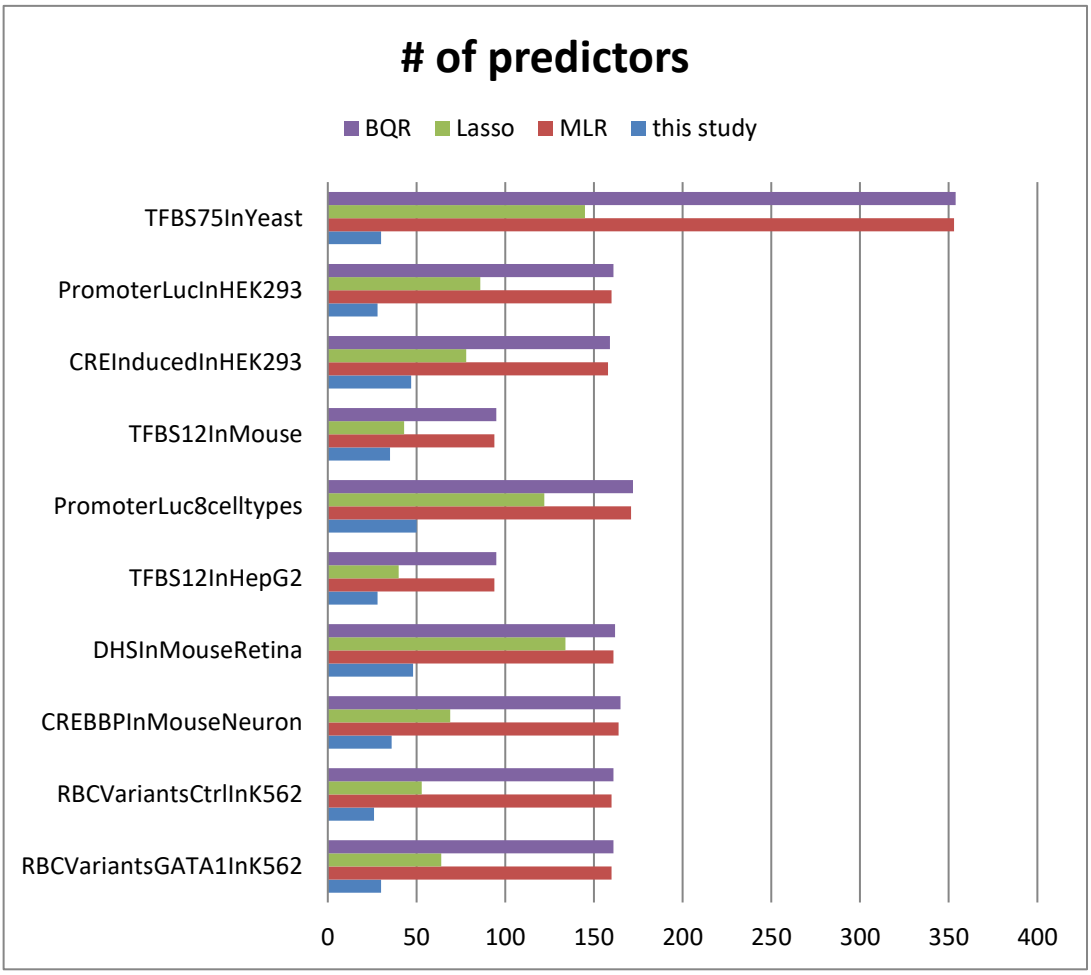


Figure 12. The number of predictors in the predictive functions estimated by the proposed method and other machine learning algorithms of MLR, Lasso regression and BQR, respectively. The number of predictors of proposed method indicate the maximum number of predictors across different clusters

3.2 Method comparisons with QSAMs

In this study, QSAMs were also considered to be introduced for method comparisons. Here, I constructed two kinds of QSAMs: 1. Conventional QSAM that encode each nucleotide into binary (0-1) code and perform linear regression; 2. After encoding all the nucleotides into binary codes, Lasso regression was employed for variable selection and regression.

I examined the predictive precisions of closed tests, open tests and the numbers of predictors for the two kinds of QSAMs. The results showed that the proposed method also had the better performances than QSAMs for both closed tests and open tests across different data sets. For the data sets of "*PromoterLucInHEK293*", "*PromoterLuc8celltypes*" and "*DHSInMouseRetina*", the sequence lengths of which are not equal, the two QSAM methods could not be applied (Figure 13-15).

For closed tests, the average correlation coefficients (Pearson's R) of the proposed method are increased average 24% and 35% as compared with the method of QSAM and QSAM combined with Lasso, respectively (Figure 13). For open tests, the average improved predictive precisions are 30% and 37%, respectively (Figure 14).

On the other hand, the QSAM basically requires more number of predictors which is three times of sequence length. We could see the simple QSAM has the average 12.3 times number of predictors than the proposed method (Figure 15). When I performed variable selection (the selections should retain the similar predictive precisions of QSAM) by Lasso, it also required approximately average 6.5 times the number of predictors than the proposed method.

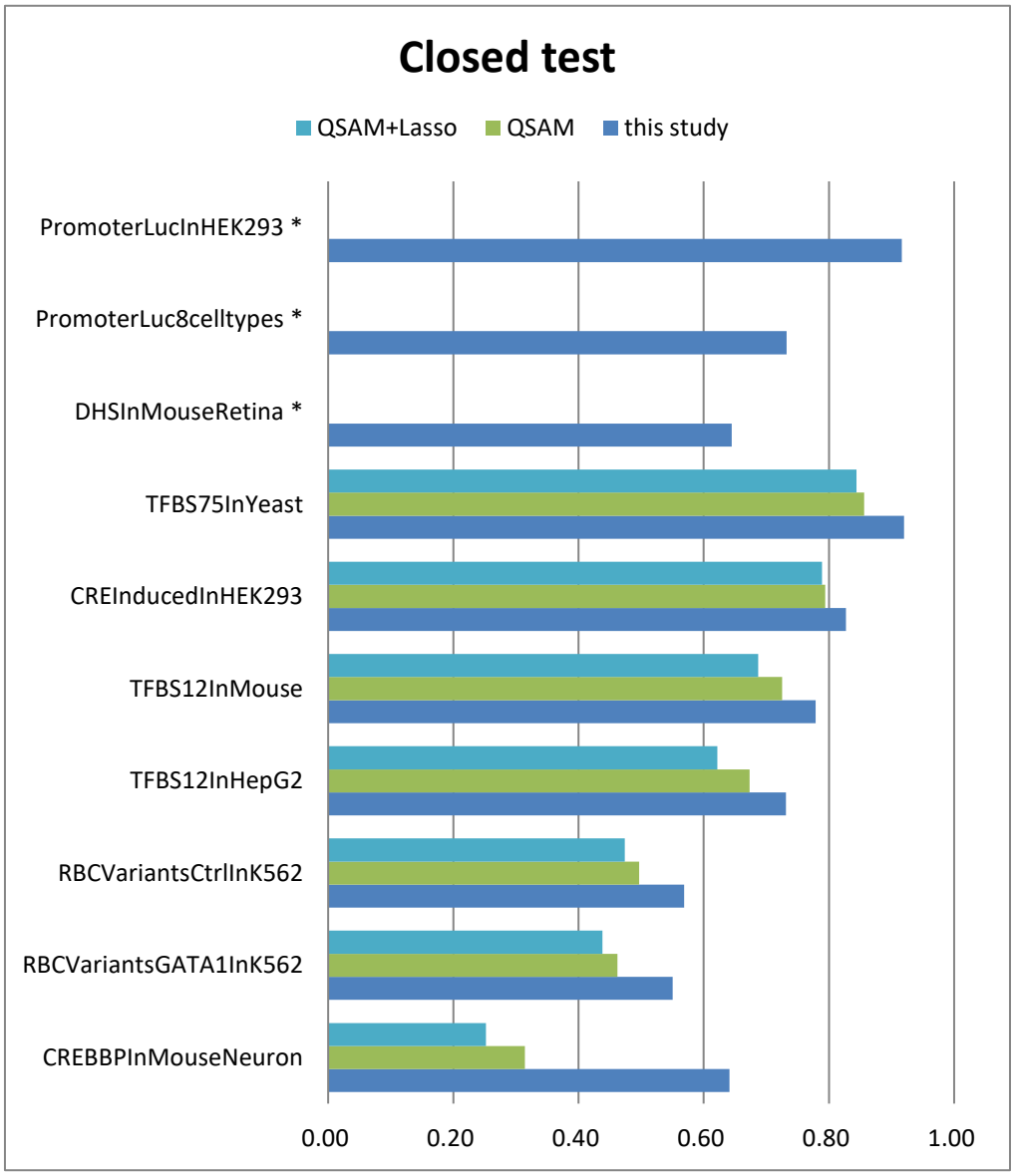


Figure 13. The closed test performances of the proposed method and QSAM and QSAM combined with Lasso regression, respectively. The x-axis indicates the correlation coefficients (Pearson's R) between predicted values and experimental values. The data sets with "" indicate the data sets could not employ QSAMs for constructing predictive functions.*

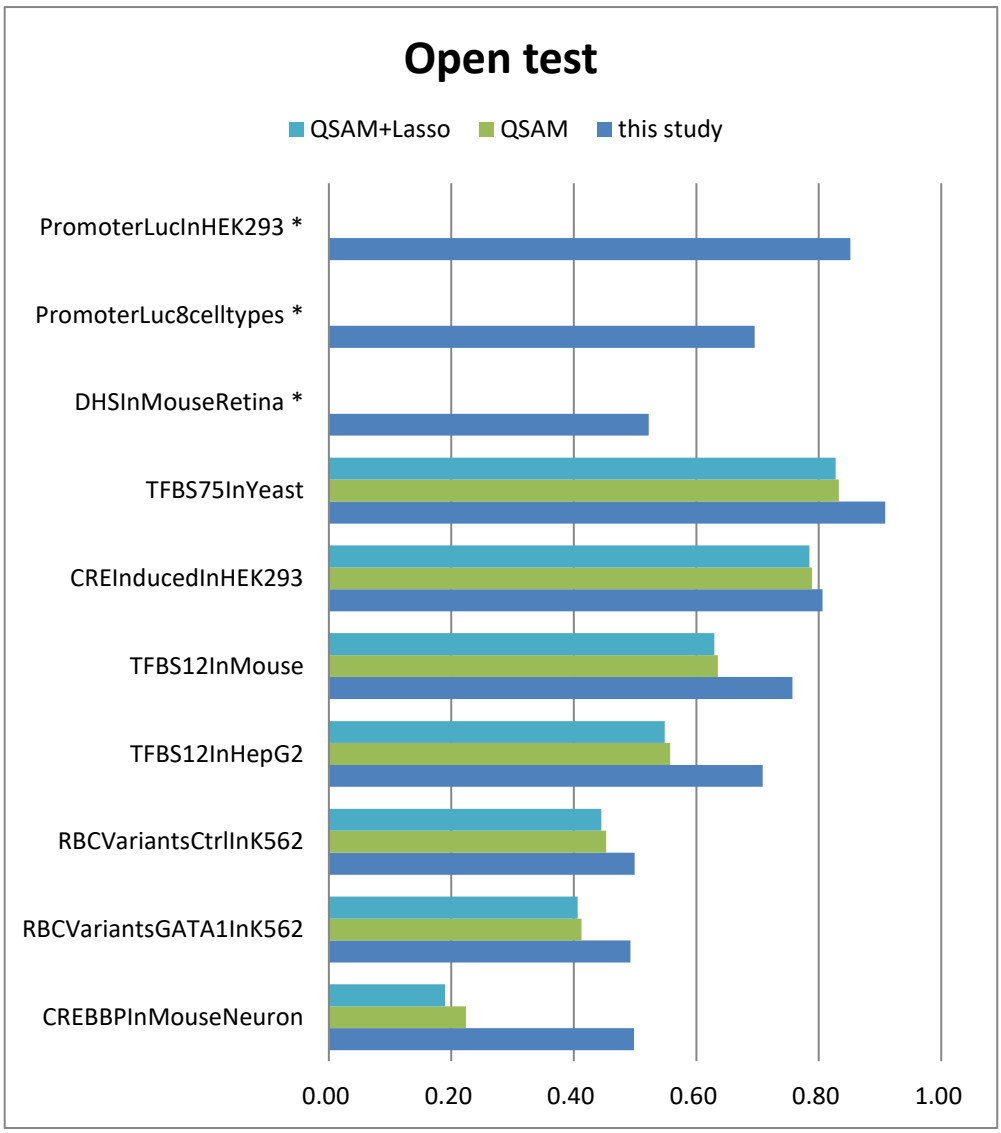


Figure 14. The open test performances of the proposed method and QSAM and QSAM combined with Lasso regression, respectively. The x-axis indicates the correlation coefficients (Pearson's R) between predicted values and experimental values. The data sets with "" indicate the data sets could not employ QSAMs for constructing predictive functions.*

