Machine Learning Study of DNA Binding by Transcription Factors from the LacI Family

Gennady G. Fedonin and Mikhail S. Gelfand

Institute for Information Transmission Problems (the Kharkevich Institute), RAS
Bolshoy Karetny per. 19, Moscow, 127994, Russia
gennady.fedonin@gmail.com,
gelfand@iitp.ru
http://www.rtcb.iitp.ru

Abstract. We studied 1372 LacI-family transcription factors and their 4484 DNA binding sites using machine learning algorithms and feature selection techniques. The Naive Bayes classifier and Logistic Regression were used to predict binding sites given transcription factor sequences. Prediction accuracy was estimated using 10-fold cross-validation. Experiments showed that the best prediction of nucleotide densities at selected site positions is obtained using only a few key protein sequence positions. These positions are stably selected by the forward feature selection based on the mutual information of factor-site position pairs.

Keywords: transcription factors, naive Bayes classifier, logistic regression, mutual information.

1 Introduction

Many biological processes involve specific interaction between DNA-binding proteins and DNA sites. The mechanisms of the sequence- and structure-specific recognition remain elusive, despite some advance coming from experimental mutagenesis studies and computational analysis of known X-ray structures of protein-DNA complexes [1], [2]. One of the reasons for that may be lack of data. Indeed, while many complexes are structurally resolved, one of the main results of the analysis has been the absence of a universal protein-DNA recognition code [3], [4], [5]. On the other hand, experimental analysis has been limited to a small number of proteins, and again, the obtained results do not seem universal [6].

A different approach is to study the protein-DNA code within large families of DNA-binding proteins [7], e.g. C2H2 zinc finger, homeodomain and bHLH domains [8] or TAL receptors [9]. At that, the data may come not only from experiment, but from comparative genomic analysis of regulatory interactions. A rich source of such data are bacterial transcription factors, e.g. the LacI family considered here. Given the data on sites bound by given proteins, one may study correlations between the amino acid sequences and corresponding DNA sites, and then to use the structures, if known, as a sanity check, verifying that the observed positions indeed form contacts in the protein-DNA complexes.

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 13–24, 2010.

One observation coming from early studies [10] has been that the correlations are not limited to pairs of positions in the protein and DNA alignment: in many cases the protein preferences to a particular nucleotide at a particular site position seemed to depend on specific residues at several protein positions. This leads to the problem of selecting the optimal model complexity. Here we address this problem using the predictive power of pattern recognition algorithms as a tool to determine the optimal number of the model parameters.

2 Materials and Methods

2.1 Data

The LacI-family bacterial transcription factor and their binding sites were selected from the LACI_DB database (O. Laikova, unpublished). The DNA-binding domain (HTH_LACI) boundaries for each protein were determined using SMART_DB [11]. The obtained sequences were aligned against the standard HTH_LACI domain alignment with minimal manual editing, resulting in an alignment of 1372 protein sequences. The resulting alignment length was 87 positions. Sixteen positions with more than 30% gaps were removed. The sample of DNA sites contained 4484 sequences. The data may be downloaded from the RegPrecise database [12].

Hence, we had a sample of protein-site pairs, and the aim was to predict the probability density of nucleotides at site positions given the protein amino acid sequence (AAS). We assumed all site positions to be mutually independent given AAS, hence each position was predicted separately.

2.2 Cross-Validation

To estimate the prediction accuracy, the initial sample was randomly split into ten sets, each of which was used as a testing set with training on the remaining nine sets. Since many proteins in the sample are closely related (and have very similar AAS) it is reasonable to require the testing set not to contain AASs too similar to an AAS in the training set. To ensure this, we grouped similar AASs by similarity into clusters never separated during splitting. At that, we calculated pairwise similarity (percent of identical amino acid) for all AAS pairs. Next, we built a full graph with AASs as vertices and edges weighted with the similarity values, and removed all edges with weight less than a fixed threshold. The similarity clusters were defined as maximal connected components.

