Technical University of Denmark

Department of Health Tecnology

22125 Algorithms in bioinformatics

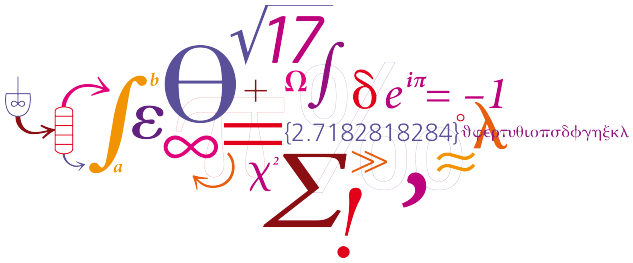
COMPARATIVE STUDY OF PSSM, ANN, AND SMM FOR PEPTIDE MHC BINDING

Students:

Paul Jeremy Simon (s202592)

Maria Vinh Thuy Tien Ta (s134042)

Huijiao Yang (s202360)



June 28, 2021

Abstract

Affinity-binding prediction of the MHC: peptide complex has long been an important research area for several purposes in the medical industry. For that reason, many computational tools have been developed to predict MHC class I binders. Even though the model of these methods has high performance, there is a low explanation (black box problem) as the pattern for why some peptide binds to the MHC are not completely understandable and predictable. In this study, the focus is made on comparing three prediction methods, two matrix-based PSSM and SMM and one neural network (ANN). To prevent overfitting the methods is evaluated via cross-validation in order to further understand the pattern for MHC binding tendencies. In general, ANN outperforms the two matrix-based prediction methods due to the ability to train on small data. However, the quality of training data plays a major role in how each methods performs overall as the models has shown to be sensitive to the distribution of data and good distributed data can therefore mean a slightly different outcome in prediction.

1 Introduction

Major Histocompatibility Complex (MHC) plays an important role in the process of antigen-presentation. In the cell, the protein of the antigen is split into small peptides which can be presented by MHC for T-cell recognition, where this interaction would trigger cell-killing or downstream signaling against the antigen. Hence, the binding of peptides to MHC molecules is necessary for T-cell recognition. The accuracy for predicting peptide:MHC binding is therefore useful for the development of vaccines, therapeutics and diagnostics for infectious and autoimmune diseases, allergy and cancer [1].

The MHC family consists of three subgroups: class I, II and III and it can be found in all vertebrates. The human MHC region is located on chromosome 6, and its class I and II are also called the HLA (Human Leukocyte Antigen) complex. MHC proteins are polymorphic which means they have many variants. In our study, we focus on 35 alleles for HLA-A and HLA-B among 9 classes of HLA. Since MHC residuals with different alleles encoding have specific clefts to allow special types of residues to enter, each MHC variant can bind special peptides which can be predicted by different methods through binding affinity [2].

In the last decades, many data-driven methods have been developed to predict peptide:MHC binding affinity. Compared with traditional methods, the model of these methods has higher performance but low explanation (black box problem) since it has too many parameters. Besides, overfitting is also very common which means that the model can have excellent performance on the training set, but poor performance on the test set due to the capture of noise in data.

The work of this report focuses on three methods. Two of them are matrix-based predictions, namely Position-Specific Scoring Matrix and Stabilized Matrix Method (thereafter called PSSM and SMM) which generate scoring matrices and the last one is Artificial Neural Network (ANN) which works on networks through the succession of linear transformations by weighted product and nonlinear transformations by activation function. To prevent overfitting of these models, we used two layers cross-validation (k=5) to generate and evaluate predictions for each of the three methods. We would then build an experiment to compare the three methods to determine which method is the best in terms of prediction performance. We also discuss the strengths and weaknesses of all these methods.

2 Materials and methods

2.1 Data

The dataset has been downloaded from the SMM exercise, and it is already split into 5 files. The raw dataset has been given by [3]. Each final file is composed of two columns, the first one is the sequence of the peptide and the second is the target value that was measured by lab experiment. The formula to convert raw values to final values is:

$$\text{Target value} = 1 - \frac{log(IC_{50})}{log(50000)}$$

where the lower the IC_{50} is, the better the peptide can bind. The range of target values is $[0, 1]$, where 0 is the worst binder and 1 is the best binder. As mentioned in [3], a cutoff to differentiate good binders from bad ones was defined. It is $IC_{50} = 500nM$, or in the new scale, Target value ≈ 0.426. This cutoff was used to find which peptides should be used as training set for the PSSM method and to compute the AUC (see thereafter). The distribution plot of the targets is shown on Figure 1 and highlights the fact that the distribution has a bias in the most of files. The preprocessing of the data is not included in our work and we assume models are reliable after training through cross-validation so that we can compare them.

