Efficient sequence kernel-based genome-wide prediction of transcription factors

Pavel P. Kuksa
NEC Laboratories America Inc.
pkuksa@nec-labs.com

Abstract

With whole genome sequences of many organisms readily available, and lack of full functional characterization of the genes, computational functional analysis of whole genomes is a target of intensive research. Of a particular interest is prediction of regulatory functions, such as regulation of gene expression by transcription factors (TFs), proteins that bind to DNA to promote or suppress transcription of their target genes. Identification of these transcription factors at the genome level (i.e. from their sequence) lays a basis for further analysis and understanding of gene regulatory networks and can serve as a starting point for targeted high-throughput experiments. In this work, we address a question of predicting whether a (uncharacterized) protein is a transcription factor (TF) or not (non-TF) given its amino acid sequence. We cast this problem as classification task: we use sequence features as input variables and output functional class (TF or non-TF). We show that our proposed method can identify with high accuracy TFs at whole genome level both within given organism and across different organisms, as well as identify novel TF families with high accuracy.

1. Introduction

Analysis of large-scale sequential data has become an important task in machine learning and pattern recognition, inspired in part by numerous scientic and technological applications such as the document and text classication or the analysis of biological sequences (DNA, proteins) or whole genome analysis and annotation. One particular way to analyze sequences is to consider them as strings of symbols (e.g., words, amino acid residues, etc.)

With the success of many whole-genome sequencing projects such as for microbial and eukaryotic species, genome annotation (e.g., functional or structural) is a necessary next step in analysis and interpretation of these raw genome sequences. Protein-level functional genome annotation aims at inferring biological/biochemical function (e.g., regulation of transcription, transport, signal communication, etc) for all the proteins in the predicted proteome.

In this work, we focus on a protein-level genome functional annotation. In particular, we aim at predicting whether a protein is involved in a transcription factor (TF) activity given its primary (amino acid) sequence. We address this problem using a class of supervised and semi-supervised string-based kernel methods. Application of the proposed methods to transcription factor activity prediction on whole genome scale yields state-of-the-art performance. We also show that our models trained one genome generalizes to other (previously unseen) species genomes and displays high accuracy in this across-organism (*trans-species*) setting thus allowing accurate annotation of newly sequenced genomes. We also demonstrate that our method can be used to identify novel families of TFs and thus can aid in annotation of newly sequenced genomes.

2. Background

A variety of methods have been proposed for solving sequence analysis tasks, including *generative*, such as HMMs, or *discriminative* approaches. Among the discriminative approaches, *kernel-based* [14] machine learning methods provide some of the most accurate results in many sequence analysis problems [2, 8, 13, 6].

The spectrum kernel methods. The spectrum kernel methods [8, 6] rely on fixed-length representations or features $\Phi(X)$ of arbitrary long sequences X modeled as the *spectra* ($|\Sigma|^k$-dimensional histogram of counts) of short substrings (k-mers) contained in X. These features are subsequently used to define a measure of similarity, or kernel, $K(X,Y) = \Phi(X)^T\Phi(Y)$ between sequences X, Y. However, computing similarity scores, or kernels, $K(X,Y)$=$\Phi(X)^T\Phi(Y)$ using these representations can be challenging, e.g., efficient $O(k^{m+1}|\Sigma|^m(|X| + |Y|))$ trie-based mismatch kernel

algorithm [8] strongly depends on the alphabet size and the number of mismatches m. On the other hand, the gapped [6] and subsequence [10] kernels that also induce spectrum-like representations have complexity independent of $|\Sigma|$, but quadratic in the sequence length (subsequence method) or show suboptimal performance compared to other methods (e.g., mismatch).

Transcription factor identification. Identification of the repertoire of transcription factors in a given genome can serve as a starting point for high-throughput experiments aimed at characterizing regulatory networks and regulation targets. A number of previous studies has focused on identifying a general class of DNA-binding proteins (e.g., [12, 9, 5, 1, 4]), which spans a wide range of proteins including transcription factors, histones, polymerases, nucleases. Most of the methods for identifying DNA-binding proteins leverage structure-sequence information. Use of structural information limits application of the methods to a smaller subset proteins with resolved structures. In this work, we target genome-wide identification of *transcription factors* using primary sequence information only. We evaluate performance of our methods on a number of genomes, and show ability of the trained models to accurately predict transcription factor repertoire across species on new (previously unseen) genomes, as well ability to identify novel (previously unseen) families of TFs.