For each split into test and training sets, all algorithms were trained and their log-likelihoods on the testing set were calculated. Log-likelihood was calculated as:

$$logL = \frac{\sum_i w_i \sum_j P(n_{ij}|S_i)}{\sum_i w_i} \ ,$$

where index i runs over all AAS, index j runs over all sites of the i–th AAS, n_{ij} is the nucleotide observed at the selected position of the j–th site of the i–th sequence, w_i is the weight of the i–th AAS. The results were averaged. The procedure was repeated ten times for better averaging.

2.3 Algorithms

Weighting amino acid and binding site sequences. The similarity clusters vary in size with some sequence motifs being overpresented. To compensate for this, protein sequences were weighted, so that closely related proteins were assigned smaller weights than proteins different from all others, using the Gerstein-Sonhammer-Chotia algorithm. Each protein weight was divided equally among all its binding sites, resulting in weights of AAS-site pairs.

These weights were used to compute amino acid residue and nucleotide frequencies for building the Bayesian classifier, computation of the mutual information, and for training the logistic regression.

Naive Bayes classifier. The Bayesian classifier [13] estimates the occurrence probability for each nucleotide at each site position using the Bayes formula:

$$P(n_i|S) = \frac{P(n_i)P(S|n_i)}{\sum_j P(n_j)P(S|n_j)} \ ,$$

where n_i is the i-th nucleotide, S is the amino acid sequence, $P(n)$ is the prior probability of nucleotide n.

The naive Bayes approach assumes all positions in AAS to be mutually independent given site position nucleotide:

$$P(S|n) = \prod_i P(a_i|n) \ ,$$

where a_i is the amino acid residue at position i. Probabilities $P(a_i|n)$ are estimated using the corresponding frequencies in the sample, with phylogenetic weights and pseudocounts.

Logistic regression. The logistic regression [14] is a popular machine learning algorithm for two-class classification tasks. The training objects are assumed to be numerical feature vectors with $\{-1, 1\}$ labels. The algorithm builds a linear decision rule, weighting each numerical feature:

$$f(x_1, \ldots, x_n) = sign\big(\sum_{i=1}^{n} \alpha_i x_i\big) \ ,$$

or in the vector form:

$$f(\boldsymbol{x}) = sign(\langle \boldsymbol{\alpha}, \boldsymbol{x} \rangle) \ ,$$

where α_i is the weight of i-th feature, x_i is the value of the i-th feature.

Learning is performed by searching for weights that optimize the quality function on the training set:

$$L(\boldsymbol{\alpha}) = \sum_{i=1}^{l} w_i \ln \sigma(y_i \langle \boldsymbol{\alpha}, \boldsymbol{x}_i \rangle) - k \sum_{i=1}^{n} \alpha_i^2 \to max_{\boldsymbol{\alpha}} \ ,$$

where index i runs over all training objects, $y_i \in \{-1, 1\}$ is the class of the i-th object, w_i is the weight of the i-th object , $\sigma(z) = \frac{1}{1+\exp(-z)}$ is the logistic (sigmoid) function, $k\sum_{i=1}^{n}\alpha_i^2$ is a regularization term, k is an a priori fixed regularization parameter.

Class probabilities given feature vector can be estimated using the sigmoid function:

$$P(y) = \frac{1}{1 + \exp(-y\langle \boldsymbol{\alpha}, \boldsymbol{x}\rangle)} \ ,$$

where $y \in \{-1, 1\}$ is the class value, $\boldsymbol{x}$ is the feature vector, $\boldsymbol{\alpha}$ is the weight vector.

The logistic regression requires numeric features. In our case all features are nominal. We used the standard binarization approach: each amino acid residue a_k at i-th position was mapped to an indicator binary feature:

$$f_i(a) = \begin{cases} 1, & \text{when } a = a_k \ ; \\ 0, & \text{otherwise } . \end{cases}$$

To predict four nucleotide probabilities, an individual classifier was trained for each nucleotide. ASS-site pairs with a given nucleotide at the given site position were used as positive training examples, all other pairs, as negative ones. The positional probability of each nucleotide was calculated as:

$$P(n_i|S) = \frac{P_i(+|S)}{\sum_{j=1}^{4} P_j(+|S)} \ ,$$

where S is the AAS for which predictions are made, $P_i(+|S)$ is the positive class probability computed by i-th classifier.

