A Computational Paradigm for More Specific TFBS Detection

Heike Sichtig [1] and Alberto Riva [1,2]

[1] Department of Molecular Genetics and Microbiology, University of Florida, Gainesville, FL, USA

[2] University of Florida Genetics Institute, University of Florida, Gainesville, FL, USA

Abstract

Motivation: One of the key challenges of current computational biology is the construction of a model of the regulatory network of a cell. The identification of regulatory patterns in genomic DNA and their relation to specific transcription factors that bind to them is vital to understanding the regulatory infrastructure of a cell.

Our paradigm is based on the combination of two biologically realistic information processing methods: third-generation artificial neural network models (spiking neural networks) are used to represent the complex structure of a binding site, while a genetic algorithm is used to optimize the network parameters during a learning phase. The networks are initially trained using known binding sites and negative examples, and are then used as classifiers to detect new TFBSs in genomic sequences.

The goal of our work is to reduce the number of false positives in the predicted TFBSs, through a more accurate modeling of the information contained in the alignments that constitute the training data.

Results: We present the evaluation of a two-neuron network topology trained to represent TFBSs for four different transcription factors. The networks were trained using real TFBS data from the TRANSFAC, JASPAR and SCPD databases, and appropriately generated negative samples, and were compared against MAPPER, TFBIND and TFSEARCH. Our results show that our paradigm has the potential to attain very high classification accuracy, with a very small number of false positives.

1 Introduction

Transcriptional regulation is one of the most basic mechanisms of gene regulation. Transcription factors (TFs) are proteins that bind to specific DNA sequences (Transcription Factor Binding Sites, TFBSs) in the promoter regions of genes, and control their transcription into mRNA through combinatorial interactions. While, in principle, knowledge of the arrangement of binding sites in the promoter of a gene would allow to determine its regulation, in practice identifying TFBSs with high sensitivity and specificity is still an open challenge. Experimental detection of functional TFBSs is complex and expensive, and cannot therefore be performed on a large scale. The alternative strategy, that is, the computational detection of TFBSs, is hindered by the scarcity of training data, by the complexity of the interaction between TFs and DNA, and by the fact that most TFBSs are short and exhibit a poorly conserved consensus sequence. The most common approaches employed for this purpose range from simple pattern-like representation of the binding sites (using regular expressions), to Probability Weight Matrices, to Hidden Markov Models [3]. All these approaches typically suffer from poor specificity, resulting in a large number of false positives.

Here we present a paradigm that is able to model complex patterns by encoding them as*spike trains*, and training a spiking neural network (SNN) to recognize them. SNNs represent the third generation of artificial neural networks and are known for their information processing and pattern recognition capabilities. If we interpret TFBS detection as an information processing problem, a more sophisticated modeling of complex relationships among nucleotides at different positions in a binding site may uncover patterns and characteristic signals that are not otherwise apparent. Given the complexity and the number of parameters of the resulting neural network, we use a genetic algorithm (GA) to optimize its performance through a learning phase. Training is performed using data about known, experimentally validated binding sites from the TRANSFAC, JASPAR and SCPD databases. The purpose of this work is therefore to test whether the use of positional pattern detection can provide a higher ability to detect TFBSs.

2 Methods

2.1 Spiking Neural Networks (SNNs)

This work is an extension of previous work on SNNs by [7, 5, 8] with Gerstner's Spike Response Model (SRM) for leaky "integrate and fire" neurons. We create SNNs as position dependent models for TFBSs. This machine learning classifier is trained with another machine learning method, called a Genetic Algorithm [6].

2.2 Genetic Algorithms (GAs)

The GA used in this work is a variation of the CHC algorithm [2]. CHC uses survival competition between parents and offspring and vigorous crossover, but only when two individuals are sufficiently distant (incest prevention that virtually eliminates genetic drift [4]).

2.3 Computational Paradigm

Figure 1 shows the steps involved a user would follow using our Computational Paradigm. First, a TF of interest has to be selected. Then, positive and negative TFBS examples are randomly paired as inputs for the simulations using a SNN and then optimized by a GA.