3. Method

In the following we describe a class of string kernel methods for TF identification applicable in both supervised and semi-supervised learning settings.

3.1. Supervised kernels for TF identification

In the supervised setting, prediction of transcription factors can be addressed using a general class of predictive models that relies on the similarity metric induced by the kernel $K(\cdot,\cdot)$ and computes the matching score between the query sequence X and the train sequences $\{X_1,\ldots,X_N\}$ whose class assignments $\{y_1,\ldots,y_n\}$ are known ($y_i \in \{-1,1\}$). The score is defined as

$$f(X) = \sum_{i=1}^{N} w_i K(X, X_i) \qquad (1)$$

with the $sign(f(X))$ indicating whether the query sequence X is a transcription factor, $f(X) > 0$, or not. The weights w_i are set by training algorithm prior to making decisions, e.g., by SVM [14].

As one of the defining characteristics of transcription factors is the presence of DNA-binding domains, which have unique structures allowing to interact with and bind DNA, one can exploit a class of kernels that have been shown to perform well on the protein structural classification tasks, e.g., remote homology prediction withing SCOP hierarchy [6, 11]. A large class of sequence kernels $K(\cdot,\cdot)$ (including spectrum/mismatch [7, 8], gapped [6], spatial kernels [3], etc.) can be written in the following general form

$$K(X,Y) = \sum_{\alpha\in X}\sum_{\beta\in Y} I(\alpha,\beta) \qquad (2)$$

where $I(\cdot,\cdot)$ is a matching function (e.g., identity) for two features α and β. Kernels in this form are essentially equivalent to cumulative comparison of all pairs of features ($\alpha \in X$, $\beta \in Y$).

Features α can be defined as k-mers (i.e. short substrings of length k) as in [6, 8, 7], or as a ordered sample of short substrings in particular spatial (positional) arrangement (i.e. α is in the form of $a_1 \overset{d_1}{\leftrightarrow} a_2 \overset{d_2}{\leftrightarrow}, \cdots, \overset{d_{t-1}}{\longleftrightarrow} a_t$) as in [3], or as short subsequences (gapped instances) as in [6].

3.1.1 Kernel computation

Kernels in the form as in Eq. 2 can be computed in two steps

1. Feature extraction (e.g., extract k-mer features, spatial samples, etc)
2. Feature counting and kernel computation (Algorithm 1)

While the step 1 is feature-type specific, the second step can be generalized and is essentially feature-type independent. The kernel computation step after feature extraction can be reduced to exact k-mer spectrum kernel computation using sorting (this is summarized in Algorithm 1). This allows to compute kernel value in *linear* time, and apply kernel computations on large scale (e.g., genome-scale analysis) as we will show in the experiments.

3.2. Semi-Supervised kernels for TF identification

While available labeled training data is often scarce, using readily available large unlabeled data sets can significantly improve prediction accuracy (cf. [15, 11]). One approach to combining unlabeled and labeled data is a sequence neighborhood approach [15]. For each sequence X its *neighborhood* $N(X)$ consists of a set of *similar* sequences obtained by querying unlabeled data set (e.g. using PSI-BLAST). Then the kernel between to sequences X and Y is defined (similarly to Eq. 2) as

Table 1. Genome-wide transcription factor prediction (C. Reinhardtii, 15497 seqs)

Method	Balanced Error, %	ROC50
Supervised		
Baseline 1: Sequence composition	42.88	1.06
Baseline 2: PSI-BLAST	45.15	0.61
Mismatch-(k=5,m=1)	24.03	26.04
SK	**25.28**	**26.89**
Semi-supervised		
Baseline 1: Sequence composition (Plant DB)	36.22	5.64
Baseline 2: Sequence composition (Plant TFDB)	23.43	15.93
Baseline 3: Profile kernel [2]	17.33	62.97
SK (Plant DB)	9.22	67.96
SK (Plat TFDB)	**6.71**	**71.91**

Algorithm 1: Kernel Computation

Input: set of N *sequences* $\mathcal{S} = \{s_1, \ldots, s_N\}$, kernel parameters