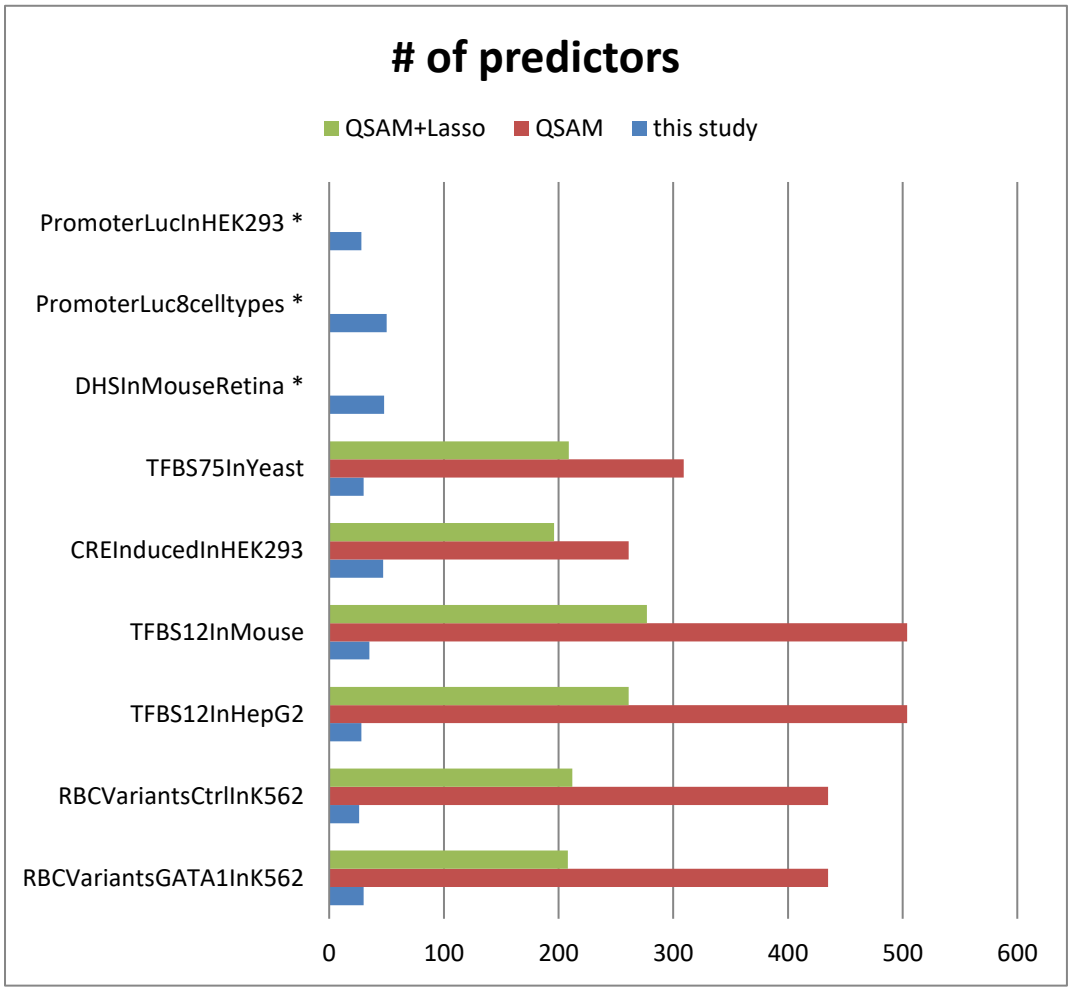


Figure 15. The number of predictors in the predictive functions estimated by the proposed method, QSAM and QSAM combined with Lasso regression, respectively. The number of predictors of proposed method indicate the maximum number of predictors across different clusters. The data sets with "" indicate the data sets could not employ QSAMs for constructing predictive functions.*

4 APPLICATIONS

4.1 Analysis of "*CREInducedInHEK293*" data set

The data set of "*CREInducedInHEK293*" has the sequences of mutant CRE enhancers with the sequence length of 87-nt and the transcriptional activities in HEK293. There are about 10% random mutations in each mutant CRE enhancer. The predictive precisions of applying the proposed method to "*CREInducedInHEK293*" data set are 0.83 and 0.81 for closed test and open test, respectively (Figure 16).

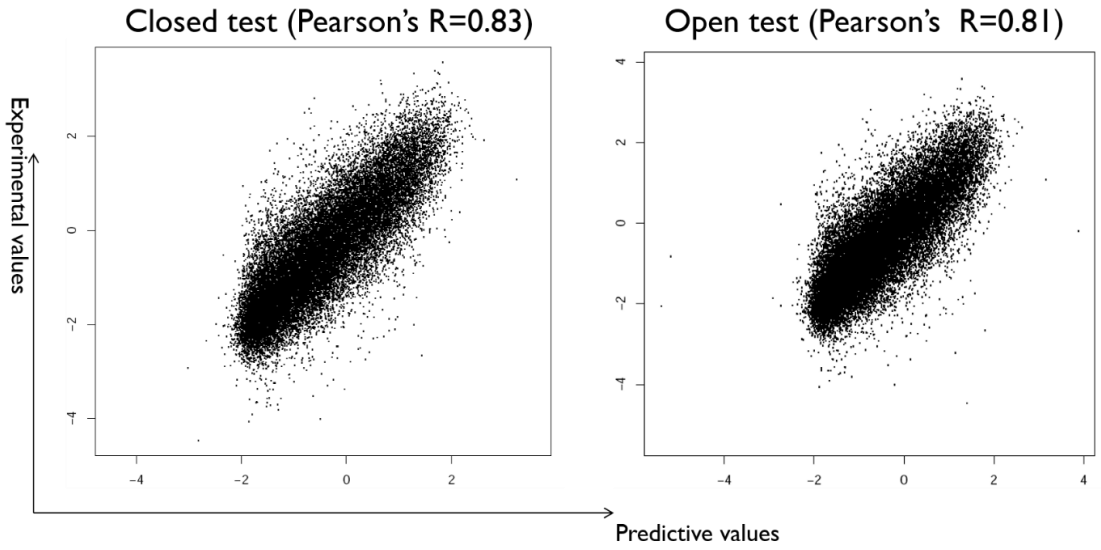


*Figure 16. Scatter plots of closed test and open test for data sets "*CREInducedInHEK293*".*

From the candidate active TFBS tree which was estimated by regression tree with "*minbucket*" formula, we could find that the TFBS of CREB (cAMP response element binding protein) appears high contributions to transcriptional activities because the different enrichment scores of CREB dominate the 4/5 tree branching (Figure 17).

In the root of the candidate active TFBS tree, enrichment scores of 3.9 means that the copy number of CREB is 4 (by applying TRANSFAC, the cut-off of "V$CREB1_Q6" is 0.866). The results are consistent with the known structure of CRE enhancer (8).

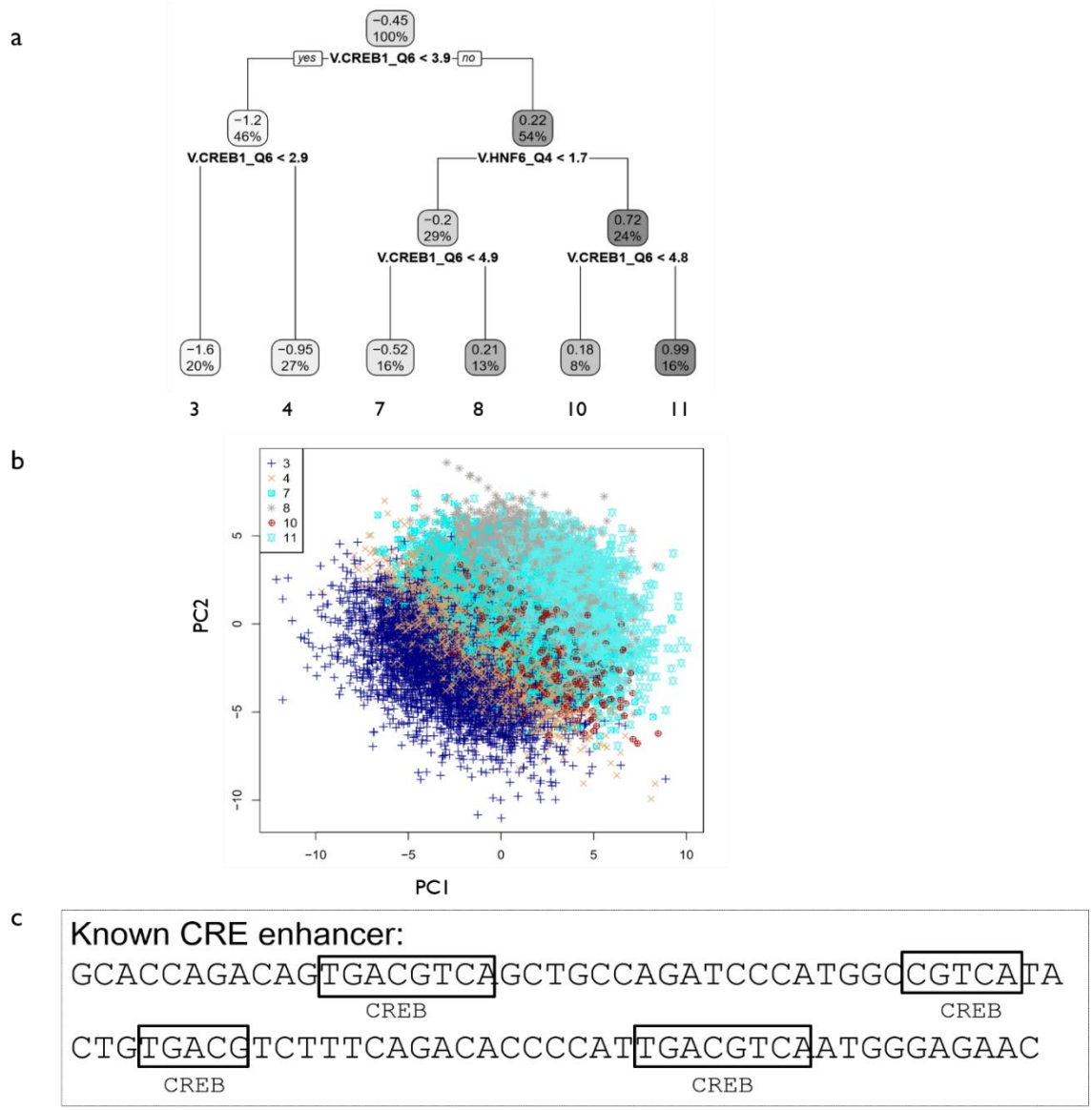


Figure 17. (a) Candidate-active TFBS trees for "CREInducedInHEK293" data sets. The values shown in each cluster indicate the average activity among samples within the corresponding cluster, and the percentages represent the sample proportions in the cluster. (b) PCA plot of "CREInducedInHEK293" with different colours indicate the different clusters showed in the candidate-active TFBS tree. (c) Known TFBSs in CRE enhancer described in the study of (8).

4.2 Analysis of "*CREBBPInMouseNeuron*" data set

In the data set of "*CREBBPInMouseNeuron*", the sequences of genomic segments that are bound by the coactivator of CREBBP (CREB binding protein) were assayed by both MPRA and STARR-seq under different experimental conditions of mouse cortical neurons and KCL (potassium chloride)-stimulated mouse cortical neurons. MPRA is generally considered as measuring promoter activities and in contrast, STARR-seq is considered as measuring enhancer activities. There are four experimental conditions in the data set and I combined all the transcriptional activities across different conditions. Here, adding the binary variables to the explanatory variable matrix was performed to distinguish the different conditions.

The proposed method was employed to predict the transcriptional activities of "*CREBBPInMouseNeuron*" for the mix of MPRA and STARR-seq activities. The predictive precisions of closed test and open test are 0.64 and 0.5, respectively (Figure 18).

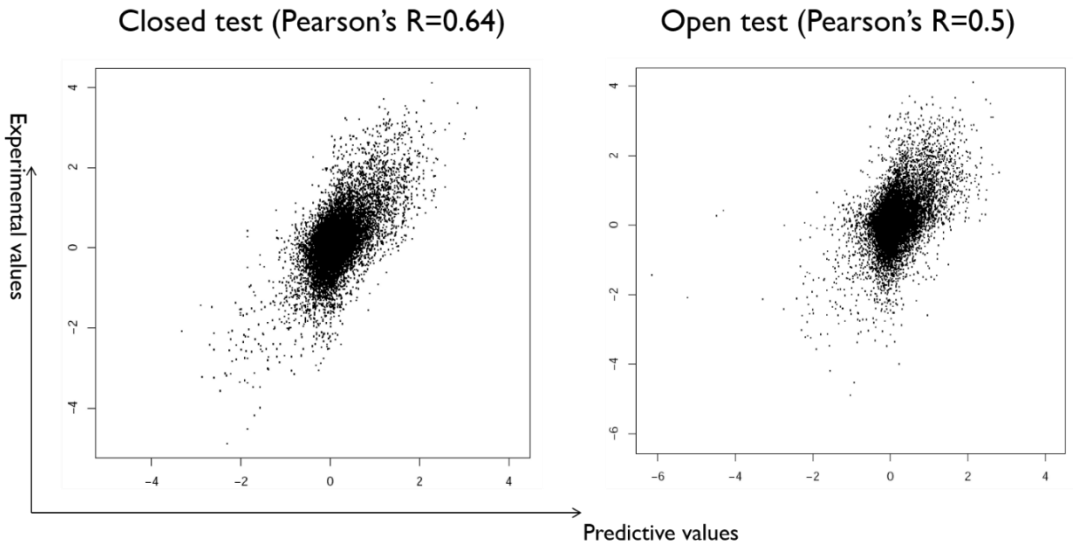


Figure 18. Scatter plots of closed test and open test for data set "CREBBPInMouseNeuron".