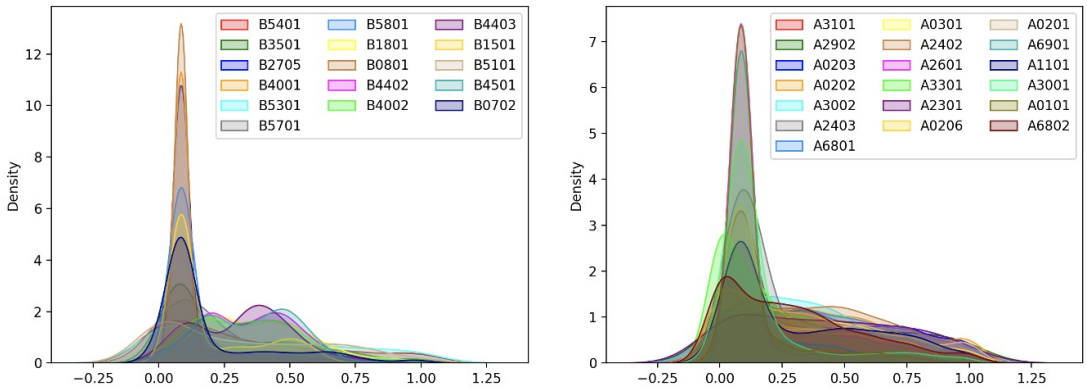


Figure 1: Distribution of data sets(x-axis is the value of target).

In order to facilitate the subsequent analysis, the distribution plots of the different files were divided into two classes, single peak or significantly high peak (peak point above 3.0), also called nested type and double peaks or width distribution.

Table 1: Distribution classes.

Nest	A0101, A3001, A6901, A2601, A0301, A3101, A2403 B0702, B0801, B5801, B4001, B2705, B1501
Width	A1101, A0201, A3301, A3002, A2902, A0206, A6802, A0203, A0202, A6801, A2402, A2301 B4501, B5101, B5701, B4403, B4002, B4402, B1801, B5301, B3501, B5401

As an example, we selected B3501 and A0201 as width data and B2705 and A0101 as nested data and represented logo plots of position importance thanks to the PSSM method (Figure 2 and Figure 3). In both classes, position 2 has the most important motif.

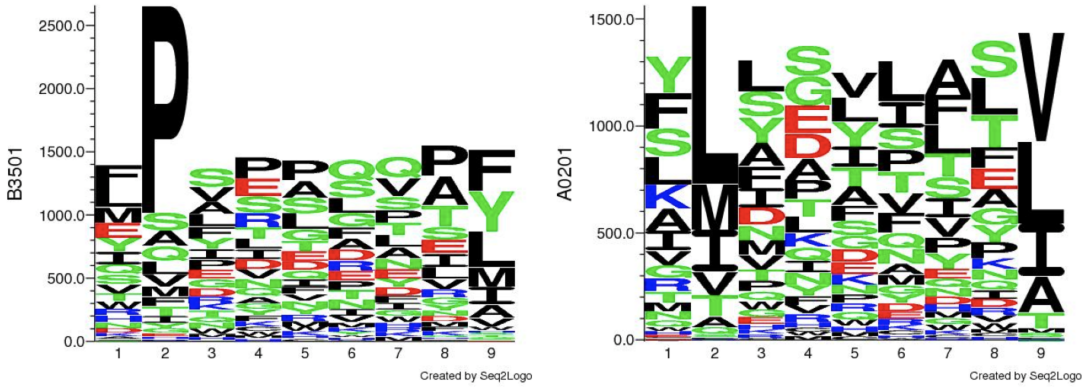


Figure 2: Width data logo plot of PSSM.