$\mathbf{K} = \mathbf{0}$;
{Step 1: build feature list $\mathbf{L}$}
$\mathbf{L} = \emptyset$;
for $i = 1, \ldots, N$ **do**
  $L_{s_i} = \bigcup_{\alpha \in s_i} (\alpha, i)$
  $\mathbf{L} = \mathbf{L} \bigcup L_{s_i}$
end for
{Step 2: feature count and kernel computation}
Sort feature list $\mathbf{L}$ using counting sort
Scan sorted list to obtain for each distinct feature α its counts $c(\alpha) = [c_{s_i}(\alpha)]_{i=1}^{N}$ and update kernel $\mathbf{K} = \mathbf{K} + c(\alpha)c(\alpha)^T$

cumulative pairwise comparison between all sequences in $N(X)$ and $N(Y)$

$$K(N(X), N(Y)) = \sum_{x \in N(X)} \sum_{y \in N(Y)} K(x, y), \quad (3)$$

where $K(x, y)$ is a kernel between two sequences x and y (Eq. 2). Algorithm 1 can then be applied to compute 3 by noting that $K(N(X), N(Y))$ can be re-written in the form of Eq. 2 as $\sum_{\alpha \in N(X)} \sum_{\beta \in N(Y)} I(\alpha, \beta)$. This approach, as we show in experiments, results in significant improvements over supervised settings.

3.3. Advantages of sequence-based TF prediction method

The proposed kernel approach to identifying TFs has the following advantages:

1. It only requires a sequence information, while other methods often need other experimental information such as solved secondary structures, 3D structures, ontology, functions, etc.
2. It works on full protein sequence (e.g., multi-domain), i.e. it does not require knowledge of protein domain.
3. The learned model is applicable to new (previously unseen) species (see experiments).
4. It can identify TFs with novel DNA-binding domains (see experiments).

4. Experimental evaluation

We evaluate our methods on three tasks testing ability of the method to (1) identify set of TFs in a given genome, (2) accurately predict TFs *across* species (i.e. for new genomes), (3) detect *new* (previously unseen) TF families. We compare with a number of other approaches including similarity search using PSI-BLAST, sequence composition, profile method [2]. We test our methods in both supervised and semi-supervised setting using a number of sequence databases (NRDB, Plant DB, Plant TFDB) as unlabeled data. We evaluate performance using balanced error rates, as well as ROC-50 (area under ROC curves up to 50 false positives) scores to estimate ranking quality.

4.1. Genome-scale transcription factor prediction

We first test our method on the task of genome-wide prediction of all TFs. We use complete genomes of C. reinhartdii (15497 seqs), A. lyrata (32670 seqs), and A. thaliana (35396 seqs) for this task. We use commonly used sequence composition and PSI-BLAST methods as baselines. We test in both supervised (genome sequences only) setting, and semi-supervised setting by using non-redundant protein database (NRDB, 0.5 million sequences), plant DB (0.9 million sequences), or plant TF database ($\sim$30K sequences). Tables 1, 2, 3 display 10-fold cross-validation error rates for each of these genomes. We observe that our method achieves higher 93-98% accuracy compared to other methods.

4.2. Cross-species TF detection

In this set of experiments, we show ability of the method to accurately identify repertoire of transcription factors in new genomes. We train our method on a known TF repertoire from one species, and then use trained model to predict a set of TFs for new (uncharacterized) species. We use as examples C. reinhartdii

Table 2. Genome-wide transcription factor prediction (Arabidopsis Lyrata, 32670 seqs)

Method	Balanced Error, %	ROC50
Supervised		
Baseline 1: Sequence composition	41.74	0.83
Baseline 2: PSI-BLAST	25.6	-
SK	**8.76**	**60.71**
Semi-supervised		
Baseline 1: Sequence composition (Plant DB)	15.93	14.17
Baseline 2: Sequence composition (Plant TFDB)	13.11	20.75
SK (Plant DB)	2.49	74.85
SK (Plant TFDB)	**1.81**	**92.58**

Table 4. Across-species transcription factor prediction

Train	Test	Method	Balanced Error, %	ROC50
C. Reinhardtii	A. lyrata	PSI-BLAST	21.67	-
A. lyrata	C. Reinhardtii	PSI-BLAST	13.09	-
C. Reinhardtii	A. lyrata	SK (Plant TFDB)	**10.08**	21.90
A. lyrata	C. Reinhardtii	SK (Plant TFDB)	**6.81**	44.83