In the study of (17) which provided the data set of "*CREBBPInMouseNeuron*", they reported that TFBSs of CREB and RFX (regulatory factor X) have both strong promoter activity and enhancer activity in the experiments. According the candidate-active TFBS tree of "*CREBBPInMouseNeuron*", I also found that TFBS of CREB occurs in the root and RFX dominates the clustering of three branches in the regression tree. It suggests that CREB and RFX play important role to regulate the transcriptional activities of different sequences (Figure 19).

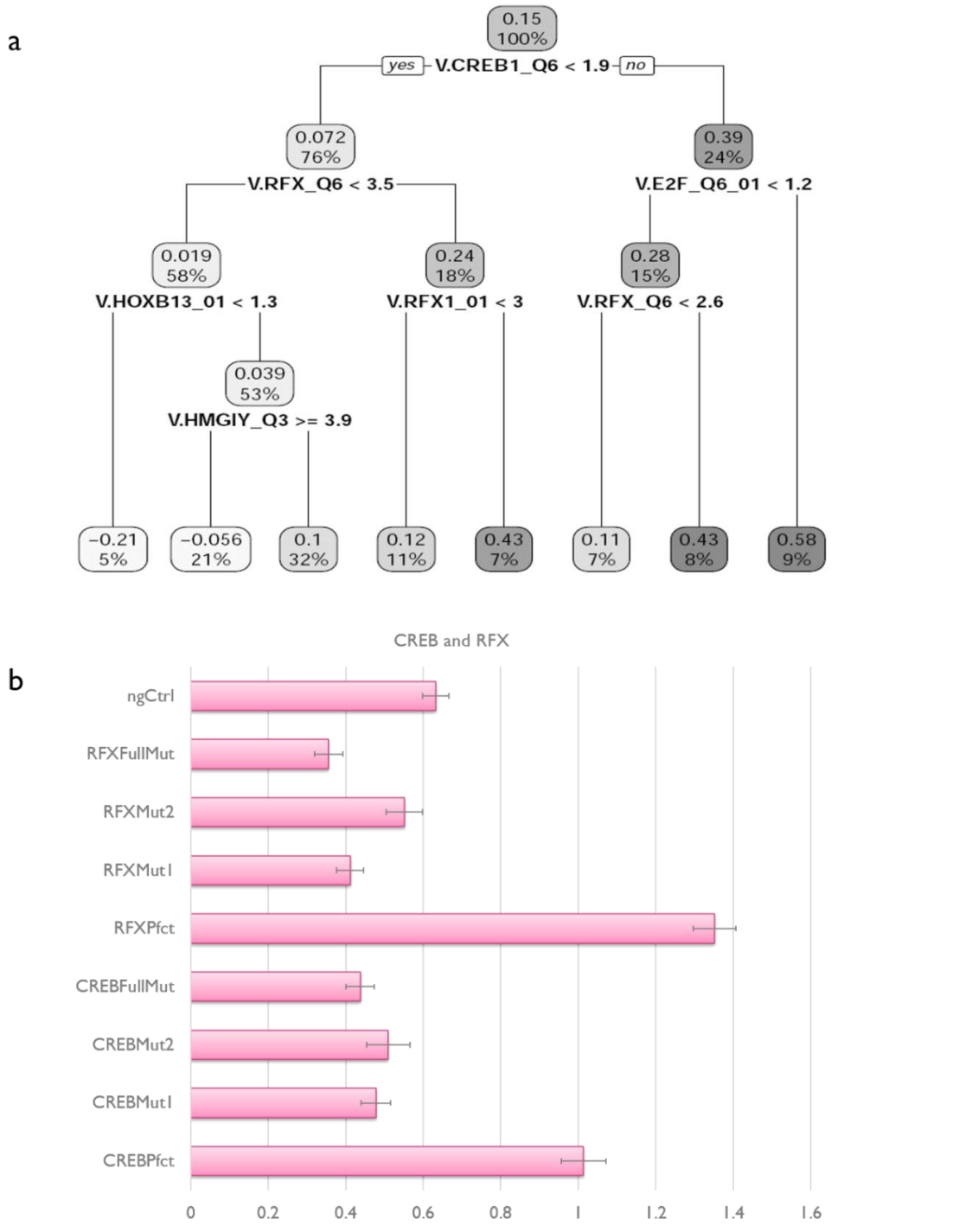


Figure 19. (a) Candidate-active TFBS trees for "CREBBPInMouseNeuron" data sets. The mean values and sample proportions of individual clusters are also given in the regression tree. (b) The mean values of transcriptional activities of CREB and RFX derived motifs. There are perfect motifs, two types of 2-bp mutant motifs, full mutant motifs and negative controls described in study of (17).

On another hand, the study of (17) also reported that TFBS of AP1 (activator protein 1) bound preferentially for enhancer activity. In the predictive functions of "*CREBBPInMouseNeuron*", there are 8 TFBSs associated enhancer-specific activity and AP1 is one of them (Table 5)

Table 5. The predictors that showed enhancer activity preferences and the estimated coefficients. The variable of "kclEnh" indicates the experimental condition (see also Methods) of enhancer activities. The predictors of "kclEnh" multiplying a hinge function of a TFBS suggest the corresponding TFBSs have enhancer preferential activities.

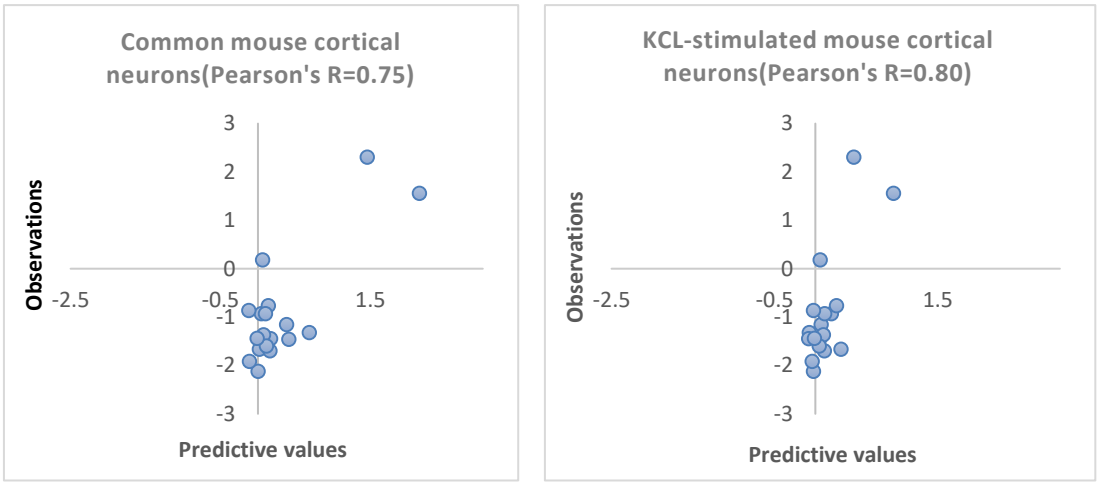
Predictor associated enhancer specific activity	Coefficients
kclEnh*h(0.958-V$CEBPA_Q6)	-1.08
kclEnh*h(1.883-V$ZFX_01)	0.10
kclEnh*h(11.239-V$ZIC1_05)	0.12
kclEnh*h(V$AP1_Q6_02-3.545)	0.59
kclEnh*h(V$BEN_01-15.139)	-1.55
kclEnh*h(V$CPBP_Q6-4.96)	-0.09
kclEnh*h(V$SP100_04-3.694)	-0.15
kclEnh*h(V$SP100_04-6.888)	-0.12
kclEnh*h(V$ZFP161_04-4.641)	-0.12
kclEnh*h(V$ZFX_01-1.883)	-1.31
kclEnh*h(V$ZIC1_05-11.239)	-0.19

In the study (17), there are other MPRA data sets which assayed 18 selected motifs under the same experimental conditions as the data set "*CREBBPInMouseNeuron*". The assayed sequences were designed in the form of corresponding motif repeats flanking by 11 bp spacers. Here, I wanted to evaluate the ability of estimating transcriptional activities for new data by predictive functions of known data sets. Hence, I used the predictive functions which estimated for "*CREBBPInMouseNeuron*" data to predict the transcriptional activities of the 18 motifs. The obtained correlation coefficients between predicted values and experimental values are approximately 0.75 and 0.80 (Figure 20a) for the assays in common mouse cortical neurons and KCL-stimulated mouse cortical neurons, respectively.

However, the relative high correlation coefficients were dragged by two motifs which have outstandingly high transcriptional activities in comparison with other motifs. If I removed the two samples, the predictive precisions were dramatically dropped (Figure 20b). It is probably because of the relative low transcriptional activities of the rest samples and the relative low sensitivity of the proposed method. The proposed method only introduces the TFBS enrichment scores as explanatory variables for predictive function training, and the other features which also have contributions to transcriptional activities such as positions and orientations were ignored. I wanted to construct a relative simple model which intend to select highly significant features for predicting transcriptional activities and this sacrificed the model sensitivity.

Furthermore, in the 16 motifs which removed the two outliners from the whole data set, there are 5 samples have relative high transcriptional activities compared to other samples (>0.15) in the data set of common mouse cortical neurons. And the predictive precision of the 5 samples is 0.56. In the data set of KCL-stimulated mouse cortical neurons, almost all the 16 motifs have the transcriptional activities near 0 (the mean value is 0.07 and the third quantile is 0.11). For these samples, the proposed method could not get a good prediction. It suggests that the proposed method could predict the transcriptional activities of new sequences by the estimated predictive functions of known data sets, although the proposed method usually performs relative low sensitivity for low activity samples.

a



b

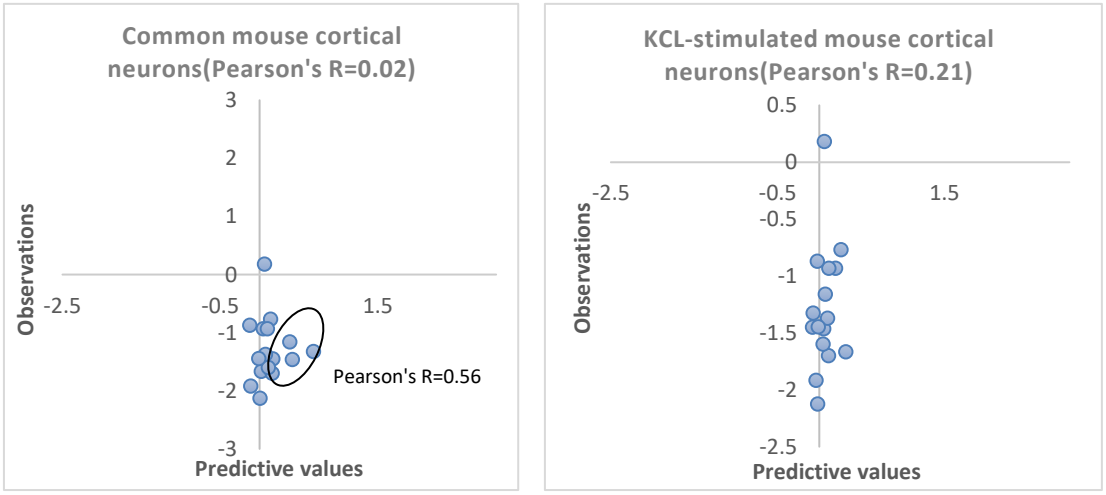


Figure 20. (a) Scatter plots between predictive values and observations of 18 individual motifs assayed in common mouse cortical neurons and KCL-stimulated mouse cortical neurons. (b) Scatter plots between predictive values and observations of the 16 individual motifs which remove the two samples with high transcriptional activities from all 18 motifs.