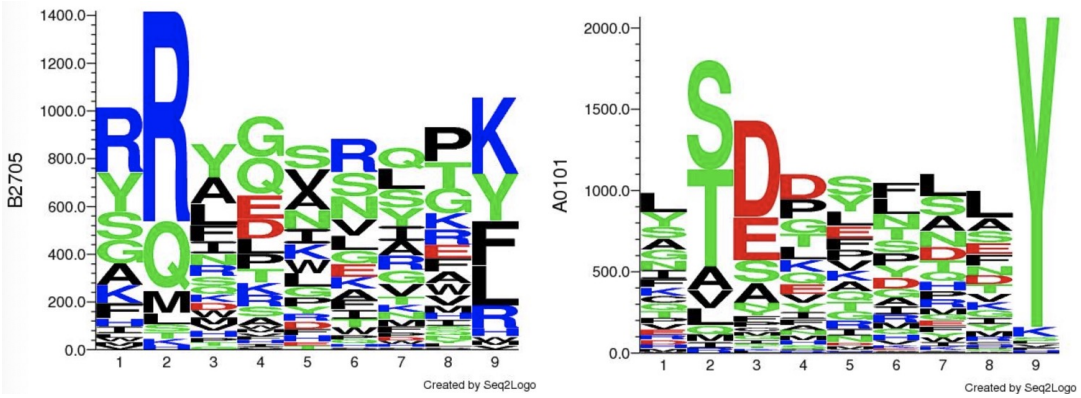


Figure 3: Nest data logo plot of PSSM.

2.2 Methods

PSSM is a statistics method to compute the likelihood for bases of each site in binder peptides to predict if a new peptide is a binder. On the other hand, SMM and ANN are machine learning methods which can learn parameters of models from training to predict a score reflecting how much a peptide is binding. SMM is a linear model whereas ANN includes non-linear steps. Theoretically, ANN is the best method for predicting performance level but it isn't easily explainable by the human.

2.2.1 PSSM

PSSM or Position-Specific Scoring Matrix specifices the scores for observing particular amino acids or nucleotides at specific positions. The score represents the quasi-probability affinity score of the peptide as a potential MHC binder, using three strategies: 1) sequence similarity, 2) motif incorpation and 3) matrix formulation [4]. The matrix is calculated from a set of peptides known to bind to a specific MHC molecule. Since peptides that bind to a given MHC complex molecule share sequence similarity, PSSM becomes quite useful in predicting whether a peptide can bind and thereby predicting possible T cell epitopes in a protein. However, PSSM is not a machine learning method as it does include any step of "learning", in other words there is no way to enhance its performance.

In PSSM, the weight matrix is calculated through the estimation of pseudo counts and frequency. This

formula can be described as followed:

$$p_a = \frac{\alpha \cdot f_a + \beta \cdot g_a}{\alpha + \beta}$$

Weight on prior (β) plays a significant role, when the dataset is small and biased. Practically, this parameter can typically vary between 50-200. In this study, β has been set to 50 throughout the prediction with PSSM. To account for data redundancy, the weights of all peptides are found with the following heuristics: $w_{kp} = \frac{1}{rs}$ where r is the number of different amino acids in the column p, and s is the number occurrence of the amino acid in that column. The α parameter is the mean of r in each column - 1. Pseudo counts are found with $g_b = \sum_a f_a \cdot q_{b|a}$ where $q_{b|a}$ is read with the BLOSUM 62 matrix.

2.2.2 SMM

Stabilized Matrix Method or SMM is a Ridge regression, that is, a linear regression where an additional penalty error value has been added through a multiplication of a regularization parameter with the norm of the weight of the models. This new error value consists of a lambda parameter that suppresses the effect of noise in the data and aims at avoiding overfitting by forcing the weights to have a small value. The error function is given by the following formula:

$$E = \frac{1}{2}\sum_{i=1}^{N}(y^i_{target} - y^i_{predicted})^2 + \frac{\lambda}{N} \cdot \sum_l w_l^2$$

where N is the length of the vector, λ is the regularization parameter and w_l are the weights. This method can be used with optimization procedures like gradient descent which is used for finding a local minimum and Monte Carlo, which by making random change to weights and by accepting to get slightly worse from an iteration to another can compute an approximation of the global minimum. SMM is an explainable method since linear regression model is self-explainable through observing coefficients [5].

