DNA Sequence Classification for Detection of Plasmid Fragments

Mehnaz Yunus[1], Sharanya Kamath[1], Nagaratna B Chittaragi[1], and Shashidhar G Koolagudi[1]

[1] Department of Computer Science and Engineering, National Institute of Technology Karnataka
Email: {16co124mehnaz, 16co140.sharanya}@nitk.edu.in

***Abstract*—DNA classification is the problem of identifying the functionality of genes using only the sequence information (ATGTGT...) automatically. To figure out the objectives of various genes and proteins, sequence classification has appealed a abundance of onsideration in genomic studies. Plasmids are round or straight dual-stranded DNA molecules which are proficient of self-governing duplication and are interchangeable between various bacteria. In silico approaches for forecasting of genomic aspects such as genes are undoubtedly varied for chromosomes and plasmids, accordingly establishing the need for separation of chromosomal from plasmid sequences in a metagenome.**

We developed a machine learning model for discrimination between plasmid-derived and chromosome-derived sequences.

***Index Terms*—DNA sequence classification, bi-directional lstm, recurrent neural networks, plasmids**

I. Introduction

Sequence classification has a wide bound of actual utilizations. In genomic studies, segregating protein arrangements into extant classes is used to determine the behaviours of a new protein. In health-studies, segregating ECG time series (the heart rates time series) lets us know if the information comes from a fit person or from a sufferer of heart ailments. In anomaly detection or intrusion detection, the arrangement of a end users system admittance actions on Unix is surveyed to catch peculiar behaviors. In information retrieval, segregating files into various topic sections has appealed a huge amount of considerations. Other fascinating examples include segregating query log sequences to recognize web robots from humans and classifying transaction sequence data in a bank for the desire of opposing money fraud.

Usually, a sequence is an systematized register of actions. An action can be defined as a symbolic assessment, a numerical real assessment, a vector of real assessments or a complex data type. In this paper, we acknowledge sequence data to be a DNA sequence composed of four amino acids A, C, G, T and a DNA segment, such as ACCCCCGT sequence. In prevailing sequence classification, all individual sequences are correlated with a single classification label and the entire sequence is accessible to a classifier before the classification.

Coming era sequencing technologies produce huge amounts of genomic sequence data in the form of RNA or DNA patterns. Study of DNA patterns is vital in avoiding the evolution of bacteria, viruses, as well as deployed to detect disease during the beginning stages.

Innovative methods such as Deep Neural Networks (DNNs) have already been adapted for genomics problems such as motif discovery, gene expression inference, and predicting the harmful nature of genetic variants. [5]

II. Literature Survey

An efficient forecasting model for the action of noncoding DNA can have huge amount advantages for both traditional science and translational studies because more than 98% of the human DNA sequence is noncoding and 93% of disease related varieties exist in these areas.

Traditionally, analyzing DNA sequences involves searching for common motifs or through phylogenetic comparison against known DNA sequences. As such, classifying a new DNA sequence relies heavily on feature engineering using prior knowledge and experts for annotation. Previous work by Asgari et al. demonstrated that word2vec vectors representing trigrams of amino acids could be trained on large amounts of protein sequence data. [8] The resulting vector representation maintained known biological relationships and were successfully used as features for protein family classification.

To bridge the growing gap between sequenced and annotated DNAs, many computational methods have been established. A larger part of these approaches use machine learning means such as ANNs (Artificial Neural Networks) and SVMs (Support Vector Machines). [2] Other in silico methods are established on the study of amino acid predispositions, physicochemical virtues and statistical ability. Equivalent approaches can be unified to form meta classifiers, which use the outcomes of other classifiers to form their own reinforced predictions.