4.3 Analysis of "RBCVariantsCtrlInK562" and "RBCVariantsGATA1InK562" data sets

The data sets of "*RBCVariantsCtrlInK562*" and "*RBCVariantsGATA1InK562*" have the sequences of red blood cell variants and the transcriptional activities were assayed in K562 cells and GATA1 overexpressing(OE) K562. I applied the proposed method to the data sets of "*RBCVariantsCtrlInK562*" and "*RBCVariantsGATA1InK562*" and obtained the predictive precisions of closed tests are 0.57 and 0.56, respectively. For the open tests, the correlation coefficients between predicted values and experimental values are 0.49 and 0.50 for "*RBCVariantsCtrlInK562*" and "*RBCVariantsGATA1InK562*", respectively (Figure 21).

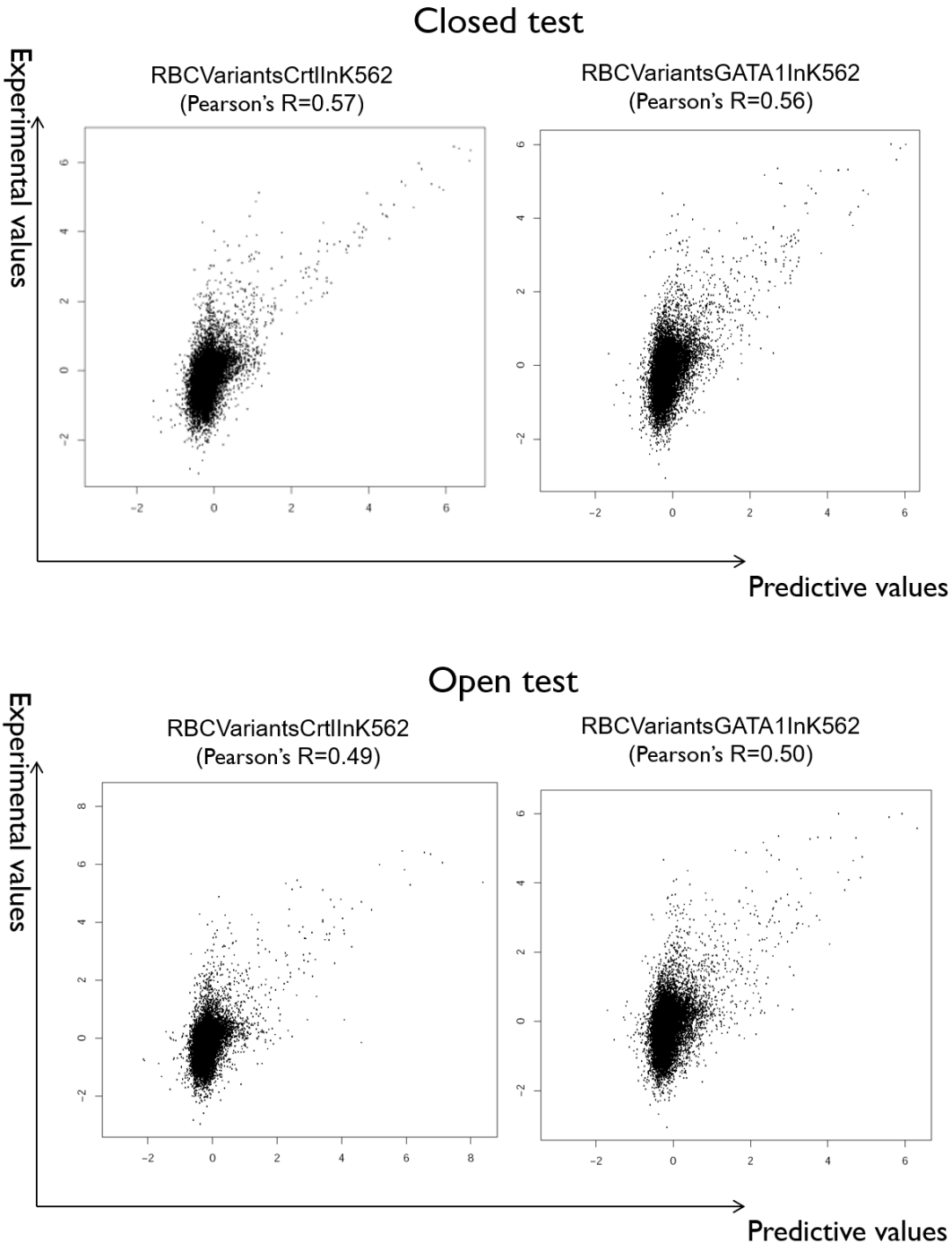


Figure 21. Scatter plots of closed tests and open tests for data sets "RBCVariantsCtrlInK562" and "RBCVariantsGATA1InK562", respectively.

From the selected TFBS frequencies across all predictors of the predictive functions for "*RBCVariantsCtrlInK562*" and "*RBCVariantsGATA1InK562*" (Figure 22), we could find that the frequency of TFBS of GATA family ("V$GATA_Q6") estimated by the proposed method for "*RBCVariantsGATA1InK562*" (the frequency is 14) is 4.67 times higher than "*RBCVariantsCtrlInK562*" (the frequency is 3). It suggests that the GATA1 overexpressing results in the TFBS of GATA family activation.

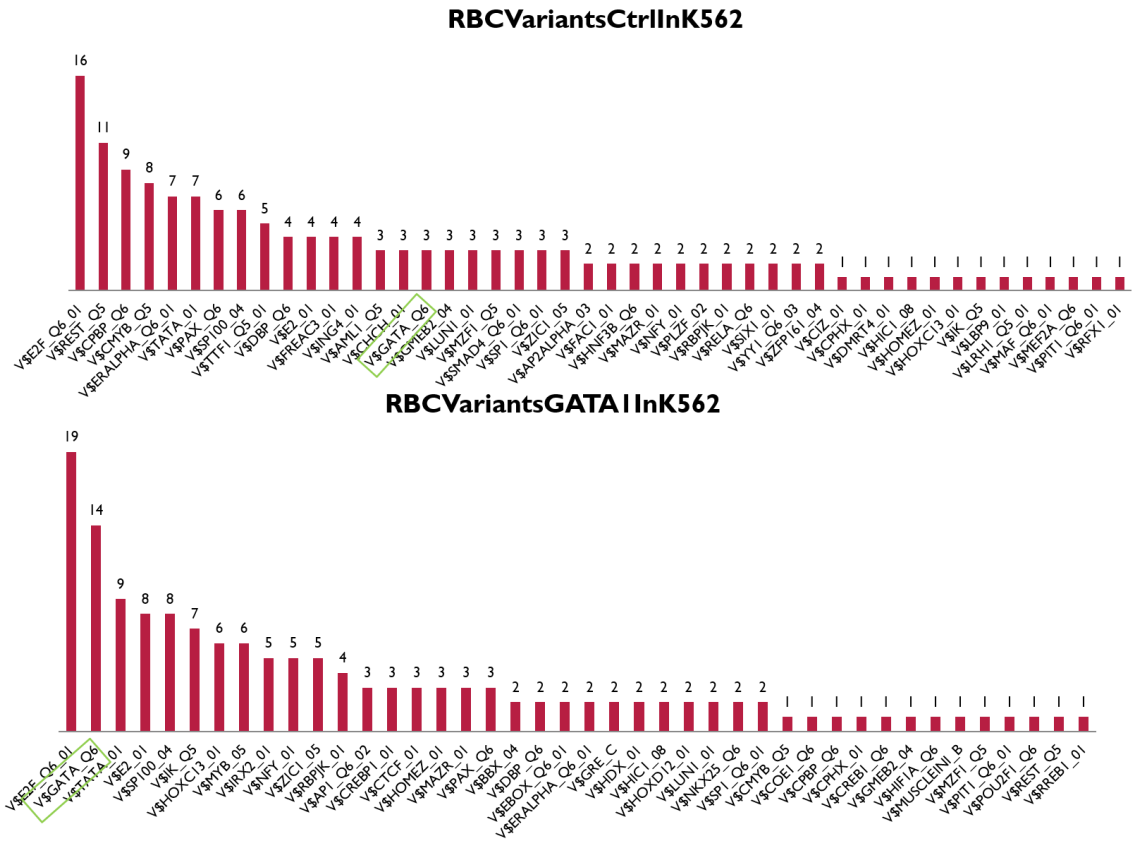


Figure 22. TFBS frequencies across all predictors of predictive function of "RBCVariantsCtrlInK562" and "RBCVariantsGATA1InK562". The frequency of TFBS "V$GATA_Q6" increased from 3 to 14 if changing the experimental conditions from common K562 to GATA1 overexpressing K562 cells.

The candidate-active TFBS trees estimated by the proposed method presented different structures between "*RBCVariantsCtrlInK562*" and "*RBCVariantsGATA1InK562*". It suggests that GATA1 OE plays relative important role in the transcriptional processes of K562. The distributions of clusters of candidate-active TFBS trees also showed different in the PCA plots for "*RBCVariantsCtrlInK562*" and "*RBCVariantsGATA1InK562*", respectively (Figure 23).

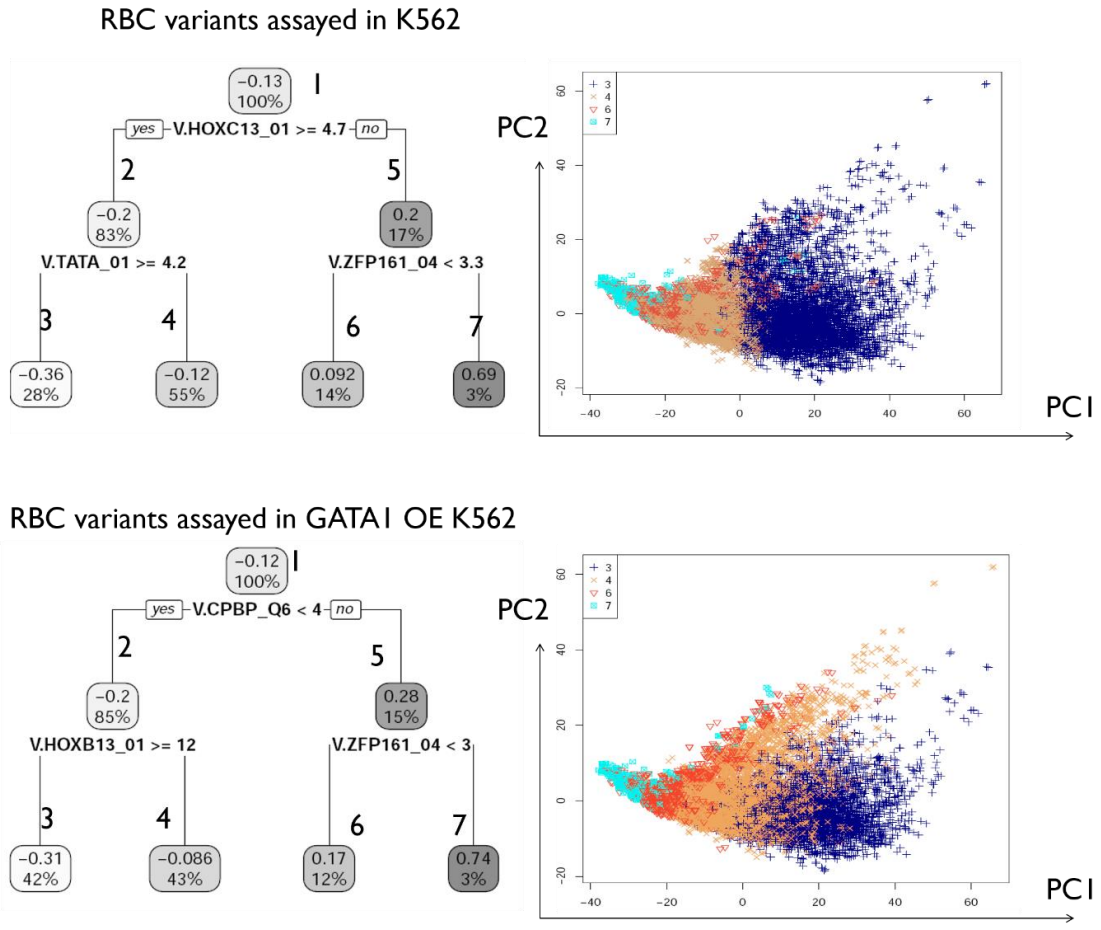


Figure 23. (Left) Candidate-active TFBS trees for data sets of "RBCVariantsCtrlInK562" and "RBCVariantsGATA1InK562". The mean values and sample proportions of individual clusters are also given in the regression trees. (Right) PCA plot of data sets of "RBCVariantsCtrlInK562" and "RBCVariantsGATA1InK562" with different colours indicate the different clusters showed in the candidate-active TFBS trees.

From the predictive functions of "*RBCVariantsCtrlInK562*" and "*RBCVariantsGATA1InK562*", the candidate-active TFBSs for the two experimental conditions could be picked up. There are 46 and 42 candidate-active TFBSs selected for "*RBCVariantsCtrlInK562*" and "*RBCVariantsGATA1InK562*" data sets, respectively. And 17 TFBSs of them are only selected by "*RBCVariantsGATA1InK562*" data and thus, they represent a GATA1 overexpression responsive property. Ten TFBSs of the 17 were reported that they have biological associations or interactions with GATA1 by several previous studies (Table 6).

