Name: Dave Mannion
Course: UCSD Machine Learning & Engineering
Date: October 23, 2022

Bootcamp Capstone Report

Gene Family Prediction & ML

Machine Learning and DNA Sequence Classification

Dave Mannion

Contents

Gene Family	Count	Class Label
G protein coupled receptors	531	1
Tyrosine kinase	534	2
Tyrosine phospotase	349	3
Synthetase	632	4
Synthase	711	5
Ion channel	240	6
Transcription factor	1341	7

Table 1: Gene Families represented by Human DNA coding sequences.

1 Introduction

Here we examine NLP algorithms to develop a Machine Learning model for making predictions on sequences of Nucleic Acid text. Specifically, we train a Multinomial Naïve Bayes classifier on Human DNA Coding Sequences to predict Gene Families. In addition to Human sequences, predictive testing was also performed on Chimpanzee and Dog DNA coding sequences. Table 1 above shows Gene Families in the dataset.

1.1 Motivation

This Capstone project was selected to provide an opportunity to explore the application of NLP methods to biological text-based sequence data.

1.2 Exploratory Work

Prior to selecting this Capstone project, recent work in ML applied to biological sequences was surveyed in order to identify the feasability and scope of a project appropriate for this course. Method feasability focused on LDA, NB and LSTM models. Some of this work is linked to Jupyter notebooks or original papers and includes protein classification[1], Sapien/Neandertal sequence classification[2], Neandertal DNA introgression[3], CRISPR gRNA design tools[4], Bacterial Classification with RNNs[5], Bacteria taxonomic

[1] https://github.com/ai-dave/ml-gene-pub/blob/main/protein-classification.ipynb
[2] https://github.com/ai-dave/ml-gene-pub/blob/main/Deep_Learning_on_Ancient_DNA.ipynb
[3] https://github.com/ai-dave/ml-gene-pub/blob/main/Deep_Learning_Neanderthal_Introgression.ipynb
[4] https://pubmed.ncbi.nlm.nih.gov/31533522/
[5] https://github.com/lelugom/wgs_classifier

classification[6], Ribosmal RNA analysis[7,8,9,10,11], Viral and Single-celled Eukaryote Taxonomy classification[12,13], Wolf/Dhole classification[14,15,16].

1.3 Capstone Project Work

- The Github repo: https://github.com/ai-dave/capstone
- This document: https://github.com/ai-dave/capstone/blob/main/report/Capstone.pdf
- Jupyter notebook: https://github.com/ai-dave/capstone/blob/main/gene-family-human-chimp-dog.ipynb
- Deployed model: http://45.33.108.171/home

1.4 Curriculum

- mec-mini-projects repo: https://github.com/ai-dave/mec-mini-projects

[6] https://www.researchgate.net/publication/348432006_Bacteria_taxonomic_classification_using_Machine_learning_models
[7] https://github.com/ai-dave/ml-gene-pub/blob/main/gnb-rdp.ipynb
[8] https://github.com/ai-dave/ml-gene-pub/blob/main/lda-compare-silva.ipynb
[9] https://github.com/ai-dave/ml-gene-pub/blob/main/lda-rdp.ipynb
[10] https://github.com/ai-dave/ml-gene-pub/blob/main/lstm-random-noise.ipynb
[11] https://github.com/ai-dave/ml-gene-pub/blob/main/lstm-rdp.ipynb
[12] https://github.com/ai-dave/ml-gene-pub/blob/main/ncbi_lstm_poc.ipynb
[13] https://github.com/ai-dave/ml-gene-pub/blob/main/Eukaryota-Amoebozoa-Discosea.ipynb
[14] https://github.com/ai-dave/ml-gene-pub/blob/main/Dhole-Wolfe-LDA.ipynb
[15] https://github.com/ai-dave/ml-gene-pub/blob/main/Dhole-Wolfe-LSTM.ipynb
[16] https://github.com/ai-dave/ml-gene-pub/blob/main/Dhole-Wolfe.ipynb