Recently, the application of deep neural networks with more than two hidden layers to proteins has permitted better learning of deep and complex relationships between sequences, structures and functions of amino acids, and advanced the accuracy of pairwise contact prediction, secondary structure and solvent accessible surface-area prediction and protein disorder prediction [5]. However, common deep learning approaches, such as RNNs and window-based ANNs, while efficient at proliferating local faults within sequence neighbours, are inefficient at modeling long-range (non-local) communications among amino acid residues that are skeletal but not arrangement neighbours. [1] Because residueresidue communications are predominated by skeletal neighbours, how to assess them is the answer for improvising sequence-based classification of protein skeletal and functional virtues.

The long-range dependence between a series of time-resolved events can be better captured by enforcing the constant error flow so that useful long-range interactions can be memorized. In this Long Short-Term Memory (LSTM) network, hidden layers are made of memory blocks containing one or more LSTM units. Every LSTM unit has the foresight to either output, input to or forget the CEC (Constant Error Carousel). The CEC ventures through each LSTM unit in the whole sequential event, substituting as a memory pillar efficiently bridging the entire sequence, in the front or the back direction. LSTM-based neural networks have been fortuitously used for to speech and image-related dilemmas for which long-range memory is the answer for definite analysis and classification.

Neural Networks (NNs) models have achieved state-of-the-art performance on language modeling tasks in recent years and are now seeing adoption for biological problems. [2] Analysis of DNA sequences is important in preventing the evolution of viruses, bacteria, and also used to diagnose disease during an early stage.From the existing work, the authors absorb some clustering algorithm and data analytics techniques like K-mean, k-mer, KNN, SVM, random forest correlation coefficient, and eigenvalue vector are used for predicting neurological disease.

Currently implemented methods for plasmid-chromosomal classification:

- Support Vector Machine
  The SVM is a machine learning technique with a strong theoretical foundation that has been used to improve classification accuracy in biological applications such as the detection of protein family members. kmer based SVM gave an average classification accuracy of 0.871625. [10]
- Random Forest.
  In random forest classification, trees are disciplined on the basis of random collection of strains and genes. Strains with similar incident patterns could get varied contribution scores. This score is an assessment of how essential a strain is to accurately predict a specific gene. Also, genes that are either existing or non-existing in all queried genes have inconsequential effects to separate genes of varying phenotypes. Random Forest gave a classification accuracy of 0.791423. [6]

  The aforementioned methods have obviously facilitated the development of this important field. However, further studies are still required. Almost all the machine learning methods require fixed length vectors as inputs. Nevertheless, the lengths of DNA sequences vary significantly. During the vectorization process, the sequence-order information and the position dependency effects are lost, and this information is critical for DNA sequence analysis and nucleic acid analysis. Although some studies attempted to incorporate this information into the predictors, it is never an easy task due to the limited knowledge of DNA sequences.

III. Problem Description

We attempt to classify DNA sequences into either plasmid or chromosomal, by learning the features present in the DNA character sequence using LSTM networks. The various tools, procedures, methodologies and algorithms available for DNA classification are to be analysed and compared to find the best and most effective.

A suitable dataset for the purpose is to be found and if not found, one is to be created by manually labelling the fetched tweets.

Improvement of performance of prediction to initial implementation of DNA classification is attempted. The accuracy of prediction of chromosomal or plasmid in the considered dataset by using the trained model is to be analysed and further improvisation of the accuracy is to be attempted.

IV. Methodology

A. Workflow

We start with character level prediction on DNA sequences. Here, we test whether an LSTM model can accurately predict the next character in a DNA sequence, for manufactured sequences. This will show whether the chosen model is suitable for the classification task or not.

To confirm that an LASTM can model the structure within a genomic sequence, we first train a simple character-level LSTM to predict one of the four possible characters given the previous string of characters. If the LSTM does not give the accuracy as expected, then we will have to more carefully tweak our model until we can identify some signal. Moreover, this simpler task will help us to choose an appropriate architecture for our model.

Once we have empirical evidence that an LSTM can capture the non-random structure within a genome, we explore a sequence classification problem. The DNA classification task into chromosomal and plasmid is carried out next.

