**PREDICTION OF LNCRNA SUBCELLULAR LOCALIZATION WITH DEEP LEARNING FROM SEQUENCE FEATURES**

Vi Ly

DATA 4395: Senior Project in Data Science

08/07/2019

|  |  |
| --- | --- |
| **Faculty Advisors:** |  |
| Dr. Benjamin Soibam: |  |
| Committee Member: |  |
| Dr. Patrick King: |  |
| Committee Member: |  |
| Dr. Dexter Cahoy: |  |
| Department Chair: |  |
| Dr. Ryan Pepper |  |

Table of Contents:

Abstract ………………………………………………………………………………………1

1. Introduction ……….………………………………………………………………………3
2. Background…………………………………………………………………
3. Datasets ………………………………………………………………..

II. Methods ………………………………………………………………….

1. Feature-based Models…………………………………………………………
   1. Feature Extraction
   2. K-Nearest Neighbor (KNN) Model
   3. Random Forest (RF) Model
2. Deep Learning Models…………………………………………………………
   1. One-hot Encoding
   2. Deep Learning Models
   3. Hyper-parameter Optimization

III. Results

* + 1. Model Evaluation
    2. Variable Importance and Motifs

IV. Conclusion

1. Discussion
2. Future Works

V. References

Key words: LncRNA, Bioinformatics, Feature-based Machine Learning Model, Deep Learning Model

**Abstract:**

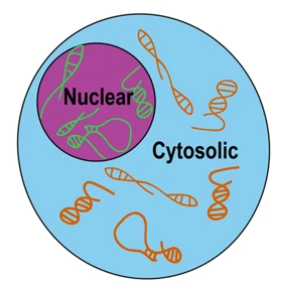
While protein subcellular localization is a well-established research field, the long non-coding RNAs (lncRNAs) localization is still fairly new. A majority of lncRNAs stay unannotated and the manual process of lncRNAs annotation of is difficult due to their low due to their low conservation and unique expression. However, the long non-coding RNAs (lncRNAs) are observed to play an active role in the biological process in the nucleus and cytosol. Therefore, LncRNA subcellular localization is useful in term of discovering the biological function of LncRN. However, but there are not many good prediction methods that are currently available1. In this research, I use the feature-based machine learning and deep learning algorithms to perform lncRNAs localization. Our input comes straight from lncRNA transcript sequences. Our dataset includes of 7259 strands of lncRNA squences classified into two class: nuclear localization and cytosolic localization. The data are processed and feed through feature-based models and the deep learning models for the task of classification based on their features and labels. This work will provide a comprehensive and quantitative study on prediction of lncRNA localization and provide insights to the sequence patterns unique to lncRNA. Each of these motifs collectively or combinatorically contribute to the prediction power of the deep learning model. The classification results will help me accurately identify the motifs that associated with lncRNA localization directly from lncRNA transcript sequence.

**I. Introduction:**

**1. Background:**

The complex interactions between DNA, RNA and proteins make up the internal system of our human cell. While mRNA solely responsible for carrying information from the synthesis of protein, lncRNAs are large RNA transcripts which do not encode proteins and are estimated to outnumber protein-coding genes within the human genome. However, as shown in Fig 1, the long non-coding RNAs (lncRNAs) are observed to locate and play an active role in the biological process in the nucleus and cytosol. Therefore, LncRNA subcellular localization is useful in term of discovering the biological function of LncRN1.

There are not many good prediction methods that are currently available. LncRNA subcellular localization depends on two main factors: sequence and structural motifs. Obtaining lncRNA structural data of motifs from manual biological process is a difficult task. However, lncRNA transcript sequences are readily available publicly in large number.1 In this research, I use the feature-based machine learning and deep learning algorithms to perform predict lncRNA subcellular localization and, therefore, obtain the motifs.

Here, we address two important thesis questions: Can we accurately indentify the motifs that associated with lncRNA localization directly from lncRNA transcript sequences?

