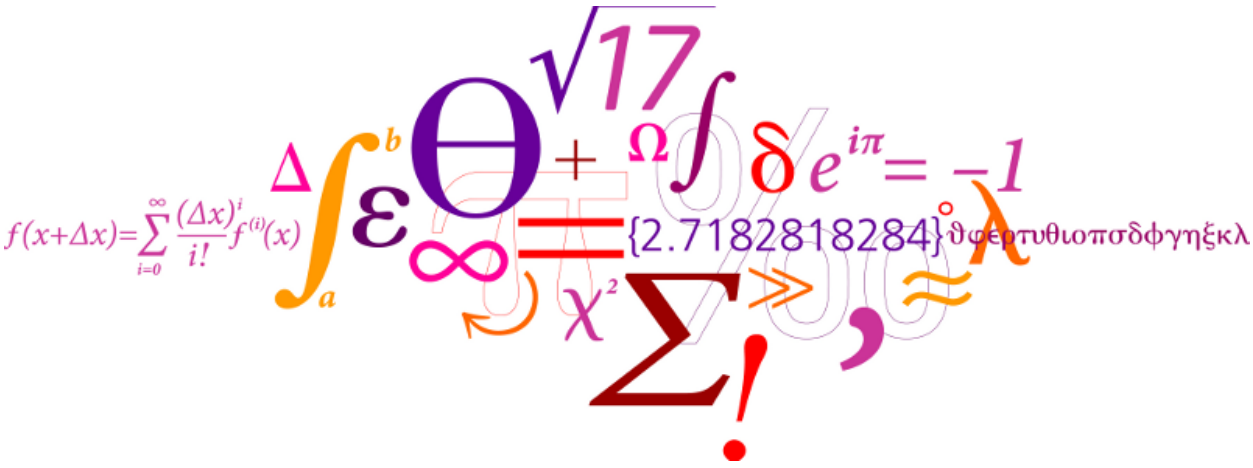
Technical University of Denmark

02456 - Deep Learning

Prediction of TCR-peptide-MHC binding

Participants:	Student Number:
Mehrdad S.Kazemi	s200090
Jake Ph.	s213060
E. Holm Bidstrup	s205980



December 24, 2021

Contents

1 Introduction

The prediction of TCR-pMHC interactions can potentially unlock many medical applications such as immunotherapy, diagnosis, vaccine and therapeutic proteins [1]. Having this knowledge, one could produce necessary antibodies to neutralize cancer, virus, or pathogens and prevent a pandemic. “There are millions to billions of different T-cells in our body that are able to bind to a large variety of antigens” (Dr. Biology, 2011) and knowing the function of each one can help us understand more about the evolution of the human immune system.

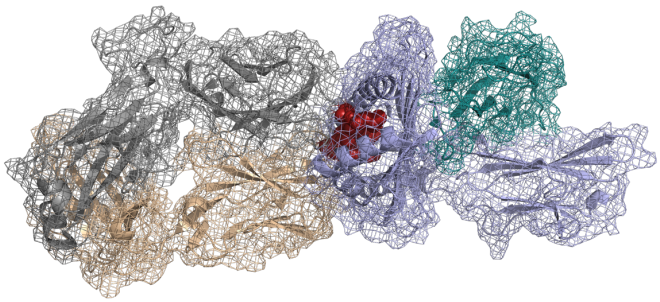


Figure 1 - 868 TCR in complex with HLA A02 presenting SLYNTIATL. The red spheres show the epitope (processed antigen) of Gag protein of HIV virus which is presented by MHC I molecule (The light blue part). TCR 868 identified the epitope and attached to the HLA A02 (MHC I molecule). The green part is Beta-2-microglobulin (B2M) of MHC I. [3]

2 Datasets

This project has been done on a dataset with binary classification; the positive binding and negative binding. Negative bindings or simply negatives are shown with the label 0 and positives with the label 1. The positive data was originally collected from two sources:

A. Immune Epitope Database (IEDB.org)
B. VDJdb

The negative dataset was mainly collected from 10X Genomics dataset with the same cutback mentioned above. The MHC was restricted to only HLA-A*02:01 where their CDR3α and CDR3β were available (8-18 amino acids length) and specific for epitopes with length of 9 amino acids. After reducing the dataset to their shared set of epitopes, the positives were divided into five partitions based on at least 95% of similarity in their CDR3α and CDR3β sequences. Then, negatives were appended to each partition. Finally, a dataset consisting of 6913 entries (TCR + epitope + HLA amino acid sequences) were obtained. In order to have the same length for all different TCRs, they were padded through adding rows of zeros. Therefore, the shape of the dataset was fixed size of (6913 420 54). The second dimension shows the number of amino acids (residues) in each entry. The last dimension represents 54 features for each residue in a given entry. The first 20 features represent the amino acid type using one-hot-encoding. For example, in order to show Alanine, the encoding is like [1, 0, 0, ... ,0] or for Valine is [0, 1, 0, 0, ... ,0].
Moreover, using Fold X 5.0 and Rosetta Energy Function tools, different energy terms have been calculated for each residue of any given entry. These energy terms were placed in the features from position 21 to 54.
The dimensions of input data are shown in the table below:

Feature	Positions	Encoding
Amino acid sequencE	1-20	One-hot-encoding
Rosetta per-residue energy terms	21-27	Per-residue
Fold X interaction energy terms	28-33	Constant values
Rosetta global energy terms complex	34-40	Constant values
Rosetta global energy terms TCR	41-47	Constant values
Rosetta global energy terms pMHC	48-54	Constant values

Table 1 – All 54 features representing every single residue

Each partition was saved in the form of Npz files, one for entries and the other for labels. We used all the 5 partitions in this project using glob module and numpy.concatenate then we made two dataset one for training our network and the other for test. The final training and test sets had the shape of (5387 420 54) and (1526 420 54), respectively. As only 25% of our training and test set was positive, so we had an unbalanced dataset.

2.1 Data Visualization

We found a big bias in terms of frequency of the epitopes in our datasets. We had 18 different epitopes which all were 9-mers. The frequency of each epitope in each dataset has been illustrated in figures 2 and 3. As it has been mentioned above and by comparing the frequencies, we can see that our data is unbalanced regarding the positive and negative data. Moreover, the top 6 epitopes are the same in both training and test set.