Table 6. The 17 selected TFBSs from predictive functions of "RBCVariantsGATA1InK562" that did not overlap with selected TFBSs of "RBCVariantsCtrlInK562". And the corresponding bound transcription factors are also shown in the table.

GATA1 over expression responsive TFBSs estimated by proposed model	Description	Previous report
V$AP1_Q6_02	AP1	(31)
V$BBX_04	Bbx	
V$COE1_Q6	COE1(EBF1)	
V$CREB1_Q6	CREB1	(32)
V$CREBP1_01	CREB-binding protein	(32)
V$CTCF_01	CCCTC-binding factor	(33)
V$EBOX_Q6_01	E-box (enhancer box)	(34)
V$GRE_C	GR(Glucocorticoid response element)	(35)
V$HDX_01	Hdx	
V$HIF1A_Q6	HIF1A	(36)
V$HOXD12_01	HOXD12	
V$IRX2_01	Irx2	
V$MUSCLEINI_B	Muscle initiator	
V$MYB_05	c-myb	(37)
V$NKX25_Q6	Nkx2-5	(38)
V$POU2F1_Q6	POU2F1	
V$RREB1_01	RREB-1	(39)

On the other hand, I tried to detect the transcription factors which interact with GATA1 (or GATA family) from the predictive functions for "*RBCVariantsGATA1InK562*" without using "*RBCVariantsCtrlInK562*" data sets. There are 8 TFBSs showing the estimated interaction with TFBSs of GATA family (Table 7) and six of them relating to GATA family are reported by previous studies (Table 8).

Table 7. Predictors in the predictive function of "RBCVariantsGATA1InK562" which take the forms of the hinge function of other TFBSs multiplied by the hinge function of the GATA family binding site ("V$GATA_Q6"). The coefficients were estimated by MPRS, and the cluster labels are shown in Figure 23.

Predictor	Coefficient	Cluster label
h(4.777-**V$GATA_Q6**)*h(24.793-V$TATA_01)	-0.00	3
h(V$COE1_Q6-0.864)*h(**V$GATA_Q6**-2.681)	2.73	4
h(**V$GATA_Q6**-2.681)*h(0.733-V$RREB1_01)	2.01	4
h(**V$GATA_Q6**-1.807)*h(V$REST_Q5-0.807)	5.19	6
h(1.807-**V$GATA_Q6**)*h(V$HOXC13_01-9.591)	-0.03	6
h(1.807-**V$GATA_Q6**)*h(9.591-V$HOXC13_01)	-0.06	6
h(V$AP1_Q6_02-3.87)*h(1.807-**V$GATA_Q6**)	-0.73	6
h(**V$GATA_Q6**-1.807)*h(V$RBPJK_01-3.398)	0.78	6
h(**V$GATA_Q6**-1.807)*h(3.398-V$RBPJK_01)	-0.15	6
h(V$CREBP1_01-4.421)*h(**V$GATA_Q6**-1.807)	0.49	6
h(4.421-V$CREBP1_01)*h(**V$GATA_Q6**-1.807)	0.24	6

Table 8. Candidate TFBSs associating with GATA family transcription factors ("V$GATA_Q6") that were selected from the predictors in Table 7 for "RBCVariantsGATA1InK562" data only.

TFBSs of interaction with V$GATA_Q6 estimated by propose model	Description	Previous report
V$AP1_Q6_02	AP1	(31)
V$CREBP1_01	CREB-binding protein	(32)
V$RREB1_01	RREB-1	(39)
V$COE1_Q6	COE1(EBF1)	
V$HOXC13_01	HOXC13	
V$RBPJK_01	RBPJ(Also known as SUH; csl; AOS3; CBF1; KBF2; RBP-J; RBPJK; IGKJRB; RBPSUH; IGKJRB1)	(40)
V$REST_Q5	REST	
V$TATA_01	TATA binding protein (TBP)	(41)

4.4 Analysis of "*TFBS12InHepG2*" and "*TFBS12InMouse*" data sets

In the data sets of "*TFBS12InHepG2*" and "*TFBS12InMouse*", transcriptional activities of 4742 sequences which have 12 liver-specific TFBSs were assayed in human HepG2 and mouse, respectively. The sequences were designed as inserting the TFBSs into template sequences according the prepared criteria such as copy numbers and TFBS permutations.

The predictive precisions of applying the proposed method to data sets of "*TFBS12InHepG2*" and "*TFBS12InMouse*" are 0.73 and 0.78, respectively. The final predictive functions have 28 and 35 predictors for to data sets of "*TFBS12InHepG2*" and "*TFBS12InMouse*", respectively. For the open tests of 100-fold cross-validation, I also got the similar predictive precisions of 0.71 and 0.76, respectively (Figure 24).

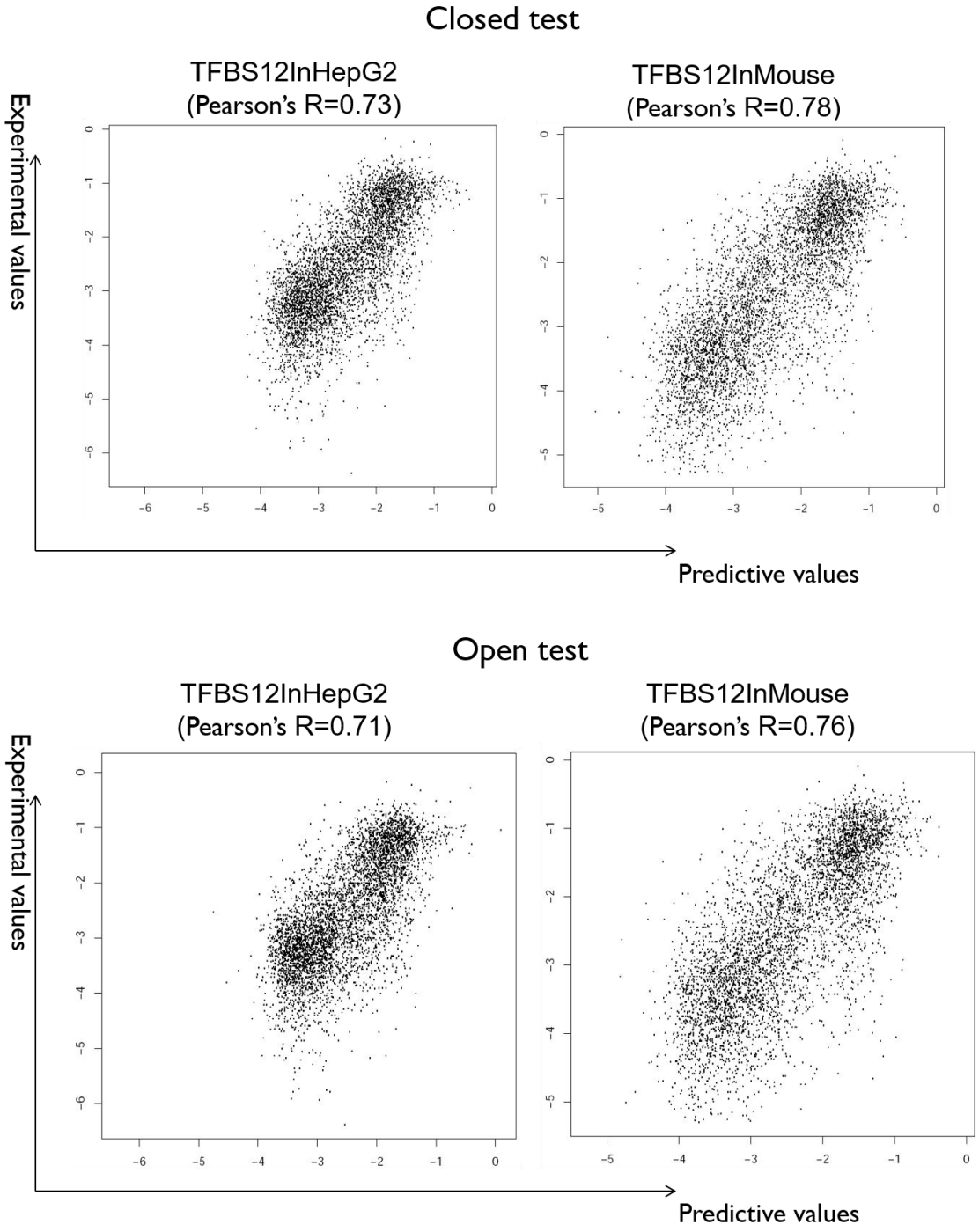


Figure 24. Scatter plots of closed tests and open tests for data sets of "TFBS12InHepG2" and "TFBS12InMouse", respectively.

The structures of regression trees for the two data sets have only one root, that is, the data sets were not clustered according the proposed feature redundancy-dependent formula (formula (2)). I investigated the estimated predictive functions for the two data sets in details and found that several estimated active-TFBSs have binding preferences for human HepG2 or mouse cells (Table 9). The motifs bound by TFs of FOXA1, FOXA2, HNF-1A, HNF-4A and HNF-1B showed different between human and mouse.

For example, four TFBSs ("V$HNF1_C", "V$HNF1_01", "V$HNF1_Q6_01" and "V$HNF1A_01") of HNF-1A (Hepatocyte nuclear factor 1-alpha) were estimated as active TFBSs in human HepG2 and mouse cells. However, from the distribution of selected TFBSs, I found the TFBS of "V$HNF1_C" preferred to be active in human HepG2 cells and the TFBSs of "V$HNF1_01" and "V$HNF1_Q6_01" are bound by HNF-1A in mouse cells. Different from the three TFBSs, "V$HNF1A_01" could be bound in both human HepG2 and mouse cells. Note that, the sequences assayed in HepG2 and mouse are identical and there is no biased background of TFBS frequencies between data sets of HepG2 and mouse.

These results are partly consistent with the previous study of (42) which reported that the binding events of FOXA2, HNF-1A and HNF-4A have diverged between human and mouse.

Table 9. The frequency of TFBSs selected by the response functions of data sets of "TFBS12InHepG2" and "TFBS12InMouse", respectively.

TFBS	Binding TFs	Mouse	HepG2
V$AHRHIF_Q6	AhR	1	
V$ATF4_Q6	ATF-4	3	5
V$CREBATF_Q6	CREB, ATF	3	
V$FOS_01	c-Fos	2	
V$FOS_02	c-Fos	6	1
V$FOS_05	c-Fos	1	
V$FOXA1_02	FOXA1		1
V$FOXA1_03	FOXA1	1	
V$FOXA1_06	FOXA1	2	4
V$FOXA2_04	FOXA2	3	
V$FOXA2_05	FOXA2		7
V$FOXA2_06	FOXA2	1	
V$HIF1A_Q6	HIF-1A		1
V$HIF1AARNT_01	HIF1A, ARNT		2
V$HNF1_C	HNF-1A		2
V$HNF1_01	HNF-1A	2	
V$HNF1_Q6_01	HNF-1A	2	
V$HNF1A_01	HNF-1A	7	5
V$HNF1B_01	HNF-1B		3
V$HNF1B_Q6	HNF-1B	1	
V$HNF3A_Q6	HNF-3A	4	
V$HNF3G_Q4	HNF-3G		3
V$HNF4A_02	HNF-4A	1	
V$HNF4A_04	HNF-4A		1
V$HNF4A_09	HNF-4A	1	
V$HNF4A_10	HNF-4A	2	
V$HNF4ALPHA_Q6	HNF-4A	7	
V$HNF4DR1_Q3	HNF4 family	2	
V$HNF6_Q4	HNF-6		2
V$LFA1_Q6	HNF-1B		2
V$NFKAPPAB50_01	NF-kappaB	2	2
V$NR2F1_04	NR2F1	2	
V$USF1_Q4	USF1		1

4.5 Analysis of "*PromoterLucInHEK293*" and "*PromoterLuc8celltypes*"data sets

There are two data sets both having the promoter sequences and assaying the transcriptional activities in different cell lines. The data set of "*PromoterLucInHEK293*" assayed transcriptional activities of 734 promoters with the median length of 1081bp in HEK293 cells; the data set of "*PromoterLuc8celltypes*" assayed transcriptional activities of 4575 promoters with the median length of 983bp in 8 tumor cell lines (Table 2).

The predictive precisions of "*PromoterLucInHEK293*" and "*PromoterLuc8celltypes*" estimated by the proposed method are 0.92 and 0.73, respectively (Figure 25). For open tests of 100-fold cross-validation, the predictive precisions for the two data sets are 0.85 and 0.70.

There are 28 selected TFBSs which have the frequencies >=5 in the predictors of the predictive functions estimated for data set "*PromoterLuc8celltypes*". On the other hand, 18 TFBSs were selected for data set of "*PromoterLucInHEK293*" and 8 of them overlap with the selected TFBSs of "*PromoterLuc8celltypes*" (frequency>=5) (Figure 26).

