NES Peptides Prediction - Scientific Report

3D Data Processing in Structural Biology Hackathon

Nil Ashkenazi, Meshi Maman

Introduction

The study of nuclear export signals (NES) within proteins has garnered significant attention due to its profound implications in cellular biology and disease mechanisms. NES are short peptide sequences that facilitate the transport of proteins from the nucleus to the cytoplasm by interacting with exportins, particularly exportin 1 (CRM1/XPO1). This process is essential for maintaining cellular homeostasis, regulating gene expression, and controlling various signaling pathways.

Proteins are transported between the nucleus and the cytoplasm through nuclear pore complexes (NPCs). To exit the NPC, large cargo molecules, such as proteins and mRNA, must bind nuclear export receptors like CRM1/XPO1 via nuclear export signals (NESs). Cargo proteins “recognize” CRM1/XPO1 through the sequence specificity of the peptide-protein interaction between the NES motif (the peptide) and the CRM1/XPO1 receptor (the protein). This specific binding allows the cargo proteins to be efficiently exported from the nucleus to the cytoplasm, playing a crucial role in cellular function and signaling.

The advancement in machine learning (ML) offers a promising solution to the challenge of identifying NES within protein sequences. By leveraging datasets of known protein sequences and their corresponding NES annotations, ML models can be trained to recognize patterns and predict the presence of NES with high accuracy. Traditional experimental methods for identifying NES are time-consuming and often not effective in discovering novel NES motifs, making ML an invaluable tool for accelerating this process.

In this hackathon, we aimed to develop a learning model capable of predicting NES in protein sequences. Our approach involves the integration of diverse biological data sources and the application of ML techniques to create a robust and generalizable prediction tool.

Methods

Preparations

After getting and looking at the dataset we generated embeddings using pre-trained ESM-2 models. For model evaluation, we split the data into training and testing sets, allocating 10% for testing.

Phase 1 - Methodology for Classification and Evaluation

We trained and tested our model on combined positive and negative peptide vectors, exploring various classification methods and strategies to enhance its performance -

1. **Minimum Distance Calculation**
   We computed the minimum Euclidean distances between test peptides and positive training peptides.

2. **K-Nearest Neighbors (KNN) Classification**
   The KNN algorithm was employed to classify the peptide sequences based on their ESM-2 embeddings. By varying the number of neighbors (k) from 1 to 15, we identified the optimal k value that maximized the area under the ROC curve (AUC). We visualized the ROC curves to compare the performance of different k values, as can be seen in the results section.

3. **K-Means Clustering**
   K-means clustering was applied to the peptide embeddings with k set to 2, corresponding to the two classes (positive and negative). The resulting clusters were analyzed to assess their correspondence with the known labels.

4. **Logistic Regression**
   Logistic regression, a linear model suitable for binary classification, was trained on the peptide embeddings. The predicted probabilities were used to plot the ROC curve and evaluate the model's performance.

5. **Cosine Similarity**
   In addition to the methods depicted above, we also explored the use of cosine similarity to classify the peptide sequences. We computed the similarities between test peptides and training peptides, and the highest similarity scores for each test peptide were used to evaluate their classification performance.

We found that certain methods performed better than others, leading us to opt for logistic regression due to its superior AUC score (detailed in the results section).

Phase 2 - Validation

To further check our predictor tool we took out all of the Snurportin 1 data from the original dataset and trained our model again on the new dataset.We knew that at least 3 Snurportin positive peptide sequences should be marked as negative as we were told, and we will check the success of the predictor against this assumption. The results we got are as follows -