**Fig. 1:** The localization of lncRNA inside the human cell. 1

**2. Dataset:**

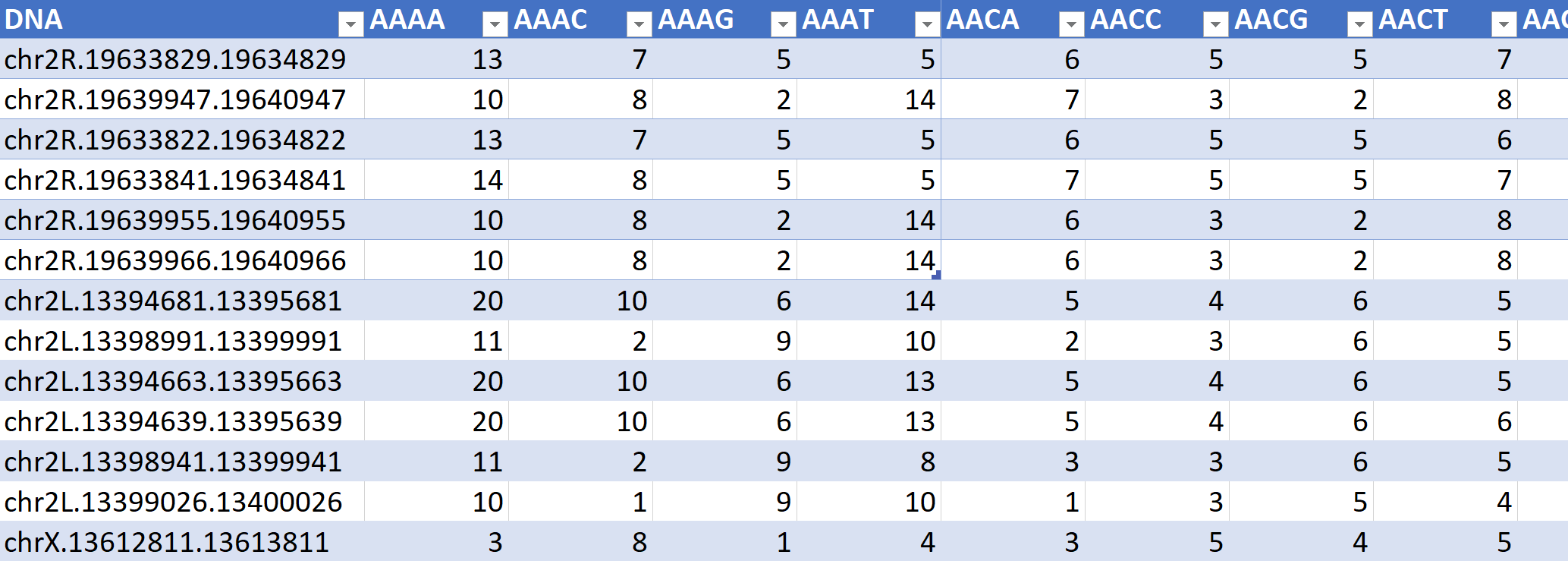
Firstly, I utilized ENCODE project’s paired-end strand-specific RNA-sequencing of human cell lines. To separate the nucleus from the cytosol, cellular fractionation was performed on these samples. For further analysis, gencode (v28) long non-coding RNA annotations were used to identify LncRNAs.1 In total, 7259 lncRNA strands sequences were found in our dataset, classifying into two classes: nuclear localization and cytosolic localization. 5806(80%) of samples strands were used to train our model. We saved 1453 (20%) other strands for evaluation process. The fasta file of the RNA sequence data can be found on this github page: <https://github.com/bgudenas/DeepLncRNA/>.

**II. Methods:**

**1. Feature-based Model:**

**a) Feature Extractions:**

We use the function oligonucleotideFrequency in the Biostrings package to slice the raw RNA sequences. The function apply kernel to slide the sequence into smaller pieces with length of k called k-mer. The length k of the kernel and the overlapping space are adjustable. We will length k of 4, 6 and 8 with the stride of 3 as described in Fig. 2. Each k-mer representing a possible features that we are hoping to extract and we select the top 500 k-mer to output to a count csv table. Two feature-based machine learning models that I use in this project are K-nearest neighbors (KNN) and random forest.



