ANALYSIS OF DNA SEQUENCES USING K-MER ENCODING UNDER NLP APPROACH

R. Phani Kishore Assistant Professor, Department of Computer Science and Engineering, Dhanekula Institute of Engineering and Technology, Vijayawada, (Affiliated to Jawaharlal Nehru Technological University, Kakinada)

P. Siva Puja, D. Geethika, G. Kusuma Navya Suseela, K. Bhuvanesh Research Scholar, Department of Computer Science and Engineering, Dhanekula Institute of Engineering and Technology, Vijayawada, (Affiliated to Jawaharlal Nehru Technological University, Kakinada)

Abstract

The DNA sequence carries and provides the information that the cell uses to make RNA molecules and proteins. The process of reducing the order of DNA bases is termed as DNA sequencing. DNA consists of a linear string of nucleotides or bases, as Adenine(A), thymine(T), cytosine(C) and guanine(G). Analyzing and Classification of DNA sequence is a crucial challenge for biomedical data and is the key to study genomes and the proteins they encode to identify- individual genes, recurrence of DNA sequences for potential drug targets. We perform it based on three divergent species DNA sequence datasets that is composed of very large and vividly combined strings of A, T, C, G which are of human, dog, and a chimpanzee. Various Machine Learning techniques implied in our model are Natural Language Processing, Naive bayes, scikit learn, k-mer encoding.

Keywords: machine learning, scikit-learn, multinomial naïve bayes-NLP, DNA sequence analysis, similar gene identification.

Introduction

Any living organism contains trillions of cells – Each cell has 2 meters of DNA and each with 3 billion DNA subunits termed as the bases (A, T, C, G). Approximately, we have 30,000 genes coded for proteins termed as coding sequence of DNA, that performs most of the life functions. Every living organism is hence associated with its own unique DNA and analyzing the DNA sequences will help us to understand their genes, diseases, complexities and functionalities.

Improvements in DNA sequencing technology leads to the heavily reading cost of DNA sequences also the amount of data to be read is vastly and rapidly increasing. How-ever, an increased dataset needs to be summarized and analysed to better understand the rich content available in genes. Modern machine learning methods give us the opportunity to better understand DNA and to verify various relations and identify them. Machine learning approaches are always fed on large datasets available. Increasing growth of data regarding DNA sequences, hence makes it easy for any algorithm to predict the required results.

In bioinformatics, k-mers are substrings of a length k contained within a biological sequence. They are primarily used within the context of computational genomics and sequence analysis, in which *k*-mers are composed of nucleotides (A, T, C, and G), *k*-mers are implemented upon to assemble DNA sequences.

Multinomial Naive Bayes, where the features are assumed to be generated from a simple multinomial distribution that describes the probability of observing counts among several categories, and thus it becomes most appropriate for methods that represent counts or count rates. In our model, multinomial naive Bayes is used for DNA sequence classification and to predict the related class values count, where the DNA sub-units are created as the bag of words using count vectorizer for feature extraction.

In our study, we consider three different datasets namely of human, dog and a chimpanzee based on protein coding sequences of their respective DNA sequence. Next, we analyse them by performing k-mer encoding and applying multinomial naïve bayes under NLP, it is further classified using scikit-learn to identify their gene similarity and predict possible relation based on machine learning approach. Our dataset contains two columns as sequence and class where class is a pre-defined integer value based on the protein coding sequence.

Literature Review

In “Classification of DNA sequences with k-mers based vector representations” paper various approaches for DNA sequence splitting into required length of vectors is performed. Four different approaches have been considered by Umit Murat which are- one-hot based with random dictionary, one-hot based with default dictionary, voss and dna2vec for vector representation, deep learning technique approach is implemented using Convolutional Neural Network (CNN) for sequence classification. For example, a sequence “AGTCACTGACG” at length 11 can be represented with a 3-mers collection as {AGT, GTC, TCA, CAC, ACT, CTG, TGA, GAC, ACG}. Each subsequent k-mer overlaps on k-1 nucleotide when stride window equals 1[1]. The k-mers, instead of the whole gene sequence, are numerically encoded into vector form and each vector representation of a k-mer in a sequence is concatenated to form a 2D numerical representation of gene sequence. Use of ANN classifier to predict the DNA sequences promoter to find out their performances. Similarly in another paper they proposed a new hybrid learning system which is used to recognize the promoters in the DNA that involve the binding of RNA to initiate the process of transcription. A promoter who frequently appears before its related gene in DNA sequence relates the expression of genes which is for identifying genes that are based on their DNA sequences.

