Artificial Intelligence in Medicine 75 (2017) 16–23

Contents lists available at ScienceDirect

Artificial Intelligence in Medicine

journal homepage: www.elsevier.com/locate/aiim

A high-order representation and classification method for transcription factor binding sites recognition in *Escherichia coli*☆

Shiquan Sun[a,b], Xiongpan Zhang[a], Qinke Peng[a,*]

[a] *Systems Engineering Institute, Xi'an Jiaotong University, 28 Xianning West Road, Xi'an, Shaanxi 710049, China*
[b] *Department of Biostatistics, University of Michigan, 1415 Washington Heights, Ann Arbor, MI 48109, USA*

ARTICLE INFO

Article history:
Received 7 June 2016
Accepted 23 November 2016

Keywords:
Tensor
Partial least squares
Transcription factor binding sites
Machine learning
Classification
Computational biology

ABSTRACT

Background: Identifying transcription factors binding sites (TFBSs) plays an important role in understanding gene regulatory processes. The underlying mechanism of the specific binding for transcription factors (TFs) is still poorly understood. Previous machine learning-based approaches to identifying TFBSs commonly map a known TFBS to a one-dimensional vector using its physicochemical properties. However, when the dimension-sample rate is large (i.e., number of dimensions/number of samples), concatenating different physicochemical properties to a one-dimensional vector not only is likely to lose some structural information, but also poses significant challenges to recognition methods.
Materials and method: In this paper, we introduce a purely geometric representation method, tensor (also called multidimensional array), to represent TFs using their physicochemical properties. Accompanying the multidimensional array representation, we also develop a tensor-based recognition method, tensor partial least squares classifier (abbreviated as TPLSC). Intuitively, multidimensional arrays enable borrowing more information than one-dimensional arrays. The performance of each method is evaluated by average *F*-measure on 51 *Escherichia coli* TFs from RegulonDB database.
Results: In our first experiment, the results show that multiple nucleotide properties can obtain more power than dinucleotide properties. In the second experiment, the results demonstrate that our method can gain increased prediction power, roughly 33% improvements more than the best result from existing methods.
Conclusion: The representation method for TFs is an important step in TFBSs recognition. We illustrate the benefits of this representation on real data application via a series of experiments. This method can gain further insights into the mechanism of TF binding and be of great use for metabolic engineering applications.

1. Introduction

Transcription factors (TFs) are one of groups of proteins that bind to specific regions on the DNA sequence, thereby activating or repressing the rate of gene transcription [1,2]. In practical bioengineering applications, an effective method for identifying new TFBSs plays an important role in providing insights into cellular behavior, and helps us further understand the complex gene regulatory networks in cells [3,4].

Generally, the method for identifying TFBSs can be roughly divided into two categories: the experimental and computational approach. However, both categories are not mutually exclusive. Experimental methods can identify binding sites in some cases, such as DNase footprinting [5,6] and electrophoretic mobility shift assays [7,8]. However, due to the relatively short length and high degrees of degeneracy of such TFBSs, showing how the specificity of protein-DNA interactions is challenging. More specifically, with the advances in high-throughput sequencing technologies, the resolution is limited in hundreds of base-pairs (bps), and the procedure to identify TFBSs is still laborious and difficult in *in vivo* protein binding across the whole genome [9].

As supplement to the experimental method, the computational method not only identifies the real TFBSs in practice, but also provides useful instructions about the distribution of probes and potential binding sites. For example, in previous studies, consensus sequence and position-specific weight matrix (PWM) have been commonly used to model the sequence motifs [10–13]. In principle, these two methods can predict the binding sites via comparing

☆ Supported by China Natural Science Fund.
* Corresponding author.
E-mail addresses: shiquan_sun@126.com (S. Sun), panpan18022@163.com (X. Zhang), qkpeng@xjtu.edu.cn (Q. Peng).

http://dx.doi.org/10.1016/j.artmed.2016.11.004
0933-3657/

test sequences and consensus sequences. However, both methods result in a low identification rate because they both assume that the relationship between the nucleotide positions is independent. To address this issue, physicochemical properties (e.g., shape) are frequently introduced to help gain more information about the original DNA sequence [14–18]. To increase the prediction power, extensive studies leverage machine learning methods to train a prediction model, providing a promising way to identify TFBSs, such as support vector machine (SVM) [19,20,14], random forest (RF) [21,22], and deep learning [23].

Therefore, we can conclude that a well-performing method for identifying TFBSs mainly depends not only on a powerful prediction model but also a good representation method, which contains as much information about sequences as possible. However, there are several potential drawbacks when a DNA sequence is represented as a one-dimensional numeric vector. Theoretically, randomly permuting (or re-ranking) features do not affect the accuracy of the prediction model. In other words, the one-dimensional numeric vector and its corresponding DNA sequence do not necessarily have one-to-one correspondence, and the different binding sites might have the same distribution pattern after we re-rank the features, which contradicts with our original intention. On the other hand, the letter features (Section 2) will become useless if the identifying procedure incorporates a feature selection step. Because four features together represent one type of nucleobases, separating the four features becomes meaningless in practice. A promising way to deal with this issue is to use multidimensional array-based representation [24,25]. This type of representation has been successfully applied to EEG signals classification in biomedical engineering [26–28], image processing in computer vision or pattern recognition [29–31], and other fields [32–34].

In this paper, moving beyond the one-dimensional representation of TFBSs, we first represent a TFBS as a multidimensional array where the rows exhibit physicochemical properties of the DNA sequence, such as shear, stretch and shift, and the columns denote the different base pair steps (k-mers) within subsequent motifs. The elements in the multidimensional array indicate the value of physicochemical features with respect to k-mers. Accompanying the multidimensional array representation, we also develop a multidimensional array-based PLS classifier (TPLSC) to predict TFBSs. The experiments were conducted on 51 TFs in *Escherichia coli* from RegulonDB, and the results demonstrate that our method can significantly improve the recognition rate, especially for the integration host factor (IHF), which is well-known to exhibit both features specific to each base and DNA structural properties [35].

The rest of the paper is organized as follows: in Section 2, we illustrate the detailed process of multidimensional array-based representation for TFBSs. In Section 3, we discuss the standard partial least squares classifier and tensor partial least squares classifier together to demonstrate the relationship between two types of classifiers. The results are given in Section 4. Some concluding remarks are presented in Section 5.

2. Materials and TFBSs representation

In this section, we illustrate the detailed process of high-order representation for TFBSs. The real data sets confirmed by experiments can be downloaded from the RegulonDB v8.0 database (http://regulondb.ccg.unam.mx/ (accessed: 10.03.16)). This database collects the *E. coli* k-12 transcription information, and aims to build a comprehensive transcription regulation network [36]. In the current study, the real transcription factor binding sites were derived from the reference sequences (*E. coli* k-12 genome MG1655 (NCBI: NC_000913.3)), according to the starting position and the ending position which were from the RegulonDB database. To make comprehensive comparison, we randomly selected 1000 sequences from background genome sequences (non-coding sequences) as the negative samples to distinguish from the known TFBSs (positive samples).