B. Dataset

In recent years, a large amount of DNA and protein sequences are available in public databases, such as GenBank, EMBL Nucleotide Sequence Database and the Entrez protein database.

The two tasks we performed used different datasets.

For character level genome prediction, our baselines were the artificial genomes. Any model that doesnt achieve results as required on these two genomic sequences will not be used for further experimentation. Two genomes were prepared:

- A length-3000 repeating genome where the repeated unit is AGCTTGAGGC.
- A length-3000 random genome.

For binary classification of DNA sequences the dataset used was from the CAMI project and was obtained from NCBI(National Center for Biotechnology Information) archives. It contained E. Coli complete genomes and plasmids. This dataset contains 3000 DNA sequences of length 500, each with a label indicating whether the sequence is plasmid or chromosomal.

C. Bidirectional LSTM (BLSTM)

A BLSTM (bi-directional long short-term memory network) is an alternative of the RNN that unifies the outcomes of two RNNs, one handling the sequence from right to left, and one from left to right. As a substitute to basic hidden cells, the two RNNs consist of LSTM blocks, which are smart network cells that can memorize a value for a random period of time. Long-term dependencies can be detected by BLSTMs and these networks have been efficient for other machine learning usages such as machine translation, human action recognition, speech recognition and phoneme classification. Even though BLSTMs are efficient for studying sequential information, they have not been applied for DNA sequences. [4]

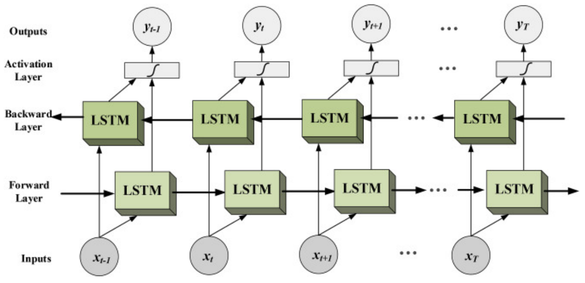

Fig. 1. BLSTM

D. Character Level Prediction

For the character level prediction, an LSTM model with 75 memory cells was used. If this model does not perform as expected on these two genomes, it would point to LSTMs being unable to properly model DNA behaviour [**?**].

- Accuracy for repeating genome: 1.0 after 25 epochs
- Accuracy for random genome: 0.25 after 25 epochs (as expected).

These experiments showed that an LSTM model may be appropriate for modeling a genomic sequence.

E. Sequence classification

The model used contains six layers. The input goes to an embedding layer in which the ACGT sequence is embedded to numerical data. This is followed by 2 sets of bidirectional layer and dropout layer which randomly drops out 20% of the neurons. Finally there is a fully connected layers which produces the final output.The input layer is designed to encode the pseudo protein by one-hot encoding. Bidirectional LSTM extracts the dependency relationships between subsequences. Superiority of every intermediate hidden value from bidirectional LSTM is taken into account for better handling of the long dimension of DNA patterns. Additional detailed dependency data can be comprehended into the hidden values by using BLSTM. Then, those intermediate hidden values are connected to the dense layer. Memory units in a single LSTM block abstract various levels of dependency data. Because of this reason, the dense layer is expected to properly weight the dependency relationships abstracted from various units. The outcomes of the dense layer are integrated into a single feature vector and is given as input into the output layer for prediction function.
This neural network enables the capturing of both the long and short term dependency data of pseudo proteins by collecting the data from each intermediate hidden value of BLSTM.

