Evaluation of Neural Networks Based Systems for DNA Motif Discovery

Milena Mišić[*], Aleksandar Stanimirović[*], Leonid Stoimenov[*]

[*] University of Niš, Faculty of Electronic Engineering, Niš, Serbia
milena.misic@elfak.ni.ac.rs, aleksandar.stanimirovic@elfak.ni.ac.rs, leonid.stoimenov@elfak.ni.ac.rs

***Abstract* — The main goal of genome analysis and DNA motif finding is to explain cell mechanisms and their influence on genetic diseases. The technological advances in the area result in large amount of data that needs to be processed in order to solve these challenges. That is why machine learning is becoming popular for genome analysis and motif discovery in particular. Some example systems that implement machine learning techniques are DeepBind, DeeperBind and DeepMotif. This work's primary objective is to assess the performance of these systems, examine their usability and behavior on new datasets and provide insights as to what causes the errors systems make. Previous evaluations are available, but do not incorporate neural network systems which are known to achieve the state-of-the-art in other areas. This research aims to address this issue and uses more common systems (e.g. Gibbs Sampler or implementations of the Expectation Maximization algorithm) to establish a baseline.**

I. Introduction

Genome analysis, combined with modern informatics approaches allows researchers to better understand the cell mechanisms and the genetic bases of drug responses and diseases. With introduction of novel deoxyribonucleic acid (DNA) sequencing techniques, the amount of available data in this field increases rapidly. Due to the growing dataset size, various machine learning systems are gaining popularity for the particular problem of DNA motif finding. Software packages commonly used for this purpose, such as statistical ones (implementing Gibbs sampling, EM algorithm or Markov models, for instance), scale poorly. To exemplify, MEME Suite's [1] implementation of Expectation Maximization algorithm scales as a quadratic function of the dataset size [2]. On the other hand, machine learning algorithms are often designed with large quantities of data in mind and thus can be used as a tool for the genome analysis.

At the moment there are a number of proposed systems that can be used as a tool for genome analysis. Some of the examples include EXTREME (an online implementation of the EM algorithm) [2], gkmSVM (an implementation of gapped-kmer support vector machine) [3], GibbsSampler [4], DeepBind (a neural network with single convolutional layer) [5], DeeperBind (extended DeepBind with additional recurrent layers) [6], DeepMotif (a deep neural network with both convolutional and recurrent layers) [7] and others. Although some of them have diverse underlying mechanisms, they share a common goal – to reduce the computational complexity of their models and to successfully handle newly available data.

Most of these systems have been recently developed and are still in the research phase. Some of them are usable in practice, while others are not. Further, it is not easy to decide which one to choose, since there are no publicly available testing results apart from the results reported by authors. Most of these systems built after 2015 have used DeepBind as a reference. However, to the best of the authors' knowledge, these systems were not compared.

The main objective of this paper is to provide a comprehensive evaluation of those systems. This evaluation could help researchers in two ways. First, they could utilize the results to choose the system for their research or other applications. Second, it could provide guidelines to researchers as to which approaches should be developed further. Additionally, certain drawbacks of the systems could be detected thus enabling their improvement.

The paper is organized in six sections. The section 2 briefly discusses related work, both systems for motif discovery and recent evaluations. The third section describes the problem and common modeling approaches. In the fourth section, the methodology is proposed for the analysis in this particular area. The results of the evaluation for the three systems (DeepBind, DeeperBind and DeepMotif) are presented in the section five. The last section contains a discussion on the work performed and provides some thoughts on future development.

II. Related work

In the literature, there are a lot of systems for finding motifs in DNA sequences. Nowadays, neural networks are the most popular due to excellent results they achieve in pattern recognition (e.g. in image classification [8]). There are also more traditional approaches based on statistical analysis and other machine learning techniques.

Systems for motif discovery can be classified according to the motif model they use. In [9], authors evaluated algorithms with the following models: k-mers, Markov models, position weight matrices and dinucleotide model. There are also hybrid models combining some of these.

Software used for this purpose also relies on different algorithms to form such a model. They include Gibbs sampling, EM algorithm, support vector machines, spectral analysis [10] and others.

