CS230

Learning low-dimensional representation for whole-genome gene regulatory elements

Zan Xu
Department of Aeronautics and Astronautics
Stanford University
SUNet ID: zanxu
zanxu@stanford.edu

1 Introduction

One of the main challenges for research in genomics is to identify functional elements in the genome,[1] especially gene regulatory elements that play a vital role in transcriptional regulation, cell differentiation, disease development. As shown in Figure 1, gene regulatory elements, such as promoters, enhancers, silencers, insulator, are short regions of non-coding DNA sequences that reside in open chromatin in a cell type-specific manner and are bound by sets of transcription factors for positive or negative transcriptional control.[2][3] This project aims to take DNA sequences as input and uses encoding/embedding techniques from NLP to learn low-dimensional representations of genetic regulatory elements in order to facilitate their downstream analysis.

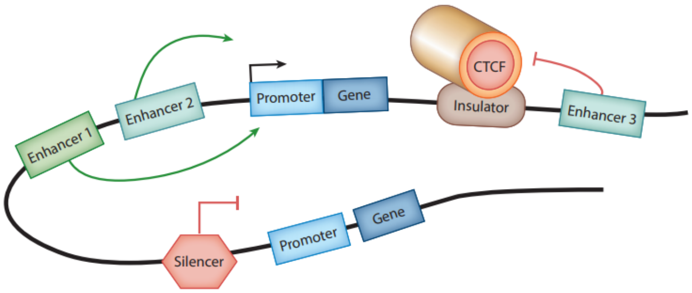

Figure 1: The different types of gene regulatory elements

2 Literature Review

The most common approach to encode a DNA fragment is one-hot encoding, which encodes the nucleotide in each position as a four-dimensional one-hot binary vector representing one type of nucleotide A, C, G, T respectively. [4][5] For predicting functionality of DNA sequences, various neural network archtectures have been explored, such as CNNs [6][7], RNNs [8] or hybrids of the two [9]. Some have also incorporated k-mer representation (which simply means subsequences of length k) and word embedding in order to improve the performance [8][10]. To build on these models, I would like to apply the attention model to the embedding representation to improve the prediction accuracy as well as to hopefully identify patterns that earlier models of neural network architectures failed to capture.

,

3 Dataset and Features

For the unsupervised learning phase, each chromosome is divided into 200 bp (base pair) bins, where each bin can be recognized as sentence and each chromosome can be recognized as corpus. For the supervised learning phase, enhancers and silencers are identified using ChromHMM[11] for Human Embryonic Stem Cell Lines (HESC) cell lines. The sequence of these identified regulatory elements can be obtained through getfasta [12], which are in the form of DNA subsequences with varying lengths on the order of $\sim 10,000$. To suit for classification task laters on, we split the different forms of regulatory elements as positive and negative samples. E.g. the identified enhancers are labelled as positive samples whereas silencers are collected as negative samples. For the scope of this study, around $2,000$ enhancers and $2,000$ silencers with highest confidence scores are collected and annotated for downstream training process.

4 Methodology

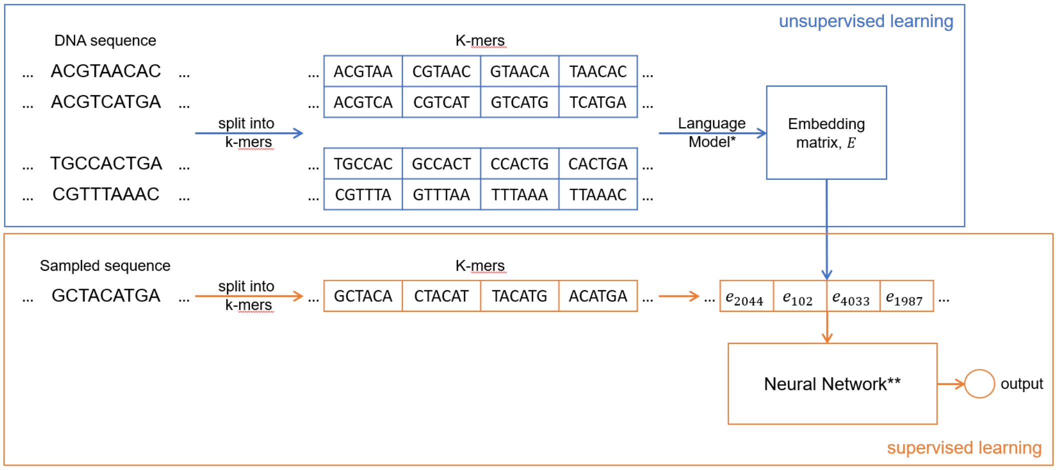

Figure 2: The computational framework consists of two components: the unsupervised learning component (Section 4.1) and the supervised learning component (Section 4.2)

4.1 Unsupervised learning phase: embedding

Given a DNA subsequence, we split it into k-mers using sliding window with stride s along a sequence, meaning that two adjacent k-mers have a distance of s bps. Thus, in general, a sequence with L bps can be split into `floor[(L-k)/s]+1` k-mers. For example, we could split "ATGCAACAC" into four 6-mers with stride $s = 1$ as "ATGCAA", "TGCAAC", "GCAACA" and "CAACAC" (Figure 2). Note that the vocabulary size for k-mers is 4^k. These k-mers are then grouped into sentences. We feed these sentences into various language models, such as Word2Vec to obtain embedding matrices of size $4^k \times n$, where n is the dimension of the embedding vector. The embedding matrix is then used in subsequent supervised learning tasks.