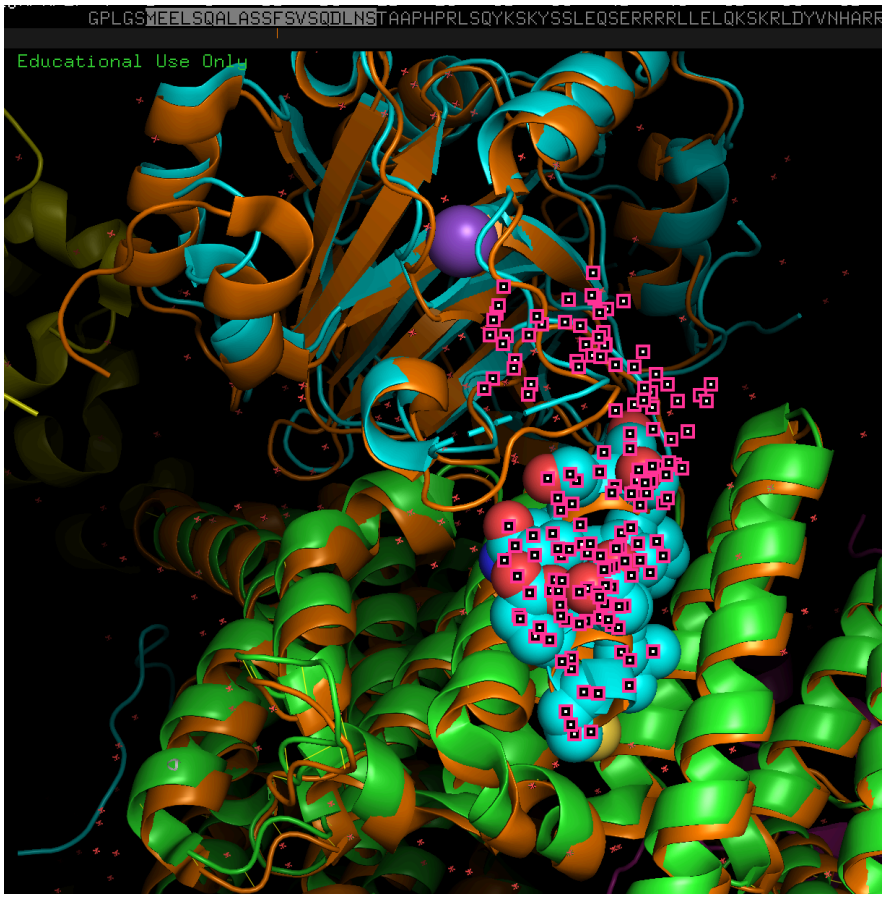
Peptide Sequence	Prediction	Confidence
EEGLGEKTKLNPFKFVGLKN	Positive*	59.7%
MEELSQALASSFSVSQDLNS	Positive	89.3%
VGKRALIVASRGSTSAYTKSGYCVN	Negative	98%
TILDCIYNEVNQTYYVLDVM	Negative	84.8%

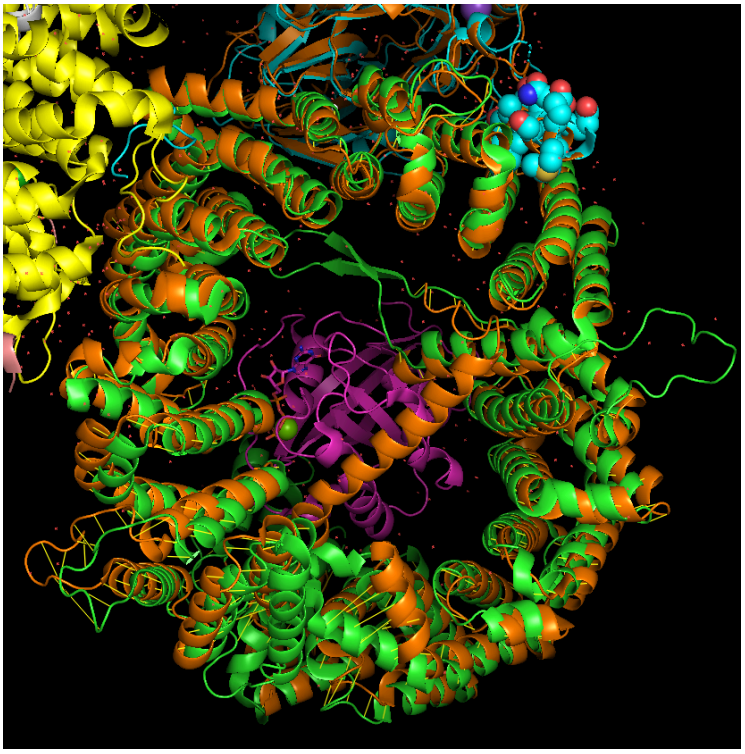
**In some cases we got a negative prediction. We assume that this is due to different sampling from the train set which took effect on the model's ability to predict and might be solved with more data.*

To sum up, the predictor managed to predict 2 out of the 3 negative peptide sequences that were marked as positive.