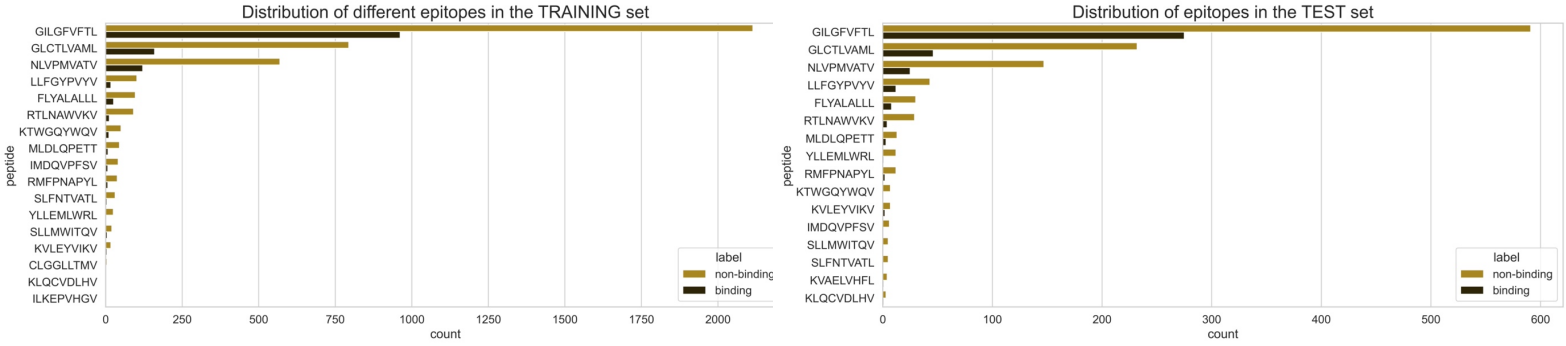


Figure 2 – Frequency of each epitope in training and test dataset

2.2 Evolutionary Scale Modeling (ESM)

The model is developed by the joint effort between Harvard University and Facebook. The repository of the ESM contains code and pre-trained wights of Transformer Protein Language Models such as ESM-1b and MSA Transformer. ESM-1b has the best performance in protein structure prediction tasks. ESM-1b model has 33 layers and 650 million parameters. The goal of this model is to extract more meaningful 3d structure information from amino acid sequences. Since our data set contains the amino acid sequences of the MHC, peptide and TCR, the ESM can generate more useful features to predict the interaction between pMHC and TCR. The output is 1280 embedding features that tell us about how the amino acid sequences interacts and the probability that they can form

chemical bonds with each other [4]. We converted the original dataset to a fasta file. Then, using the codes below we extracted the ESM embeddings:

```
$ pip install fair-esm
$ python extract.py esm1b_t33_650M_UR50S examples/some_proteins.fasta examples/
some_proteins_emb_esm1b/ \
    --repr_layers 0 32 33 --include mean per_tok
```

2.3 Principal Component Analysis

In order to compress information of our high dimensional data and visualize it in a 2D space, we did PCA on our dataset in different states. As the shape of our train dataset was (5387 420 54), we had to reshape it to get a 2D matrix for doing PCA on the it. Therefore, we convert the dataset and got a new one with the shape of (5387 22680). Using the ScikitLearn, we got the plot in figure 3. The variance explained by PC1 and PC2 were 0.455 and 0.149, respectively. It is obvious from the plot that binding and non-binding data points had a large overlap. This makes difficult to train a good model to predict the binding effectively. We decided to train our model using this overlapped dataset and evaluate the trained model on the test set to see if we can achieve a good performance. We call this dataset as D1. At the same time, we tried to find a way to make our training dataset better. A closer look at the dataset, shows that we have many repetitive values in features 28 to 54. It means that almost the half of features are the same for different entries. We decided to remove this repetitive part and see how the PCA results and model performance affected by that. The variance explained by PC1 and PC2 were 0.087 and 0.052, respectively. We call this dataset as D2. The output plot is presented in figure 4.

In addition to D1 and D2 dataset, we have one more dataset resulted from ESM-1b which had the shape of (6913 1280). We call this transformed dataset as D3. The PCA plot of D3 is shown in figure 5.