4.2 Supervised learning phase: classification prediction

To analyze the generated embedding matrix, we encode DNA subsequences of gene regulatory elements and use the encoded representation in a classification task to test for performance. The idea is analogous to sentiment analysis. First, DNA subsequences of gene regulatory elements are pre-processed into sentences of k-mers, similar to the embedding pre-processing step. The embedded vectors of kmers in each sentence are then retrieved and fed into a classifier for prediction.

4.3 Integration

Though language models provide embedding matrices that not only reduce the dimenisonality of the inputs, but also improve the performance of classifiers, they can be integrated into the classifier

Table 1: Performance metrics under different NN architectures

	AUC	Accuracy
Averaging	0.615	0.6185
CNN	0.613	0.6210
LSTM	0.582	0.6063
LSTM with attention	0.619	0.6208

as a set of trainable hyperparameters to be adjusted for particular tasks. In this project, different forms of classifiers are used to train the model as well as the embedding matrix in order to obtain the low-dimensional representation of the embedding matrix.

5 Evaluation

As the aim of this project is to explore the low-dimensional embedding performance in various classification tasks, the major hyperparameter to choose is the neural network architecture. The performance of the overall model is evaluated based on AUC and accuracy. The most simplistic model uses a simple averaging layer to average the embedding vectors in each sentence. The average is then fed into a single neuron with sigmoid activation for classification. Since averaging can be considered as a special case of convolutional operation, another model incorporating CNN and pooling operations is also tested. If the $k$mers are treated as sequence data, RNN or its variants are suitable candidates as well. Hence a sequence model using LSTM is tested. In addition, an attention mechanism is added to explore the emphasis among different $k$mers. The results of various models are summarized in Table 1 and their ROC curves plotted in Fig 3. The overall accuracy among different model architectures is not high, around 0.61. LSTM with attention model outperforms other models by a small margin. It is noted that these models are used as baseline comparisons and none of them are exploited with large number of hyperparameters. Therefore, the comparison in Table 1 shows that no one particular architecture has a competitive edge in this case though exploring a larger hyperparamter space can potentially improve the performance at the expanse of computational costs.

Qualitatively, the embedding matrix from the best performing test case is projected non-linearly to a low-dimensional space and visualized using t-SNE. The plot (Fig. 4) show that there exists some form of clustering among different $k$mers induced by the classification task and there is potentially an underlying relationship among $k$mers that can be exploited to aid in gene regulatory element prediction tasks.