**Fig. 2:** (a) Raw lncRNA sequences;

(b) Process of kernel slicing for K-mer

features extraction; (c) K-mer feature

count table

**b) K-Nearest Neighbors (KNN) Model:**

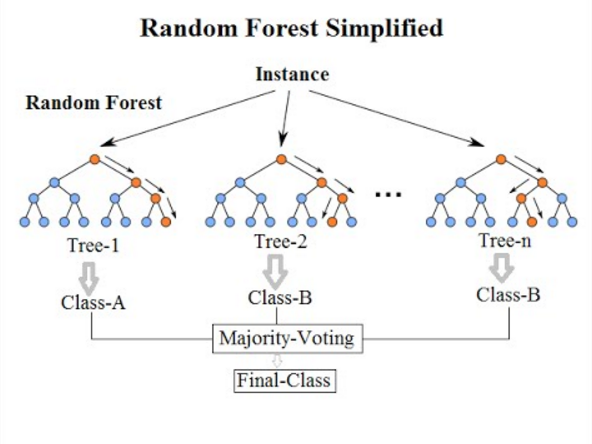
**Fig. 3:** A sample visualization of KNN. 2

K-Nearest Neighbor (also known as KNN) is part of the supervised machine learning algorithms family, in which a new class of data point are predicted from a labeled dataset. The KNN model find the closest neighbors to the target point from training examples in the feature space. 2 To implement KNN model, I use Caret package in R Programming. The k parameters tuned are in range 1 to 20. The best model is selected based on the value of the accuracy.

**c) Random Forests (RF) Model:**

A decision tree is a flowchart-like structure in which each internal node represents a “test” on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label. 3 In a RF, each decision tree produces a class prediction, after receiving the results of the forest, the most voted class is selected as the RF model’s prediction.4

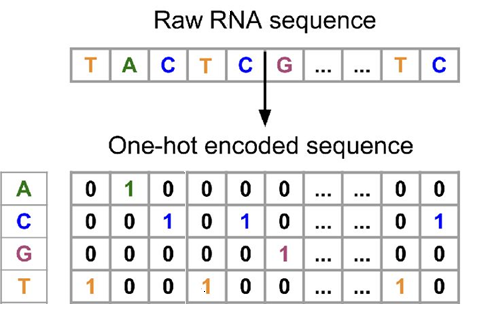
Number of variables available for splitting at each tree node. This is referred to as the mtry parameter The mtry is in range from 409 to 4090 splits. The accuracy is in consideration for taking the best model.



**Fig. 4:** A simplified structure of random forest algorithm.

**2. Deep Learning Model:**

**a) One-hot Encoding:**

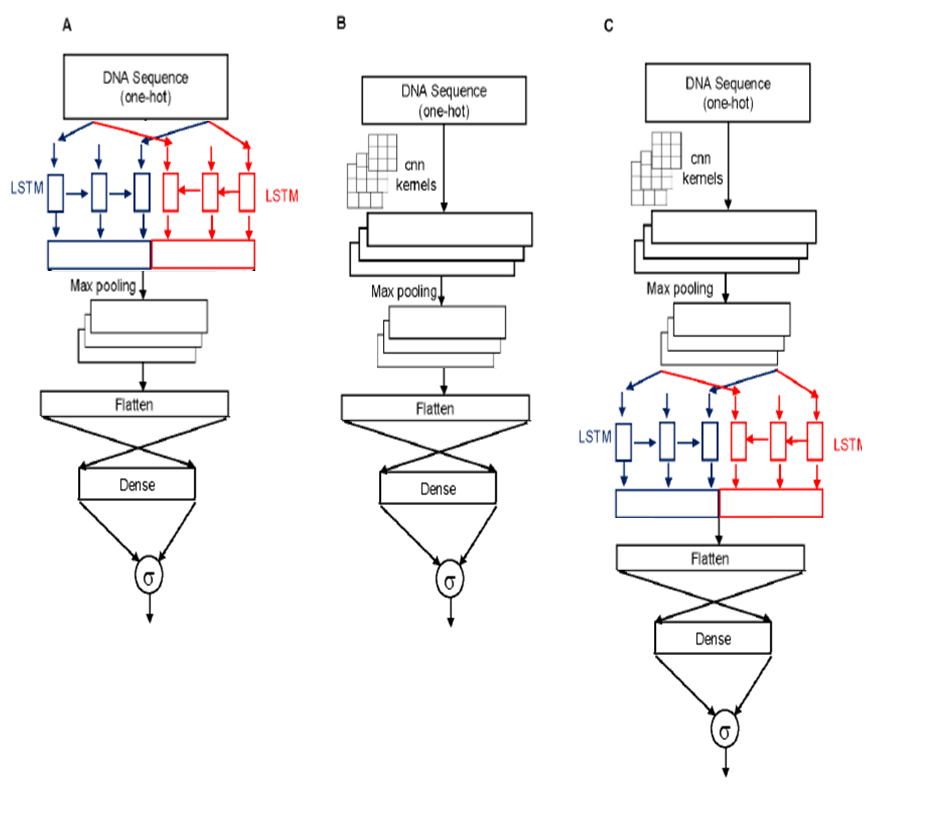
One-hot vectors can be used to represent sequences, these sequences can conserve the position information of each nucleotide within the sequences. This popular model was created based on an inspiration of a deep learning text classification model. Performance improvements have been achieved in several validation datasets using this model.5 The mechanism of one hot encoding is