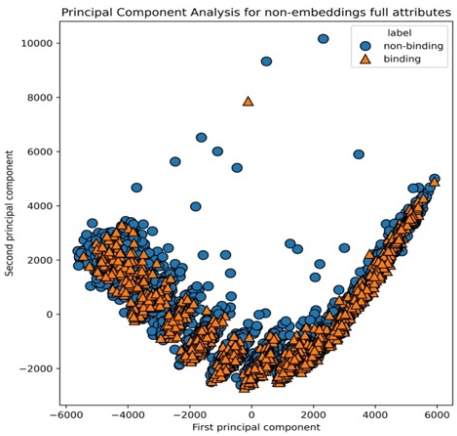


Fig 3 – PCA plot of D1 dataset

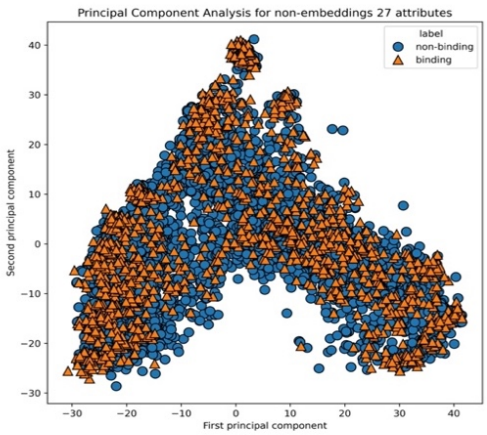


Fig 4 – PCA plot of D2 dataset

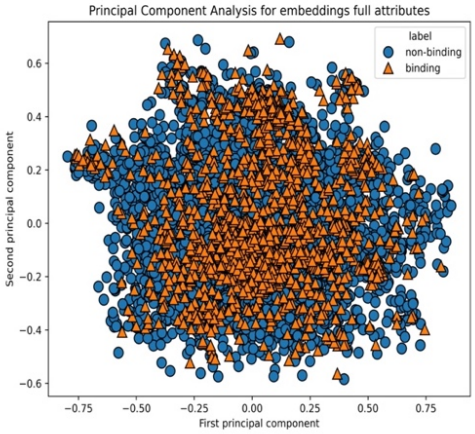


Fig 5 – PCA plot of D3 dataset

In the field of machine learning and artificial intelligence, neural network models are very effective and powerful tools in recognizing hidden patterns and modelling highly complicated data. Neurons of biological nervous system was the inspiration in developing deep learning algorithms [5].

3.1 Model's architecture

The basic task of this project was classification. We used a CNN-biLSTM-FFN neural network on a supervised dataset of 6913 TCR-pMHC sequences.

We used CNN because of its ability to finding hidden pattern in data. Moreover, biLSTM is a proper algorithm for sequential datasets like protein sequences. It empowers the model

training by double traversing through the input data. In our model architecture we used BatchNorm, Rectified Linear Unit (ReLU) for non-linearity inside the architecture, MaxPooling, Dropout, and Sigmoid for non-linearity after the fully connected layer.
Our network design is shown in the figure 6.

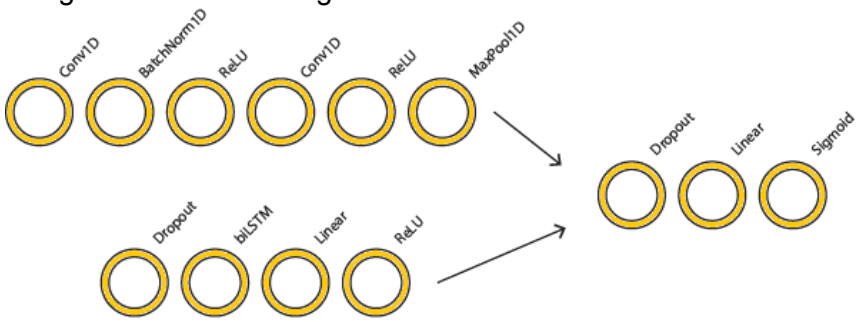


Figure 6 – The architecture of our model: CNN-biLSTM-FFN

3.2 Network's performance measure

In order to evaluate our model performance on test datasets, we used Matthew's correlation coefficient (MCC). Compared to other statistical evaluation rates for binary classification tasks, such as accuracy and F_1 score which may produce overoptimistic inflated results, especially on imbalanced datasets, MCC is a more trustable statistical rate. MCC shows a good result only when predictor indicates good results in all of four confusion matrix estimators [6]. The equation of MCC is as below:

$$MCC = \frac{TN.TP - FN.FP}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

MCC ranges from -1 to +1. MCC equal to +1 means the model performance was completely true. MCC equal to 0 means that the model performance is no better than random. When the MCC is -1 means that there is a total disagreement between predictions and the real values.
We also used Area Under the Curve (AUC) to visualise the performance of our model.
The probability that a model ranks a random positive sample over a random negative one is AUC of the model [7]. In this project we assumed a good AUC is over 0.75 intuitively.