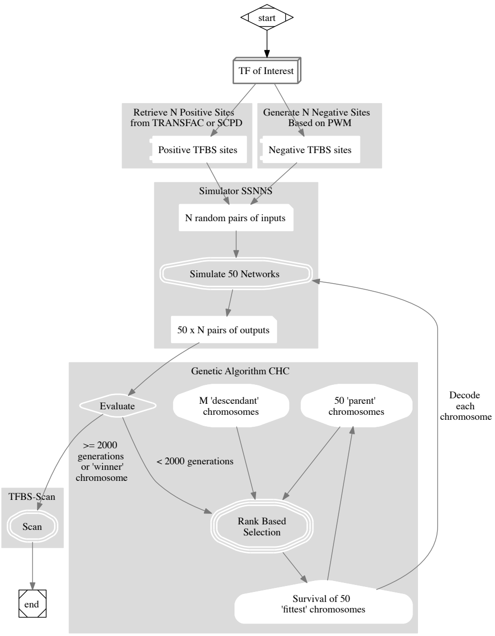

Figure 1: Computational Paradigm. Graphical representation of steps involved a user needs to execute to computationally detect TFBSs for a TF of interest.

2.4 TFBS Scanner

To evaluate the performance of the best evolved network, we create a random sequence of nucleotides (or we extract a fragment of a known genome), and we embed real binding sites into it at known locations, using the RSAT program [11, 9]. We then scan the resulting sequence using the network: we feed the first n nucleotides to the network (where n is the length of the binding site) and record its output spike(s), then move to the second nucleotide and repeat the process. Ideally, we expect to see activity only when the scanned subsequence overlaps with a known binding site.

2.5 Computational Model

We use a simple topology based on two neurons, each of which receives as input the spike train corresponding to the sequence to be classified. We stipulate that when the input sequence is a real TFBS, one neuron should spike and the other neuron should remain quiet, while if the input sequence is not a TFBS, the neuron should not spike while the other neuron should spike. The behavior of the network is simulated using an event-driven version of the previously developed Sichtig's Spiking Neural Network Simulator (SSNNS) [7]. During the training phase, we generate a large population of networks whose tunable parameters are initialized at random, and we select an appropriate *fitness function* that is based on the network's ability to correctly classify the training sequences. The GA is then responsible for evolving the parameters of the population of networks until the desired classification accuracy level is reached. The best network produced by the GA is therefore selected as the optimal one to detect the TFBSs under study.

2.6 Data Sets

The networks were trained using real TFBS data from the TRANSFAC, JASPAR and SCPD databases, and appropriately generated negative samples. We selected four specific TFs as the positive sets: GAL4 in yeast, ABF1 in *Arabidopsis thaliana*, and p53 and cMyc in human. We generate negative examples (sequences that are known not to be binding sites) based on characteristics of known TFBS sequences, such as their probability weight matrix (PWM), to generate new sequences that are similar to the known sites, but at the same time different enough to be suitable as negative examples for training.

3 Results

We tested our method on four TFs (Methods) within three different organisms. Evolved models for all four TFs were scanned using the TFBS Scanner (Methods) with random and genomic DNA sequences. The classification accuracy reached 83% or more for all four TFs tested.

3.1 Comparison with other methods

The following two tables report sensitivity and specificity of the network that was tested with SNN-GA in comparison with MAPPER by [3], TFBIND by [10] and TFSEARCH by [1, 12], three well-known tools to dicover TFBSs. Table 1 shows the counts of true positives (true binding sites). Table 2 shows the count of false positives (negative binding sites that are classified as true binding sites).

Table 1: Comparison with MAPPER, TFBIND and TFSEARCH (sensitivity)

TF (#s)	SNN-GA	MAPPER	TFBIND	TFSEARCH
GAL4[1] (13)	0.92	0.92	0.00	0.23
GAL4[2] (9)	1.00	1.00	0.00	0.44
ABF1 (45)	0.93	1.00	0.00	0.00
p53 (16)	1.00	1.00	1.00	1.00
c-Myc (40)	0.93	0.95	1.00	0.70

[1] binding sites from TRANSFAC database
[2] binding sites from SCPD database

Table 2: Comparison with MAPPER, TFBIND and TFSEARCH (specificity)

TF (#s)	SNN-GA	MAPPER	TFBIND	TFSEARCH
GAL4[1] (13)	0	5	0	0
GAL4[2] (9)	0	2	0	0
ABF1 (45)	12	24	0	0
p53 (16)	0	10	13	2
cMyc (40)	7	17	29	3

[1] binding sites from TRANSFAC database
[2] binding sites from SCPD database

Table 1 shows that the SNN-GA method exhibits high sensitivity, detecting most of the TFBSs for the four TFs. The SNN-GA approach has a low number of false positives while discovering most TFBSs. MAPPER is also able to discover most TFBSs of the four factors; however, it also detected a high number of false positives. TFBIND has high sensitivity sensitivity for the two factors (p53 and c-Myc), but also contains a very high number of false positives. TFSEARCH has the lowest sensitivity, detecting less than half of the TFBSs of both GAL4 TFs but with zero false positives. At the same time, TFSEARCH can detect all p53 TFBSs, but also retrieves two false positives. For the c-Myc TF, TFSEARCH discovers only 28 TFBSs and three false positives.