2.2.3 ANN

Artificial Neural Network or ANN is a model to predict targets with high performance through transforming from input layer to output layer based on sum of weighted products and non-linear activation functions. Each layer is composed of units, also called neurons and from one layer to another, all neurons are connected and weighted. There are at least 2 layers: an input layer which has the length as in the input and an output layer. In our case, the output layer is of size one because one number has to be predicted. The data flow is transformed through a non-linear activation function. There are many types of these functions, the one used in this study is the sigmoid which outputs numbers in the $[0, 1]$ interval and is defined by:

$$\phi(z) = \frac{1}{1 + e^{-z}}$$

According to the direction of data flow, two procedures can be identified: Forward Propagation (FP) and Back Propagation (BP). FP is the calculation of the outputs of all nodes, based on the input and the weights. BP serves as corrector of parameters through gradient descent based on a loss function. This model has a high number of hyperparameters, that is to say, parameters that are chosen by the user and that are not learnt by the algorithm such as the number of units for each hidden layer. By modifying those hyperparameters, the predicted performance of the model can be increased. However, hyperparameters tuning is not handled in the work presented here. Since there are many parameters, ANN is more likely to overfit. On the other hand, ANN has high performance in this regression task to predict the degree of binding. Theoretically, ANN can fit any actual situation since the network is flexible and it can represent complexity of the real. In our case, the model has one hidden layer consisting of 5 units.

SMM and ANN are machine learning methods, implying that they are working only on numbers. The inputs from the dataset are peptides of length 9, therefore, the letters have to be encoded to numbers. Each amino acid is represented by 20 numbers. The numbers used depend on the encoding scheme chosen. Here, we chose to use the BLOSUM 50 encoding, meaning that each amino acid is encoded by the row of the BLOSUM 50 matrix corresponding to its letter. This type of encoding ensures there are not too many zeros in the encoding of the input contrary to a sparse encoding. This encoding scheme was therefore applied to both methods.

Both in SMM and ANN, Stochastic Gradient Descent (SGD) was used to update parameters to compute the local minimum. It means that when the model was trained, we input data points one by one randomly. The formula is the following:

$$w_i^{'} = w_i - \epsilon \cdot \frac{\partial E}{\partial w_i} \text{ where } E = \frac{1}{2}(y^i_{target} - y^i_{predicted})^2$$

In the gradient descent process, there are different loss functions. In this case, MSE (Mean Square Error) is used to find differences between predicted value and real value to update the parameters of model through back propagation process since the target is a continue variable. The parameter ϵ is a positive real number referred to as learning rate.

2.3 Metrics for comparison

2.3.1 Root Mean Square Error (RMSE)

The main error metric chosen to compare the methods is the Root Mean Square Error (RMSE). It is given by the following formula :

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y^i_{target} - y^i_{predicted})^2}{N}}$$

where N is the number of points to predict. This metric was chosen because it has the same dimension as a distance, which makes it easy to understand at first sight. It also allows all the results to be on the same scale. On the contrary, with the Mean Square Error (MSE), the values of errors would have ranged from approximately 0 to more than 100, which would have made any plot almost unreadable.

2.3.2 Area Under the Curve (AUC)

The Receiver Operating Characteristic (ROC) curve is used to measure the ability of predictions to classify peptides into binders ($IC_{50} < 500nM$) or non-binders ($IC_{50} > 500nM$). When choosing a cutoff for the predicted value, the peptides used for prediction are set into a positive and negative subset. This allows the calculation of the number of true-positive and false-positive predictions. The ROC Curve is the plot that visualize the rates between these two predictions with the rate of true positive (y-axis) against the rate of false negative (x-axis) [3]. The information used to make a ROC Curve is given by the following formula:

$$\text{True Positive Rate} = \frac{TP}{TP + FN} \text{ and False Positive Rate} = \frac{FP}{FP + TN}$$

where TP is the number of true positive, FN is for the number of false negative, FP is the number of false positive and TN is the number of true negative in the experimental dataset. The area under the curve (AUC) is then used to determine to the range in which the classification of binders by the method occurs randomly (AUC = 0.5) to perfectly (AUC = 1) [6].

2.3.3 Accuracy

Another simple metric is defined to compare the results, it is the accuracy. This other metric is also working on binary data. The accuracy is defined by $\frac{TP+TN}{N}$. This gives a quick overview on how much a method can predict if a peptide is a binder, assuming that the cutoff for these methods is the same is the cutoff defined in the target dataset.

2.4 The algorithm

All of the scripts were written in Python and the final ranking plot was made with Excel. The work is divided into 4 files, three of them being method specific. They include two functions, one for training and one for the evaluation. The main script is implementing the cross-validation by calling the three other scripts. The algorithm parameters are given in Table 2.

Table 2: Parameters chosen for each method.