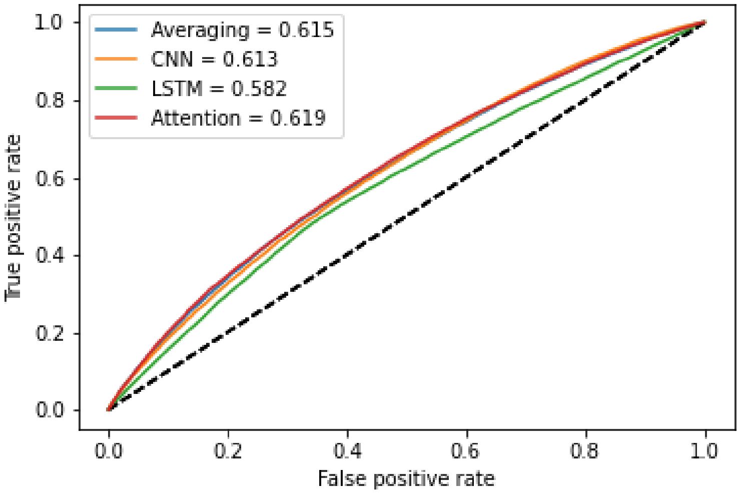

Figure 3: Performance metric ROC of different NN architectures

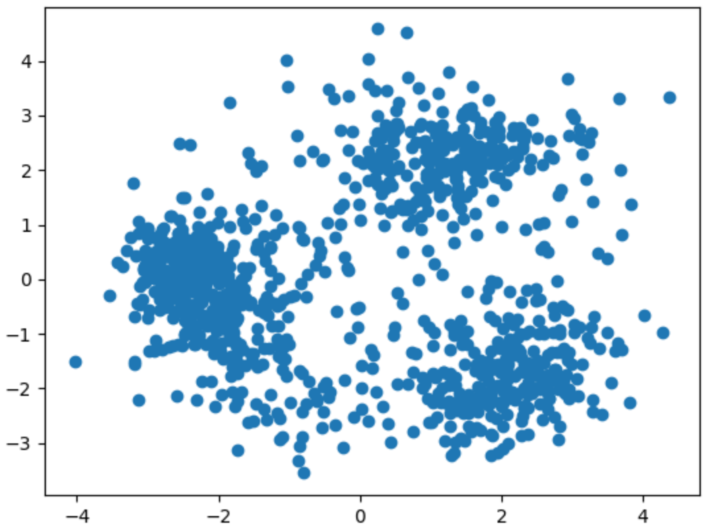

Figure 4: t-SNE plot of embedding matrix

6 Conclusion

In this work, I try to apply word embedding and sentence embedding methods in NLP to genomic data. In the unsupervised phase, the results show that using k-mer as words and taking the deep learning-based embedding framework, the embedding vectors of different types of regulatory elements can be extracted in a lower-dimensional form successfully. In the supervised phase, the results show that deep learning methods however, do not always perform well. The performance of embedding matrix in classification task seems to be insensitive to the model architecture under the choices of hyperparameters considered. Some future work includes expansion to larger hyperparameter spaces in order to explore the performance of various classification models. In addition, more advanced language models such as GloVe, Transformer, BERT can be used to pre-train the embedding. Lastly, network architecture such as graph convolutional neural network can be potentially helpful as well to capture the 3D effect of entanglement of proteins.

References

[1] Y. Li, C.-y. Chen, A. M. Kaye, and W. W. Wasserman, “The identification of cis-regulatory elements: A review from a machine learning perspective,” *Biosystems*, vol. 138, pp. 6–17, 2015, ISSN: 0303-2647. DOI: `https://doi.org/10.1016/j.biosystems.2015.10.002`. [Online]. Available: `http://www.sciencedirect.com/science/article/pii/S0303264715001604`.

[2] S. Chatterjee and N. Ahituv, “Gene regulatory elements, major drivers of human disease,” *Annual Review of Genomics and Human Genetics*, vol. 18, no. 1, pp. 45–63, 2017, PMID: 28399667. DOI: `10.1146/annurev-genom-091416-035537`. eprint: `https://doi.org/10.1146/annurev-genom-091416-035537`. [Online]. Available: `https://doi.org/10.1146/annurev-genom-091416-035537`.

[3] G. A. Maston, S. K. Evans, and M. R. Green, “Transcriptional regulatory elements in the human genome,” *Annual Review of Genomics and Human Genetics*, vol. 7, no. 1, pp. 29–59, 2006, PMID: 16719718. DOI: `10.1146/annurev.genom.7.080505.115623`. eprint: `https://doi.org/10.1146/annurev.genom.7.080505.115623`. [Online]. Available: `https://doi.org/10.1146/annurev.genom.7.080505.115623`.

[4] X. Min, W. Zeng, S. Chen, N. Chen, T. Chen, and R. Jiang, “Predicting enhancers with deep convolutional neural networks,” *BMC bioinformatics*, vol. 18, no. 13, p. 478, 2017.

[5] J. Zhou and O. G. Troyanskaya, "Predicting effects of noncoding variants with deep learning–based sequence model," *Nature methods*, vol. 12, no. 10, pp. 931–934, 2015.

[6] H. Zeng, M. D. Edwards, G. Liu, and D. K. Gifford, "Convolutional neural network architectures for predicting dna–protein binding," *Bioinformatics*, vol. 32, no. 12, pp. i121–i127, 2016.

[7] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey, "Predicting the sequence specificities of dna-and rna-binding proteins by deep learning," *Nature biotechnology*, vol. 33, no. 8, pp. 831–838, 2015.

[8] Z. Shen, W. Bao, and D.-S. Huang, "Recurrent neural network for predicting transcription factor binding sites," *Scientific reports*, vol. 8, no. 1, pp. 1–10, 2018.

[9] D. Quang and X. Xie, "Danq: A hybrid convolutional and recurrent deep neural network for quantifying the function of dna sequences," *Nucleic acids research*, vol. 44, no. 11, e107–e107, 2016.

[10] W. Zeng, M. Wu, and R. Jiang, "Prediction of enhancer-promoter interactions via natural language processing," *BMC genomics*, vol. 19, no. 2, pp. 13–22, 2018.

[11] J. Ernst and M. Kellis, "Chromhmm: Automating chromatin-state discovery and characterization," *Nature methods*, vol. 9, no. 3, pp. 215–216, 2012.

[12] A. R. Quinlan and I. M. Hall, "Bedtools: A flexible suite of utilities for comparing genomic features," *Bioinformatics*, vol. 26, no. 6, pp. 841–842, 2010.