The weights for the negative objects were set to the weight of the corresponding AAS, and for positive objects, the same weight, multiplied by the frequency of the given nucleotide at the given site position.

Feature selection using mutual information. The mutual information (MI, [15]) of the AAS-site position pair is the measure of correlation of these positions, allowing for a quick estimation of the predicting power of the AAS position for the nucleotide at the site position. Calculating the MI is fast, making it convenient for the feature selection.

To offset for unreliable estimations of the frequencies of rare residues and nucleotides (at a given position), we used pseudocounts, adding small values for rare events.

The effective frequency of residue a at position i was defined as:

$$f_i(a) = \frac{N_i(a) + k\frac{\sum_b N_i(b)P(b\to a)}{\sqrt{N}}}{N + k\sqrt{N}} \ ,$$

where $N_i(a)$ is the total weight of AASs with a in position i, N is the total weight of all AASs in the sample. The transition probabilities $P(b \to a)$ were obtained from BLOSUM60 [16].

The effective frequency of nucleotide n at position j was:

$$f_j(n) = \frac{N_j(n) + k\frac{\sum_m N_j(m)P(m\to n)}{\sqrt{N}}}{N + k\sqrt{N}} = \frac{N_j(n) + 0.25k\frac{\sum_m N_j(m)}{\sqrt{N}}}{N + k\sqrt{N}} \ ,$$

where $N_j(n)$ is the total weight of sites with n at position j, N is the total weight of the sample sites.

The observed effective frequency of 'amino acid - nucleotide' pair:

$$f^{\mathrm{o}}_{ij}(a,n) = \frac{N_{ij}(a,n) + k\sqrt{N}f^{\mathrm{e}}_{ij}(a,n)}{N + k\sqrt{N}} \ ,$$

where $N_{ij}(a,n)$ is the total weight of pairs with a at position i of the AAS and n at the site position j, N is the total weight of sample pairs, $f^{\mathrm{e}}_{ij}(a,n)$ is the expected effective frequency of pair (a,n) defined as

$$f^{\mathrm{e}}_{ij}(a,n) = f_i(a)f_j(n) \ ,$$

where $f_i(a)$ and $f_j(n)$ are the effective frequencies of residue a at position i and nucleotide n at position j, respectively.

The mutual information was computed as

$$I_{ij} = \sum_a \sum_n f^{\mathrm{o}}_{ij}(a,n) \log \frac{f^{\mathrm{o}}_{ij}(a,n)}{f^{\mathrm{e}}_{ij}(a,n)} \ .$$

Greedy forward feature selection. Another strategy for feature selection is searching through subsets of features, training algorithms using feature subsets on parts of the training set, estimating error on remaining objects and selecting the subset with the minimal error.

In practice, the exhaustive search is computationally intractable, so we used the greedy algorithm, successively adding each of the remaining features to the current best subset and selecting the feature which provides the best classifier. This feature then is added to the best-feature subset and the process is repeated.

The greedy strategy takes into account feature dependency, but still can lead to suboptimal subsets. On the other hand, this strategy is the fastest after the MI-based feature selection.

3 Results and Discussion

We report only the performance of two simple algorithms: Naive Bayes classifier (NB) [13] based on amino acid frequencies estimation and logistic regression (LR) [14] with simple AAS encoding to feature vectors. We also tried using amino acid pairs' frequencies (and corresponding binarisation) with these algorithms, but the prediction quality was the same. The reason of this might be the data sparseness, which makes it impossible to estimate frequencies of complex

events robustly. We also tried linear SVMs with these feature vectors, but the performance was poor. SVM with a linear kernel is the fastest in training SVM algorithm, but it is very slow compared with NB and LR. Using SVMs based on nonlinear kernels for feature selection required computational resources not available for this study.