4 Results

The performance of our model on three different datasets are presented in table 2

	D1 dataset	D2 dataset	D3 dataset
Model training parameters			
Number of epochs	30	30	30
Criterion	Binary Cross Entropy	Binary Cross Entropy	Binary Cross Entropy
Optimizer	SGD	SGD	SGD
Learning rate	10^{-2}	10^{-2}	10^{-2}
Weight decay	10^{-3}	10^{-3}	10^{-3}
Model evaluation results			
MCC of training set	0.455	0.921	0.901
MCC of test set	0.087	0.693	0.692
AUC of training set	0.68	0.95	0.93

AUC of test set	0.58	0.81	0.81

Table 2 – Training parameters and the model performance on D1 dataset (with 54 features for every amino acid), D2 (with the first 27 features) and D3 (embeddings resulted from ESM-1b approach)

The AUC curves on train and test sets of D1 and D2, are illustrated in the figure 7.

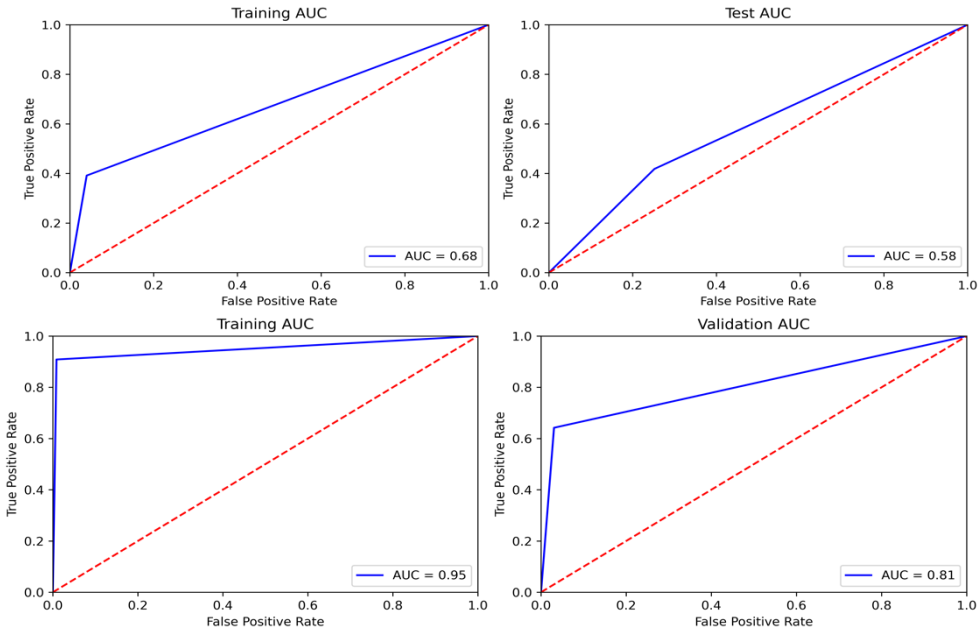


Fig 7 – AUC of model on D1 (top plots) and D2 (down plots) datasets.