Table 1
The combinations of different properties, and their corresponding values were collected from [19,37]. n is the number of binding sites for a specific TF, and the number n can be found in Fig. 4.

Combination	Description	Dimension
Di	All possible 2-mers properties	$n \times 111 \times 35$
DiL	All possible 2-mers properties and the letter features	$n \times 115 \times 35$
Mu	3-mers, 4-mers, and 7-mers properties	$n \times 70 \times 35$
MuL	3-mers, 4-mers and 7-mers properties, and the letter features	$n \times 74 \times 35$
DiMu	2-mers, 3-mers, 4-mers, and 7-mers properties	$n \times 181 \times 35$
DiMuL	2-mers, 3-mers, 4-mers and 7-mers properties, and the letter features	$n \times 185 \times 35$

Briefly, we summarized two ways to represent TFBSs from previous studies: base pair steps (e.g., 2-mer, 3-mer, and 7-mer), and geometrical parameters of base pairs (e.g., shear, stretch, and shift). In this paper, we focused on the physicochemical properties recorded as 2-mers to characterize the specific TFBSs, and the extended physicochemical properties recorded as 3-mers, 4-mers, and 7-mers from two recent studies [37,19]. For 2-mers, we collected all dinucleotide properties from DiProDB database (http://diprodb.fli-leibniz.de/ShowTable.php (accessed: 10.03.16)), and the total number of corresponding properties was 110. For k-mers (k = 3, 4, 7), all dinucleotide properties were collected from the Additional Materials in the paper [19] and the total number of corresponding properties for 3-mers was 62, 4-mers was 6, and 7-mers was 2. The papers have not provided the properties for 5-mers, 6-mers or other base pair steps; therefore, we left out these features in our study. Additionally, we also incorporated the letter features to provide the same information as used in PWM-based approaches. As described in previous studies [37,19], letter features were generated by designating the four kinds of nucleotides – A, C, G, and T – as mutually orthogonal 4D vectors (1,0,0,0), (0,1,0,0), (0,0,1,0), and (0,0,0,1), respectively.

We extended the length of all TFBSs with flanking nucleotides to 41 base pairs. As shown in the first step of Fig. 1, if we slide a subwindow from left to right on a 41 base pairs sequence, it will generate 35 features for 7-mers, 40 features for 2-mers, 39 features for 3-mers, and 38 features for 4-mers. To make a unified representation, we symmetrically discarded the nucleotides from both sides to ensure all k-mers with the same length (35). To clearly show the process of tensor representation, we take a binding site from AgaR TF as an example (Fig. 1), for 3-mers, we have 62 physicochemical properties and 35 features which form a 62×35 matrix, and the element $a_{1,1}$ in the matrix indicates the value of the physicochemical properties (such as 'shear') with respect to the first 3-mer feature, TTA; for 4-mers, 6 physicochemical properties and 35 features which form a 6×35 matrix; for 7-mers, 2 physicochemical properties and 35 features which form a 2×35 matrix. Then we simply concatenate the three matrices to form a tensor $\mathbf{X}^{(\mathbf{n})}$ ($1 \times 70 \times 35$). Assuming there are 11 binding sites for AgaR TF, therefore, we can obtain a third order tensor $\mathcal{X}$ in which the order is number of binding sites $\times$ number of physicochemical properties $\times$ Number of features ($11 \times 70 \times 35$). We did not illustrate the 2-mers (110×35 matrix) and the letter features (4×35 matrix) in Fig. 1. However, the process is similar to what we described above. The dimensionality of each tensor $\mathcal{X}$ is shown in Table 1.

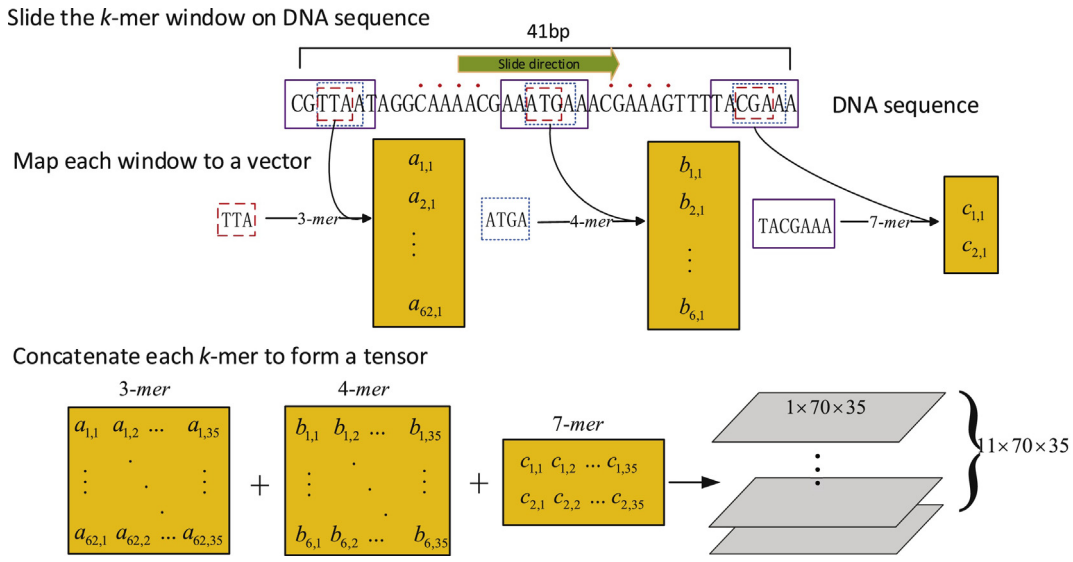

Fig. 1. Schematic diagram of the process from a raw DNA sequence to a tensor. This is an example for one of binding sites from AgaR TF. The total number of properties for 3-mers, 4-mers, and 7-mers are 62, 6, and 2, respectively. We concatenated the k-mers with the same length (35) to form a tensor ($1 \times 70 \times 35$). The total number of binding sites for AgaR TF is 11; therefore, we can obtain a tensor ($11 \times 70 \times 35$) for AgaR TF.

We can summarize the process to form a tensor as the following three steps:

- Obtaining different k-mers, and starting at different positions to ensure the same length (35);
- Mapping the k-mers to a vector using its physicochemical properties;
- Concatenating each matrix from k-mers to forming a tensor.

3. Methods overview

3.1. Notation

Throughout this paper, N-dimensional vectors are denoted by lowercase boldface letters, e.g., $\mathbf{x} \in \mathbb{R}^N$; $I_1 \times I_2$ order matrices are denoted by uppercase boldface letters, e.g., $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2}$; and N-order tensors are denoted by calligraphic letters, e.g., $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. Indices are denoted by lowercase letters and span the range from 1 to its uppercase version, for example, $i_N = 1, 2, \cdots, I_N$.

Definition 1. The n-model product of a tensor $\mathcal{X}$ and a matrix $\mathbf{B} \in \mathbb{R}^{J_n \times I_n}$ can be defined as:

$$\mathcal{A} = \mathcal{X} \times_n \mathbf{B}$$

where $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times J_n \times I_{n+1} \times \cdots \times I_N}$, $a_{i_1 i_2 \cdots i_{n-1} j_n i_{n+1} \cdots i_N} = \sum_{i_n} x_{i_1 i_2 \cdots i_n \cdots i_N} b_{j_n i_n}$.