(PyMol visuals in the next page)

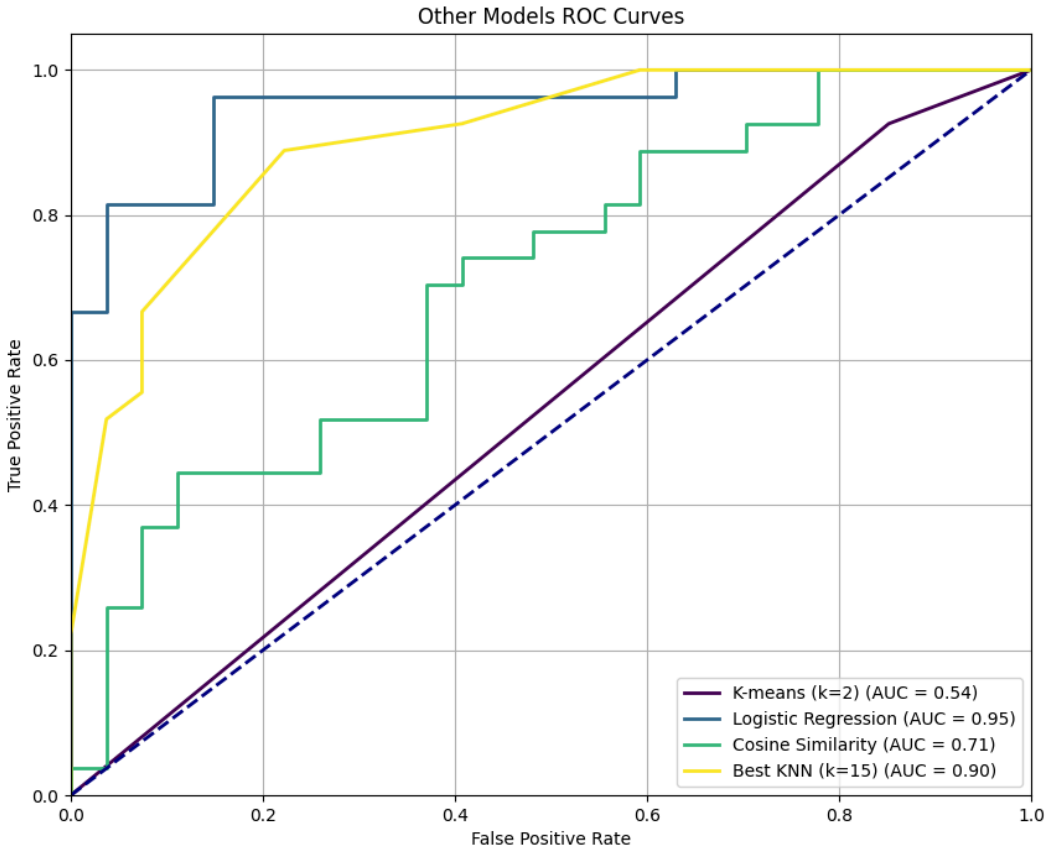
Example for one of the predicted peptides - as we can see in the results from PyMol, the peptide MEELSQALASSFSVSQDLNS (pink) that we predicted as positive with 89% confidence is indeed positive as it binds well to the CRM1 in orange (and protein in green)-