3.1 Selecting Site Alignment Positions

Different site positions can be predicted with different accuracy. In this study we used those site positions, for which significantly correlated AAS positions were found [10]. We used the mutual information to measure correlation. As one can see in the heat map in Fig. 1, significant correlations are observed for positions 5, 6, 7, 9 and the symmetric ones. Below we consider only these four positions.

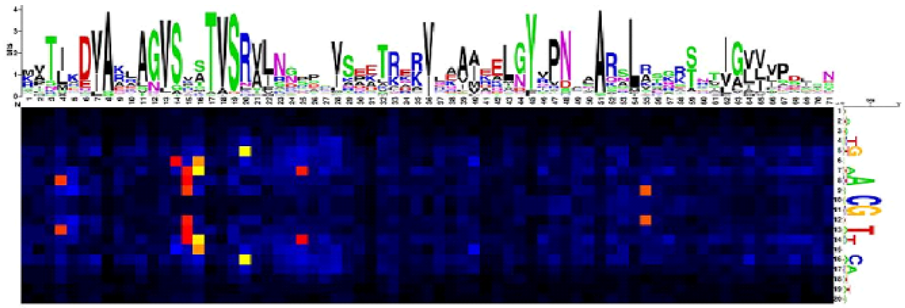

Fig. 1. Mutual information of AAS-site position pairs [10]. Light colors correspond to significant correlations.

3.2 Selection of Significant Positions

Selection was performed using two methods. Using the MI-based selection, twenty positions were selected for each of three site positions. Positions were selected successively, starting from the most informative one. On each iteration, the classifiers were trained using the current position set and the prediction quality (testing set log-likelihood) was estimated. The greedy selection was organized in the same way, but only for ten AAS positions for each site position. In both cases the process was repeated for different sample splits during 10-fold cross validation (2.2).

The prediction quality values for different feature set lengths were plotted on a graph. The selected positions were tabulated. The selected positions may vary for different sample splits. Hence we can only report the frequencies of given positions in position sets selected at algorithm iterations, i.e. the frequencies in the selected sets of sizes ranging from 1 to 20. To visualize the tables, we ordered all positions by the total frequency (the sum of frequencies in sets of all lengths) and report the top ones.

Only few positions are stably selected by both algorithms, i.e. these positions are selected with almost any sample split. The maxima of the test set log-likelihood plots often correspond to these position sets. Further increase of the position set size leads to overfitting. The selection stability and existence of well-defined maxima on the log-likelihood plots can be treated as a proof of connection between the selected AAS positions and the site positions.

While the prediction quality shows large variation, dependent on the split of the data into training and test sets, the overall results from different runs (position of the local maxima, selected positions, relative quality of predictions by different algorithms) are consistent.

3.3 AAS-Position Selection for Position 9 of the Site Alignment

The log-likelihood values obtained on the testing set for position 9 by various algorithms and selection strategies are plotted in Fig. 2. Well-defined maxima are obtained on three positions by all methods.

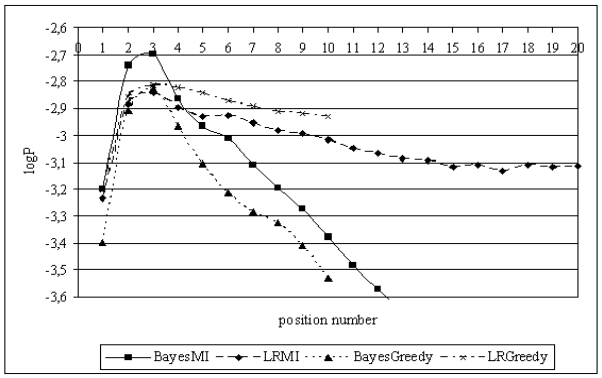

Fig. 2. The log-likelihood values against the number of selected positions for position 9 of the site alignment

Table 1 features the most frequent positions. The column numbers are the position numbers starting from the most frequent one. The row numbers are the selected set sizes. The MI-based search and greedy naive Bayes search stably select three positions 55, 15 and 5. The greedy logistic regression stably selects the same three positions, and frequently position 27.

The maximum prediction quality is achieved by using three positions. Therefore, positions 55, 15 and 5 of the amino acid alignment are significantly linked to position 9 of the site alignment.

