VERTICALLY INTEGRATED

PROJECT (VIP) REPORT

NEUROMORPHIC SOLUTIONS TO HGIH DIMENSIONAL DNA SUBSEQUENCE MATCHING VIA ALIGNMENT DATA

*Ethan Jacob Moyer*

ENGR 370: Applications, Algorithms, & Architecture for

Neuromorphic Computing

*Dr. Anup Kumar Das*

Drexel University

4 May 2020

**INTRODUCTION**

BACKROUND:

Sequence alignment of biological molecules, such as deoxyribonucleic acid (DNA), focuses on identifying regions that are of high similarity due to common factors. Typically, these factors are derived from functional, structural, or evolutionary relationships shared between the sequences being aligned [1]. More specifically, the field of bioinformatics is concerned with sequence alignment algorithms as longer sequences become increasingly difficult to align by hand. One such algorithm is Pairwise Sequence Alignment (PSA). In this algorithm, each element in a query sequence (the sequence being aligned) is compared each element of a reference sequence (the sequence to which the query is being aligned). Although it has mainly two different dynamic programming implementations, such as through the Needle-Wunsch (NW) [2] and Smith-Waterman (SW) [3] techniques, the general theme of alignment remains constant. A positive score is assigned to matching elements and a negative score is assigned to nonmatching elements. In this way, the optimal alignment between a query and a reference is identified by the highest scoring segment [4].

In the case of this research, DNA subsequence matching is defined as the matching of short sequences (of 150 nucleotides or less) based on percent similarity of local sequence alignment scores. Since the SW technique focuses primarily on local alignment while the NW technique focuses on global alignment, the former will be used as the bases for alignment scores.

[4] Salamat, S. and Rosing, T., 2020. FPGA Acceleration Of Sequence Alignment: A Survey. UC San Diego: Department of Computer Science and Engineering, pp.2,3.

[1] Mount DM. (2004). Bioinformatics: Sequence and Genome Analysis (2nd ed.). Cold Spring Harbor Laboratory Press: Cold Spring Harbor, NY.

[2] S. B. Needleman and C. D. Wunsch, “A general method applicable to the search for similarities in the amino acid sequence of two proteins,” Journal of molecular biology, vol. 48, no. 3, pp. 443–453, 1970.

[3] T. F. Smith, M. S. Waterman et al., “Identification of common molecular subsequences,” Journal of molecular biology, vol. 147, no. 1, pp. 195–197, 1981.

ANALOGY:

An analogous problem is asking to classify similar words across repeats of the same sentence. This might be based on high dimensional quantifiers, such as properties of being a noun, verb, conjunction, etc., or context indicators that the English language takes for granted, such as verb-noun agreement, implied antecedents, etc. Using context cues, a computer could generate disjoint data for different words. For instance, one sentence copy would be classified solely based on a specific class of closely related nouns, such as the noun ‘CAR.’ An algorithm might place different subtypes of cars into that word class, like ‘TRUCK’ or ‘VAN.’ A separate sentence copy would be classified based on another English quantifier or context indicator. In this way, sentence data would be organized into discrete classes, each containing disjoint information about its respective word group. As one could imagine, the task of classifying words is trivial for short sentences and when there are relatively few classes. The task becomes more intensive once both of these attributes increase.

Such an increase is observed when applying this problem structure to DNA subsequence matching. It is similar to word classification problem for spoken languages in that both problems are based on text classification. One difference between the two is the convention by which text is interpreted. In English, sentences are composed of words, each of which is comprised of an arrangement of 26 unique letters. In the genetic language, these sequences are composed of a composition of four unique bases. This distinction between *an arrangement* in a language like English and *a composition* in the genetic language is imperative of the increased complexity when applying word classification to DNA subsequence classification.

An arrangement is operationally defined as grouping of characters where only specific permutations carry meaning. For example, the word ‘ACR’ has little meaning compared to ‘CAR’ and ‘ARC.’ A composition is operationally defined as a grouping of characters where generally all permutations may carry meaning. In the case of DNA nucleotides, ‘ATCAGTC’ caries as much meaning as ‘TTCCAA.’ As displayed by the provided examples, the concept of an arrangement pairs nicely with English words and the concept of a composition pairs nicely with the genetic language. In most cases, the length of subsequences is much greater than the length of words. As previously mentioned, it becomes increasingly difficult to classify subsequences as the length of each subsequence and the number of possible classes increases.

For these reasons, DNA subsequence matching is much more complex than language word classification; nonetheless, high dimensional word classification serves as a perfect analogy from DNA subsequence matching can be built.

Moreover, the need for the neural network in the analogous word classifying problem is strikingly similar to our problem in DNA subsequence matching. If the number of quantifiers is low, there really isn’t much need for a network to do prediction. However, as these quantifiers increases increase, the complexity of classifying words (or in our case, subsequences) increases exponentially. Similar to how my MATLAB-produced files containing disjoint information about different subsequence matches, so too could a ‘dictionary’ file contain disjoint information about words in different contexts for a variety of quantifiers.

One aspect that the word classification model does not directly address is the ability to determine (and predict) where matches occur in a sentence. This is of course more critical in DNA subsequence matching as the position of subsequence matches, as opposed to the quality of having a match, is of much greater importance. Regardless, this will be achieved through careful curation of the data and the use of Long Short-Term Memory (LSTM) Recurrent Neural Networks (RNNs).

RATIONAL:



ALGORITHM:

Through the use of the MATLAB Bioinformatics Toolbox, a modified SW alignment technique is used through the function *localalign* [q].