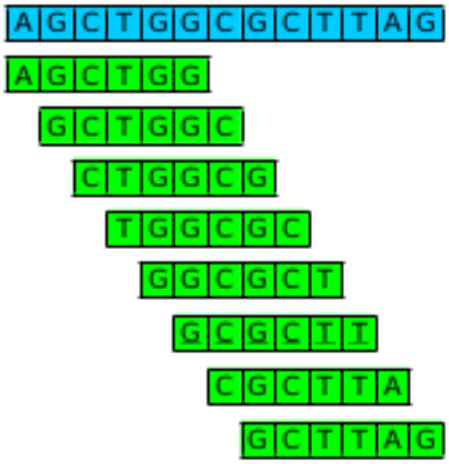

Figure 1: 6-mer "*word*" list production from original sequence.

2 Data

2.1 Source

The data for this project came from a Kaggle notebook, *DNA Sequencing with Machine Learning*[17]. DNA Coding Sequences are mostly comprised of exons and, in this corpus, range in length from ∼1000 to ∼3,500 bases.

2.2 Labeling

Each sequence in the corpus of sequence data was annotated with one of seven Gene Families. The Gene Family annotation was translated into an integer class label c in $\{1, 2, 3, 4, 5, 6, 7\}$

2.3 Preprocessing: Sequence to K-mer List

The nucleotide bases comprise an *alphabet* of the four letters b in $\{A, G, C, T\}$. A *K-mer* is a k-letter word comprised from *letters* of this *alphabet*. Here we let k=6. The *dictionary* contains up to 4^6 *words*, each 6 letters in length.[18]

Figure 1 above depicts 6bp "*words*" derived from a sequence from the original corpus. From this point on, all vectorization is based on a corpus of lists of *6-mer* "*words*" rather than the original sequences.

[17] https://www.kaggle.com/code/nageshsingh/demystify-dna-sequencing-with-machine-learning/data
[18] Dictionary size calculated as $alphabetSize^{wordLength} \Rightarrow 4^6 \Rightarrow$ 4,096 words.

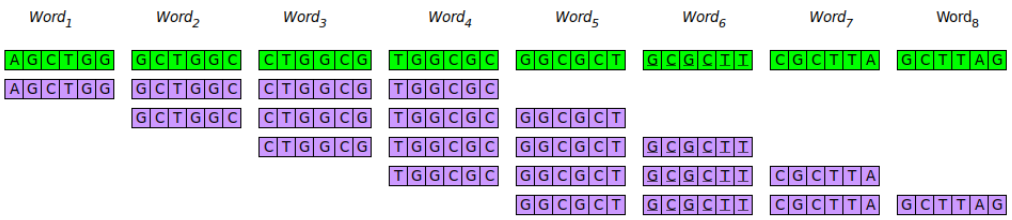

Figure 2: 4-word n-gram (*4-gram*) phrases generated from 6-mer words representing the original nucleic acid sequence.

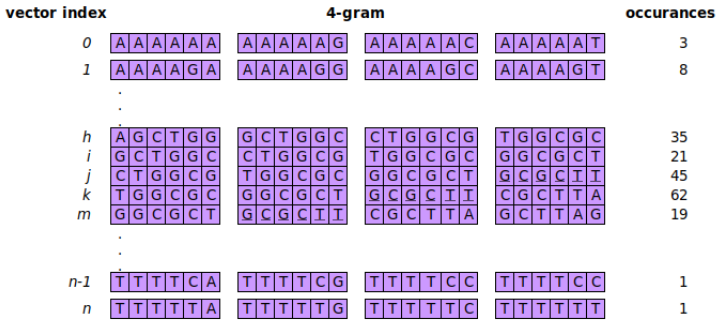

Figure 3: Counts of (*4-gram*) phrases generated from 6-mer words comprising a Document Vector of a single nucleic acid sequence.