Further, CNN model basically includes convolutional layers, pooling layers, and fully connected layers. Convolutional layers convolve raw data using filters or kernels. Each convolutional layer tries to extract a higher-level feature than the previous layer. An activation function is applied for growth of non-linearity for every convolution layer under CNN. Pooling layers simply follow every convolution layer for down-sampling of features to prevent overfitting. Connected layers come after the last pooling layer. Further, features are combined through the fully connected layers, creating a classification model.

In the paper mentioned above, various representation methods are simply termed as- One-hot vector is a bit vector that has a single dimension with one “1” and the rest with “0”. The vector length depends on the k value in k-mer and the possible number of nucleotides(n). The curse of dimensionality is a problem for the one-hot vector representation of k-mers. Voss representation is a simple and widely used mapping scheme for DNA sequences. Dna2vec was inspired by word2vec it is trained by estimating the set of adjacent k-mers surrounding the targeted k-mer.

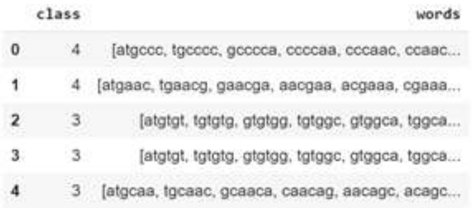
Methodology

Any Genomic sequence**,** protein structure, gene expression, and gene regulatory functions are some of the application areas described. Since the work relates to processing vast amounts of incomplete biological data, we can provide the learning ability of the Machine learning techniques to solve these kinds of problems. The machine learning techniques train the model to classify the genes data. They solve the problem that occur in the biological areas, there is a need for modern techniques which handles the genes data. There are many machine learning methods which are used for identification, selection, prediction, recognition and in classification of the DNA Sequences.

A. k-mer encoding

Any Long DNA sequences are expressed as a collection of shorter sub-sequences at length k and each of these sub sequences is called k-mer. K-mers are equivalent to words in natural language processing (NLP) and it helps to understand the DNA sequences and simplifies the computation for analysis.

In our model k-mer encoding is used to generate hexamer “words” which is arbitrary and word length can be tuned to suit the possible situation. The length of word and overlapping is to be determined empirically for any application considered. In genomics, we identify these types of manipulations as “k-mer counting” or counting the occurrences of each possible k-mer sequence. There are specialized tools for this, but the natural language processing tools of python makes it super easy. Later, we **define a function (get_Kmers) to collect all possible overlapping k-mers of specified length from any sequence string.** Later we convert our data sequences into repeating k-mers of length six. Similarly, we perform the repeated actions for each species DNA sequences of our dataset we have using our get_Kmers function.



	class	words
0	4	[atgccc, tgcccc, gcccca, ccccaa, cccaac, ccaac...
1	4	[atgaac, tgaacg, gaacga, aacgaa, acgaaa, cgaaa...
2	3	[atgtgt, tgtgtg, gtgtgg, tgtggc, gtggca, tggca...
3	3	[atgtgt, tgtgtg, gtgtgg, tgtggc, gtggca, tggca...
4	3	[atgcaa, tgcaac, gcaaca, caacag, aacagc, acagc...

Fig. 1. Representation of DNA sequence after performing k-mer encoding

B. Scikit-learn approach

Any As we are going to implement scikit-learn natural language processing tools to do the k-mer counting, for which we need to convert the lists of k-mers of each gene into string sentences of words so that the count vectorizer can easily deal with it. We shall also create a variable to assign the class labels of our dataset for individual DNA sequence. The same steps are recursively repeated for other datasets as well. Various activities performed during scikit-learn implementation are-

- The coding region recognition & gene identification process
- Sequence classification & feature extraction to be identified.