The two data sets have similar sequence types and could be considered that using the estimated predictive functions of one data set to predict the transcriptional activities of the sequences of another data set.

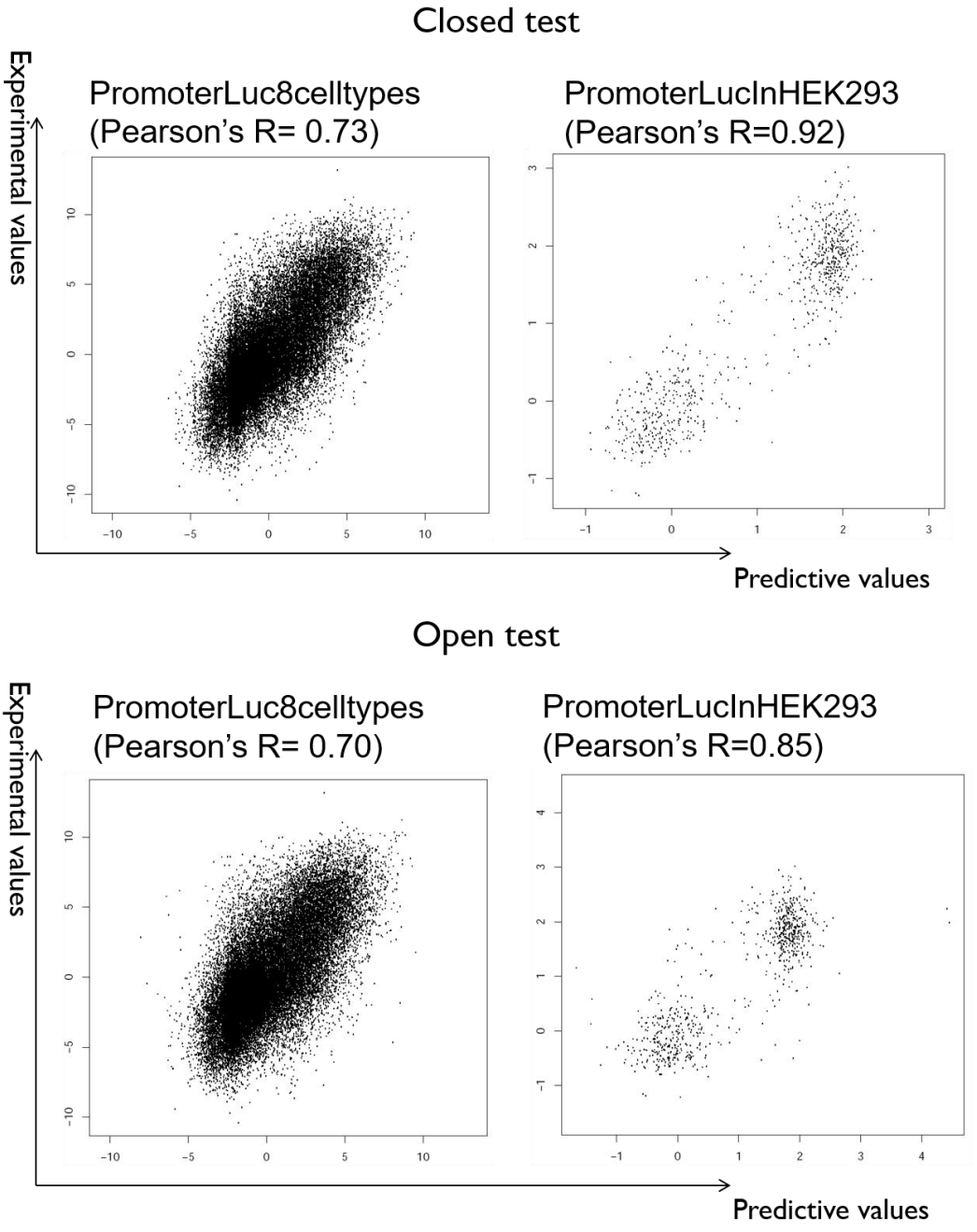


*Figure 25. Scatter plots of closed tests and open tests for data sets of "*PromoterLucInHEK293*" and "*PromoterLuc8celltypes*", respectively.*

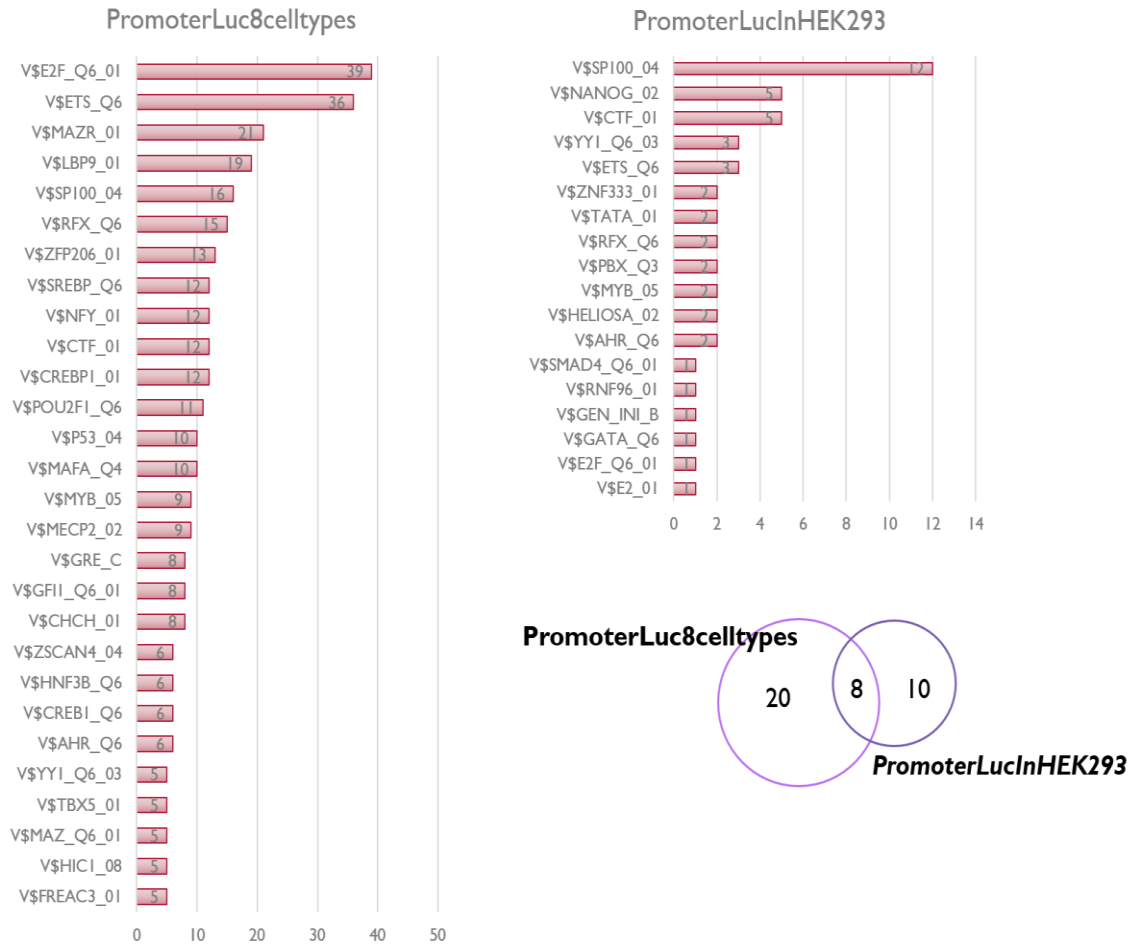


Figure 26. The selected TFBSs and the corresponding frequency. (Left) Frequency of selected TFBSs that occur >=5 times in the predictors within the predictive functions estimated by modelling "PromoterLuc8celltypes". (Right) Frequency of all selected TFBSs of "PromoterLucInHEK293". The number of selected TFBSs of "PromoterLuc8celltypes" (frequency>=5) and "PromoterLucInHEK293" are also given.

Here, I tried to predict the transcriptional activities of "*PromoterLucInHEK293*" by the predictive functions estimated for "*PromoterLuc8celltypes*" data set. There is a generally good prediction (the correlation coefficient is approximately 0.68) between predicted values and experimental values of "*PromoterLucInHEK293*" (Figure 27).

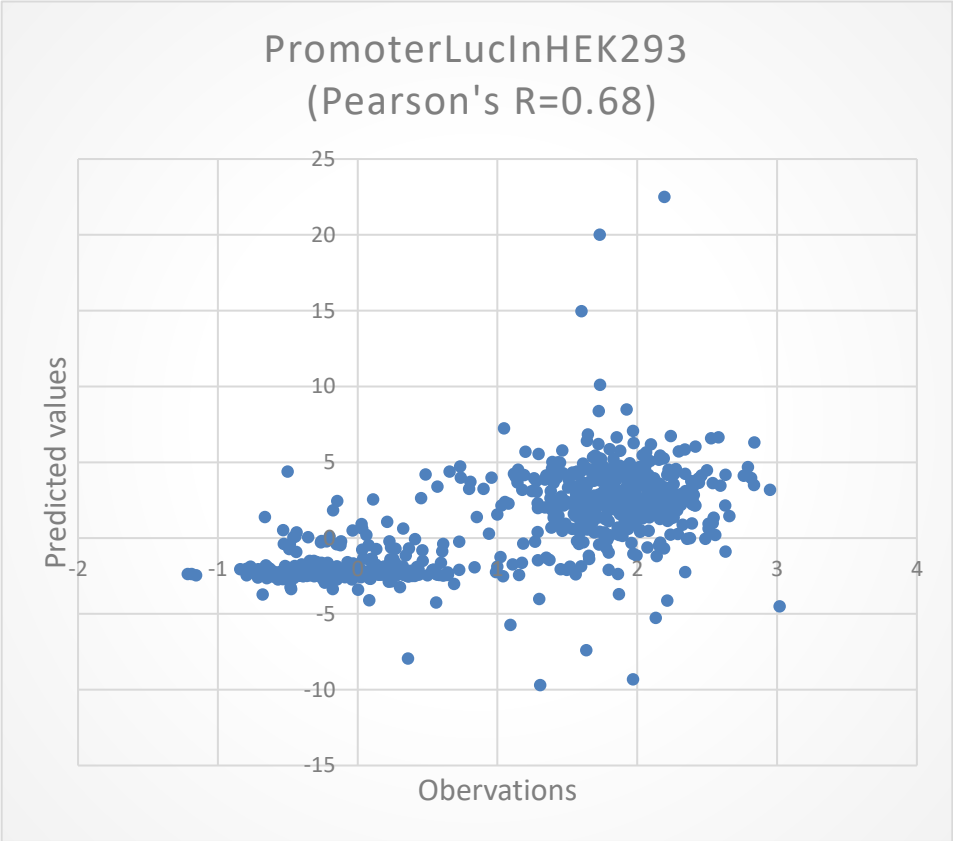


Figure 27. Plots between predicted transcriptional activities of "PromoterLucInHEK293" that were estimated by predictive functions of "PromoterLuc8celltypes" and the observations of "PromoterLucInHEK293".

I also investigated the samples of "*PromoterLucInHEK293*" which are not well-predicted by the predictive function of "*PromoterLuc8celltypes*". First, the samples of 5% most-over estimated and 5% most-under estimated were picked up and compared with the samples of 10% most predicted samples; second, 13 TFBSs were selected in the 10% outliners which have the fold

changes of TFBS enrichment scores>= 2 as compared with the 10% well-predicted samples. There are several TFBSs in the 13 TFBSs having tumor cell line-specific behaviours such as E2F family, EGR-1 and HIF-1alpha (43–47) (Table 10).

Table 10. TFBSs in which fold-change of enrichments >=2 in the 10% worst predicted samples by the predictive functions of "PromoterLuc8celltypes" and their mainly binding proteins were also shown.

TFBS label	Transcription factor	Previous reports
V$AHR_Q6	AhR	
V$E2F_Q6_01	E2F family	(44)
V$EGR1_Q6	EGR-1	(47)
V$HIF1A_Q6	HIF-1alpha	(43)
V$MAZ_Q6_01	MAZ	
V$MAZR_01	MAZ related factor	
V$MECP2_02	MECP2	
V$NANOG_01	Nanog	(45)
V$RNF96_01	RNF96	
V$SP1_Q6_01	Sp1 family	(46)
V$SP100_04	Sp100	
V$ZFP161_04	ZF5	
V$ZNF333_01	ZNF333	

4.6 Summary of different applications of the proposed method

In this study, I designed a new computational method to predict transcriptional activities of diverse MPRA data sets as well as luciferase reporter assays. The method constructs its predictive functions based on TRANSFAC database and machine learning algorithms of regression tree and MARS. I also proposed a feature redundancy-dependant formula for conventional regression tree for being adaptive to diverse data types.