Neural networks utilized for motif discovery usually consist of multiple hidden layers, but their architectures vary. Some systems employ convolutional and fully-connected layers (e.g. DeepBind), while other add recurrent layers between the two (e.g. DeeperBind and DeepMotif). The motivation behind the use of specialized layers is that convolutional layers act as motif detectors [5], while recurrent layers allow the network to capture long-term dependencies which are not easily found otherwise.

Although quite extensive and conducted as a part of the DREAM5 challenge, [9] does not include neural networks in its assessment. Other evaluations are also available, such as [11]. However, although [11] discusses machine learning based motif discovery tools, it almost exclusively relies on different implementations of genetic algorithms (such as MOGAMOD [12] or GARPS [13] [14]) and do not account for neural networks.

III. DNA MOTIF DISCOVERY

DNA consists of nucleotides whose components are nitrogen bases (adenine, guanine, thymine and cytosine). The nucleotides (with one corresponding nitrogen base) are arranged in a sequence. Certain sequences of bases are recognized by some proteins (i.e. transcription factors - TFs) which then bind to that sequence in the DNA. Those sequences are called motifs and are of great importance for transcription and gene regulation processes.

Discovery of the motifs can be executed through experimental procedures, but the more frequent and convenient approach nowadays is to apply machine learning algorithms or statistical analysis in order to identify them. Computational methods impose two questions: how the motifs should be represented and which algorithm to use to infer the motif.

Representations include consensus sequences, positional weight matrices (PWMs), k-mers or dinucleotide models. Motifs are discovered by EM algorithm, Gibbs sampling (which are standard approaches [15]), SVM implementations, genetic algorithms and most recently, neural networks. Neural networks differ from other approaches in that they do not model the motif explicitly, but rather classify the sequence as positive, when it contains the motif, or negative, when it does not.

IV. METHODOLOGY

In this section, the evaluation methodology of systems for motif discovery will be described. First, a short overview of the systems used for evaluation will be presented. In the following subsection, we will propose an evaluation pipeline. Finally, the implementation details will be presented.

As a dataset for the analysis 108 K562 cell ENCODE ChIP-Seq is chosen. It contains a separate dataset and test set for each transcription factor. In this analysis, transcription factors used are ATF1, ATF3, CEBPB, BACH1 and BHLHE. Further examinations were performed for ATF1, ATF3 and CEBPB. This dataset was used in the original DeepBind and DeepMotif papers.

A. *Neural networks systems*

The three systems to be analyzed are DeepBind, DeeperBind and DeepMotif.

DeepBind was developed so that it can process both *in vitro* and *in vivo* data in order to discover motifs. The probability that a transcription factor will bind to a given sequence is given by the equation 1. The equation indicates that the network consists of a convolutional layer, rectified linear unit, pooling layer and fully-connected layers on top of the pooling layer.

$$f(s)=net_w(pool(rect_b(conv_m(s)))) \quad (1)$$

DeeperBind was developed as an improved solution over the DeepBind. It includes a recurrent layer on top of convolutional layer. Its significance originates from the existence of loops and the ability to map the history of previous inputs on to a current output [16]. The recurrent layer in question uses a long-short term memory units. The idea of such layers is important because, unlike the standard approaches where base pairs in the motifs are assumed to be independent, these models are able to capture long-term dependencies among the bases within the motif.

DeepMotif is a multi-layered neural network which includes three different architectures. The first one uses only convolutional layers, the second only relies only on recurrent ones, while the third implements a combination of those two types. In the original paper [7], the authors report that the best results are achieved with hybrid architecture, so that one will be used in this paper. What makes this implementation stand out is the explanatory aspect – along with the neural network, additional software is provided that aims to explain how the motif was discovered and thus improve the reliability and applicability of the system.

B. *Evaluation pipeline*

In order to perform the evaluation, [9] emphasizes the importance of the following:

- The output of any system for motif discovery should be a numerical score indicating the preference for the sequence,
- Area under the ROC curve should be used as a measure to describe if positive sequences (sequences to which the proteins will bind) can be discriminated from the negative sequences (sequences to which the proteins will not bind).

With these two factors in mind, we propose the pipeline consisting of five basic steps and an optional training step (Fig. 1). Those steps are data loading, data processing, execution, performance measurements and result comparisons.

The training step, if performed, should be the first one. If systems are available in the form of source code, as it is the case for the systems to be analyzed, it is necessary to train them.