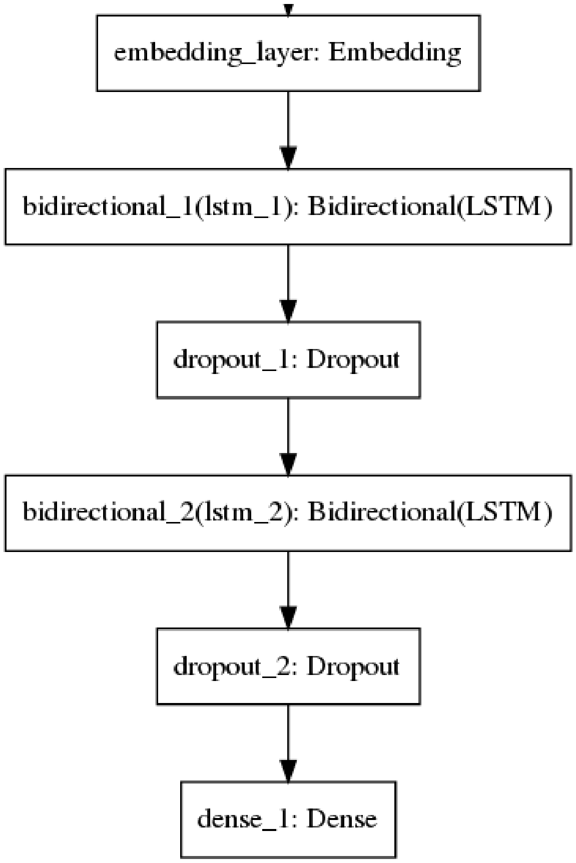
embedding_layer: Embedding
bidirectional_1(lstm_1): Bidirectional(LSTM)
dropout_1: Dropout
bidirectional_2(lstm_2): Bidirectional(LSTM)
dropout_2: Dropout
dense_1: Dense

Fig. 2. Model

The learning rate was determined in initial training, where larger learning rates of ¿0.001 did not enable the network to converge. We employed the step learning rate decay technique to reduce the learning rate by 1% per epoch. That is, the learning rate was initialized at 0.001, and then systematically annealed to 6*10^-4 within 50 epochs. This allowed the model to learn finer detail as it progressed through training. Finally, the two outputs of this network are squeezed into a probability distribution through the use of the softmax function.

We used 5-fold cross-validation in which the original sample is randomly partitioned into 5 equal sized sub-samples. Of the 5 sub-samples, a single sub-sample is retained as the validation data for testing the model, and the remaining 4 sub-samples are used as training data. The cross-validation process is then repeated 5 times, with each of the 5 sub-samples used exactly once as the validation data. The 5 results are then averaged to produce a single estimation. The leverage of this approach over replicated arbitrary subsampling is that all conclusions are deployed for training and validation both, and each conclusion is deployed for validation only once.

V. Results and Discussion

Each output from the classifier can be sorted into one of four outcomes depending on the label of the sample: True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). Sensitivity (Se = TP/(TP+FN)) and specificity (Sp = TN/(TN+FP)) are two metrics which measure the performance of each class in binary classification. Sensitivity illustrates the classifiers ability to correctly allocate samples into the disordered (or positive) class, whereas specificity does the same for the ordered (or negative) class. These two measures are often combined into a single metric to form the balanced accuracy measurement (Acc = (Se+Sp)/2).

The results of the character level prediction fully match the expected results, giving 1.0 for the repeated sequence and 0.25 for the random sequence. We thus proceeded to the next task.

For the DNA classification, performance is assessed through the analysis of binary labels and raw prediction values. The raw prediction probabilities are obtained at the output of the network through the use of the softmax function. The discrete labels are generated by the comparison of these probabilities with a pre-calculated threshold T. The loss on the train dataset decreases as the number of epochs decreases, as expected, and achieves a minimum value of 0.29. The maximum an average validation accuracy obtained were 0.89 and 0.729 respectively. The graphs for the same are shown in figures 2 and 3.

VI. Conclusion

From the preliminary experiments, we were able to deduce that LSTM models are suitable to model DNA sequences effectively. Proceeding to the DNA classification, we developed an LSTM based architecture to classify given DNA sequences into plasmid or chromosomal. The architecture gives an accuracy of 0.89 which performs better than other models used for the same task, such as support vector machine and random forest classifier.