C. Multinomial Naive Bayes classifier

Any Multinomial Naive Bayes classifier is based on Bayes theorem and an attribute independence assumption. Despite its simplicity it can often outperform more sophisticated classification methods [3]. The DNA Replication Protein (DNARP) prediction is a typical binary classification problem. Let F = {f1, f2, ⋯, fn} be a feature vector for a given protein sequence. The targeted class set is defined as C = {c1, c2}, where c1 denotes the sample that is predicted as a DNARP and c2 denotes non-DNARP. The Multinominal Naïve Bayes classifier is to find the targeted class with maximum probability given the feature set F, which can be described as follows: For the binary classification, based on Bayes theorem, the posterior probability is defined as Equation (8).

Assume that each feature fi is conditionally statistical independent of every other feature fj, then Equation (8) is as Equation (9).

$$\frac{P(c_1|F)}{P(c_2|F)} = \frac{P(c_1)}{P(c_2)} \frac{\prod_{i=1}^{n} P_i(f_i|c_1)}{\prod_{i=1}^{n} P_i(f_i|c_2)}. \qquad (9)$$

$$\begin{cases} P(c_1|F) = \frac{P(F|c_1)P(c_1)}{P(F)} \\ P(c_2|F) = \frac{P(F|c_2)P(c_2)}{P(F)} \end{cases} \Rightarrow \frac{P(c_1|F)}{P(c_2|F)} = \frac{P(F|c_1)P(c_1)}{P(F|c_2)P(c_2)}. \qquad (8)$$

Thus, for a given protein sample P, if P (c1|F) P (c2|F) > 1, P belongs to class c1 otherwise belongs to class c2.

Hence for our model a multinomial naive Bayes classifier is created using naïve bayes library. Some parameter tuning is done well in hand to find out the n-gram size of- 4, which is also reflected in the countvectorizer instance, and an alpha parameter was determined by grid search, which resulted a model alpha of (α =0.1) that did the work best.

Experiments and Results

A. Datasets

To compare the performance of the learning model with different sequence representations three datasets are used (Table 1). The first two data sets are dog and chimpanzee datasets, which are the datasets collected from UC Irvine machine learning repository. For the third data set, human dataset is collected from National Centre for Biotechnology Information (NCBI) gene sequences dataset repository.

The three datasets contain the information of DNA sequences that encode proteins, they are termed as the coding sequences of complete DNA strand that which works for required protein generation in humans. The reason for collecting coding sequence is that it helps us to easily classify our DNA sequences and identify the gene similarities as required while comparison with other DNA sequences.

After divergence of their ancestor lines, human and chimpanzee genes gone through multiple changes including single nucleotide substitutions, deletions and duplications of DNA strands of varying size, insertion of transposable elements and chromosomal rearrangements. Human specific single nucleotide alterations constitute of 1.23% of human DNA, whereas added deletions and insertions cover ~3% of our gene. Moreover, much higher proportion is made by differential chromosomal inversions and translocations comprising several mega base-long regions or even whole chromosomes.

TABLE 1. Information about included datasets and their column descriptions as considered in our machine learning model

Dataset	**Sequence length**	**Class count**
DNA sequences of Dog	820	0-6
DNA sequences of Chimpanzee	1682	0-6
DNA sequences of Human	4380	0-6

B. Model configuration and Evaluation

The collected datasets information is in the form of large continuous strings of DNA sub-units (A, T, C, G), to reduce the composition and make the machine learn easily we have implemented k-mer encoding analysis that splits and forms the substrings of required length. They are represented in the form of vectors or lists, later to prepare the model for NLP processing we further convert the vectors to basic string format of matching lengths. Each dataset is split as training and test. 80% of the samples in a dataset are used for training and the remaining 20% for testing.