In the data load phase, it is necessary to analyze the dataset which will be used for further analysis. Some of important features are the sequence length, class distribution, the motif representation format.

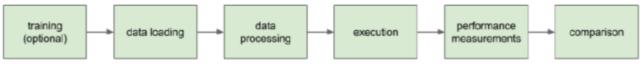

Figure 1 – Evaluation pipeline

Data processing refers to adapting the data format for each system separately. Apart from simple adaptations, such as adjusting the sequence length, this can also refer to adaptation of data types for a particular system. This problem is much harder and, to our best knowledge, has not yet been automated. The problem originates from the fact that the data can be obtained from different laboratory experiments (e.g. PBM or ChIP-Seq) and thus cannot be directly mapped from one experiment to another.

The last two phases are measuring the performance of the systems and result comparison. The first one refers to evaluation of each system individually and in regard to the test set. The comparison indicates that these results should be further analyzed and some insight into the errors they made should be provided.

C. Implementation details

For the purpose of implementing the proposed methodology and for testing the aforementioned systems, a few aspects of those systems had to be taken into consideration.

For conducting this analysis, the first requirement is to determine data type and dataset. To this end, we have chosen to use data originating from ChIP-Seq experiment available in FASTA format from 108 K562 cell ENCODE ChIP-Seq dataset. The data is organized as follows: for each transcription factor, there is a separate dataset and thus a separate neural network model to be trained. The dataset consists of sequences which are labeled as positive (containing a motif for transcription factor) or negative. With training examples labeled in this manner, the problem becomes binary classification. All sequences are 101 base pairs long. In the test set for each transcription factor there are is a total of 1000 sequences evenly distributed between the positive and negative class.

In such setup, it is assumed that systems can handle ChIP-Seq data in FASTA format. That is not applicable for DeeperBind. To solve that issue, the input layer of the network has been adapted. The input vector length has been changed from 35 base pairs to 101 base pairs represented by one-hot encoding. The labels were changed to 1 or 0, for positive and negative sequences, respectively. It is important to note that the network architecture was not altered in any way.

TABLE I.
SPEARMAN CORRELATION COEFFICIENT FOR DIFFERENT TRANSCRIPTION FACTORS

Systems / TFs	DeepBind	DeeperBind	DeepMotif	Dataset size
ATF1	0.76	0.75	**0.77**	20110
ATF3	0.57	0.57	**0.76**	21715
CEBPB	0.84	**0.85**	0.84	53501
BACH1	**0.84**	0.80	0.83	4628
BHLHE	0.77	**0.79**	0.77	30796
Average	0.76	0.75	**0.79**	/

To analyze the results, in addition to dataset name and type and transcription factor, we also stored neural network parameter values (dropout probability, learning rate, learning rate decay etc.). This evaluation takes into account all classification results on the sequence level in order to provide a list of misclassified sequences for further analysis.

Correlation and other metrics were calculated using Scikit-learn Python library [17].

V. RESULTS

A common metrics for assessing how well the predictions resemble the real labels for the motif discovery is Spearman coefficient. In the motif discovery settings, it was used as metrics in [6]. The Spearman coefficient was calculated for five different transcription factors within the dataset: ATF1, ATF3, CEBPB, BACH1 and BHLHE. The results are given in table 1.

In the table 1, dataset size is also displayed. It ranges from about 4500 training examples to over 53000. Test sets for each transcription factor consist of 1000 sequences, half of which are positive.

By comparing Spearman correlation coefficients, it is evident that for the given setting, DeepMotif has the best average performance. For ATF1 and ATF3 it scores the highest.

The average performances of DeepBind and DeeperBind differ only by 0.01. There might be a few reasons behind this. The first one is that the recurrent layers do not add to the predictive value of the network for this particular problem. This does not seem likely since another neural network with the same architecture constantly exhibits the best performance. The other possible reason might be the state of the software. DeepBind is trained by its authors and available as such online. Additionally, it is a more mature solution being available since 2015. On the other hand, DeeperBind is still in a research phase. Its source code can be downloaded, but it is up to a research to train the model. That could be the reason for its poorer performance.