The results can be found here: https://github.com/Hypertyz/tcr-pmhc

5 Conclusion

In this project we used CNN-biLSTM architecture in our model. Protein sequences and their interactions are like natural languages with 20 alphabets (i.e., there is only 20 different amino acids). In many studies fusing CNN to attention-based biLSTM resulted in achieving good models, namely in NLP studies, stock price predictions, interpretation of electro cardiogram rhythm and even covid-19 infection detection [8]–[11]. In this project we used the mentioned architecture as it showed good results in NLP modelling and we expected that including the energy terms derived from FoldX and Rosetta global improve the model performance. Interestingly, the model performance with all features (54 attributes) was poor, whereas with removing Rosetta global energy term and FoldX, we could get a model with high performance in predicting the bindings. This happened probably because the values of global and FoldX energies were the same between all entries. The repetitive values decrease the differences in data points and make it hard to find a model that can effectively separate two classes. It also increases the risk of overfitting. ESM-1b model transformed our 3D dataset to a 2D dataset so that each entry had 1280 features. As ESM model has been trained on a huge number of protein datasets, we expected a better performance. Indeed, ESM consider many important aspects and determining factors in binding two amino acid sequences such as the hydrophobicity and the spatial configurations. As we expected, the model trained more effective and the performance was remarkably better with ESM embeddings compared to the original dataset. However, with removing repetitive values of features 28 to 54, the performance of model was the same as ESM embeddings, even slightly better. We didn't prepare an ESM embedding using the first 27 features. We probably could get a higher performance with ESM embeddings if we did ESM on the D2 dataset. We also expect better generalisation capabilities on a larger dataset. The dataset was relatively small and included many repetitive peptides which made the network prone to overfitting. We tried to overcome this problem by adding dropout and weight decay.

References:

[1] K. Dhusia, Z. Su, and Y. Wu, “A structural-based machine learning method to classify binding affinities between TCR and peptide-MHC complexes,” *Molecular Immunology*, vol. 139, pp. 76–86, Nov. 2021, doi: 10.1016/J.MOLIMM.2021.07.020.
[2] S. McComb, A. Thiriot, B. Akache, L. Krishnan, and F. Stark, “Introduction to the Immune System,” *Methods in molecular biology (Clifton, N.J.)*, vol. 2024, pp. 1–24, 2019, doi: 10.1007/978-1-4939-9597-4_1.
[3] D. K. Cole *et al.*, “Dual molecular mechanisms govern escape at immunodominant HLA A2-restricted HIV epitope,” *Frontiers in Immunology*, vol. 8, no. NOV, Nov. 2017, doi: 10.3389/FIMMU.2017.01503.
[4] A. Rives *et al.*, “Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences,” *bioRxiv*, p. 622803, Dec. 2020, doi: 10.1101/622803.
[5] N. Kriegeskorte and T. Golan, “Neural network models and deep learning,” *Current Biology*, vol. 29, no. 7, pp. R231–R236, Apr. 2019, doi: 10.1016/J.CUB.2019.02.034.
[6] D. Chicco and G. Jurman, “The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation,” *BMC Genomics*, vol. 21, no. 1, pp. 1–13, Jan. 2020, doi: 10.1186/S12864-019-6413-7/TABLES/5.
[7] J. M. Iii, “ROC and AUC with a Binary Predictor: a Potentially Misleading Metric,” 2019, doi: 10.1007/s00357-019-09345-1.
[8] M. Rhanoui, M. Mikram, S. Yousfi, and S. Barzali, “A CNN-BiLSTM Model for Document-Level Sentiment Analysis,” *Machine Learning and Knowledge Extraction 2019, Vol. 1, Pages 832-847*, vol. 1, no. 3, pp. 832–847, Jul. 2019, doi: 10.3390/MAKE1030048.
[9] W. Lu, J. Li, J. Wang, and L. Qin, “A CNN-BiLSTM-AM method for stock price prediction,” *Neural Computing and Applications*, vol. 33, no. 10, pp. 4741–4753, Nov. 2020, doi: 10.1007/S00521-020-05532-Z/TABLES/3.
[10] X. Xu, S. Jeong, and J. Li, “Interpretation of Electrocardiogram (ECG) Rhythm by Combined CNN and BiLSTM,” *IEEE Access*, vol. 8, pp. 125380–125388, 2020, doi: 10.1109/ACCESS.2020.3006707.
[11] M. F. Aslan, M. F. Unlersen, K. Sabanci, and A. Durdu, “CNN-based transfer learning–BiLSTM network: A novel approach for COVID-19 infection detection,” *Applied Soft Computing*, vol. 98, p. 106912, Jan. 2021, doi: 10.1016/J.ASOC.2020.106912.