4 Discussion and Conclusion

The computational identification of TFBSs is an inherently challenging task, due both to the complexity of the problem and to the limited amount of training data available to develop reliable models. In this paper we have described a computational paradigm for TFBS detection that combines the SNN's ability to represent complex patterns with the efficient heuristic optimization performed by a genetic algorithm. In the four cases tested (GAL4 in yeast, ABF1 in *Arabidopsis thaliana* and p53 and c-Myc in human) our method showed high sensitivity (consistently over 92%) and specificity, in comparison with three other common TFBS-finding tools.

The major limitation of our method is currently its high computational cost: training a single network on a relatively small number of binding site examples can take several hours on a medium-sized workstation. Consequently, current evaluation of the paradigm is limited by the high computational cost and scarcity of training data. We are aware that random pairing of all TFBS/ NON-TFBS sites for training may lead to over-estimate the predictive power of the model.

Future work on improving speed of algorithms and parallelizing the GA will lower the computational cost. It should be noted, though, that the training phase only needs to be performed once for each transcription factor. Once the network parameters that provide optimal classification ability have been identified, they are hard-coded into a specialized network that can then be used to scan new sequences very efficiently.

Although our results are still preliminary, they demonstrate that an information-processing framework able to capture higher amounts of information from binding site alignments can improve our ability to reliably detect novel TFBSs. The results presented in this paper show that the novel SNN-GA approach to TFBS discovery is promising. Comparison with existing methods for TFBS detection highligh its potential for attaining high sensitivity and specificity.

References

[1] Yutaka Akiyama. Tfsearch: Searching transcription factor binding sites, 2011.

[2] L. J Eshelman. The chc adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In *Foundations of Genetic Algorithms*, volume 1, pp. 265–283. Morgan Kaufmann, 1991.

[3] Voichita D Marinescu, Isaac S Kohane, and Alberto Riva. Mapper: a search engine for the computational identification of putative transcription factor binding sites in multiple genomes. *BMC Bioinformatics*, 6:79, 2005.

[4] J. David Schaffer, Murali Mani, Larry Eshelman, and Keith Mathias. The effect of incest prevention on genetic drift. In Reeves Banzhaf, editor, *Foundations of Genetic Algorithms*, volume 5, pp. 235–243. Morgan Kaufmann, 1998.

[5] J. David Schaffer, Heike Sichtig, and Craig Laramee. A series of failed and partially successful fitness functions for evolving spiking neural networks. In *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*, pp. 2661–2664, Montreal, Québec, Canada, 2009. ACM.

[6] H. Sichtig, J.D. Schaffer, and A. Riva. Evolving spiking neural networks for predicting transcription factor binding sites. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pp. 1 –8, 2010.

[7] Heike Sichtig. *The SGE framework: Discovering spatio-temporal patterns in biological systems with spiking neural networks(S), a genetic algorithm (G) and expert knowledge (E)*. PhD thesis, State University of New York at Binghamton, http://gradworks.umi.com/33/59/3359709.html, 2009.

[8] Heike Sichtig, J. David Schaffer, and Craig B. Laramee. Ssnns -: a suite of tools to explore spiking neural networks. In *GECCO '08: Proceedings of the 2008 GECCO conference companion on Genetic and evolutionary computation*, pp. 1787–1790, New York, NY, USA, 2008. Atlanta, GA, USA, ACM.

[9] M. Thomas-Chollier, O. Sand, J.V. Turatsinze, R. Janky, M. Defrance, E. Vervisch, S. Brohee, and J. van Helden. Rsat: regulatory sequence analysis tools. *Nucleic Acids Res.*, 2008.

[10] T.Tsunoda and T.Takagi. Estimating transcription factor bindability on dna. *BIOINFORMATICS*, 15(7/8):622–630, 1999.

[11] J. van Helden. Regulatory sequence analysis tools. *Nucleic Acids Res*, 31(13):3593–6, July 2003.

[12] Edgar Wingender. The transfac project as an example of framework technology that supports the analysis of genomic regulation. *Brief Bioinform*, 9(4):326–332, July 2008.