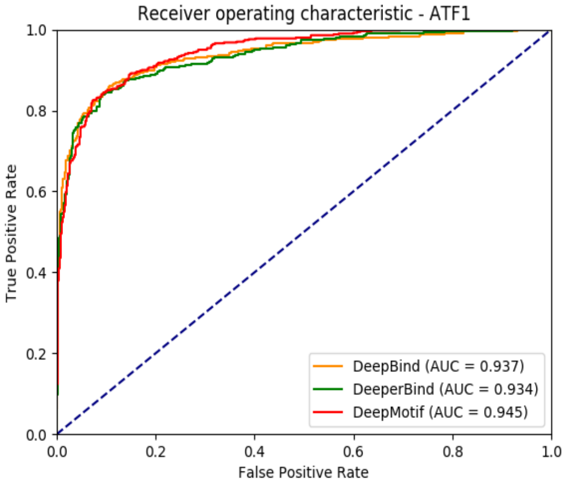

Figure 2 – ROC curve for ATF1 transcription factor

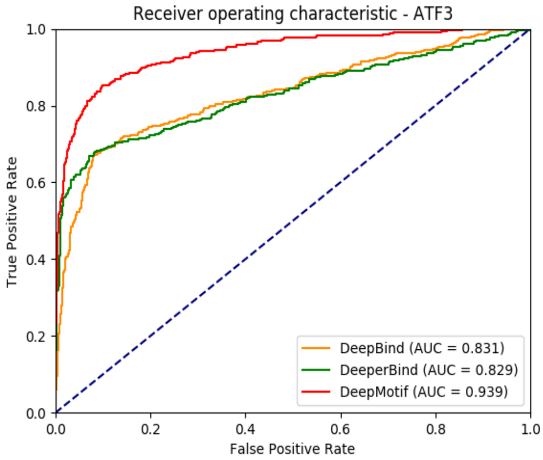

Figure 3 – ROC curve for ATF3 transcription factor

The figures 2 and 3 illustrate behavior of DeepBind, DeeperBind and DeepMotif on two different datasets. It is important to note that, even though different systems are the best ones for different transcription factors, DeepMotif does not show significant fall in performance even when it is not the most accurate solution. The same is not applicable to DeepBind and DeeperBind.

In table 2, AUC is given for each of the three systems and for different datasets (ATF1, ATF3 and CEBPB). The highest average AUC is observed for DeepMotif which is 0.96.

Another analysis was conducted in order to explore the correlation between the predictions of these systems. To that end, Spearman correlation was calculated for all pairs of systems. The results are given in the table 3.

The highest correlation is constantly noted between DeepBind's and DeeperBind's predictions across all transcription factors (ATF1, ATF3 and CEBPB). This implies that they might be making the same errors in the process of classification. In order to tackle this problem further, a list of erroneously classified sequences is extracted and available for analysis.

VI. Conclusion

The proposal of the procedure to analyze systems is significant because if proved beneficial in practice, it could become an evaluation framework for machine learning systems used in the process of motif discovery. As a consequence of their diversity, a well-defined approach has to be established and this work might be a good step in that direction.

Using this methodology for analysis of DeepBind, DeeperBind and DeepMotif, the obtained results indicate that DeepMotif consistently gives the most accurate predictions of binding affinity.

TABLE II
AUC SCORES FOR THE SYSTEMS

TFs / System	ATF1	ATF3	CEBPB	Average
DeepBind	0.937	0.831	**0.988**	0.92
DeeperBind	0.934	0.829	**0.988**	0.92
DeepMotif	**0.945**	**0.939**	**0.988**	**0.96**

TABLE III
SPEARMAN CORRELATION COEFFICIENT BETWEEN PREDICTIONS OF PAIRS OF SYSTEMS

System pairs / TFs	DeepBind and DeeperBind	DeepBind and DeepMotif	DeeperBind and DeepMotif
ATF1	**0.825**	0.805	0.797
ATF3	**0.742**	0.689	0.729
CEBPB	**0.844**	0.829	0.834

The most relevant issue encountered during the evaluation is the lack of information integration. Most of the systems are constructed to work almost exclusively with the datasets provided by the authors. That fact makes it hard to test and compare systems. In this work, that issue was ameliorated by adapting some of the systems to work with different datasets.

Another problem with machine learning systems is their interpretability. Probabilistic models or classification algorithms provide an explanation for decisions made by the system. However, if neural networks are to be used in biomedical applications, some means for explanation and justification need to be available.