Method	β	Optimization function	λ	ϵ	Epochs	Encoding scheme	Number of hidden units
PSSM	50	-	-	-	-	-	-
SMM	-	Gradient descent	0.01	0.01	100	BLOSUM 50	-
ANN	-	Gradient descent	-	0.05	100	BLOSUM 50	5

The main algorithm is written as pseudo code in Algorithm 1. To calculate the evaluation error, a nested cross-validation was chosen. This method was chosen because it avoids overfitting for small test files, which is the case here because some file have down to 59 peptides and 11 binders. Briefly, the outer cross validation is 5-fold and the inner cross validation is 4-fold. For each data file, 1/5 of the file is consecutively used as an evaluation set which is never parsed before the actual evaluation of the model. The 4 other parts of the file undergo a classic cross-validation. Each inner fold will serve as training set to stop the learning early and therefore avoid overfitting. During this process, 4 models are trained (1 for each test set). Finally, they are combined to predict the target values on the evaluation set. Once all evaluation sets have been used, the RMSE and AUC are computed by applying the formula previously mentioned. It is worth noticing that the PSSM is not a machine learning method and therefore it does not require any test set for early stopping like the other two methods. That is why this method is "trained" outside of the inner loop.

3 Results

The main algorithm took approximately 15 hours to run. Most of the time was used to train the ANN models. Indeed, our implementation of this algorithm is somewhat basic and does not include any optimization. The output of the algorithm are plots made with the matplotlib library. The first one is showing the number of binders for each MHC type against RMSE errors for the three compared methods. It can be seen on Figure 4.

Algorithm 1 Main program

input: HLA files
Define threshold for a binder (0.426)
Load the dataset
for each HLA file **do**
  Create a 5 fold partition of the data
  for each outer $fold_i$ **do**
    $fold_i$ is the evaluation set
    for each inner $fold_j$ **do**
      $fold_j$ is the test test
      all other folds (except i and j) constitute the train set
      Train SMM and ANN
    end for
    Construct PSSM model with peptides defined as binders on all folds except i
    Predict the score on $fold_i$ with SMM and ANN by an average of the 4 trained models
    Predict the score on $fold_i$ with PSSM
  end for
  for each method (PSSM,SMM,ANN) **do**
    Calculate RMSE between prediction by the method and target value in the dataset
    Calulate AUC and Accuracy for the three models thanks to the predictions and binder threshold
  end for
  Count the number of binders in the current dataset
end for
Plot Number of binders against RMSE for the three methods
Plot Number of binders against AUC for the three methods
Plot the ranks for the three methods according to Accuracy and RMSE

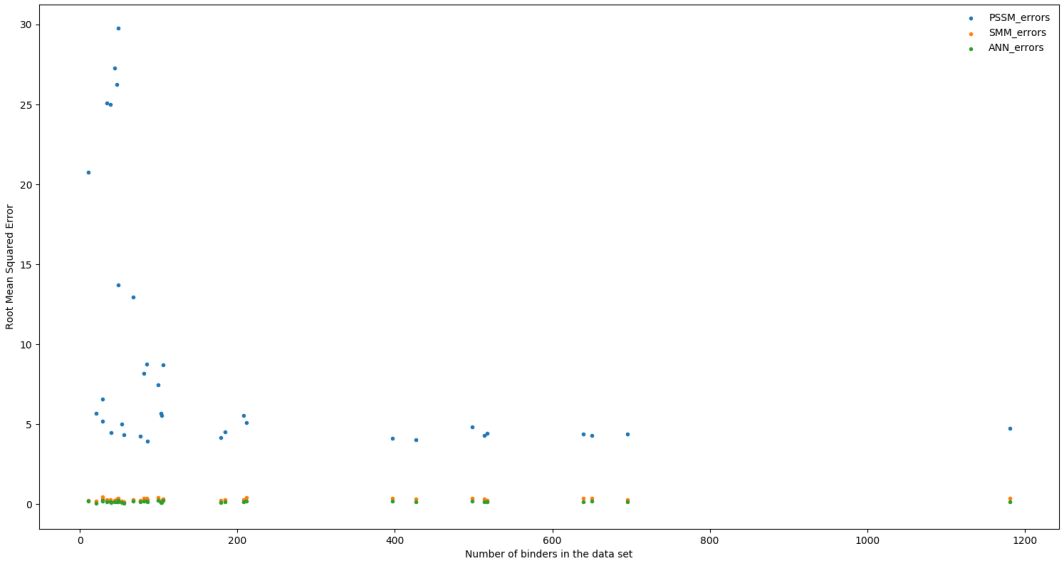


Figure 4: Number of binders against RMSE for all three methods.