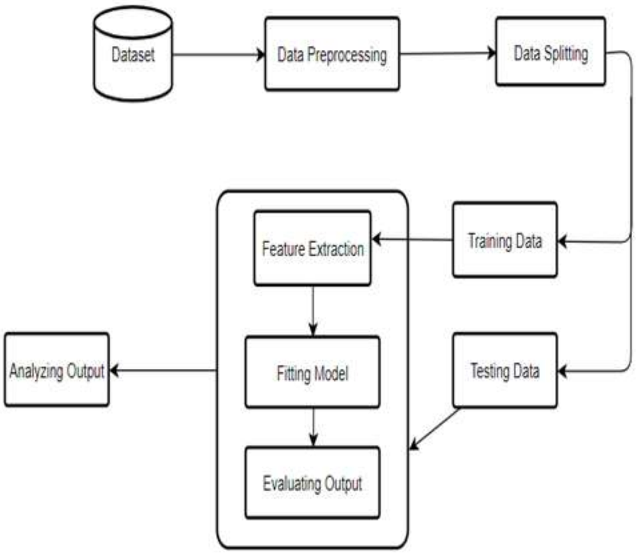


Fig. 2. Architectural representation of model

For feature extraction and learning, an NLP model tool is constructed with Python using scikit-learn and evaluations are made on colab, with scikit-learn DNA sequence features get extracted as simply as text feature extraction, thanks to the advanced developments of python for data science (machine learning) as evaluations and work of implementation become easy. We perform it by creating the bag of words using NLP CountVectorizer technique, later we do the required fitting transformations on individual datasets and identify the shapes of every dataset to know more information about refined sequence lengths and column values. Our model is built using multinomial naïve bayes classifier, where we fit our training dataset using our calculated attributes xtrain and ytrain to classify the data and later predict our test dataset values by fitting the model prediction with x_test and y_test attributes.

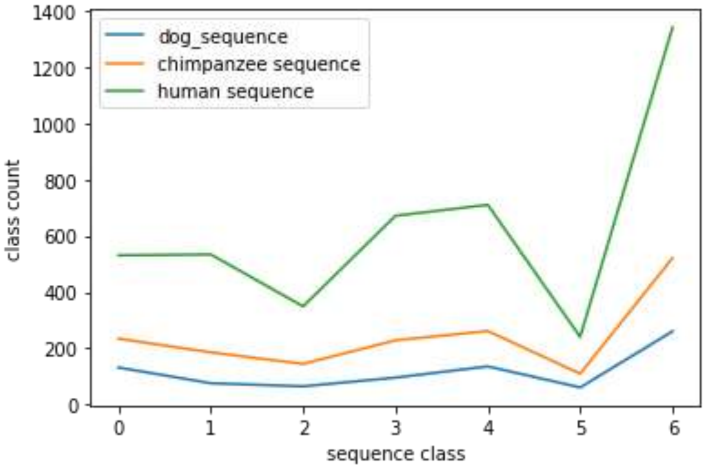


Fig. 3. Represents class information of the collected DNA sequences.

The above graph makes it clear that human coding DNA sequences are of high range and most of them belong to predefined class with the value of six. The next highest DNA sequence range is of chimpanzee with the class count range between 200-400, also they are mostly under the class value of six itself, similarly dog DNA sequence is identified under the range of 0-200 class count values, the least DNA sequence range due to its least available data then compared to the other two datasets information.

C. Results

The Three varying accuracy rates are predicted for individual dataset, also the close relation between predicted values leads to the identification of similarity of genes present in them, in our model the accuracy rate for human DNA coding sequence is given as 98.40%, for chimpanzee DNA coding sequence the accuracy rate is predicted as 99.34% and for dog DNA coding sequences it is stated as 92.56%. Hence, based on the predicted accuracy rates, we can imply that human and chimpanzee are very closely related to each other based on their DNA sequence analysis and classification. We have performed the above calculations by taking various classification and sequence analysis algorithms like k-mer analysis, scikit-learn, Multinomial Naïve Bayes (under NLP approach).

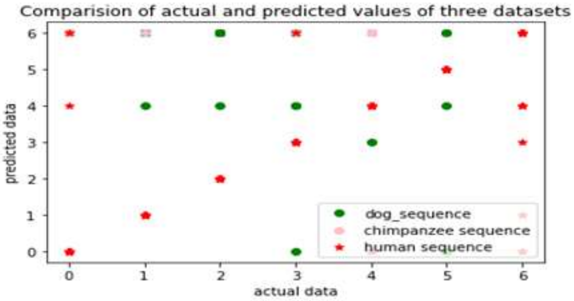


Fig. 4. Predicting accuracy based on training and testing dataset values of DNA sequences

The above graph of comparison of actual and predicted values of three datasets clearly visualizes that as both chimpanzee and human DNA sequences are closely relative to each other, most of the points in the scatter plot got overlapped and even some of the points that are depicted as star marker are colored in pink with which we can easily identify through the above graph itself that the coding sequences of both human and chimpanzee datasets are related in similar mannered ways.