3.4 AAS-Position Selection for Position 7 of the Site Alignment

The log-likelihood values obtained on the testing set for position 7 by various algorithms and selection strategies are plotted in Fig. 3. Well-defined maxima are

Table 1. Frequencies of six most frequent positions in MI-selected, greedy naive Bayes classifier (NB) and greedy logistic regression (LR) sets of varying lengths for prediction of site position 9 (in %)

	MI-selected						NB						LR					
Set size	55	15	5	68	56	16	55	15	5	1	70	26	55	15	5	27	49	56
1	100	0	0	0	0	0	91	9	0	0	0	0	96	4	0	0	0	0
2	100	100	0	0	0	0	100	100	0	0	0	0	100	100	0	0	0	0
3	100	100	90	0	0	0	100	100	96	0	0	0	100	100	90	9	0	0
4	100	100	90	20	35	39	100	100	99	36	5	6	100	100	96	82	5	4
5	100	100	95	50	64	57	100	100	99	52	23	23	100	100	98	94	38	37
6	100	100	97	79	80	80	100	100	99	68	42	40	100	100	99	96	64	54

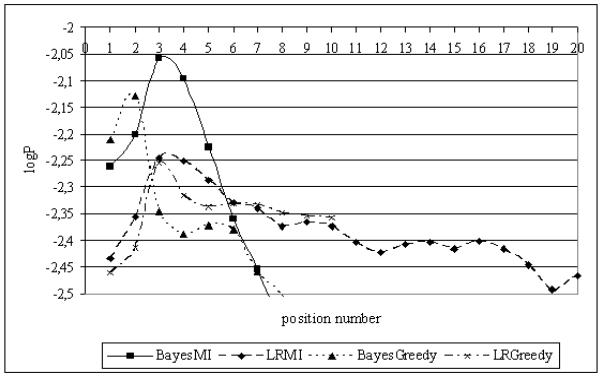

Fig. 3. The log-likelihood values against the number of selected positions for position 7 of the site alignment

obtained on three positions by all methods, except the greedy Bayes classifier, which has maximum on two positions.

The most frequent positions are listed in Tab. 2, with the notation as above. The MI-based search stably selects three positions 16, 25 and 15, and sometimes position 68. The greedy logistic regression stably selects the same three positions, whereas the greedy Bayes classifier based search makes a mistake on the third step, stably selecting position 49, which, as seen on the log-likelihood plot, leads to a dramatic decrease of the prediction quality.

The maximum prediction quality is achieved by using three positions. Therefore, positions 16, 25 and 15 of the amino acid alignment are significantly linked to position 7 of the site alignment.

3.5 AAS-Position Selection for Position 6 of the Site Alignment

The log-likelihood values are plotted in Fig. 4. The naive Bayes classifier with the MI-based selection has two maxima at one and three positions, while the greedy strategy has maxima at one and seven positions. The logistic regression

Table 2. Frequencies of six most frequent positions in MI-selected, greedy naive Bayes classifier (NB) and greedy logistic regression (LR) sets of varying lengths for prediction of site position 7 (in %)

	MI-selected						NB						LR					
Set size	16	25	15	68	5	46	16	15	49	68	50	19	16	15	25	49	68	50
1	100	0	0	0	0	0	100	0	0	0	0	0	100	0	0	0	0	0
2	100	96	4	0	0	0	100	97	0	2	0	0	100	69	30	0	3	0
3	100	100	100	0	0	0	100	97	71	11	9	2	100	99	99	0	0	0
4	100	100	100	84	5	3	100	98	89	59	33	7	100	100	100	56	20	5
5	100	100	100	94	25	18	100	98	92	94	75	12	100	100	100	78	57	16
6	100	100	100	97	38	46	100	99	93	100	86	64	100	100	100	89	84	42