The proposed method could be applied to diverse MPRAs as well as conventional luciferase reporter assay data sets despite of different transfected cell types (human, mouse and yeast), different sequence lengths (several ten bp to more than 1k bp), different number of sequences (several hundred to more than several ten thousand) and different sequence types (promoters, enhancers, artificial sequences, ChIP-seq peak regions and genomic variants). I applied the proposed method to the 10 data sets of MPRAs and luciferase assays and analyzed their candidate-active TFBSs according to the corresponding predictive models.

4.6.1 Investigating candidate active TFBSs

From the analysis of "*CREInducedInHEK293*" and "*CREBBPInMouseNeuron*", the candidate-active TFBSs estimated by the proposed method are characterized via tree structure. The TFBS trees have the advantages of simply understandable and could provide the biological information for data set clustering. The TFBSs which are selected by regression trees could be qualitatively considered as active TFBSs. And I also found that several candidate active TFBSs estimated by the proposed method were consistent with several previous studies.

4.6.2 Detecting experimental condition-specific TFBSs

In the analysis for the data sets "*RBCVariantsCtrlInK562*" and "*RBCVariantsGATA1InK562*", the proposed method could pick up the candidate-active TFBSs that response to GATA1 overexpression. The candidate-active TFBS trees and the sample clusters estimated by the proposed method also show different structures for different experimental conditions. The proposed method estimated several GATA1 overexpressing-responsive TFBSs and approximately 59% of them were identified by several previous studies. It suggests that the proposed method could detect the experimental condition-specific TFBSs.

4.6.3 Predicting transcriptional activities of unknown sequences by known data sets

Using the predictive functions for data set of "*CREBBPInMouseNeuron*", I predicted the transcriptional activities of new sequences of 18 motifs which had the same experimental condition as "*CREBBPInMouseNeuron*". For the data set of "*PromoterLucInHEK293*", the transcriptional activities could be estimated, in some extents, by the predictive functions of "*PromoterLuc8celltypes*". This suggests that using the proposed method, the transcriptional activities of unknown sequences could be predicted by known data sets, despite of the cell types.

5 DISCUSSION

In this study, I designed a new computational method to decipher regulatory code of transcription via estimating the relation between DNA sequences and the transcriptional activities of diverse MPRA data sets. The proposed method mainly consists of four steps: 1. Data pre-processing to format different MPRA data sets; 2. TRANSFAC database searching to encode sequences into explanatory variables and construct the explanatory variable matrix; 3. Variables clustering to assemble variables into more compact subpopulations by regression tree; 4. Performing MARS in different clusters to construct predictive functions (Figure 4).

The proposed method could be applied to diverse MPRA as well as to luciferase reporter assay data sets despite different transfected cell types, different sequence lengths (several ten bp to more than 1k bp), different number of sequences (several hundred to more than several ten thousand) and different sequence types (promoters, enhancers, artificial sequences, ChIP-seq peaks and genomic variants) (Table 2).

However, the proposed method employs the TRANSFAC database to encode sequences into explanatory variables and the estimated results are dependent on TRANSFAC database in some extents. In other words, unknown TFBSs which are not annotated in TRANSFAC database could not be characterized by the proposed method. On the other hand, the explanatory variables are encoded as the form of TFBS enrichment scores rather than several TFBS features such as positions and orientations. The missing information which should be provided by such trivial features probably cause the relative low sensitivity of this method. And the future work of this study should consider reducing these limitations.

The performances of the proposed method applying to different data sets showed different (Table 4). I found that for the data sets of "*CREInducedInHEK293*", "*PromoterLucInHEK293*" and "*TFBS75InYeast*", all the methods described in this study could obtain good predictions. This probably because the sequence patterns and/or transcriptional processes of these data sets are simple. And the data sets, the predictive precisions of which were lower than 0.7 estimated by

the proposed method, are all genomic segments except for promoters. It suggests that the chromatin contexts are more complex than the artificially designed sequences and promoters probably have relative simple transcriptional regulatory processes than enhancers.

According the applications of "*PromoterLucInHEK293*" and "*PromoterLuc8celltypes*" data sets, I found that the cell line specific TFBSs have contributions to transcriptional activities assayed in different cell lines. However, the promoters of unknown transcriptional activities could be estimated using known transcriptional activities despite of the different cell types, in some extents. It suggests that the common TFBSs make higher contributions to transcriptional activities across different cell types.

ACKNOWLEDGMENTS

First, I am grateful to my supervisor professor Suzuki for supervising. Thanks for the kindly teaching and supporting me to complete the postgraduate research. Thanks for training my scientific thought and behavior during my doctor course which would give me a great effect on future career. And thanks for giving me the happy time in Suzuki laboratory.

Besides my supervisor, I would like to thank Prof. Yada who supervised me in the master course and gave me great help during my Ph.D. study.

I would also like to thank the rest of my thesis committee: professor Kasahara, professor Morishita, and professor Tsuda for their insightful comments and encouragement.

My sincere thanks also go to Dr. Irie for giving me careful instructions of experimental skills although I was a layman of biological experiments before joining Suzuki laboratory.

I would like to K. Imamura, T. Horiuchi, T. Arauchi, Y. kuze, K. Shimizu and H. Wakaguri for their technical assistance. We are grateful to M. Seki, S. Meakawa for their useful advice of my Ph.D. study.

I also want to thank the Ministry of Education, Culture, Sports, Science, and Technology (MEXT) of the Japanese government for offering me the scholarship during my Ph.D. course.

Last but not the least, I would like to thank my parents for supporting me study in Japan.

REFERENCES

1. Crick,F. (1970) entral dogma of molecular biology. *Nature*, **227**.

2. ENCODE Project, Bernstein,B.E., Birney,E., Dunham,I., Green,E.D., Gunter,C. and Snyder,M. (2012) An integrated encyclopedia of DNA elements in the human genome. *Nature*, **489**, 57–74.

3. Shen,S.Q., Myers,C.A., Hughes,A.E.O., Byrne,L.C., Flannery,J.G. and Corbo,J.C. (2016) Massively parallel cis -regulatory analysis in the mammalian central nervous system. *Genome Res.*, **26**, 238–255.

4. White,M. a, Myers,C. a, Corbo,J.C. and Cohen,B. a (2013) Massively parallel in vivo enhancer assay reveals that highly local features determine the cis-regulatory function of ChIP-seq peaks. *Proc. Natl. Acad. Sci. U. S. A.*, **109**, 11952–7.

5. Ulirsch,J.C., Nandakumar,S.K., Wang,L., Giani,F.C., Zhang,X., Rogov,P., Melnikov,A., McDonel,P., Do,R., Mikkelsen,T.S., *et al.* (2016) Systematic functional dissection of common genetic variation affecting red blood cell traits. *Cell*, **165**, 1530–1545.

6. Kwasnieski,J.C., Fiore,C., Chaudhari,H.G. and Cohen,B.A. (2014) High-throughput functional testing of ENCODE segmentation predictions. *Genome Res*, **24**, 1595–1602.

7. Smith,R.P., Taher,L., Patwardhan,R.P., Kim,M.J., Inoue,F., Shendure,J., Ovcharenko,I. and Ahituv,N. (2013) Massively parallel decoding of mammalian regulatory sequences supports a flexible organizational model. *Nat. Genet.*, **45**, 1021–8.

8. Melnikov,A., Murugan,A., Zhang,X., Tesileanu,T., Wang,L., Rogov,P., Feizi,S., Gnirke,A., Callan,C.G., Kinney,J.B., *et al.* (2012) Systematic dissection and optimization of inducible enhancers in human cells using a massively parallel reporter assay. *Nat. Biotechnol.*, **30**, 271–7.

9. Arnold,C.D., Gerlach,D., Stelzer,C., Boryń,Ł.M., Rath,M. and Stark,A. (2013) Genome-wide quantitative enhancer activity maps identified by STARR-seq. *Science*, **339**, 1–4.

10. Hakim,O. and Misteli,T. (2012) SnapShot: Chromosome conformation capture. *Cell*, **148**, 16–18.

11. Shen,S.Q., Myers,C.A., Hughes,A.E.O., Byrne,L.C., Flannery,J.G. and Corbo,J.C. (2016) Massively parallel cis -regulatory analysis in the mammalian central nervous system. *Genome Res.*, 10.1101/gr.193789.115.

12. Kellner,W.A., Bell,J.S.K. and Vertino,P.M. (2015) GC skew defines distinct RNA polymerase pause sites in CpG island promoters. *Genome Res.*, **25**, 1600–1609.

13. Sarda,S., Das,A., Vinson,C. and Hannenhalli,S. (2017) Distal CpG islands can serve as alternative promoters to transcribe genes with silenced proximal promoters. 10.1101/gr.212050.116.

14. Sonay,T.B., Carvalho,T., Robinson,M.D., Greminger,M.P., Krützen,M., Comas,D., Highnam,G., Mittelman,D., Sharp,A., Marques-bonet,T., *et al.* expression divergence GC skew defines distinct RNA polymerase pause sites in CpG island promoters variation across humans of unwanted transcripts reflecting genetic background Extensive de novo mutation rate variation between individuals and across the ge.

15. Kwasnieski,J.C., Mogno,I., Myers,C.A., Corbo,J.C. and Cohen,B.A. (2012) Complex effects of nucleotide variants in a mammalian cis-regulatory element. *Proc. Natl. Acad. Sci.*, **109**, 19498–19503.

16. Sharon,E., Kalma,Y., Sharp,A., Raveh-Sadka,T., Levo,M., Zeevi,D., Keren,L., Yakhini,Z., Weinberger,A. and Segal,E. (2012) Inferring gene regulatory logic from high-throughput measurements of thousands of systematically designed promoters. *Nat. Biotechnol.*, **30**, 521–530.

17. Nguyen,T.A., Jones,R.D., Snavely,A.R., Pfenning,A.R., Kirchner,R., Hemberg,M. and Gray,J.M. (2016) High-throughput functional comparison of promoter and enhancer activities. *Genome Res.*, **26**, 1023–1033.

18. Dao,L.T.M., Galindo-albarrán,A.O., Castro-mondragon,J.A., Andrieu-soler,C., Medina-rivera,A., Souaid,C., Charbonnier,G., Griffon,A., Vanhille,L., Stephen,T., *et al.* (2017) Genome-wide characterization of mammalian promoters with distal enhancer functions. *Nat. Publ. Gr.*, **49**, 1073–1081.

19. Lenhard,B., Sandelin,A. and Carninci,P. (2012) Regulatory elements: Metazoan promoters: emerging characteristics and insights into transcriptional regulation. *Nat. Publ. Gr.*, **13**, 233–245.

20. Shlyueva,D., Stampfel,G. and Stark,A. (2014) Transcriptional enhancers: from properties to genome-wide predictions. *Nat. Rev. Genet.*, **15**, 272–86.

21. Breiman,L., Friedman,J., Stone,C.J. and Olshen,R.A. (1984) Classification and Regression Trees.

22. Friedman,J. (1991) Multivariate Adaptive Regression Splines. *Ann. Stat.*, **19**, 1–141.

23. Wilkinson, G. N. and Rogers,C.E. (1973) Symbolic Description of Factorial Models for Analysis of Variance. *Appl. Stat.*, **22**, 392–399.

24. Tibshirani,R. (1996) Regression Selection and Shrinkage via the Lasso. *J. R. Stat. Soc. B*, **58**, 267–288.

25. Yu,K. and Moyeed,R.A. (2001) Bayesian quantile regression. *Stat. Probab. Lett.*, **54**, 437–447.

26. Wingender,E. (2008) The TRANSFAC project as an example of framework technology that supports the analysis of genomic regulation. *Brief. Bioinform.*, **9**, 326–332.

27. R. KNÜPPEL, P. DIETZE, W. LEHNBERG, K. FRECH, and E.W. (2009) TRANSFAC Retrieval Program: A Network Model Database of Eukaryotic Transcription Regulating Sequences and Proteins. *J. Comput. Biol.*, **1**, 191–198.

28. D.Stormo,G., D.Schneider,T. and Gold,L. (1986) Quantitative analysis of the relationship between nucleotide sequence and functional activity. *Nucleic Acids Res.*, **14**, 6661–6680.

29. Irie,T., Park,S.J., Yamashita,R., Seki,M., Yada,T., Sugano,S., Nakai,K. and Suzuki,Y. (2011) Predicting promoter activities of primary human DNA sequences. *Nucleic Acids Res.*, **39**.

30. Landolin,J.M., Johnson,D.S., Trinklein,N.D., Aldred,S.F., Medina,C., Shulha,H., Weng,Z. and Myers,R.M. (2010) Sequence features that drive human promoter function and tissue specificity. *Genome Res.*, **20**, 890–898.

31. Walters,M. and Martin,D.I.K. (1992) Functional erythroid promoters created by interaction of the transcription factor GATA-1 with CACCC and AP-1 / NFE-2 elements. *Proc. Natl. Acad. Sci. U. S. A.*, **89**, 10444–10448.