At first glance the PSSM method had the largest amount of error in all cases. It is yet not clear to see how this is compared to SMM and ANN. Therefore a second plot is made with the two other methods in Figure 5.

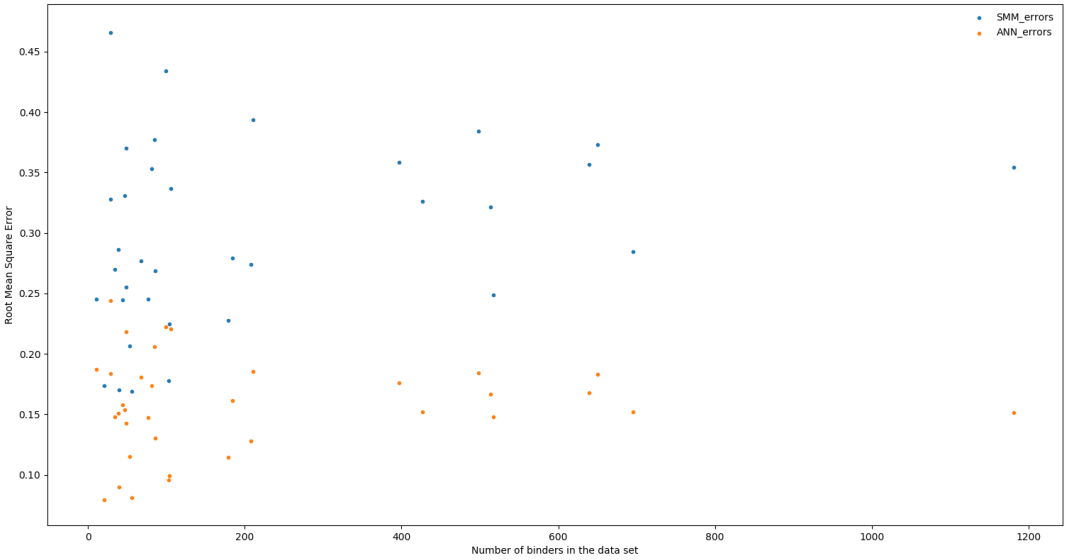


Figure 5: Number of binders against RMSE for ANN and SMM.

On the plot, it is not always obvious which method is the best for each HLA allele type. Therefore, a ranking of the methods was made by simply comparing the errors for each HLA allele type. It turns out that for all 35 types, ANN is always performing better than SMM which in turn is performing better than PSSM. This result was expected because as explained before, the score predicted with the PSSM matrix has really larger bounds than the ones predicted with ANN and PSSM. Moreover, ANN is the most flexible model so the learning is enhanced.

To overcome the fact that the RMSE always ranked the methods in the same way, a similar plot was drawn by replacing RMSE with AUC against the number of binders. This could potentially allow PSSM to be compared in an other way. Even if the score predicted can be really far from the evaluation score, there is a possibility that the AUC of this model is good. This would mean that PSSM can distinguish between binders and non binders, provided that one finds the good cutoff score. The plot comparing AUC is drawn on Figure 6.