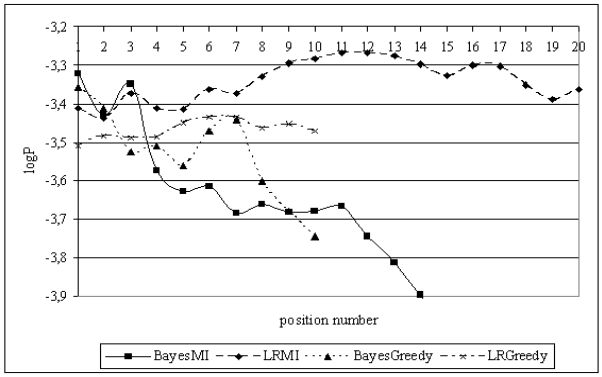

Fig. 4. The log-likelihood values against the number of selected positions for position 6 of the site alignment

curves slowly grow, having many local maxima with highest values around six and eleven positions for the greedy and MI-based search, respectively.

Table 3 features the most frequent positions. The MI-based selection has one absolutely stable position, 16, and two additional stable positions, 25 and 15, which are interchangeable at the second selection step. The greedy strategies select two positions, absolutely stable 16 and strongly stable 15. Further selection is unstable.

In prediction of position 6 in binding sites, different algorithms behave differently: the naive Bayes classifier has two maxima, while the logistic regression seems to overfit. However, all methods stably select position 16 of the AAS alignment that is significantly connected with position 6 in the site alignment.

3.6 AAS-Position Selection for Position 5 of the Site Alignment

The log-likelihood values obtained on the testing set for position 5 by different algorithms and selection strategies are plotted in Fig. 5. For the naive Bayes classifier, both MI-based and greedy, the maximum is reached when only one

Table 3. Frequencies of five most frequent positions in MI-selected, greedy naive Bayes classifier and greedy logistic regression sets of varying lengths for prediction of site position 6 (in %)

	MI-selected					NB					LR				
Set size	16	25	15	68	26	16	15	20	27	49	16	15	27	25	49
1	100	0	0	0	0	100	0	0	0	0	100	0	0	0	0
2	100	60	40	0	0	100	90	0	10	0	100	85	5	8	0
3	100	96	91	0	8	100	94	61	28	6	100	93	65	19	4
4	100	98	95	45	29	100	94	82	64	21	100	95	78	35	22
5	100	100	97	66	58	100	97	89	82	68	100	98	86	55	47

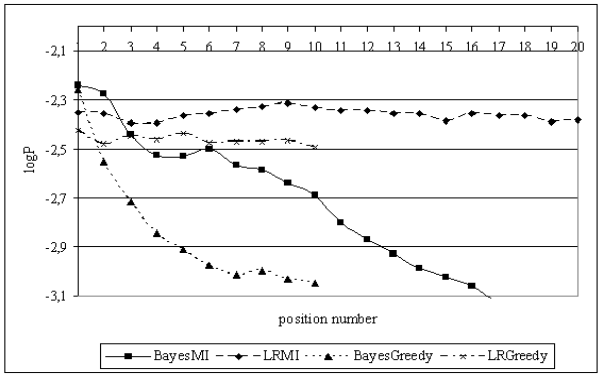

Fig. 5. The log-likelihood values against the number of selected positions for position 5 of the site alignment

Table 4. Frequencies of five most frequent positions in MI-selected, greedy naive Bayes classifier and greedy logistic regression sets of varying lengths for prediction of site position 5 (in %)

	MI-selected					NB					LR				
Set size	20	25	27	68	16	20	27	15	69	50	20	25	16	50	27
1	100	0	0	0	0	100	0	0	0	0	100	0	0	0	0
2	100	95	3	2	0	100	55	2	20	0	100	54	33	0	13
3	100	96	35	41	21	100	61	28	58	18	100	87	69	16	25
4	100	99	62	62	53	100	62	48	60	28	100	94	73	60	44
5	100	99	85	83	77	100	64	67	61	49	100	100	75	85	62

position is used for prediction. The logistic regression algorithm plots do not have a marked maximum.

The most frequent positions are tabulated in Tab. 4. Position 20 is absolutely stable, position 25 is stable for the MI-based search. Further selection is unstable.