32. Blobel,G. a, Nakajima,T., Eckner,R., Montminy,M. and Orkin,S.H. (1998) CREB-binding protein cooperates with transcription factor GATA-1 and is required for erythroid differentiation. *Proc. Natl. Acad. Sci. U. S. A.*, **95**, 2061–2066.

33. Manavathi,B., Lo,D., Bugide,S., Dey,O., Imren,S., Weiss,M.J. and Humphries,R.K. (2012) Functional Regulation of Pre-B-cell Leukemia Homeobox Interacting Protein 1 (PBXIP1 / HPIP) in Erythroid. *J. Biol. Chem.*, **287**, 5600–5614.

34. Anderson,K.P., Crable,S.C. and Lingrel,J.B. (2000) The GATA-E box-GATA motif in the EKLF promoter is required for in vivo expression. *Blood*, **95**, 1652–1655.

35. Tj,C., Bm,S., Waxman,S. and Scher,W. (1993) Inhibition of mouse GATA-1 function by the glucocorticoid receptor : possible mechanism of steroid inhibition of erythroleukemia cell differentiation . *Mol. Endocrinol.*, **7**, 528–542.

36. Zhang,F., Shen,G., Liu,X., Wang,F., Zhao,Y. and Zhang,J. (2012) Hypoxia-inducible factor 1 – mediated human GATA1 induction promotes erythroid differentiation under hypoxic conditions. *J. Cell. Mol. Med.*, **16**, 1889–1899.

37. Bartůnek,P., Králová,J., Blendinger,G., Dvorák,M. and Zenke,M. (2003) GATA-1 and c-myb crosstalk during red blood cell differentiation through GATA-1 binding sites in the c-myb promoter. *Oncogene*, **22**, 1927–35.

38. Caprioli,A., Koyano-Nakagawa,N., Iacovino,M., Shi,X., Ferdous,A., Harvey,R.P., Olson,E.N., Kyba,M. and Garry,D.J. (2011) Nkx2-5 represses gata1 gene expression and modulates the cellular fate of cardiac progenitors during embryogenesis. *Circulation*, **123**, 1633–1641.

39. Chen,R.-L., Chou,Y.-C., Lan,Y.-J., Huang,T.-S. and Shen,C.-K.J. (2010) Developmental silencing of human zeta-globin gene expression is mediated by the transcriptional repressor RREB1. *J. Biol. Chem.*, **285**, 10189–97.

40. Ross,J., Mavoungou,L., Bresnick,E.H. and Milot,E. (2012) GATA-1 Utilizes Ikaros and Polycomb Repressive Complex 2 To Suppress Hes1 and To Promote Erythropoiesis. *Mol. Cell. Biol.*, **32**, 3624–3638.

41. Papadopoulos,P., Gutiérrez,L., Demmers,J., Scheer,E., Pourfarzad,F., Papageorgiou,D.N., Karkoulia,E., Strouboulis,J., van de Werken,H.J.G., van der Linden,R., *et al.* (2015) TAF10 interacts with GATA1 transcription factor and controls mouse erythropoiesis. *Mol. Cell. Biol.*, **35**, MCB.01370-14.

42. Odom,D.T., Dowell,R.D., Jacobsen,E.S., Gordon,W., Danford,T.W., Macisaac,K.D., Rolfe,P.A., Conboy,C.M., Gifford,D.K. and Fraenkel,E. (2007) Tissue-specific transcriptional regulation has diverged significantly between human and mouse. *Nat. Genet.*, **39**, 2006–2008.

43. Chiavarina,B., Whitaker-Menezes,D., Migneco,G., Martinez-Outschoorn,U.E., Pavlides,S., Howell,A., Tanowitz,H.B., Casimiro,M.C., Wang,C., Pestell,R.G., *et al.* (2010) HIF1-alpha

functions as a tumor promoter in cancer associated fibroblasts, and as a tumor suppressor in breast cancer cells: Autophagy drives compartment-specific oncogenesis. *Cell Cycle*, **9**, 3534–3551.

44. Nevins,J.R. (2001) The Rb/E2F pathway and cancer. *Hum. Mol. Genet.*, **10**, 699–703.

45. Jeter,C.R., Yang,T., Wang,J., Chao,H.-P. and Tang,D.G. (2015) NANOG in Cancer Stem Cells and Tumor Development: An Update and Outstanding Questions. *Stem Cells*, **33**, 2381–2390.

46. Li,L. and Davie,J.R. (2010) The role of Sp1 and Sp3 in normal and cancer cell biology. *Ann. Anat.*, **192**, 275–283.

47. Anja Krones-Herzig, Shalu Mittal, Kelly Yule, Hongyan Liang, Chris English, Rafael Urcis, Tarun Soni, Eileen D. Adamson, Dan Mercola1 (2005) Early Growth Response 1 Acts as a Tumor Suppressor In vivo and In vitro via Regulation of p53. *Cancer Res.*, **65**, 12.

APPENDIX

A-1 R packages used in this study and the corresponding parameters

Algorithm	R package	Method	Specified parameters
Regression tree	rpart	rpart	Control.rpart (minbucket=*minbucket*) cp=0.01 (for multiple conditions data, cp=0.005)
MARS	earth	earth	degree=2
MLR	base	lm	default
Lasso	glmnet	glmnet	s=0.01
BQR	bayesQR	bayesQR	quantile=0.5

A-2 Source code of the proposed method

```
library("earth")

library("rpart")

library("rpart.plot")


minSet<-20

dg<-2

outputFolder<-""  ## setting the output folder


##### calculate the value of "minbucket" using the formula(2) ######

autoPCA<-function(matrix){

  pcaX<-princomp(matrix)

  tmp<-cumsum(pcaX$sdev^2 / sum(pcaX$sdev^2))[1]

  orgPara<-tmp

  auto<-(2^(-tmp)/(length(exp)^2))*10000000

  list(auto=auto,orgPara=orgPara,

       pc1=pcaX$scores[,1],pc2=pcaX$scores[,2])

}

##### calculate the value of "minbucket" over #######
```

```
######## 100-fold cross-validation #########

treeEarthCVAuto<-function(mydata,folder,outputfile,dg){

  nsamples<-nrow(mydata)

  nf<-100

  #set.seed(200)


  folds <- cut(sample(c(1:nsamples),nsamples, FALSE), breaks=nf,
labels=FALSE)


  cvPred=rep(0,nsamples)

  for(tt in 1:nf){

    #   print(tt)

    testIndex<-which(folds==tt)

    traindata<-mydata[-testIndex,]

    testdata<-mydata[testIndex,]

    auto<-autoPCA(traindata)$auto


    trainTree<-rpart(exp~.,data=traindata,   method   =   "anova",
model=TRUE,      control=      rpart.control(minbucket           =
max(nrow(traindata)*auto,minSet), cp=cpSet ))


    trainTreeValue<-predict(trainTree,traindata[,-1])

    testTreeValue<-predict(trainTree,testdata[,-1])
```

```
    trainCluster<-trainTree$where
    testCluster<-character(length(testTreeValue))

    pair<-unique(data.frame(trainTreeValue,trainCluster))
    for(rr in 1:nrow(pair)){
      tmp<-which(pair[rr,"trainTreeValue"]==testTreeValue)
      testCluster[tmp]<-pair[rr,"trainCluster"]
    }
    for(cc in 1:nrow(pair)){
      clusterNum<-pair[cc,"trainCluster"]
      sub<-traindata[which(trainCluster==clusterNum),]
      earthModel<-earth(exp~.,data=sub,degree=dg)
      tmp<-which(testCluster==clusterNum)
      if(length(tmp)>0){
       cvPred[testIndex[tmp]]<-predict(earthModel,testdata[tmp,])
      }
    }
  }
cvPred
}
```

########## 100-fold cross-validation over ###########

```
############ closed test #########
groupEarthClose<-function(group,dataframe,dg,output){
  groupLabels<-names(table(group))
  groupPred<-rep(0,length(exp))
  closeR<-tmp<-rep(0,length(groupLabels))
  count<-1
  for(gg in groupLabels){
    groupIndex<-which(group==gg)
    sub<-mydata[groupIndex,]
    earthModel<-earth(exp~.,data=sub,degree=dg)
    closeR[count]<-cor(predict(earthModel),exp[groupIndex])
    groupPred[groupIndex]<-predict(earthModel)
    tmp[count]<-length(earthModel$coefficients)
    write.table(cbind(as.matrix(earthModel$coefficients),gg),
output,
                append = TRUE, quote = FALSE, sep = ",",
                row.names = TRUE, col.names = FALSE)
    count<-count+1
  }
  clsSize<-paste(min(tmp)-1,"-",max(tmp)-1)
  list(groupPred=groupPred,closeR=closeR,clsSize=clsSize)
}
```

```
########### closed test over ############


############# main process ###########

setwd("")

folder<-"2.dataNew"


fileList<-c("DHSInMouseRetina",

          "RBCVariantsCtrlInK562",

          "RBCVariantsGATA1InK562",

          "TFBS75InYeast",

          "TFBS12InHepG2LVR",

          "TFBS12InMouseLVR",

          "CREBBPPeakInMouse",

          "CREInducedInHEK293",

          "Promoter8celltype",

          "PromoterLucInHEK293"

          )


orgPara=modPara=closeR=openR=bucket=clsSize<-rep(0,length(fileLi
st))

count<-1

for(ff in fileList){
```

```
print(ff)
file<-paste("TFMatrix_exp_minFN_",ff,".csv",sep="")
outputfile<-paste(outputFolder,"/",ff,sep="")
data<-read.table(paste(folder,"/",file,sep=""),
                 header=TRUE,sep=",",fill=TRUE)
ava<-which(!is.na(data[,ncol(data)]))

if(length(ava)>0){
  data<-data[ava,]
}
exp<-data[,3]

TFMatrix<-data[,c(4:ncol(data))]
ptm <- proc.time() ######### time cost

pca<-autoPCA(TFMatrix)
auto<-pca$auto
orgPara[count]<-pca$orgPara
modPara[count]<-auto

if(length(table(TFMatrix[,1]))==2){
```

```
    cpSet<-0.005
  }else{
    cpSet<-0.01
  }

  bucket[count]<-length(exp)*auto
  mydata<-data.frame(exp,TFMatrix)

  rpartModel<-rpart(exp~.,data=mydata,method = "anova",
model=TRUE,
                control=
rpart.control(minbucket=max(length(exp)*auto,minSet),
                                      cp=cpSet))
  group<-rpartModel$where

output<-paste(outputfile,"_group_coefficients_D",dg,".csv",sep="
")
  groupModel<-groupEarthClose(group,mydata,dg,output)
  print(proc.time() - ptm)  ####### time cost
#  pdf(paste(outputfile,"_tree_boxplot.pdf",sep=""))
#  boxplot(exp~group,cex.axis=1)
#  deV$off()
```

```
print(rpartModel)


pdf(paste(outputfile,"_PCA_tree_scatterplot.pdf",sep=""))
plot(pca$pc1,pca$pc2,pch=group,main=ff,
    col=colors()[as.numeric(group)*10] )
label<-as.numeric(names(table(group)))
legend("topleft", legend = label,
      col = colors()[label*10],
      pch = label)
deV$off()


if(length(table(group))>1) {
  pdf(paste(outputfile,"_tree_plot.pdf",sep=""))
  rpart.plot(rpartModel,type=2)
  deV$off()


}
closePredValue<-groupModel$groupPred
closeR[count]<-cor(closePredValue,exp)
clsSize[count]<-groupModel$clsSize
```

```
  pdf(paste(outputfile,"_allClosePlot_closeD",dg,".pdf",sep=""))
  plot(closePredValue,exp,pch=".",cex.lab=1,cex.axis=1,
       xlim=range(c(closePredValue,exp)),
ylim=range(c(closePredValue,exp)),
       xlab="activities",ylab="predictive values")
  deV$off()

  cvPred<-treeEarthCVAuto(mydata,folder,outputfile,dg)

  pdf(paste(outputfile,"_allCVPlot_openD",dg,".pdf",sep=""))
  plot(cvPred,exp,pch=".",cex.lab=1,cex.axis=1,
       xlim=range(c(cvPred,exp)), ylim=range(c(cvPred,exp)),
       xlab="activities",ylab="CV predictive values")
  deV$off()

  openR[count]<-cor(cvPred,exp)

write.table(cbind(ff,orgPara[count],modPara[count],bucket[count]
,
                  closeR[count],openR[count],clsSize[count]),
paste(outputFolder,"/",outputFolder,"_r.csv",sep=""),
```

```
            append = TRUE, quote = FALSE, sep = ",",

            row.names = FALSE, col.names = FALSE)


  count<-count+1

}
```

PAPER LIST

A new computational method to predict transcriptional activity of a DNA sequence from diverse datasets of massively parallel reporter assays

Ying Liu, Takuma Irie, Tetsushi Yada and Yutaka Suzuki

Nucleic Acids Res (2017) 45 (13): e124. DOI: https://doi.org/10.1093/nar/gkx396