DNA Classification with HMMs and ANNs Project Proposal

Katherine Perry

Background, goals, and significance:

In my project, I will investigate the classification and prediction of DNA sequences and structure, as they relate to Hidden Markov Models. The two strategies for gene prediction are sequence similarity searches and signal-based searches. The current methodology addresses the issue using a Generalized Hidden Markov Model called AUGUSTUS, and techniques such as Dynamic Programming and Neural Networks[6].

HMMs use a transition probability matrix and emission probability matrix as parameters. The inputs are sequences of DNA or other genetic material. The Markov Models can be trained with sequences that are already classified to augment the predictive ability of the parameters. Extrinsic evidence can be integrated into the models to improve accuracy, and it is sometimes comprised of database sequences that aid in comparative gene prediction.

One limitation is finding the balance between intrinsic and extrinsic evidence, in addition to evaluating the efficiency of methods of the integrated model. Additionally, the extrinsic evidence lacks consistency because it results from different initial queries and sequence alignment tools. The biggest limitation with HMMS is lack of knowledge about gene structures, especially for unsequenced genomes.

My approach is comparing the existing Hidden Markov Model, Neural Networks, and dynamic programming for speed, efficacy, and ability to use BLAST and multiple sequence alignment programs such as CLUSTAL as a tool for integrating extrinsic evidence. I will compare the different probabilistic methods for balancing evidence types. This may lead to a new method or analysis of combining BLAST with a GHMM, which is likely to be successful because both are well documented and successful on their own.

This project could lead to newly created techniques that allow for program crossover and complement. In addition, examining the outputs of the HMMs and NNs could provide insight into the benefits of certain amounts of noncoding DNA. Analyzing the accuracy and other metrics of the different machine learning algorithms on the same datasets will show which algorithms are best at nucleotide prediction and DNA classification. This would augment gene prediction and our knowledge of genetic variation and the genetic basis for diseases.

Datasets:

In terms of data, I will utilize the UCI Molecular Biology (Promoter Gene Sequences) Data Set as described in a Markov Model, kNN, SVM tutorial I found[1]. The University of California, Irvine has a machine learning repository with ample datasets of DNA sequences. This particular dataset has a list of 56 sequences of nucleotides that have been classified as promoter regions of DNA. A promoter is a region to which proteins bind to start the process of transcription. Each of the sequences are originally formatted as strings but are then manipulated into boolean data. For index i, there are 4 columns i_a, i_c, i_t, and i_g which represent which nucleotide is at that index (indicated by a value of 1). Also, there are class and instance labels for each row that need to be parsed.

Additionally, I will utilize the UCI Molecular Biology Splice-Junction Sequences dataset. The dataset has 3190 sequences of nucleotides that have been classified as splice junctions, which are boundary regions between introns and exons. The dataset is of a similar format as the previous with strings of sequences and also an indication for each sequence whether the region was an EI (exon-intron site) or IE (intron-exon site).