Definition 2. The n-model cross covariance between $\mathcal{X}$ and $\mathbf{Y}$ can be defined as:

$$\mathrm{cov}_{\{n;1\}}(\mathcal{X}, \mathbf{Y}) = \mathcal{X} \times_n \mathbf{Y}^T$$

where $\mathcal{X}$ and $\mathbf{Y}$ have the same size on the nth mode.

In the current study, each TF is represented as a 3-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, and the matrix $\mathbf{Y} \in \mathbb{R}^{J_1 \times J_2}$ is encoded as class membership in binary form ($J_2 = 2$, with each column denoting one class).

3.2. Standard PLS classifier

Partial least squares (PLS) shares the characteristics of canonical correlation analysis (CCA) and principal component analysis (PCA), and be applied in situations where the number of observed variables (features) D is significantly greater than the number of observations (instances) I (i.e., $I \ll D$, multicolinear problem) [38–41].

The goal of PLS is to optimize the mathematical model formulated as follows:

$$\max_{\mathbf{w},\mathbf{q}} [\mathrm{cov}(\mathbf{Xw}, \mathbf{Yq})]^2, \quad \text{s.t.} \quad \mathbf{w}^T\mathbf{w} = 1, \quad \mathbf{q}^T\mathbf{q} = 1. \tag{1}$$

Essentially, to solve this optimization problem, we are required to seek a set of *latent vectors* (also called score vectors) $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_R]$, and *loading vectors* $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_R]$ (related to $\mathbf{X} \in \mathbb{R}^{I \times D}$) and $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_R]$ (related to $\mathbf{Y} \in \mathbb{R}^{I \times K}$) to reconstruct the original data $\mathbf{X}$ and $\mathbf{Y}$, i.e.,

$$\mathbf{X} = \mathbf{TP}^T + \mathbf{E} = \sum_{r=1}^{R} \mathbf{t}_r \mathbf{p}_r^T + \mathbf{E} \tag{2}$$

$$\mathbf{Y} = \mathbf{TDQ}^T + \mathbf{F} = \sum_{r=1}^{R} d_{rr} \mathbf{t}_r \mathbf{q}_r^T + \mathbf{F} \tag{3}$$

where $\mathbf{T} = \mathbf{XW}$ ($\mathbf{W}$ is *weight vectors*). $\mathbf{D}$ is a diagonal matrix, $\mathbf{D} = \mathrm{diag}(d_{11}, \ldots, d_{RR})$. The matrices $\mathbf{E}$ and $\mathbf{F}$ are the residual of $\mathbf{X}$ and $\mathbf{Y}$, respectively.

Multiple ways have been developed to solve standard PLS [42], and we can select one of them according to the intention of practical applications when the data is presented by one-dimensional vector. However, when the data is represented by a tensor, their corresponding mathematical model and algorithm should be different (Section 3.3).

Although PLS is not inherently designed for classification, it can be easily modified for this purpose, and is routinely used for classification [43,44]. To predict the new data $\mathbf{X}^{new}$, the procedure can be performed by

$$\mathbf{Y}^{new} = \mathbf{X}^{new}\mathbf{WDQ}^T \tag{4}$$

Once we obtain the predicted value for the new data $\mathbf{Y}^{new}$, two ways can be used to identify which class they belong to. Firstly, we can determine the maximum value of each column directly. Secondly, we can use Bayesian discrimination to find the optimum

threshold in the training procedure, and then use the threshold in the testing procedure. In the current study, we used the latter. For the final decision of membership, the standard PLS and tensor PLS are the same after obtaining $\mathbf{Y}^{new}$.

3.3. Tensor PLS classifier

Tensor PLS has been already proven useful in QSAR [45], brain computer interface [46,47] and other applications [48,49]. Similar to optimizing the PLS model described above, tensor PLS can be reformulated as follows:

$$\max_{\{\mathbf{P}^{(n)}\},\mathbf{q}} \left[\text{cov}\left(\mathcal{X} \times_{(2)} \mathbf{P}^{(1)T} \times_{(3)} \cdots \times_{(N)} \mathbf{P}^{(N-1)T}, \mathbf{Yq}\right)\right]^2, \quad \text{s.t.} \quad \mathbf{P}^{(n)T}\mathbf{P}^{(n)} = \mathbf{I},\ \mathbf{q}^T\mathbf{q} = 1. \tag{5}$$

The tensor data $\mathcal{X}$ can be decomposed as the sum of rank-$(1, L_2, \ldots, L_N)$ tensors (Fig. 2(a)), i.e.,

$$\mathcal{X} = \sum_{r=1}^{R} \mathcal{R}_r \times_1 \mathbf{t}_r \times_2 \mathbf{P}_r^{(1)} \times_3 \cdots \times_N \mathbf{P}_r^{(N-1)} + \mathcal{E} \tag{6}$$

where $\mathcal{E}$ is the residual of $\mathcal{X}$ and R is the number of latent vectors. $\mathcal{R}$ is a core tensor and $\mathbf{P}^{(n)}$ is a factor matrix ($n = 1, 2, \ldots, N-1$). Eq. (6) is the *Tucker model* tensor decomposition [50,51]. The class membership matrix $\mathbf{Y}$ can be also approximated by Eq. (3), which is the same as standard PLS (Fig. 2(b)). To predict the new tensor $\mathcal{X}^{new}$, the procedure is performed by

$$\mathbf{Y}^{new} = \mathcal{X}^{new}_{(1)}\mathbf{WDQ}^T \tag{7}$$

After obtaining the predicted class membership matrix $\mathbf{Y}^{new}$ using Eq. (7), we utilize the Bayesian discrimination to determine the class membership of TFBSs, this procedure is the same as that in standard PLS classifier. To clearly show the procedure for TFBSs recognition, the TPLSC algorithm is outlined in Algorithm 1. Moreover, the MATLAB source code and all TFBSs data sets are freely available at https://github.com/sqsun/HOPLSC_TFBSs (accessed 01.06.16).

Algorithm 1. Tensor partial least squares classifier (TPLSC) for TFBSs recognition

Data: The data $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, $\mathbf{Y} \in \mathbb{R}^{J_1 \times J_2}$, and $\mathcal{X}^{new}$
Result: The predicted data $\mathbf{Y}^{new}$

```
Initialization: E_1 = X, F_1 = Y;
for r in 1 to R do
    repeat
        C_r = C_r ×_1 F_r ;
        C_r = R_r ×_1 q_r ×_2 P_r^(1) ×_3 P_r^(2);
        t_r = (E_r ×_2 P_r^(1) ×_3 P_r^(2))_(1) (vec^T(R_r))^+;
        Update R_r;
        d_rr = (F_r q_r)^T t_r;
        Deflation E_r and F_r;
    until || E_r ||< ε and || F_r ||< ε;
end
Y^new = X^new W D Q^T ;
```

4. Experiments and results

We performed a series of experiments on 51 real TFs to compare the performance of TPLSC with four other popular machine learning-based recognition methods: support vector machine with linear kernel (SVML); support vector machine with linear kernel as well as incorporating feature selection (SVML_FS); support vector machine with RBF kernel as well as incorporating feature selection (SVMR_FS); and random forest with feature selection (RF_FS). The parameter settings of SVM variants were the same as the previous work [19], i.e., the penalty parameter (C) and the RBF kernel function parameter (γ) were performed with 2D grid search to find the optimal parameters; the range of C was set to $\{2^1, 2^2, 2^3, \ldots, 2^{15}\}$ and the range of γ was set to $\{2^{-10}, 2^{-9}, 2^{-8}, \ldots, 2^{-1}\}$; and for each pair of C and γ, cross-validation was conducted on the training data to evaluate the performance of parameter pairs, $\{C, \gamma\}$. For random forest, we also used the default parameter setting, i.e., the number tree was 500.