The maximum prediction quality is achieved by using only one position and addition of the second position considerably decreases it. Therefore, only position 20 of the amino acid alignment is significantly connected with position 5 of the site alignment.

4 Conclusions

Experiments showed that knowledge of only a few key protein sequence positions is sufficient for prediction of nucleotide densities at selected site positions. These positions form significantly correlated pairs with corresponding site alignment positions, having high mutual information values. Moreover, the selected pairs of positions are largely the same for different methods (for any given site position) and correspond to the contacts in protein-DNA complexes [10]. On the other hand, the results show that the dependencies are not limited to simple pairs of contacting positions. Overall, these observations support the existence of protein family-specific protein-DNA recognition code. Analysis of other transcription factor families will show what features of this code are universal.

Acknowledgements

We are grateful to Olga Laikova and Dr. Alexandra B. Rakhmaninova who provided the data, and to Yuri Korostelev for sharing preliminary results and providing Fig. 1. This study was partially supported by RAS (program "Molecular and Cellular Biology"), RFBR (grains 09-04-92745 and 10-04-00431), and state contract 2.740.11.0101.

References

1. Luscombe, N.M., Laskowski, R.A., Thornton, J.M.: Amino acid-base interactions: a three-dimensional analysis of protein-DNA interactions at an atomic level. Nucleic Acids Research 29, 2860–2874 (2001)
2. Baker, C.M., Grant, G.H.: Role of aromatic amino acids in protein-nucleic acid recognition. Biopolymers 85, 456–470 (2007)
3. Suzuki, M., Brenner, S.E., Gerstein, M., Yagi, N.: DNA recognition code of transcription factors. Protein Engineering 8, 319–328 (1995)
4. Benos, P.V., Lapedes, A.S., Stormo, G.D.: Is there a code for protein-DNA recognition? Probab(ilistical)ly. Bioessays 24, 466–475 (2002)
5. Luscombe, N.M., Thornton, J.M.: Protein-DNA interactions: amino acid conservation and the effects of mutations on binding specificity. Journal of Molecular Biology 320, 991–1009 (2002)
6. Luscombe, N.M., Austin, S.E., Berman, H.M., Thornton, J.M.: An overview of the structures of protein-DNA complexes. Genome Biology 1, REVIEWS001 (2000)
7. Sandelin, A., Wasserman, W.W.: Constrained binding site diversity within families of transcription factors enhances pattern discovery bioinformatics. Journal of Molecular Biology 338, 207–215 (2004)

8. Mahony, S., Auron, P.E., Benos, P.V.: Inferring protein-DNA dependencies using motif alignments and mutual information. Bioinformatics 23, i297–i304 (2007)
9. Moscou, M.J., Bogdanove, A.J.: A simple cipher governs DNA recognition by TAL receptors. Science 326, 1501
10. Korostelev, Y., Laikova, O.N., Rakhmaninova, A.B., Gelfand, M.S.: Correlations between amino acid sequences of transcription factors and their DNA binding sites. In: Abstr. First RECOMB Satellite Conference on Bioinformatics Education, San Diego, USA (2009)
11. Kalinina, O.V., Novichkov, P.S., Mironov, A.A., Gelfand, M.S., Rakhmaninova, A.B.: SDPpred: a tool for prediction of amino acid residues that determine differences in functional specificity of homologous proteins. Nucleic Acids Research 32, W424–W428 (2004)
12. Novichkov, P.S., Laikova, O.N., Novichkova, E.S., Gelfand, M.S., Arkin, A.P., Dubchak, I., Rodionov, D.A.: Nucleic Acids Research 38, D111–D118 (2010)
13. Domingos, P., Pazzani, M.: On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning 29, 103–137 (1997)
14. Hosmer, D., Lemeshow, S.: Applied Logistic Regression, 2nd edn. Wiley, Chichester (2000)
15. Peng, H.C., Long, F., Ding, C.: Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence 27, 1226–1238 (2005)
16. Henikoff, S., Henikoff, J.G.: Amino Acid Substitution Matrices from Protein Blocks. Proc. Natl. Acad. Sci. USA 89, 10915–10919 (1992)