# Preliminary Results

**Mica Haney, Camrie Johnson**

University of North Texas

## 1 Data Preprocessing

All datasets were split into training and testing sets at an 80 to 20 percent ratio. No set aside validation set was created. It was determined that the limited dataset size would work best with a larger test set and trained using cross-fold validation.

### 1.1 Dataset Modification

The root dataset was considered to be the pair of the target gene sequence, the start location of the binding site, the end location of the binding site, and the siRNA sequence that bound to the binding site.

Two other datasets were generated from the original dataset. One was a subset where only records with an activity value over 0.7 - indicating more effective binding - were kept. The other kept some of the numeric features included in the original dataset. The retained features are normalized to values between 0 and 1.

### 1.2 Windowing

The Naive Bayes and Markov Chain models require the input data to be a sequence of bases followed by a single base as a categorical output. To accommodate this, each target gene and siRNA sequence pair was windowed to generate a series of vectors, each containing a unique subset of the target gene and containing subsets from the target gene’s binding site plus the preceding bases within the window size. Each vector had the corresponding base of the siRNA sequence attached to its window as the target category for prediction.

For the Naive Bayes model the additional preprocessing step of one-hot encoding was performed on the window vectors before training.

### 1.3 Vectorized ID Encoding

The LSTM and Transformer models require inputs to be pairs of a sequence of bases from the target gene paired with the siRNA sequence. For each siRNA sequence, the target gene was windowed to contain the binding site as well as the preceding bases within the window size.

After windowing each target gene subset and each siRNA were ID encoded. Both the target gene window and the siRNA sequence had a start token added to the beginning of each sequence and an end token added to the end of each sequence.

For the LSTM model the siRNA sequences were padded with trailing 0s in order to match the length of the target gene sequences.

## 2 Model Architectures

The Naive Bayes model was implemented using Scikit-Learn’s Categorical Naive Bayes model, implemented with the default values. A single predictor element was trained over all of the training data and prediction sequences were generated by predicting on all of the windows of one input record.

The Markov Chain has not been finished. It will generate a transition matrix computed over all of the training data and predictions will be done in the same fashion as the Naive Bayes model.

The LSTM has not been finished. The architecture was implemented using Tensorflow, and was an encoder-decoder model with two LSTM layers in total, one for the encoder and one for the decoder. The embedding dimension was set to 128 and the number of LSTM units per layer was set to 512. Currently the architecture is mostly complete and debugging is the next step before training.

|  | Acc. | Pre. | Rec. | F1 | BLEU | R. 1 | R. 2 | R. L |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Naive Bayes | 0.3014 | 0.3014 | 0.3014 | 0.3014 | 0.2761 | 0.6531 | 0.4331 | 0.5148 |
| Markov Chain | - | - | - | - | - | - | - | - |
| LSTM | - | - | - | - | - | - | - | - |
| Transformer | - | - | - | - | - | - | - | - |
| Table 1: Results for average accuracy (Acc.), average precision (Pre.), average recall (Rec.), average F1, BLEU, ROUGE 1 (R. 1), ROUGE 2 (R. 2), and ROUGE L (R. L) for each model on the root dataset. | | | | | | | | |

The Transformer has not been finished. The architecture will mirror the LSTM model with two units total, an embedding dimension of 128, and the number of units set to 512.

## 3 Training Details

### 3.1 Setup and Implementation

The training hyperparameters were set to 10 epochs, batch size of 20, and 5-fold cross-validation, and a window size of 5. The training hyperparameters epochs and batch size were not used due to not having completed models to train besides the Naive Bayes.

### 3.2 Evaluation

All of the models were evaluated with the metrics of accuracy, precision, recall, F1, BLEU, ROUGE 1, ROUGE 2, and ROUGE L scores. For accuracy, precision, recall, and F1, each sequence prediction is evaluated as a batch of individual classifications. Scores are averaged across all sequence “batches” for reporting.

### 3.3 Results

Since only the Naive Bayes model was completed, only that model’s scores were reported. Results are reported on the dataset without additional feature engineering or feature selection.

The low scores in all metrics indicate that the Naive Bayes model has a poor ability to generate a siRNA sequence. This is most likely due to the destruction of sequence structure during training, and is likely to be significantly influenced by hyperparameter selection. Modifying the window size and using k-mers with k values over 1 are planned for the next phase of this project.