[q] Barton, G. (1993). An efficient algorithm to locate all locally optimal alignments between two sequences allowing for gaps. CABIOS 9, 729–734.

**DATA**

SELECTION:

Accession numbers of relativity short genes ranging from 1 to 5 kilobases (kb) were selected under the family of Mycobacterium from Nucleotide on GenBank [j].

MATLAB getgenbank function

STRUCTURE:

**Figure 1:** Data Structure

Query (Sq)

|  |  |
| --- | --- |
| Reference (Sr) | Logical (L) |
| k0 | m0 |
| k1 | m1 |
| . | . |
| . | . |
| . | . |
| ki | mi |
| . | . |
| . | . |
| . | . |
| kn-1 | mn-1 |
| kn | mn |

where ki represents an individual k-mer segment of the reference sequence Sr and mi represents the logical value of a match of the query sequence Sq against ki.

PIPELINE:

**Figure 1:** Data Pipeline

S0, S1, … , Sj, … , Sn-1, Sn

Model trained on specific set at 95% similarity

L0, L1, … , Lj, … , Ln-1, Ln

where Sj represents a list of k-mer fragments of a unique gene sequence and Lj represents a list of logical values corresponding to the occurrences of those k-mer fragments within the specific set.

CURATION:

This data curation along with other data analysis were carried out with several MATLAB functions built to efficiently separate and analyze the data. These functions along with their documentation are included on the project’s GitHub repository.

FEATURE ANALYSIS:

**MODEL**

PROPOSED MODEL:

<https://github.com/philippmuench/dna_lstm>

Input Embedding Layer

Output Dense Layer

Time Distributed LSTM Layer

Dropout Layer

Dropout Layer

Time Distributed LSTM Layer

**MEHTODS**

Long-Short Term Memory (LSTM) Recurrent Neural Network (RNN)

1. Background
2. Results
3. Alternatives

**DISSCUSION**

FALSE POSITIVES/FALSE NEGATIVES

COMPARISON

CONCLUSION

FUTURE RESEARCH

**REFERENCES**

Anderson, J Christopher et al. “BglBricks: A flexible standard for biological part assembly.” Journal of biological engineering vol. 4,1 1. 20 Jan. 2010, doi:10.1186/1754-1611-4-1. Accessed Web. 24 Oct 2019.

#### Breiman L., Cutler A."Random Forests - Classification Description." *Stat.berkeley.edu*. Accessed Web. 28 Feb. 2020.

Cooper, Geoffrey. "DNA Replication." *Sinauer Associates* (2000). Accessed Web. 27 Feb. 2020.

"CRISPR Timeline." *Broad Institute*. 2015. Accessed Web. 3 Mar. 2020.

Cutler A., and J. R. Stevens. 2006. Random forests for microarrays. Methods in Enzymology 411:422-432. Accessed Web. 28 February 2020.

Deshpande, Adit. "A Beginner's Guide To Understanding Convolutional Neural Networks." Adeshpande3.github. 2016. Accessed Web. 29 Feb. 2020.

Devetyarov D., Nouretdinov I. (2010) Prediction with Confidence Based on a Random Forest Classifier. In: Papadopoulos H., Andreou A.S., Bramer M. (eds) Artificial Intelligence Applications and Innovations. AIAI 2010. IFIP Advances in Information and Communication Technology, vol 339. Springer, Berlin, Heidelberg. Accessed Web. 28 February 2020.

Goodfellow, Ian, et al. Deep Learning. The MIT Press, 2017. p. 326

Hughes, Randall A, and Andrew D Ellington. “Synthetic DNA Synthesis and Assembly: Putting the Synthetic in Synthetic Biology.” Cold Spring Harbor perspectives in biology vol. 9,1 a023812. 3 Jan. 2017, doi:10.1101/cshperspect.a023812. Accessed Web. 24 Oct 2019.

"Keras Conv1d: Working With 1D Convolutional Neural Networks In Keras - Missinglink.Ai." *MissingLink.ai.* Accessed Web. 2 Mar. 2020.

New England Biolabs. “Restriction Endonucleases.” NEB, www.neb.com/products/restriction-endonucleases. Accessed Web. 29 Oct 2019.

Nucleotide [Internet]. Bethesda (MD): National Library of Medicine (US), National Center for Biotechnology Information; 2004 – Accessed Web. 24 Jan. 2020. Available from: https://www.ncbi.nlm.nih.gov/nuccore/

Support Vector Machines: Data Analysis, Machine Learning and Applications, edited by Brandon H. Boyle, Nova Science Publishers, Incorporated, 2011. ProQuest Ebook Central, https://ebookcentral-proquest-com.ezproxy2.library.drexel.edu/lib/drexel-ebooks/detail.action?docID=3021500. Accessed Web. 27 February 2020.

#### Scikit-Learn SVM. "1.4. Support Vector Machines — Scikit-Learn 0.22.1 Documentation." *Scikit-learn.org*. Accessed Web. 27 Feb. 2020.

Scikit-Learn Random Forest."3.2.4.3.1. Sklearn.Ensemble.Randomforestclassifier — Scikit-Learn 0.22.1 Documentation." Scikit-learn.org. Accessed Web. 28 Feb. 2020.

Tang, Nicholas, Siying Ma, and Jingdong Tian. "New Tools For Cost-Effective DNA Synthesis." Synthetic Biology (2013): 3-21. Accessed Web. 1 Mar. 2020.

"The Human Genome Project." Genome.gov. *National Human Genome Research Institute*. 2020. Accessed Web. 3 Mar. 2020.