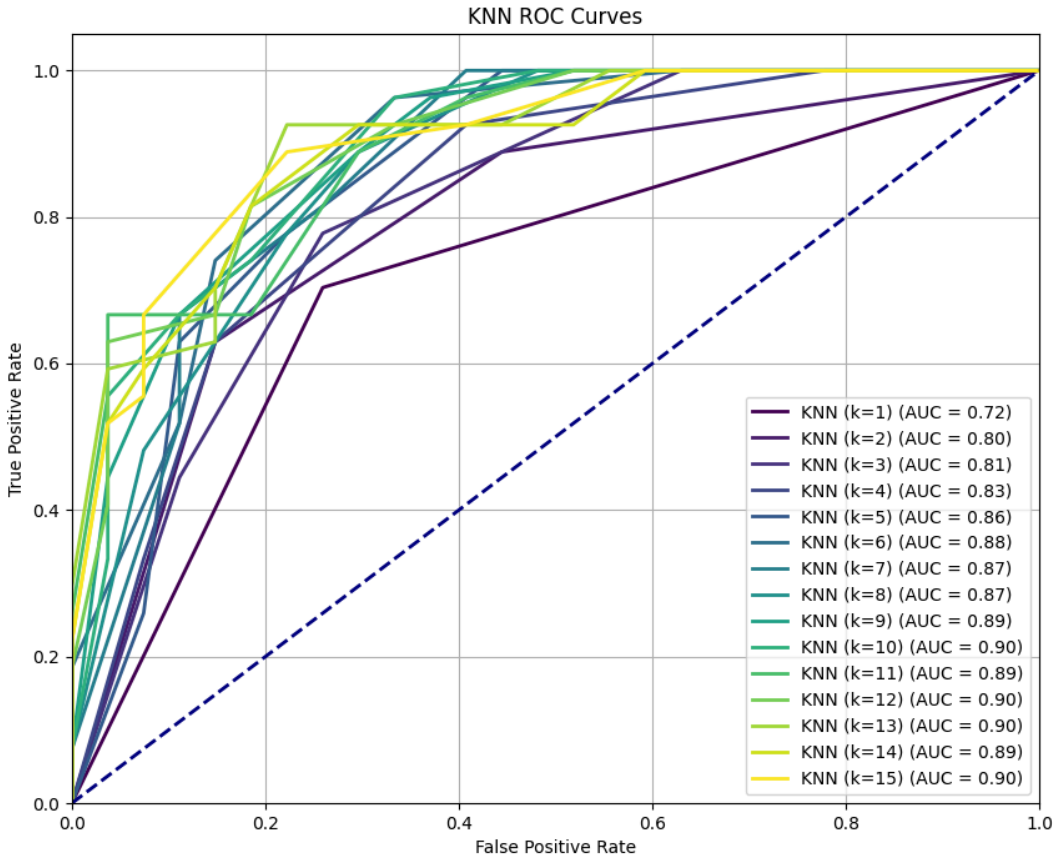
Results

1. After exploring multiple classification methods, we chose to use logistic regression since it had the best AUC score of 0.95, as seen in the ROC curve graph below:

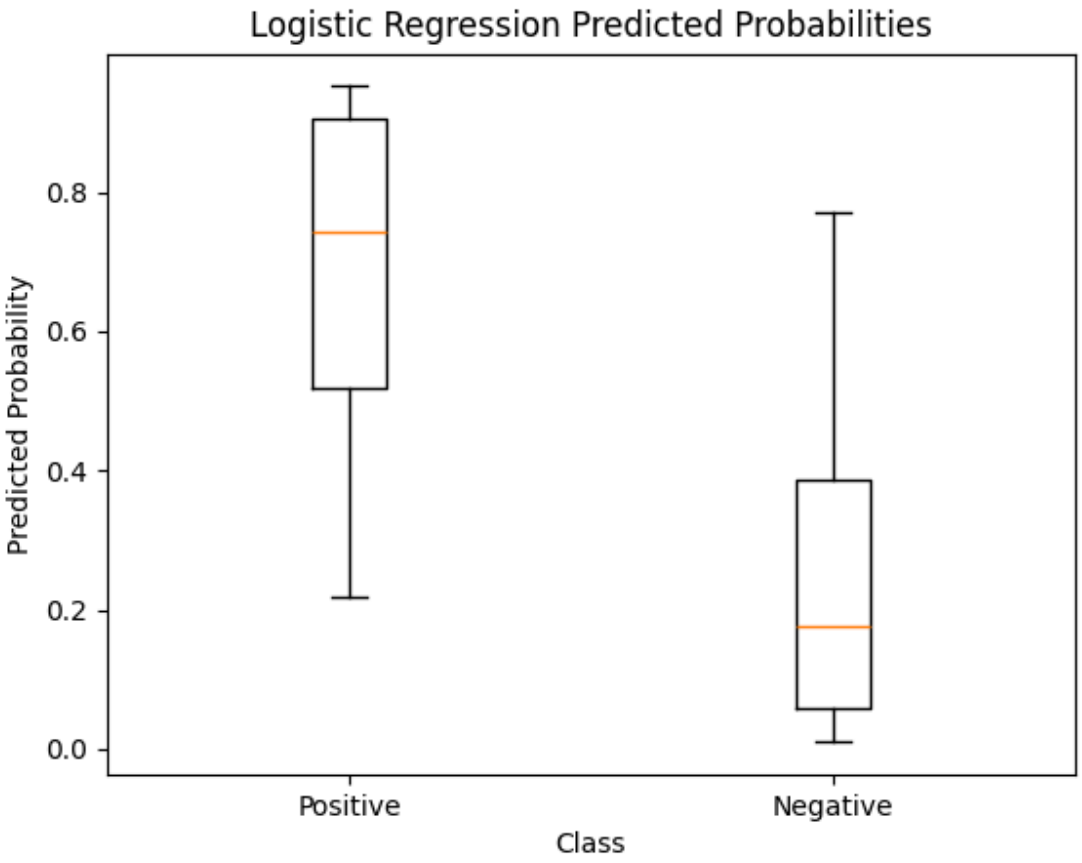


<u>Note:</u> the plots are based on the test data.

2. As for KNN, each time we run the model, we select the best k value from 1 to 15. The results for the optimal k vary with each run, but the general trend remains consistent.



3. After we decided using linear regression we added the next box plot:



There is a clear distinction between positive and negative classes and high confidence in its predictions. These findings indicate that the logistic regression model is an appropriate and effective choice for the classification of peptide sequences in this study.

Positive Class:

- The median predicted probability for the positive class is approximately 0.7, demonstrating the logistic regression model's high confidence in identifying positive samples.
- The interquartile range (IQR) is relatively narrow, indicating that the majority of positive samples are assigned high predicted probabilities.
- The lower whisker extends to around 0.2, suggesting a few positive samples have lower probabilities, but they remain substantially higher than those of the negative class.

Negative Class:

- The median predicted probability for the negative class is approximately 0.2, highlighting the model's effectiveness in recognizing negative samples.
- The IQR is also relatively narrow, suggesting that the majority of negative samples receive low predicted probabilities.
- The upper whisker extends to around 0.5, indicating that while a few negative samples have higher probabilities, they are still generally lower compared to the positive class.

Discussion

During this experiment, we encountered many trials and errors. The first thing that came to mind when trying to optimize our model was to use the cosine angle between the peptides embedded vectors and the peptide vector we want to predict, as we thought vector similarity was crucial (especially in high-dimensional spaces). After trying this and getting poor results, we tried increasing the embedding dimension but we couldn't see an improvement, even then.

Then, we thought of trying different methods and figured out which of them worked best by visualizing the AUC. We added a logistic regression model (which worked best, as can be seen in the results section), KNN model, and K-means model. Where in KNN we first tried running on different k sizes from k=1 to k=15 in order to find the k which provides the best performance, and in K-means we set k=2 in order to create only 2 labels - positive and negative.

For future work, we planned to do the following:

1. Check the CRM1 structure and identify the coordinates of the designated positions for the peptide's Cβ side chains to bind.
2. Review all the positive peptides in the received database.
3. Check the distance between the peptide Cβ side chains and their designated positions in CRM1 (after alignment).
4. Save this distance and add it as a new column in the dataset.

This approach will help us determine if the positive peptides in the given data are indeed positive (if the distance is small enough) or should be reclassified as negative. It will also validate our model's predictions by checking if we correctly predicted a peptide as negative when the distance is large, even if it was labeled as positive in the data.

Another idea we had was trying to enhance the embedding part, by adding:

1. Incorporating attention mechanisms into sequence embeddings to selectively focus on relevant parts of the sequence. This method allows the model to focus on relevant regions that are pivotal for identifying NES motifs.
2. Use Neural Networks to capture complex relationships and dependencies within biological sequences and structures.