In the training procedure, we performed 3-fold cross validation (3-CV) on each TF data set, i.e., the training data set was randomly split into three parts, one of which was a test set and the remaining parts were training sets. The reason we used 3-fold cross-validation was that some TFs had only 5 or 6 positive samples in the training procedure. All results were assessed by average F-measure over 10 independent runs. The F-measure is a commonly used measurement to assess the performance of the classifier when the number of positive samples is small [19]; it is formulated as follows:

$$F\text{-measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

where recall = TP/(TP + FN), precision = TP/(TP + FP). Here, TP, FP, and FN are true positive, false positive, and false negative, respectively.

Our first experiment was designed to investigate how these properties affect the performance of TFBSs recognition. We conducted the experiment on three possible groups of DNA properties: the conformational properties recorded as 2-mers [37] (i.e., dinucleotide properties); the physicochemical properties recorded as 3-mers, 4-mers, and 7-mers [14,19]; and their combinations. For simplicity, we denoted different combinations of the properties as simple names, as shown in Table 1. The first two properties (Di and DiL) represent dinucleotide properties, and dinucleotide properties and the letter features, respectively. The next two (Mu and MuL) denote multiple nucleotide properties, and multiple nucleotide properties and the letter features, respectively. The last two (DiMu and DiMuL) are their combinations. Intuitively, combining the two group properties is expected to provide increased power in predictions because they characterize the different aspects using k-mers for DNA sequences.

As shown in Fig. 3, Mu (blue) or MuL (magenta) outperforms other properties for most data sets. The average performance of Di, DiL, Mu, MuL, DiMu, and DiMuL across 51 TFs are 0.1879, 0.1935, 0.4587, 0.4696, 0.2004, and 0.2093, respectively. Unexpectedly, combining Di (or DiL) and Mu (or MuL) properties shows just slightly better performance than using Di (or DiL) alone. Moreover, we can see that incorporating the letter features into the combinations does not consistently increase the prediction power (F-measure). For AraC (the second TF in Fig. 3), incorporating the letter feature into Di and Mu can improve the performance while it decreases the performance for ArcA (the third TF in Fig. 3). The number of TFs improved by MuL compared with Di, DiL, DiMu, and DiMuL are 46 (with improvement roughly more than 90%), 45 (roughly more than 88%), 47 (roughly more than 92%), and 45 (roughly more than 88%), respectively.

Our second experiment was designed to assess the performance of TPLSC with several other popular machine learning-based methods in TFBSs recognition. In our first experiment, the results illustrated that Mu or MuL can provide more increased power than other properties. Therefore, in this experiment, we only focused on Mu and MuL. To illustrate how the letter features affect the performance of each method, TPLSC method was carried out on both Mu and MuL, but other methods were carried out on MuL because of incorporating feature selection step.

All results of the methods based on the average F-measure are summarized in Table 2. As shown in Table 2, the best performance is TPLSC on Mu (denoted as TPLSC) or TPLSC on MuL (denoted as TPLSC-Letter). For specific TF, some binding sites could not be identified completely by existing methods but can be recognized by the proposed method (TPLSC), such as CytR,

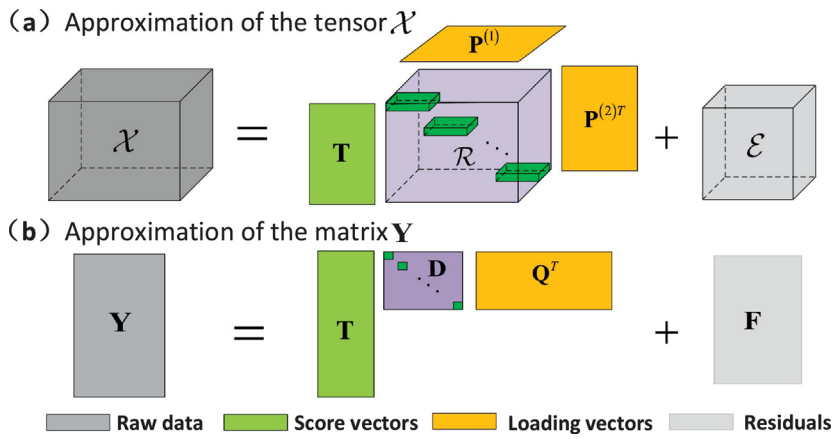

Fig. 2. The Tucker decomposition process for tensor $\mathcal{X}$ and matrix **Y**. (a) Approximation of the tensor $\mathcal{X}$ which can be decomposed as three parts: latent matrix **T**, factor matrix **P**, and core tensor $\mathcal{R}$. (b) Approximation of the matrix **Y**. The decomposition process of **Y** in TPLSC is the same as that in PLS.

Table 2
Comparison of six methods over 10 independent training runs on 51 TFs in *E. coli*. The performance of each method was evaluated by average *F*-measure and standard deviation.