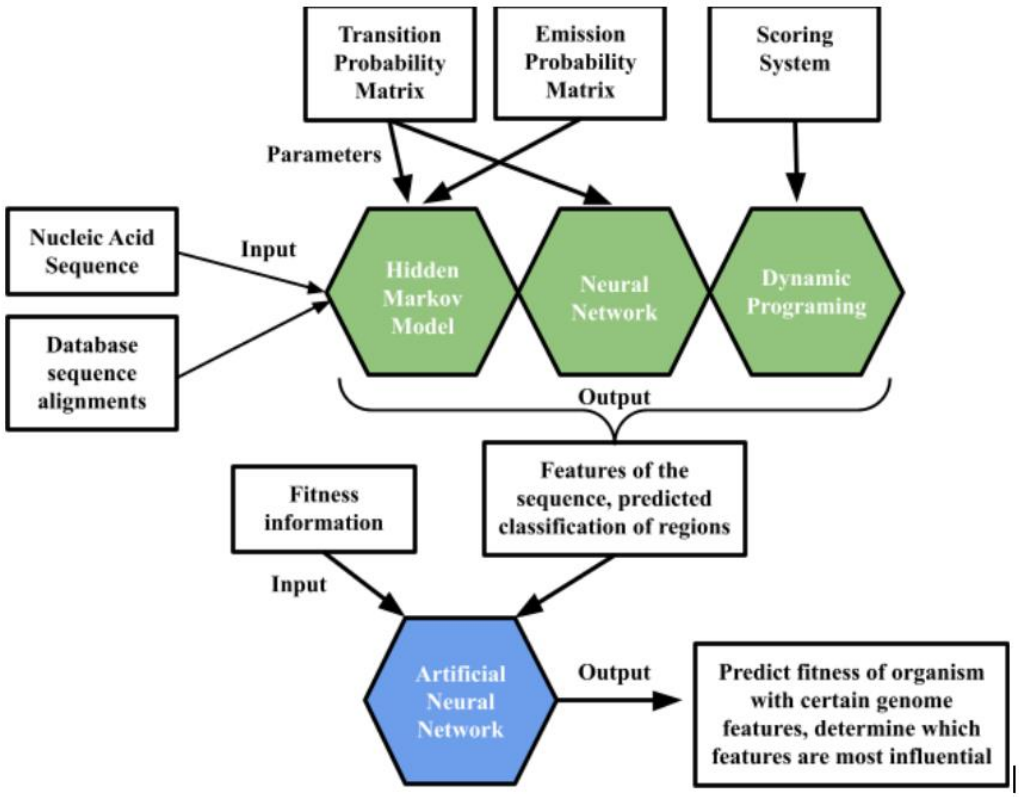
Finally, I have access to a lot of unused data generated by a member of the MSU Devolab about the genomes of digital organisms and how they evolve when certain instructions are replaced with NOPs. If I become comfortable with inputting new datasets to the code for the above algorithms, I could use a Hidden Markov Model to analyze the coding and noncoding sequences of that data. It would answer questions such as: what ratio of coding to noncoding DNA leads to the highest fitness? How do the differences in states/transitions of Hidden Markov Models and Neural Networks manifest in DNA sequence classification?

Computational Methods/Approaches:

The computational methods I will use are Hidden Markov Models and Neural Networks. These are both unsupervised machine learning algorithms, which means they don't rely on external input for training and evaluation. The tutorial for the Promoter Region dataset implemented a variety of machine learning and training algorithms, but not an HMM or NN. I will likely import and reference code from Neural Networks Demystified[7] in order to implement an Artificial Neural Network on the sequence datasets. I will import and reference code from the hmmlearn python library, which used to be part of scikit-learn in order to develop a Hidden Markov Model for the sequence datasets[4].

Software packages such as numpy, pandas, and sklearn will be crucial to my project. They provide methods for data manipulation, analysis, and baseline steps of machine learning. The Kaggle tutorial provided an example of how to use these libraries to load in a dataset and create suitable inputs for the algorithms.

One analytical method I will use is just a plain comparison of the accuracy metrics for the different models. This is a preliminary result that allows for an evaluation without statistical confirmation. However, I will then accumulate average values for multiple runs of each algorithm and use a statistical test, such as a t-test, to see if there is a significant difference between the mean accuracy of the two different algorithms run on the same dataset.



Evaluation Plan:

The library sklearn has a plethora of built-in metrics that allow evaluation of a model based on actual vs predicted training and testing data. The values of accuracy, precision (tp / (tp + fp)), and recall (tp / (tp + fn)) will enable me to evaluate how well the algorithm is tuned to the dataset and if it has any predictive power.

Also, I will observe how adjusting aspects of the HMM and NN, such as features, labels, sizes of hidden layers, number of states, and transition probabilities, affects the metrics and outputs of the models. For instance, there are (at least) two types of Hidden Markov Models – standard and convolved. The difference between the two is the number of hidden layers, one and two hidden layers for a total of two and three layers respectively[2].

To ensure that my model is biologically meaningful, I'll want to prevent the number of hidden layers and states from growing too large and introducing too much complexity. Initially, the models will be predicting the value of a nucleotide at a particular location in a sequence with a given classification. Expanding the models to be able to predict whether a sequence is or is not of a given classification (promoter, splice junction) would probably be more biologically important.