**Fig. 5:** A sample of one-hot encoded RNA sequence.

In Figure 5, we see an encoded RNA sequence with the dimension of (n, 4). N is the length of the sequence and 4 is the dimension of encoded nucleotides. Each nucleotide is now binary encoded A is {1, 0, 0, 0}, C is {0, 1, 0, 0}, G is {0, 0, 1, 0} and T is {0, 0, 0,1}. N is {0, 0, 0, 0} but it is later removed out of the sequence. The encoded matrix is used as input to the neural network model.

**b) Model Training:**

After perform one-hot encoding on the data. I input the feature matrix into deep learning models such as convolutional neural networks (CNN) and recurrent neural networks (RNN). These models do not need a feature count table; they can automatically learn the important features from the actual encoded sequences.

****

**Figure 5.** Three representative deep learning models. Panel (**A**) shows the 1-layer LSTM model, which is comprised of an input layer, LSTM layer, pooling layer, a flatten layer, a dense layer and output layer. Panel (**B**) shows the 1-layer CNN model, which is comprised of an input layer, LSTM layer, pooling layer, a flatten layer, a dense layer and output layer. Panel (**C**) shows the 1-layer CNN\_LSTM consisting of an input layer, a CNN layer, a pooling layer, a LSTM layer, a flatten layer, a dense layer and output layer. The output layer has one neuron with sigmoid activation function indicated by σ.

These systems detect patterns and learn at each layer of the network. The system then runs the information back through the network (back-propagation) or feeds the data through the next layer of the network via a CNN.7 lncRNA sequences will be run through a neural network that transforms the data at each layer. Each transformation allows features to be extracted from the data. These data extractions give insight to the lncRNA sequences and which motifs help on the classification process.

**c) Hyper-parameter Optimization:**

Talos allows you to use Keras models exactly as you would otherwise. Talos used random, and probabilistic hyperparameter optimization strategies that help on maximizing the flexibility, efficiency, and result of random strategy.6 With minimum efforts, Talos also allows me to configure, perform, and evaluate hyperparameter optimization my deep learning experiments that yield best possible model across different combination of parameters. All the model experiments on different hyperparameters are exported in to a zip file that includes csv log and the best model with the weights.

(A)

|  |  |  |
| --- | --- | --- |
|  | **Number of Layers** | **Parameters/Hyper-paprameters** |
| **MiniCNN** | 1x{CNN, Maxpooling} | Learning rate (1, 0.1, 10)  Kernel size (6, 8, 10, 12, 14)  Number of kernels (32, 64, 96, 120)  Optimizer (Adam, RMSprop, Adagrad, Adadelta) |
| **SmallCNN** | 2x{CNN, Maxpooling} |
| **MediumCNN** | 3x{CNN, Maxpooling} |
| **LargeCNN** | 5x{CNN, Maxpooling} |
| **VeryLargeCNN** | 8x{CNN, Maxpooling} |

(B)

|  |  |  |
| --- | --- | --- |
|  | **Number of Layers** | **Parameters/Hyper-paprameters** |
| **SmallLSTM** | 1x{LSTM, Maxpooling} | Learning rate (1, 0.1, 10)  Kernel size (6, 8, 10, 12, 14)  Number of kernels (32, 64, 96, 120)  Optimizer (Adam, RMSprop, Adagrad, Adadelta) |
| **MediumLSTM** | 2x{LSTM, Maxpooling } |
| **LargeLSTM** | 3x{LSTM, Maxpooling } |