TF	SVML	SVML_FS	SVMR_FS	RF_FS	TPLSC	TPLSC-Letter
AgaR	0.4179 ± 0.0895	0.3474 ± 0.0502	0.2822 ± 0.0563	0.1622 ± 0.0183	0.5259 ± 0.0850	0.4550 ± 0.0681
AraC	0.4685 ± 0.0939	0.4284 ± 0.0842	0.4880 ± 0.0605	0.3448 ± 0.0646	0.5545 ± 0.0525	0.6177 ± 0.0684
ArcA	0.3334 ± 0.0375	0.3794 ± 0.0529	0.3905 ± 0.0166	0.2551 ± 0.0175	0.6582 ± 0.0494	0.6507 ± 0.0492
ArgR	0.8122 ± 0.0302	0.8031 ± 0.0528	0.7781 ± 0.0540	0.3268 ± 0.0318	0.7928 ± 0.0294	0.7616 ± 0.0374
CpxR	0.2372 ± 0.0564	0.2403 ± 0.0310	0.2645 ± 0.0306	0.3312 ± 0.0210	0.5756 ± 0.0464	0.5849 ± 0.0272
CRP	0.8065 ± 0.0138	0.8074 ± 0.0148	0.8538 ± 0.0191	0.8100 ± 0.0101	0.9829 ± 0.0041	0.9176 ± 0.0112
CytR	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0250 ± 0.0116	0.1232 ± 0.0194	0.1496 ± 0.0288
DeoR	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0333 ± 0.0154	0.0000 ± 0.0000
DnaA	0.0190 ± 0.0402	0.1333 ± 0.0497	0.0000 ± 0.0000	0.0222 ± 0.0022	0.1881 ± 0.0083	0.2697 ± 0.0442
FadR	0.6897 ± 0.0785	0.5827 ± 0.0807	0.6476 ± 0.0586	0.5690 ± 0.0229	0.6979 ± 0.0456	0.7295 ± 0.0399
Fis	0.4752 ± 0.0177	0.4719 ± 0.0186	0.5591 ± 0.0181	0.3287 ± 0.0107	0.8848 ± 0.0164	0.8830 ± 0.0142
FlhDC	0.1887 ± 0.0988	0.2424 ± 0.0545	0.1533 ± 0.0236	0.2045 ± 0.0116	0.3608 ± 0.0607	0.3079 ± 0.0273
FNR	0.7349 ± 0.0196	0.7302 ± 0.0194	0.7525 ± 0.0208	0.7612 ± 0.0104	0.8602 ± 0.0239	0.8647 ± 0.0279
Fur	0.7385 ± 0.0125	0.7039 ± 0.0248	0.7505 ± 0.0146	0.4164 ± 0.0211	0.8653 ± 0.0208	0.8649 ± 0.0134
GadE	0.0000 ± 0.0000	0.0444 ± 0.0099	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0889 ± 0.0463	0.0833 ± 0.0416
GalR	0.2689 ± 0.0122	0.3390 ± 0.0406	0.3244 ± 0.0000	0.3556 ± 0.0192	0.4319 ± 0.0344	0.4886 ± 0.0148
GalS	0.3235 ± 0.0175	0.3048 ± 0.0246	0.3003 ± 0.0115	0.3956 ± 0.0138	0.4932 ± 0.0073	0.4944 ± 0.0112
GcvA	0.3044 ± 0.0234	0.2578 ± 0.0155	0.0444 ± 0.0000	0.0167 ± 0.0028	0.1967 ± 0.0074	0.1246 ± 0.0068
GlpR	0.1309 ± 0.0200	0.1881 ± 0.0353	0.1453 ± 0.0187	0.1917 ± 0.0103	0.3857 ± 0.0338	0.3821 ± 0.0118
GntR	0.0622 ± 0.0088	0.1581 ± 0.0087	0.0533 ± 0.0184	0.0644 ± 0.0047	0.4027 ± 0.0064	0.4040 ± 0.0068
H-NS	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.1552 ± 0.0362	0.1440 ± 0.0438
IclR	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0778 ± 0.0099	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0229 ± 0.0049
IHF	0.0722 ± 0.0101	0.1053 ± 0.0149	0.1084 ± 0.0142	0.0381 ± 0.0105	0.4947 ± 0.0195	0.4707 ± 0.0114
IscR	0.0933 ± 0.0144	0.1600 ± 0.0116	0.0990 ± 0.0000	0.0533 ± 0.0047	0.1672 ± 0.0065	0.2095 ± 0.0066
LexA	0.8148 ± 0.0190	0.7922 ± 0.0350	0.8155 ± 0.0155	0.8378 ± 0.0321	0.8355 ± 0.0351	0.8630 ± 0.0372
Lrp	0.0085 ± 0.0015	0.0324 ± 0.0100	0.0562 ± 0.0013	0.2004 ± 0.0026	0.3843 ± 0.0026	0.4215 ± 0.0013
MalT	0.2896 ± 0.0067	0.2906 ± 0.0069	0.2568 ± 0.0028	0.6155 ± 0.0107	0.5203 ± 0.0019	0.5337 ± 0.0016
MarA	0.0074 ± 0.0234	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0981 ± 0.0214	0.1933 ± 0.0380	0.1596 ± 0.0337
MelR	0.4084 ± 0.0244	0.4556 ± 0.0221	0.5674 ± 0.1659	0.2411 ± 0.0216	0.2184 ± 0.0310	0.3406 ± 0.0191
MetJ	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0133 ± 0.0141	0.1710 ± 0.0057	0.1934 ± 0.0024	0.2675 ± 0.0178
MetR	0.3900 ± 0.0652	0.4244 ± 0.0521	0.0778 ± 0.0004	0.1444 ± 0.0207	0.6163 ± 0.0577	0.6015 ± 0.0380
ModE	0.0556 ± 0.0907	0.2000 ± 0.0944	0.0000 ± 0.0000	0.0000 ± 0.0000	0.2500 ± 0.0474	0.3133 ± 0.0310
Nac	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0389 ± 0.0083
NagC	0.9029 ± 0.0279	0.8884 ± 0.0436	0.8869 ± 0.0249	0.9226 ± 0.0109	0.8843 ± 0.0350	0.8885 ± 0.0386
NanR	0.7489 ± 0.0638	0.7200 ± 0.0633	0.7822 ± 0.5132	0.7454 ± 0.0232	0.7508 ± 0.0330	0.8292 ± 0.0407
NarL	0.1870 ± 0.0362	0.2189 ± 0.0452	0.2443 ± 0.0233	0.1189 ± 0.0126	0.4649 ± 0.0249	0.5353 ± 0.0324
NarP	0.0324 ± 0.0419	0.0524 ± 0.0481	0.0167 ± 0.0162	0.0345 ± 0.0220	0.3079 ± 0.0138	0.3405 ± 0.0391
NtrC	0.8871 ± 0.0426	0.8875 ± 0.0193	0.8980 ± 0.0205	0.8954 ± 0.0314	0.8812 ± 0.0511	0.8732 ± 0.0756
OmpR	0.1274 ± 0.0121	0.1529 ± 0.0102	0.2351 ± 0.0288	0.5598 ± 0.0138	0.4323 ± 0.0165	0.4437 ± 0.0104
OxyR	0.7209 ± 0.0473	0.6761 ± 0.0423	0.7441 ± 0.0356	0.0940 ± 0.0119	0.7230 ± 0.0314	0.7304 ± 0.0221
PhoB	0.3440 ± 0.0655	0.4368 ± 0.0767	0.3937 ± 0.0501	0.5875 ± 0.0372	0.5073 ± 0.0512	0.4910 ± 0.0481
PhoP	0.4166 ± 0.0611	0.4686 ± 0.0720	0.4519 ± 0.0615	0.2913 ± 0.0226	0.5888 ± 0.0405	0.6217 ± 0.0445
PspF	0.5222 ± 0.0403	0.4222 ± 0.0648	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0333 ± 0.0805	0.0333 ± 0.0354
PurR	0.8872 ± 0.0323	0.8772 ± 0.0419	0.9208 ± 0.0523	0.9636 ± 0.0304	0.9121 ± 0.0533	0.9251 ± 0.0412
RcsAB	0.0611 ± 0.0100	0.1778 ± 0.0086	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0250 ± 0.0056	0.0000 ± 0.0000
Rob	0.0000 ± 0.0000	0.0711 ± 0.0655	0.0000 ± 0.0000	0.0133 ± 0.0318	0.2306 ± 0.0598	0.1100 ± 0.0532
SoxS	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0720 ± 0.0429	0.0487 ± 0.0658	0.1445 ± 0.0516
TorR	0.2410 ± 0.0773	0.2889 ± 0.0497	0.0756 ± 0.0428	0.2133 ± 0.0126	0.4869 ± 0.0416	0.5274 ± 0.0578
TrpR	0.8189 ± 0.0641	0.8186 ± 0.0710	0.7692 ± 0.0467	0.5975 ± 0.0166	0.6822 ± 0.0648	0.7838 ± 0.0608
TyrR	0.4156 ± 0.0231	0.4541 ± 0.0380	0.4692 ± 0.0240	0.4822 ± 0.0178	0.6025 ± 0.0246	0.5272 ± 0.0361
UxuR	0.8370 ± 0.0080	0.8343 ± 0.0097	0.6800 ± 0.0060	0.7533 ± 0.0082	0.7002 ± 0.0072	0.7263 ± 0.0048
Avg.	0.3393	0.3543	0.3221	0.2947	0.4587	0.4696