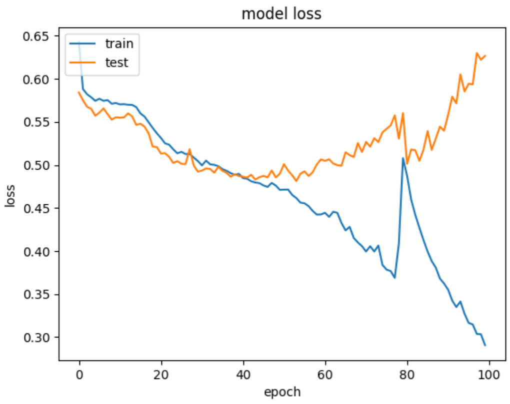

Fig. 3. Loss

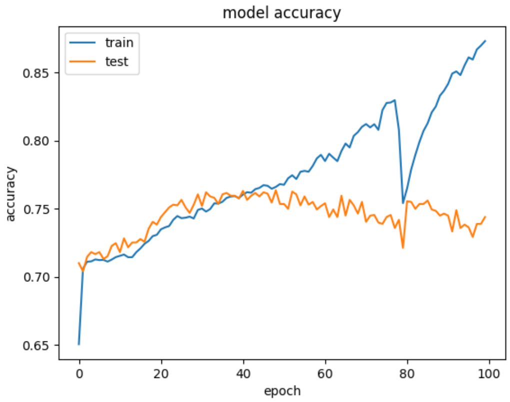

Fig. 4. Accuracy

VII. Future Interests

There are several possibilities for future interests. First, we can endeavor to broaden the model to process genetic variants and then predict their functional ramifications. Second, the model can be made completely recurrent so it can process sequences of discretionary length, such as entire chromosome sequences, to produce sequential outputs.

In contrast, our current setup can only process sequences of constant length with static outputs. A completely recurrent architecture may also further our effort to study variants since it would allow us to explore the long-range consequences of genetic variants.

Secondly, we can attempt incorporating more data to test on longer sequences. Fine-tuning hyperparameters, for example, changing the number of layers, increasing the number of hidden layers, and testing different activation functions can also be tried for increasing accuracy.

References

[1] A deep learning approach to pattern recognition for short DNA sequences, Akosua Busia, George E. Dahl, Clara Fannjiang, June 22 2018.

[2] Deep Recurrent Neural Network for Protein Function Prediction from Sequence, Xueliang Liu, Cornell University, 28 Jan 2017.

[3] DeepCpG: accurate prediction of single-cell DNA methylation states using deep learning, Christof Angermueller, Heather J. Lee, Wolf Reik and Oliver Stegle, Genome Biology (2017)

[4] Improving protein disorder prediction by deep bidirectional long short-term memory recurrent neural networks Jack Hanson, Yuedong Yang, Kuldip Paliwal and Yaoqi Zhou, Oxford, October 26, 2016.

[5] Daniel Quang and Xiaohui Xie. Danq: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. bioRxiv, page 032821, 2015.

[6] Genotype-phenotype matching analysis of 38 Lactococcus lactisstrains using random forest methods,Jumamurat R Bayjanov, Marjo JC Starrenburg, Marijke R van der Sijde, BMC Microbiology 2013.

[7] Qingda Zhou, Qingshan Jiang, Dan Wei, "A new method for classification in DNA sequence", Computer Science & Education (ICCSE) 2011 6th International Conference on, pp. 218-221, 2011.

[8] A Brief Survey on Sequence Classification, Zhengzheng Xing, Jian Pei, Eamonn Keogh, ACM SIGKDD Explorations Newsletter, Volume 12 Issue 1, June 2010

[9] In Silico Detection and Typing of Plasmids, Alessandra Carattoli, a Ea Zankari.

[10] Gene identification using a support vector machine for ORF classification, Lutz Krause, Alice C. McHardy Tim W. Nattkemper.