Potential Challenges/Alternative Approaches:

One potential challenge is the aptitude of the selected datasets for application of Hidden Markov Models and Neural Networks. I may have to do extensive data manipulation or reformatting to ensure the inputs to the algorithms are compatible. Examining the examples on the Github pages of both libraries will aid me in understanding what the acceptable parameter values and data structures are. I can additionally search for other datasets that may lend themselves better to my selected algorithms and libraries.

Another limitation of the UCI datasets is that they actually aren't that large. One had 56 sequences, while the other had 3,190. While potentially enough to train a decent model, those aren't very big numbers given the amount of DNA in the world and other databases. The size of the training data set has a significant impact on accuracy of the trained model[3]. This could also make it difficult to extrapolate the utility of the models for the real world and any dataset. Again, time-permitting, finding other datasets of the same classification (ex. Promoter or Splice-junction regions) would aid in extrapolation evaluation.

If I delve into the realm of the dataset of digital organism genomes, I face even greater challenges and assumptions. That data is of a completely different format than the previous datasets. Its "sequences" are actually just computer programs with a given sequence of instructions (instructions being functions like AND, OR, XOR, NOP, etc.). I would be making the assumption that these sequences are comparable to the Promoter and Splice-junction region data, given the difference in genome structure and also lack of initial classification of regions, other than realizing that NOP instructions are essentially placeholders or non-coding DNA.

If the last challenge proves insurmountable, I will focus on the first two datasets and honing their HMMs and NNs, and perhaps discuss hypothetically the application of HMMs and NNs to digital organism genomes.

Milestones:

One milestone which I have already completed is going through the tutorial of the UCI dataset and getting the machine learning algorithms working (a). I also examined the outputs and the varying metric values for each (b).

The next milestone is importing and familiarizing myself with the code to implement an HMM and NN (c).

Then I will need to load in and manipulate the second UCI Molecular Biology dataset to ensure its compatibility with sklearn, splitting into training/testing data and features/labels, etc. (d).

A significant milestone will be running the HMM and NN on the dataset, which will be followed up by analyzing the metrics and adjusting code and parameters (e).

The last milestones of the project will be having well trained models for both datasets (f) and performing a statistical analysis of the differences in results (g).

References:

[1] Bulentsiyah. "Classifying DNA Sequences-Markov Models-KNN-SVM." *Kaggle*, 17 May 2020, www.kaggle.com/bulentsiyah/classifying-dna-sequences-Markov-models-knn-svm/execution

[2] D. V. Lindberg and H. Omre, "Inference of the Transition Matrix in Convolved Hidden Markov Models and the Generalized Baum–Welch Algorithm," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 12, pp. 6443-6456, Dec. 2015, https://ieeexplore-ieee-org.proxy2.cl.msu.edu/document/7147792, doi: 10.1109/TGRS.2015.2440415.

[3] Grewal, Jasleen K., et al. "Markov Models -- training and evaluation of Hidden Markov Models." *Nature Methods*, vol. 17, no. 2, 2020, p. 121+. *Gale OneFile: Health and Medicine*, https://go-gale-com.proxy2.cl.msu.edu/ps/i.do?p=HRCA&u=msu_main&id=GALE%7CA613230415&v=2.1&it=r&sid=summon Accessed 12 Feb. 2021.

[4] Hmmlearn. "Hmmlearn." *GitHub*, 5 Feb. 2021, github.com/hmmlearn/hmmlearn.

[5] Noordewier, Michiel, et al. "Training Knowledge-Based Neural Networks to Recognize Genes in DNA Sequences." *Proceedings of the 3rd International Conference on Neural Information Processing Systems*, 1990, pp. 530–36, proceedings.neurips.cc/paper/1990/file/8efb100a295c0c690931222ff4467bb8-Paper.pdf.

[6] Wang, Zhuo, et al. "A Brief Review of Computational Gene Prediction Methods." *Genomics, Proteomics & Bioinformatics*, vol. 2, no. 4, 2004, pp. 216–221., https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5187414/#:~:text=There%20are%20mainly%20two%20classes,as%20ab%20initio%20gene%20finding. doi:10.1016/s1672-0229(04)02028-5.

[7] Welch, Stephen. "Neural-Networks-Demystified." *GitHub*, 9 Mar. 2020, github.com/stephencwelch/Neural-Networks-Demystified.