Some more traditional approaches, i.e. optimization of position weight matrices (PWMs) through EM algorithm or Gibbs sampling, do exist, but typically solve the problem of generating PWMs as motif models which could be later used for scoring the sequence. The problem with systems which use PWMs stems partially from the assumption that bases within the motif are independent. Neural networks do not assume that independence and that might be one of the reasons why they provide better results.

Another issue which is also frequently encountered in systems relying on PWM and statistics is scalability. It is natural for neural networks to work with large datasets, while the EM algorithm, for instance, required an online implementation at the very least in order to be used with the novel data. On the other hand, these systems provide more insight into how the decision was made, since the motif model is explicitly available. As previously noted, this does not hold true for neural networks.

Authors of this paper extracted the misclassified sequences to attempt to ameliorate the problem. These sequences could be examined by a domain expert in order to confirm or deny the existence of common characteristics between the sequences. If there are such characteristics, it would be possible to adapt the system to the specific problem. If that is not the case, then we might conclude that the behavior is caused by imperfect models and continue to work on them.

References

[1] T. L. Bailey, J. Johnson, C. E. Grant, and W. S. Noble, “The MEME Suite,” *Nucleic Acids Res.*, vol. 43, no. W1, pp. W39–W49, 2015.

[2] D. Quang and X. Xie, “EXTREME: An online em algorithm for motif discovery,” *Bioinformatics*, vol. 30, no. 12, pp. 1667–1673,

2014.

[3] M. Ghandi, M. Mohammad-Noori, N. Ghareghani, D. Lee, L. Garraway, and M. A. Beer, “GkmSVM: An R package for gapped-kmer SVM,” *Bioinformatics*, vol. 32, no. 14, pp. 2205–2207, 2016.

[4] W. Thompson, E. C. Rouchka, and C. E. Lawrence, “Gibbs Recursive Sampler: Finding transcription factor binding sites,” *Nucleic Acids Res.*, vol. 31, no. 13, pp. 3580–3585, 2003.

[5] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey, “Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning,” *Nat. Biotechnol.*, vol. 33, no. 8, pp. 831–838, 2015.

[6] H. R. Hassanzadeh and M. D. Wang, “DeeperBind: Enhancing prediction of sequence specificities of DNA binding proteins,” in *Proceedings - 2016 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2016*, 2017, pp. 178–183.

[7] J. Lanchantin, R. Singh, B. Wang, and Y. Qi, “Deep Motif Dashboard: Visualizing and Understanding Genomic Sequences Using Deep Neural Networks,” in *Pacific Symposium on Biocomputing 22*, 2017, pp. 254–265.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Adv. Neural Inf. Process. Syst.*, pp. 1–9, 2012.

[9] M. T. Weirauch *et al.*, “Evaluation of methods for modeling transcription factor sequence specificity,” *Nat. Biotechnol.*, vol. 31, no. 2, pp. 126–34, 2013.

[10] N. Colombo and N. Vlassis, “FastMotif: spectral sequence motif discovery.,” *Bioinformatics*, vol. 31, no. 16, pp. 2623–31, 2015.

[11] A. Makolo, “A Comparative Analysis of Motif Discovery Algorithms,” *Comput. Biol. Bioinforma.*, vol. 4, no. 1, p. 1, 2016.

[12] M. Kaya, “MOGAMOD: Multi-objective genetic algorithm for motif discovery,” *Expert Syst. Appl.*, vol. 36, no. 2 PART 1, pp. 1039–1047, 2009.

[13] H. Huo, Z. Zhao, V. Stojkovic, and L. Liu, “Optimizing genetic algorithm for motif discovery,” *Math. Comput. Model.*, vol. 52, no. 11–12, pp. 2011–2020, 2010.

[14] Y. Fan, W. Wu, R. Liu, and W. Yang, “An iterative algorithm for motif discovery,” in *Procedia Computer Science*, 2013, vol. 24, pp. 25–29.

[15] K. Y. Yip, C. Cheng, and M. Gerstein, “Machine learning and genome annotation: A match meant to be?,” *Genome Biology*, vol. 14, no. 5. 2013.

[16] A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*, 1st ed. Springer-Verlag Berlin Heidelberg, 2012.

[17] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2012.