3 Feature Engineering

After converting contiguous nucleotide sequences into 6-mer words, *Bag of N-grams* model was used to produce document vectors.

3.1 n-grams

We use *CountVectorizer*[19] to convert k-merized sequences to a matrix of 4-gram counts. Figure 2 above shows 4-gram phrases generated from a series of 6-mers of the original sequence.

3.2 Frequency Vectorization

The frequencies of ordered 4-grams from each single nucleic acid k-merized sequence will produce a histogram based on the occurance of the 4-gram. Figure 3 above shows a Count Vector formed from 4-gram frequencies. Each Document Vector is a representation of a single nucleic acid sequence.

[19] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

3.3 Implementation

Python code implementing *CountVectorizer* and *pickle*:

```
cv = CountVectorizer(ngram_range=(4,4), lowercase=False) # 4-word n-gram
X = cv.fit_transform(human_texts)
X_chimp = cv.transform(chimp_texts)
X_dog = cv.transform(dog_texts)

pickle.dump(cv, open('CountVectorizer-human.pkl','wb'))
```

Predicted Actual	1	2	3	4	5	6	7
1	113	0	0	0	1	0	10
2	0	82	0	0	0	0	3
3	0	0	65	0	0	0	4
4	0	0	0	125	2	0	1
5	3	0	0	0	121	0	0
6	1	0	0	0	0	38	0
7	1	0	0	0	1	0	233

Table 2: Confusion matrix for predictions on human test DNA sequence.

accuracy	0.966
precision	0.968
recall	0.966
f1	0.966

Table 3: Accuracy, precision, recall

4 Modeling

4.1 Training

We are building a Multinomial Naive Bayes classifier model using *MultinomialNB*[20] and training it on Human DNA sequences *k-merized* and embedded as 4-gram frequency histogram vectors:

```
X_train, X_test, y_train, y_test = train_test_split(X, y_human, test_size=0.2, random_state=42)


classifier = MultinomialNB(alpha=0.1)
classifier.fit(X_train, y_train)
pickle.dump(classifier, open('model-human.pkl','wb'))
```

4.2 Performance

Table 2 and Table 3 above show Confusion Matrix and accuracy/precision/recall values on the Test set of Human DNA.

[20] https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html

Predicted Actual	1	2	3	4	5	6	7
1	224	0	0	0	3	0	6
2	0	182	0	0	0	0	3
3	0	0	137	0	0	0	7
4	0	0	0	222	3	0	7
5	2	0	0	0	259	0	0
6	1	0	0	0	0	108	0
7	0	0	0	0	1	0	521

Table 4: Confusion matrix for predictions on chimpanzee DNA sequences.

accuracy	0.984
precision	0.984
recall	0.984
f1	0.984

Table 5: Accuracy, precision, recall

5 Results: Chimpanzee DNA

Table 4 and Table 5 above show results of testing chimpanzee DNA Coding sequences against a model trained on Human DNA for the seven Gene Families:

Predicted Actual	1	2	3	4	5	6	7
1	119	0	0	0	1	0	7
2	0	60	0	0	0	0	14
3	0	0	45	0	1	0	16
4	0	0	0	77	2	0	16
5	6	0	0	1	117	0	7
6	3	0	0	0	1	51	5
7	0	0	0	0	0	0	254

Table 6: Confusion matrix for predictions on dog DNA sequences.

accuracy	0.900
precision	0.916
recall	0.900
f1	0.900

Table 7: Accuracy, precision, recall

6 Results: Dog DNA

Table 6 and Table 7 above show results of testing dog DNA Coding sequences against a model trained on Human DNA for the seven Gene Families.

7 Deployment

7.1 Architecture

- Server HTTP framework: Flask
- Deployed platform: Docker

7.2 Resources

- Deploy from: `https://github.com/ai-dave/capstone/tree/main/deploy`
- Deployed model: `http://45.33.108.171/home`