(C)

|  |  |  |
| --- | --- | --- |
|  | **Number of Layers** | **Parameters/Hyper-paprameters** |
| **MiniCNN-LSTM** | 1x{CNN, Maxpooling} + 1x{LSTM, Maxpooling} | Learning rate (1, 0.1, 10)  Kernel size (6, 8, 10, 12, 14)  Number of kernels (32, 64, 96, 120) - Optimizer (Adam, RMSprop, Adagrad, Adadelta) |
| **SmallCNN-LSTM** | 1x{CNN, Maxpooling} + 2x{LSTM, Maxpooling} |
| **MediumCNN-LSTM** | 1x{CNN, Maxpooling} + 3x{LSTM, Maxpooling} |
| **LargeCNN-LSTM** | 2x{CNN, Maxpooling} + 3x{LSTM, Maxpooling} |

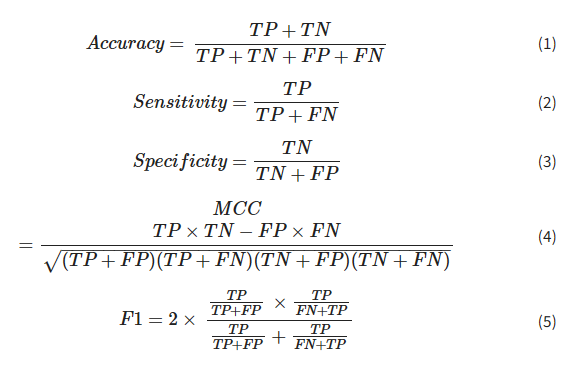
**Table 1:** (A) Five different permutation sets of layers and hyper-parameters for CNN model. (B) Three different permutation sets of layers and hyper-parameters for LSTM model. (B) Four different permutation sets of layers and hyper-parameters for CNN-LSTM model.

The use of deep learning is an integral part of the research being conducted. It allows for learning from vast amounts of data in a very short period. Deep learning models utilize many layers of artificial neural networks, which are inspired by the workings of the human brain. It is a powerful tool in computing and bioinformatics. With it, we are able to build on methods used in genomic work such as strand comparison and feature extraction of lncRNA. Neural networks of all types have become the primary tool for handling data of this magnitude.1

**IV. Results:**

* + 1. **Model evaluation:**

In this work, we develop a DeepLncRNA to identify lncRNAs to be enriched in the nucleus (positive class) or cytosol (negative class). We use the common machine learning metrics such as accuracy, sensitivity, specificity and Matthews correlation coefficient for classifier performance evaluation. TP is the number of true positives; TN is the number of true negatives; FP is the number of false positives; and FN is the number of false negatives.

****

**Table 2:** Formulas for accuracy (acc), Matthews correlation coefficient (mcc), sensitivity, specificity, F1 score 1

The best trained model from each machine learning algorithm is used to perform prediction on the testing dataset. The predicted results are compared with the labels to output the number of TP, TN, FP and FN. Those values are then used to calculate the 5 metrics of area under the curve (auc), accuracy (acc), Matthew (mcc), precision (specificity), recall (sensitivity) and F1 score using the formulas in table 5. We also use TP, TN, FP and FN to plot a graph called a Receiver Operating Characteristic curve (or ROC curve) and then calculate the area under the curve (or AUC).