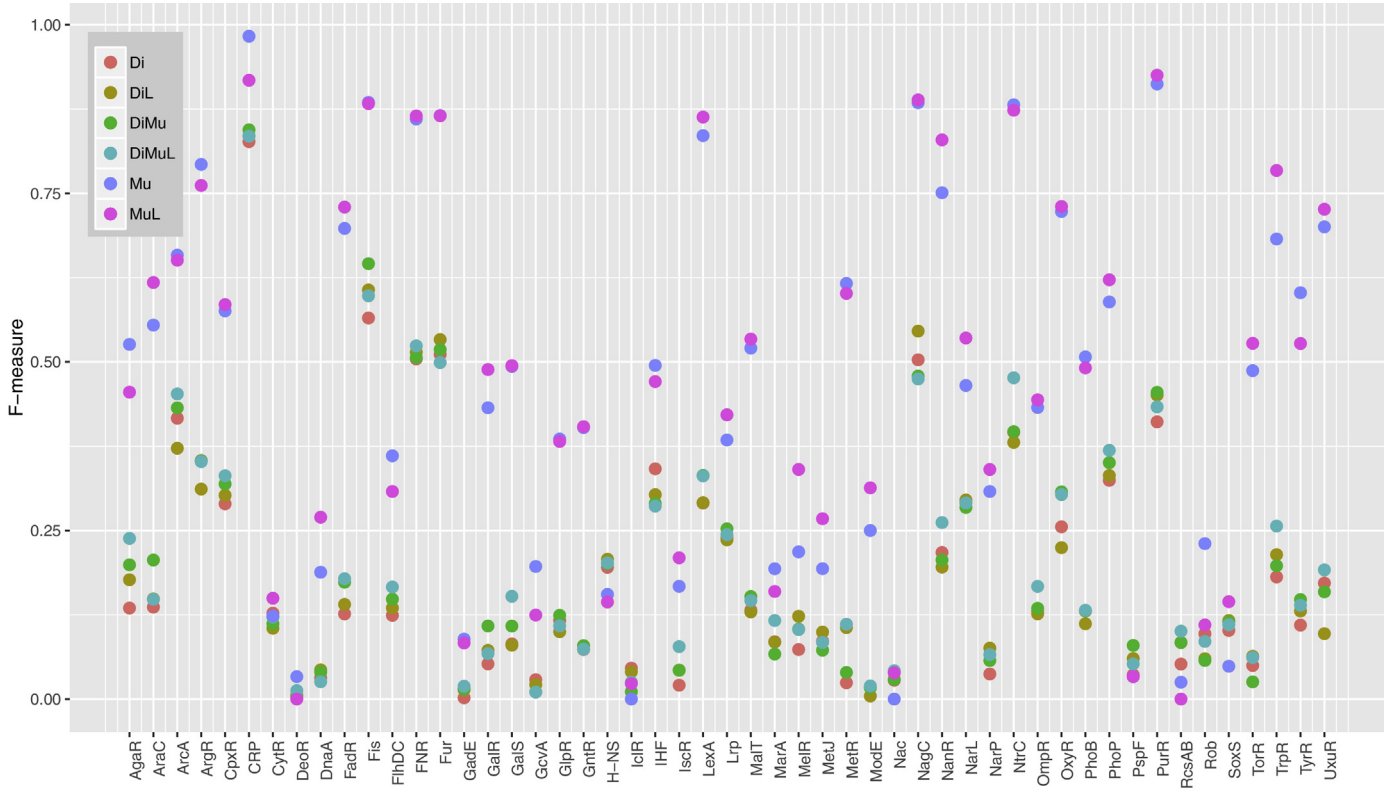

Fig. 3. Comparison of average *F*-measure over 51 TFs data sets. The performance of each combination was assessed by TPLSC. The dots with different colors denote the different combinations of DNA properties. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

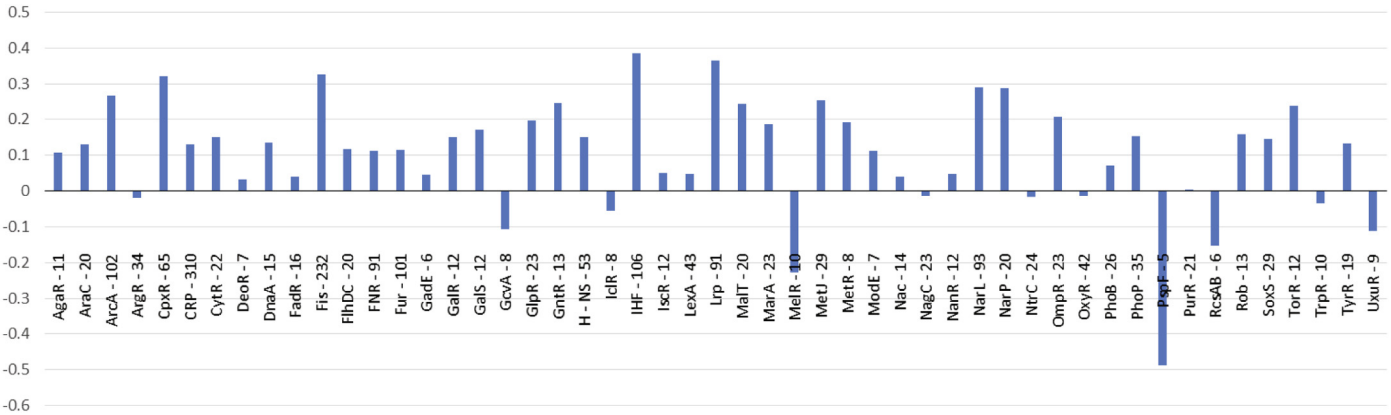

Fig. 4. The improved magnitude from our method compared with the best results reported by existing methods. The bar indicates the magnitude of improvement, the positive bar illustrates the improvements, while the negative bar shows the deteriorations. The numbers after TF names connected by symbol '-' indicate the number of binding sites (or sample sizes).

DeoR, Nac, and SoxS. Furthermore, Fis, IHF, Lrp, and CpxR can achieve more than 30% improvements compared with other methods, while the recognition rate of CRP can achieve as high as 98.29%. The last row of Table 2 reports the average *F*-measure over 51 TFs. TPLSC and TPLSC-Letter achieve 0.4587 and 0.4696, respectively. The second best algorithm is SVML_FS (average value 0.3543).