Table 3. Genome-wide transcription factor prediction (Arabidopsis Thaliana, 35396 seqs)

Method	Balanced Error, %	ROC50
Baseline 1: Sequence Composition	37.13	4.00
SK	8.13	41.84
SK (Plant TFDB)	**7.08**	**44.75**

Table 5. Novel TF detection (C. Reinhardtii)

Experiment	#experiments	Mean Error, %
Hold-one-TF-family out	52	20.1
Hold-two-TF-families out	50	26.39
Hold-three-TF-families out	50	22.57

and A. lyrata species to illustrate trans-species detection of TFs. As can be seen from results in Table 4, the method can identify a set of all TFs with 90-93% accuracy across species (we note that tested genomes have low average sequence identity of only 17%).

4.3. Novel TF detection

We simulate detection of transcription factors with novel DNA-binding domains by holding out TF samples with specific DNA-binding domain(s). TF samples belonging to these held-out TF families are then presented to the model that only contains TF samples from the remaining "known" TF families. We evaluate novel TF detection by holding out one, two or three TF families and compute average performance (detection accuracy) on these held-out families (Table 5). As shown in the table, average accuracy of detecting novel TF families with previously unseen DNA-binding domains is around 80%.

5. Conclusions

We presented and evaluated sequence-based methods for accurate genome-wide transcription factor (TF) identification. The methods predict with high (93-98%) accuracy TFs for a number of genomes, identify with 90-93% accuracy TFs in novel genomes, as well as identify with 80% accuracy new (previously unseen) TF families. These results show promise in using presented methods in annotating newly sequenced genomes.

References

[1] M. Gao and J. Skolnick. Dbd-hunter: a knowledge-based method for the prediction of dnaäiprotein interactions. *Nucleic Acids Research*, 36(12):3978–3992, 2008.

[2] R. Kuang, E. Ie, K. Wang, K. Wang, M. Siddiqi, Y. Freund, and C. S. Leslie. Profile-based string kernels for remote homology detection and motif extraction. In *CSB*, pages 152–160, 2004.

[3] P. P. Kuksa and V. Pavlovic. Spatial representation for efficient sequence classification. In *ICPR*, 2010.

[4] S. K. Kummerfeld and S. A. Teichmann. Dbd: a transcription factor prediction database. *Nucleic Acids Research*, 34(suppl 1):D74–D81.

[5] R. E. Langlois and H. Lu. Boosting the prediction and understanding of dna-binding domains from sequence. *Nucleic Acids Research*, 38(10):3149–3158, 2010.

[6] C. Leslie and R. Kuang. Fast string kernels using inexact matching for protein sequences. *J. Mach. Learn. Res.*, 5:1435–1455, 2004.

[7] C. S. Leslie, E. Eskin, and W. S. Noble. The spectrum kernel: A string kernel for SVM protein classification. In *Pacific Symposium on Biocomputing*, pages 566–575, 2002.

[8] C. S. Leslie, E. Eskin, J. Weston, and W. S. Noble. Mismatch string kernels for SVM protein classification. In *NIPS*, pages 1417–1424, 2002.

[9] W.-Z. Lin, J.-A. Fang, X. Xiao, and K.-C. Chou. idna-prot: Identification of dna binding proteins using random forest with grey model. *PLoS ONE*, 6(9):e24756, 09 2011.

[10] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using string kernels. *J. Mach. Learn. Res.*, 2:419–444, 2002.

[11] I. Melvin, E. Ie, J. Weston, W. S. Noble, and C. Leslie. Multi-class protein classification using adaptive codes. *J. Mach. Learn. Res.*, 8:1557–1581, 2007.

[12] G. Nimrod, M. Schushan, A. Szilagyi, C. Leslie, and N. Ben-Tal. idbps: a web server for the identification of dna binding proteins. *Bioinformatics*, 26(5):692–693, 2010.

[13] S. Sonnenburg, G. Rätsch, and B. Schölkopf. Large scale genomic sequence svm classifiers. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 848–855, New York, NY, USA, 2005. ACM Press.

[14] V. N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, September 1998.

[15] J. Weston, C. Leslie, E. Ie, D. Zhou, A. Elisseeff, and W. S. Noble. Semi-supervised protein classification using cluster kernels. *Bioinformatics*, 21(15):3241–3247, 2005.