The results of all testings are shown in the Table 4 below:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **auc** | **acc** | **mcc** | **precision** | **recall** | **fscore** |
| LargeCNN | 0.685139 | 0.343686 | 0 | 0.17184318 | 0.5 | 0.255779 |
| KNN8 | 0.5143 | 0.5269 | 0.0265 | 0.4314 | 0.595 | 0.5 |
| SmallCNN | 0.6043 | 0.6043 | 0.6043 | 0.6043 | 0.6043 | 0.6043 |
| VeryLargeCNN | 0.579304 | 0.655804 | -0.01633 | 0.32806928 | 0.499612 | 0.396064 |
| LargeRNN | 0.664436 | 0.67057 | 0.144938 | 0.69384868 | 0.527092 | 0.460804 |
| MediumCNN | 0.76955 | 0.676171 | 0.18039 | 0.75094978 | 0.532417 | 0.467196 |
| MediumRNN | 0.683711 | 0.684318 | 0.236781 | 0.64163575 | 0.59896 | 0.599458 |
| RF8 | 0.726 | 0.6999 | 0.3588 | 0.495 | 0.8492 | 0.6254 |
| KNN6 | 0.5128 | 0.7114 | 0.0266 | 0.4 | 0.5732 | 0.4712 |
| KNN4 | 0.6701 | 0.7114 | 0.3206 | 0.604 | 0.7168 | 0.6556 |
| RF4 | 0.7716 | 0.7114 | 0.3925 | 0.6986 | 0.5324 | 0.6043 |
| SmallRNN | 0.731412 | 0.713849 | 0.313167 | 0.69078262 | 0.628515 | 0.63296 |
| LargeCNN\_RNN | 0.739359 | 0.720978 | 0.338514 | 0.69488297 | 0.647001 | 0.654404 |
| SmallCNN\_RNN | 0.7400 | 0.7210 | 0.3536 | 0.6891 | 0.6653 | 0.6721 |
| MedCNNRNN | 0.740019 | 0.720978 | 0.353621 | 0.68906709 | 0.665348 | 0.672141 |
| RF6 | 0.7941 | 0.7259 | 0.416 | 0.5299 | 0.8581 | 0.6095 |
| MiniCNN\_RNN | 0.774359 | 0.737271 | 0.411633 | 0.70814042 | 0.703519 | 0.705619 |
| MiniCNN | 0.812864 | 0.760183 | 0.477734 | 0.7352909 | 0.742497 | 0.738378 |

**Table 3: Deep Learning versus feature-based models.** Twelve best models of deep learning model permutations vs six best models of feature-based model permutations.

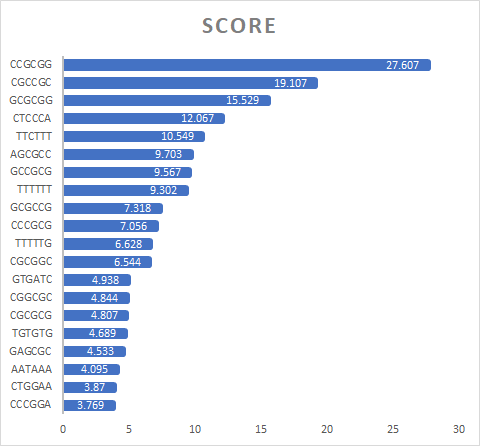
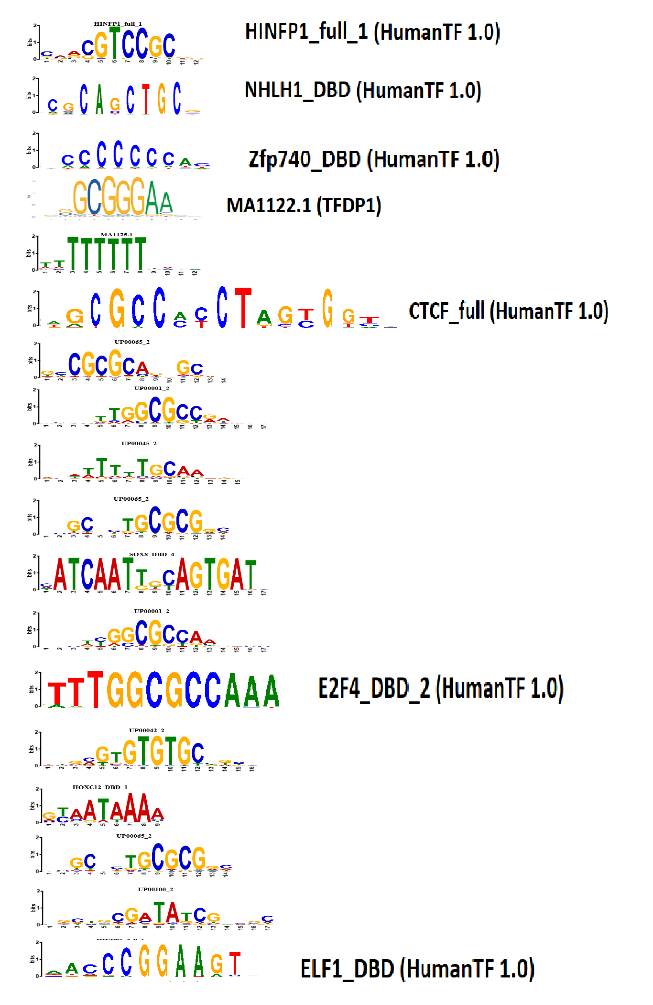