Fig. 4 illustrates the difference between the results obtained by TPLSC and the best results achieved by existing methods. As shown in Fig. 4, the positive sample size can dramatically affect the performance of all methods. With large sample sizes of TFs, such as Fis (232), IHF (106), Lrp (91), TPLSC can significantly improve the performance. It should be noted that the methods may lead to unrealistic results when the positive sample size of TFs is less than 10. Particularly, for PspF TF, SVML can achieve 0.5222 while SVMR_FS and RF_FS can not identify it completely, and our method can only achieve 0.0333. The performance of TPLSC on other small sample size data sets is also not very well, such as TF GcvA (8), MelR (10), RcsAB (6), PspF (5), and UxuR (9).

In terms of computational complexity, RF_FS method depends on the number of trees, while SVM-based methods are very time consuming, especially for searching the optimal parameters for RBF kernel (C and γ). The average running time of SVM-based motif models [19] is roughly one day over 5 independent runs. However, the running time of our method is no more than 9 minutes over 10 independent runs.

5. Discussion and conclusion

In this paper, we have proposed a promising way to represent the DNA sequence for TFBSs, and developed a tensor-based PLS classifier for the identification of TFBSs. The experimental results demonstrate that our approach can significantly improve the identification rate compared with existing methods. The results also indicate that the performance of classifiers with dinucleotide properties (Di or DiL) is inferior to those with multiple nucleotide properties (Mu or MuL). However, incorporating the feature selection step into training model procedure does not improve the performance of the SVM-based motif models, while incorporating the letter features into the physicochemical properties can slightly improve the performance. For example, TPLSC with the letter features achieves 0.4696 while without the letter features, it can achieve 0.4587. The increased power obtained by our approach is due to two key benefits, which are summarized as follows:

- Tensor-based representation has the ability to capture more structural information of DNA sequences than vector representation. Mapping a given DNA sequence to a one-dimensional feature vector is likely to lose some structural information of DNA sequences. This issue is similar to vector representation in the image processing field, in which reshaping image data into vectors commonly loses the neighborhood characteristics of the image. The tensor representation used in this study may capture the potential interaction among physicochemical properties of DNA sequences.
- With high-dimensional, small sample size data sets, tensor-based representation also alleviates the limitations of the recognition method, such as the risk of over-fitting in the training model procedure. Vectorial representation can significantly increase the dimensionality of training data sets. For example, the physicochemical properties are recorded as 56 properties for each 3-mer nucleotides and the length of the sequence is 39. Thus, we can obtain a 2184 (56×39) dimensions vector each binding site. However, in our paper, we only obtain a $1 \times 56 \times 39$ tensor.

With small sample size data sets (less than 10), the methods may lead to unrealistic results. Fortunately, with the rapid development of sequencing techniques, the cost of sequencing is decreasing. The database for TFBSs is updating and more and more positive samples will be available for analysis. In future studies, we plan to extend our method to other TFBS recognition for further insights into cellular behavior and the complex gene regulatory networks in cells [52,53]. How to represent the raw data as tensor data is an open problem. In the current study, we only concatenate different mers with a two-order tensor to recognize binding sites. A very promising direction is to model our experimental data as three-order tensor data in the molecular dynamics phase.

Acknowledgments

This work was jointly supported by the National Natural Science Foundation of China (grant numbers 61173111 and 60774086) and the Ph.D. Programs Foundation of Ministry of Education of China (grant number 20090201110027). We thank Dr. Kirsten Herold at the University of Michigan, Ann Arbor, who provided very helpful comments about the writing.

References

[1] Latchman DS. Transcription factors: an overview. Int J Biochem Cell Biol 1997;29(12):1305–12.
[2] Sun S, Hood M, Scott L, Peng Q, Mukherjee S, Tung J, et al. Differential expression analysis for RNAseq using Poisson mixed models; 2016, bioRxiv:073403.
[3] Gusmao EG, Dieterich C, Zenke M, Costa IG. Detection of active transcription factor binding sites with the combination of DNase hypersensitivity and histone modifications. Bioinformatics 2014;30(22):3143–51.
[4] Zhou X, Cain CE, Myrthil M, Lewellen N, Michelini K, Davenport ER, et al. Epigenetic modifications are associated with inter-species gene expression variation in primates. Genome Biol 2014;15(12):1.
[5] Galas DJ, Schmitz A. DNase footprinting – simple method for detection of protein–DNA binding specificity. Nucleic Acids Res 1978;5(9):3157–70.
[6] Hampshire AJ, Rusling DA, Broughton-Head VJ, Fox KR. Footprinting: a method for determining the sequence selectivity affinity and kinetics of DNA-binding ligands. Methods 2007;42(2):128–40.
[7] Fried MG. Measurement of protein–DNA interaction parameters by electrophoresis mobility shift assay. Electrophoresis 1989;10(5–6):366–76.
[8] Smith AJP, Humphries SE. Characterization of DNA-binding proteins using multiplexed competitor EMSA. J Mol Biol 2009;385(3):714–7.
[9] Berezikov E, Guryev V, Cuppen E. CONREAL web server: identification and visualization of conserved transcription factor binding sites. Nucleic Acids Res 2005;33:W447–50.
[10] Stormo GD. DNA binding sites: representation and discovery. Bioinformatics 2000;16(1):16–23.
[11] Kel AE, Gossling E, Reuter I, Cheremushkin E, Kel-Margoulis OV, Wingender E. MATCH(tm): a tool for searching transcription factor binding sites in DNA sequences. Nucleic Acids Res 2003;31(13):3576–9.
[12] Osada R, Zaslavsky E, Singh M. Comparative analysis of methods for representing and searching for transcription factor binding sites. Bioinformatics 2004;20(18):3516–25.
[13] Yang L, Zhou T, Dror I, Mathelier A, Wasserman WW, Gordân R, et al. TFBSshape: a motif database for DNA shape features of transcription factor binding sites. Nucleic Acids Res 2014;42(D1):D148–55.
[14] Bauer AL, Hlavacek WS, Unkefer PJ, Mu FP. Using sequence-specific chemical and structural properties of DNA to predict transcription factor binding sites. PLoS Comput Biol 2010;6(11).
[15] Meysman P, Thanh HD, Laukens K, De Smet R, Wu Y, Marchal K, et al. Use of structural DNA properties for the prediction of transcription-factor binding sites in *Escherichia coli*. Nucleic Acids Res 2011;39(2).
[16] Zhou T, Yang L, Lu Y, Dror I, Dantas Machado AC, Ghane T, et al. DNAshape: a method for the high-throughput prediction of DNA structural features on a genomic scale. Nucleic Acids Res 2013;41(W1):W56–62.
[17] Chiu T-P, Yang L, Zhou T, Main BJ, Parker SCJ, Nuzhdin SV, et al. GBshape: a genome browser database for DNA shape annotations. Nucleic Acids Res 2015;43(D1):D103–9.
[18] Yang J, Ramsey SA. A DNA shape-based regulatory score improves position-weight matrix-based recognition of transcription factor binding sites. Bioinformatics 2015;31(21):3445–50.
[19] Maienschein-Cline M, Dinner AR, Hlavacek WS, Mu F. Improved predictions of transcription factor binding sites using physicochemical features of DNA. Nucleic Acids Res 2012;40(22):e175.
[20] Zhou T, Shen N, Yang L, Abe N, Horton J, Mann RS, et al. Quantitative modeling of transcription factor binding specificities using DNA shape. Proc Natl Acad Sci U S A 2015;112(15):4654–9.
[21] Smitha CS, Saritha R. Computational transcription factor binding prediction using random forests. In: Glan Devadhas G, editor. 2014 international conference on control, instrumentation, communication and computational technologies (ICCICCT). Kanyakumari, India: IEEE; 2014. p. 577–83.
[22] Hooghe B, Broos S, van Roy F, De Bleser P. A flexible integrative approach based on random forest improves prediction of transcription factor binding sites. Nucleic Acids Research; 2012.
[23] Weirauch Babak Alipanahi MT, Andrew D, Frey BJ. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. Nat Biotechnol 2015;33(8):831–8.
[24] Mažgut J, Tiňo P, Bodén M, Yan H. Dimensionality reduction and topographic mapping of binary tensors. Pattern Anal Appl 2014;17(3):497–515.
[25] Haiping L, Plataniotis KN, Anastasios V. Multilinear subspace learning: dimensionality reduction of multidimensional data. London, New York: CRC Press; 2013.
[26] Lu H, Eng H-L, Guan C, Plataniotis KN, Venetsanopoulos AN. Regularized common spatial pattern with aggregation for EEG classification in small-sample setting. IEEE Trans Biomed Eng 2010;57(12):2936–46.
[27] Li J, Zhang L, Tao D, Sun H, Zhao Q. A prior neurophysiologic knowledge free tensor-based scheme for single trial EEG classification. IEEE Trans Neural Syst Rehabil Eng 2009;17(2):107–15.
[28] Chen Z-Y, Fan Z-P, Sun M. A SVM ensemble learning method using tensor data: an application to cross selling recommendation. In: Chen J, editor. 2015 12th international conference on service systems and service management (ICSSSM). Guangzhou, China: IEEE; 2015. p. 1–4.
[29] Yan S, Xu D, Yang Q, Zhang L, Tang X, Zhang H-J. Multilinear discriminant analysis for face recognition. IEEE Trans Image Process 2007;16(1):212–20.
[30] Wang J, Barreto A, Wang L, Chen Y, Rishe N, Andrian J, et al. Multilinear principal component analysis for face recognition with fewer features. Neurocomputing 2010;73(10):1550–5.
[31] Itoh H, Imiya A, Sakai T. Dimension reduction and construction of feature space for image pattern recognition. J Math Imaging Vis 2016:1–31.
[32] Sun J, Tao D, Papadimitriou S, Yu PS, Faloutsos C. Incremental tensor analysis: theory and applications. ACM Trans Knowl Discov Data (TKDD) 2008;2(3):11.
[33] Panagakis Y, Kotropoulos C, Arce GR. Non-negative multilinear principal component analysis of auditory temporal modulations for music genre