TABLE 2. Approximate comparison of proposed work with other algorithms

Classifiers	**Accuracy (%)**
Support Vector Machine (SVM)	96.50
Artificial Neural Networks (ANN)	90.10
Adaboost	82.60
Logistic	78.82
Multinomial Naïve Bayes	99.32

In our work, we utilized the k-mer encoding and multinomial naïve bayes algorithms, hence we worked on similar datasets by functioning with both the models at once. Some of the other classifiers mentioned in the above table used different datasets. The above comparison table approximates other accuracies of different algorithms as calculated.

Also, we can see that the obtained results for our algorithms that we considered to build our model play a better role than the other existing algorithm based on the results as shown in table II, for better classification and analysis of DNA sequences.

Conclusion

In this research, coding sequence representation methods are used on three different datasets and classification performances are compared based on the predicted accuracy. The importance of the representation of the sequences related to the model performance is exhibited. A new classification methodology is proposed in this paper. As k-mer encoding analysis employed during the DNA sequence analysis and classification process, is a python library that is exclusively built for the DNA sequence analysis due to its large varying composition of strings, also our model predicts well with an accuracy rate of 98.40% and identifies gene similarities in best way possible.

At the same time, we can see that the application range of the classification method is not wide enough. And the DNA sequences feature extraction as well as the representation method need to be optimized. It could reflect the structure characteristics of the DNA sequence more precise and more comprehensive.

References

1. Umit Murat Akkaya, Habil Kalakan-Classification of DNA Sequences with k-mers based vector representations- vol. 9, no. 5, pp. 280–286, 2021

2. Pooja Dixit, Ghanshyam I. Prajapati- Machine Learning in Bioinformatics: A Novel Approach for DNA Sequencing-DOI 10.1109/ACCT.2015.73/IEEE

3. Qingda Zhou, Qingshan Jiang- A New Method for Classification in DNA Sequence-978-1-4244-9718-8/11/$26.00/IEEE

4. S. Alhalem, et al.DNA Sequences Classification with Deep Learning: A Survey, Menoufia Journal of Electronic Engineering Research, vol. 30, 10.21608/mjeer.2021.146090, 2020.

5. J. D. Washburn et al., "Evolutionarily informed deep learning methods for predicting relative transcript abundance from DNA sequence," Proceedings of the National Academy of Sciences. vol. 116, pp. 5542– 5549, 2019.

6. H. Zhang, C. Hung, M. Liu, X. Hu, and Y. Lin, "NCNet: Deep Learning Network Models for Predicting Function of Non-coding DNA," Frontiers in genetics, vol. 10, 2019.

7. N. G. Nguyen, V. A. Tran, D. L. Ngo et al., "DNA sequence classification by convolutional neural network," Journal of Biomedical Science and Engineering, vol. 9, no. 5, pp. 280–286, 2016.

8. D. B. Waz, "Four-component spectral representation of DNA sequences," Journal of Mathematical Chemistry, vol. 47, pp.41–51.

9. R. F. Voss, "Evolution of long-range fractal correlations and 1/f noise in DNA base sequences," Physical review letters, vol.68, pp. 3805- 3808.

10. T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," ICLR, 2013.

11. Dewey FE, Pan S, Wheeler MT, Quake SR, Ashley EA. DNA sequencing clinical applications of new DNA sequencing technologies. Circulation. 2012; 125:931–44.

12. H. K. Kwan, and S. B. Arniker, Numerical representation of DNA sequences, 2009 IEEE International Conference on Electro/Information Technology, pp. 307-310, DOI: 10.1109/EIT.2009.5189632, 2009.

13. Y. Li, Y. Sun, J.C. Hines, D.S. Ray, Identification of New Kinetoplast DNA Replication Proteins in Trypanosomatids Based on Predicted Sphase Expression and Mitochondrial Targeting, Eukaryot Cell, vol. 6, 2007, pp 2303-2310.

14. A.G. de Brevern, New assessment of a structural alphabet, In Silico Biol, vol. 5, 2005, pp 283-289.