**Fig. 2:** Model evaluation through line charts of 6 metrics.

**2. Variable Importance and Motifs:**

After obtaining the best model in each scenario, we ranked the features by their importance in prediction, which is calculated for each parameter in the model. In order to identify the most important k-mer pattern that differentiate between the lncRNA at the Nucleus vs the Cytosol, we will quantify and rank the variable importance of each k-mer. After that, we use match the kmer pattern with the correct DNA and protein motifs using TOMTOM service.

A) (B)



**Fig. 6:** (A) shows the top 20 of 6-mers with highest VarImp score. (B) shows the matched motifs output of the top 6-mers from TOMTOM .

In human (and eukaryotes in general), transcription factors are transcribed inside of the nucleus and they are then move to be translated in the cell's cytosol. These factors that are active in the nucleus contain nuclear localization signals that direct them to the nucleus and this is a key point in their regulation. A nuclear localization signals is an amino acid sequence that marks a protein to be imported into the cell nucleus by nuclear transport. 8

**V. Conclusion:**

**1. Discussion:**

In summary, using feature-based and deep learning methods based on the acquired RNA-features information, we could predict the lncRNA localization in the nucleus or cytosol up to the accuracy o 76%. By analyzing both feature-based models and deep learning models, we can see that MiniCNN with only one convolutional layer perform he best on all metrics. In contrast, the complex model of large CNN constantly rank the lowest in all the metrics. Therefore, we can conclude that training the deep learning model on simple patterns extracted from the first layer of CNN help is efficient enough to classify precisely the location of RNA. Top three models with the best metrics are miniCNN, miniCNN-RNN and RF6, which all have very simple architecture. While the architecture of the model become more complex, the learning power of the models decrease and they classify less accurately on the localization task.

After extracting the features with highest score of variable importance to match with actual motifs in human cells, these results suggest that specific gene orientation of transcription factors could be useful for future RNA localization research.

* + 1. **Future research:**

From the lncRNA sequence alone, we can have achieve an 81% on AUC in the RNA localization. This performance can be improved by adding other features such as length of the sequence. To fully decipher the nature of interaction between these motifs is beyond the scope of this paper and will be addressed in future studies.

**V. References:**

1. Gudenas, B. L., & Wang, L. (2018, November 06). Prediction of LncRNA Subcellular Localization with Deep Learning from Sequence Features. Retrieved from <https://www.nature.com/articles/s41598-018-34708-w>
2. Pros and Cons of K-Nearest Neighbors. (2018, September 25). Retrieved from <https://www.fromthegenesis.com/pros-and-cons-of-k-nearest-neighbors/>
3. Decision tree. (2019, July 27). Retrieved from <https://en.wikipedia.org/wiki/Decision_tree>
4. Tonester. (2019, August 04). Understanding Random Forest. Retrieved from https://towardsdatascience.com/understanding-random-forest-58381e0602d2
5. Abapihi, B., Faisal, M. R., Lumbanraja, F. R., Phan, D., Ngo, D. L., Tran, V. A., . . . Kubo, M. (2016, April 27). DNA Sequence Classification by Convolutional Neural Network. Retrieved from <https://m.scirp.org/papers/65923>
6. Introduction. (n.d.). Retrieved from <https://autonomio.github.io/docs_talos/#introduction>
7. Henderson, J., Ly, V., Olichwier, S., Chainani, P., Liu, Y., & Soibam, B. (2019, July 26). Accurate prediction of boundaries of high resolution topologically associated domains (TADs) in fruit flies using deep learning. Retrieved from <https://www.ncbi.nlm.nih.gov/pubmed/31049567>
8. Transcription factor. (2019, July 25). Retrieved from <https://en.wikipedia.org/wiki/Transcription_factor>