classification. IEEE Trans Audio Speech Lang Process 2010;18(3):576–88.
[34] Fanaee-T H, Gama J. Tensor-based anomaly detection: an interdisciplinary survey. Knowl Based Syst 2016;98:130–47.
[35] Steffen NR, Murphy SD, Tolleri L, Hatfield GW, Lathrop RH. DNA sequence and structure: direct and indirect recognition in protein–DNA binding. Bioinformatics 2002;18(1):22–30.
[36] Salgado H, Peralta-Gil M, Gama-Castro S, Santos-Zavaleta. RegulonDB v8.0: omics data sets, evolutionary conservation, regulatory phrases, cross-validated gold standards and more. Nucleic Acids Res 2013;41(D1):D203–13.
[37] Friedel M, Nikolajewa S, Suhnel J, Wilhelm T. DiProDB: a database for dinucleotide properties. Nucleic Acids Res 2009;37:D37–40.
[38] Wold S, Ruhe A, Wold H, Dunn WJ. The collinearity problem in linear-regression – the partial least-squares (PLS) approach to generalized inverses. SIAM J Sci Stat Comput 1984;5(3):735–43.
[39] Burguillo FJ, Corchete LA, Martin J, Barrera I, Bardsley WG. A partial least squares algorithm for microarray data analysis using the VIP statistic for gene selection and binary classification. Curr Bioinform 2014;9(3):348–59.
[40] Sun SQ, Peng QK, Shakoor A. A kernel-based multivariate feature selection method for microarray data classification. PLOS ONE 2014;9(7).
[41] Rahman A, Kondo N, Ogawa Y, Suzuki T, Kanamori K. Determination of *k* value for fish flesh with ultraviolet–visible spectroscopy and interval partial least squares (IPLS) regression method. Biosyst Eng 2016;141:12–8.
[42] Andersson M. A comparison of nine PLS1 algorithms. J Chemom 2009;23(9–10):518–29.
[43] Gottfries J, Blennow K, Wallin A, Gottfries CG. Diagnosis of dementias using partial least squares discriminant analysis. Dementia 1995;6(2):83–8.
[44] Brereton RG, Lloyd GR. Partial least squares discriminant analysis: taking the magic away. J Chemom 2014;28(4):213–25.
[45] Goodarzi M, Freitas MP. On the use of PLS and N-PLS in MIA-QSAR: azole antifungals. Chemom Intell Lab Syst 2009;96(1):59–62.
[46] Eliseyev A, Aksenova T. Recursive N-way partial least squares for brain–computer interface. PLOS ONE 2013;8(7):e69962, 07.
[47] Eliseyev A, Moro C, Faber J, Wyss A, Torres N, Mestais C, et al. L1-penalized N-way PLS for subset of electrodes selection in BCI experiments. J Neural Eng 2012;9(4):045010.
[48] Andersson CA, Bro R. The N-way toolbox for MATLAB. Chemom Intell Lab Syst 2000;52(1):1–4.
[49] Ouertani SS, Mazerolles Gérard, Boccard J, Rudaz S, Hanafi M. Multi-way PLS for discrimination: compact form equivalent to the tri-linear PLS2 procedure and its monotony convergence. Chemom Intell Lab Syst 2014;133:25–32.
[50] Zhao QB, Caiafa CF, Mandic DP, Chao ZC, Nagasaka Y, Fujii N, et al. Higher order partial least squares (HOPLS): a generalized multilinear regression method. IEEE Trans Pattern Anal Mach Intell 2013;35(7):1660–73.
[51] Cong F, Lin Q-H, Kuang L-D, Gong X-F, Astikainen P, Ristaniemi T. Tensor decomposition of EEG signals: a brief review. J Neurosci Methods 2015;248:59–69.
[52] Sun S, Peng Q, Zhang X. Global feature selection from microarray data using Lagrange multipliers. Knowl Based Syst 2016;110:267–74.
[53] Hore V, Viñuela A, Buil A, Knight J, McCarthy MI, Small K, et al. Tensor decomposition for multiple-tissue gene expression experiments. Nat Genet 2016;48(9):1094–100.