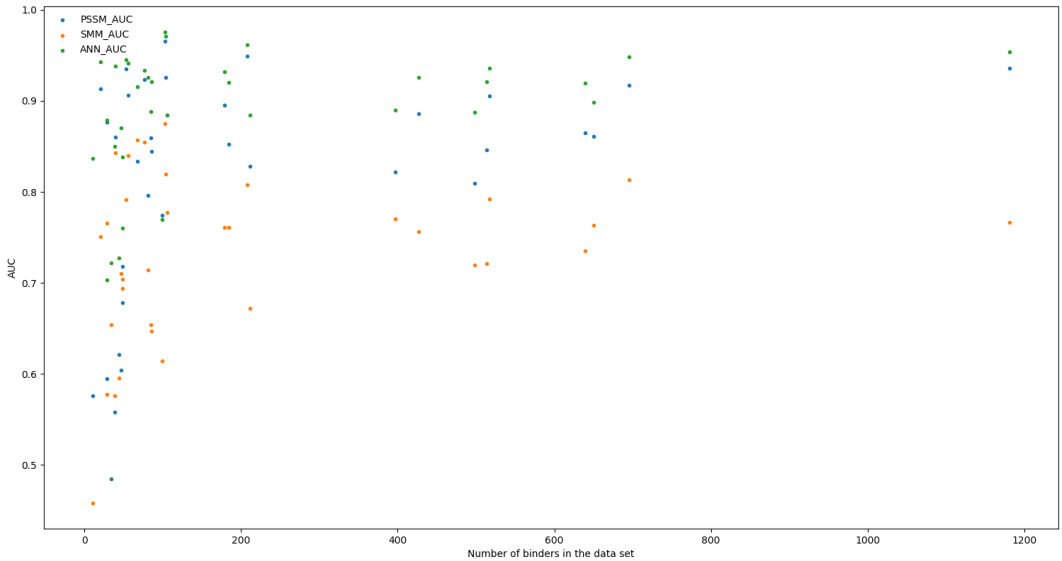


Figure 6: Number of binders against AUC for all three methods.

	Accuracy				RMSE	
PSSM	SMM	ANN	Rank	PSSM	SMM	ANN
B0702	B0801	B0801	1	A6901	B2705	B0801
B0801	B4001	B4001	2	A3101	B4001	B2705
A0201	B2705	B2705	3	A6802	B0801	B4001
A3001	A2601	A0101	4	B1501	A0101	A0101
A0101	A0101	A2601	5	A3001	A2601	B5801
A1101	A3001	B5801	6	A0206	B5801	B1501
A2403	B5801	A3001	7	A0202	B1501	A2601
A2402	A6901	B0702	8	B2705	B4402	B0702
B3501	A2403	A6901	9	A0203	A3001	A6901
A0202	A3301	A2403	10	A1101	B5701	B4501

Figure 7: Top 10 rank table of Accuracy (left) and RMSE (right). The nested datasets are noted by blue.

This time, the rankings are different and the PSSM seems to be performing better than the SMM overall. In 28/35 HLA allele types, ANN is performing better than PSSM which is performing better than SMM. In 5/35 allele types, ANN is performing better than SMM which is performing better than PSMM. Finally, in 2/35 allele types, PSSM is performing better than ANN which is performing better than SMM. This means that, for a really large majority, PSSM is performing better than SMM. This result seems be counter-intuitive because the PSSM is not a machine learning method, which means a linear regression should be able to outperform it in most cases. This result can be explained by the way SMM is trained in our script which is explained in Discussion.

Although the observed results of the RMSE are the ones expected, the rank of performance in data sets is different. So we made a robustness check, in other words, we performed a further analysis to try to explain the results in the light of the two distributions that were identified previously. The results show that nested distribution data sets have better performance in most cases for each of the three methods Figure 8. On the y axis, each label is created with the type of distribution (N:nest, W:width) followed by the number of binders in this distribution. The label is defined as rank number, 1 is the best and 35 is the worst. This figure shows that the top ranks of performance based on RMSE are mostly found for the nested data distribution and the low ranks are obtained with width data. For some of the nested data sets, we observe a better rank for SMM and ANN than for PSSM. Especially, ANN is more sensitive for nested data than SMM which is better than PSSM. For example, by selecting top 10 ranks, 5/10 in PSSM, 8/10 in SMM and 9/10 in ANN are nested data sets, as one can see on Figure 7. For the best rank in each method (A6901 for PSSM, B2705 for SMM and B0801 for ANN), all are nested distribution and B0801 have less binders which could indicate ANN can capture weak signal due to its sensitivity.

Similarly, we also have Accuracy ranks. There are some similarities with RMSE ranks. For example, by selecting top 10 ranks, 5/10 in PSSM, 9/10 in SMM and 10/10 in ANN are nested data sets (Figure 7). However, there are some differences between Accuracy and RMSE ranks, especially in PSSM method. By comparing the ranks with the two metrics, we found there are some similarity in files: 8/10 files are same in ANN, 6/10 in SMM and only 3/10 in PSSM. It indicates that model trained by PSSM has good performance in RMSE even if it can be worse in terms of Accuracy.

PSSM is stable in different distribution types of data because it is constructed on a binary result but it can be unstable in different evaluation methods. On the contrary, the two machine learning methods are stable in different evaluation matrix, this could be because learning process can get stable and have high performance.

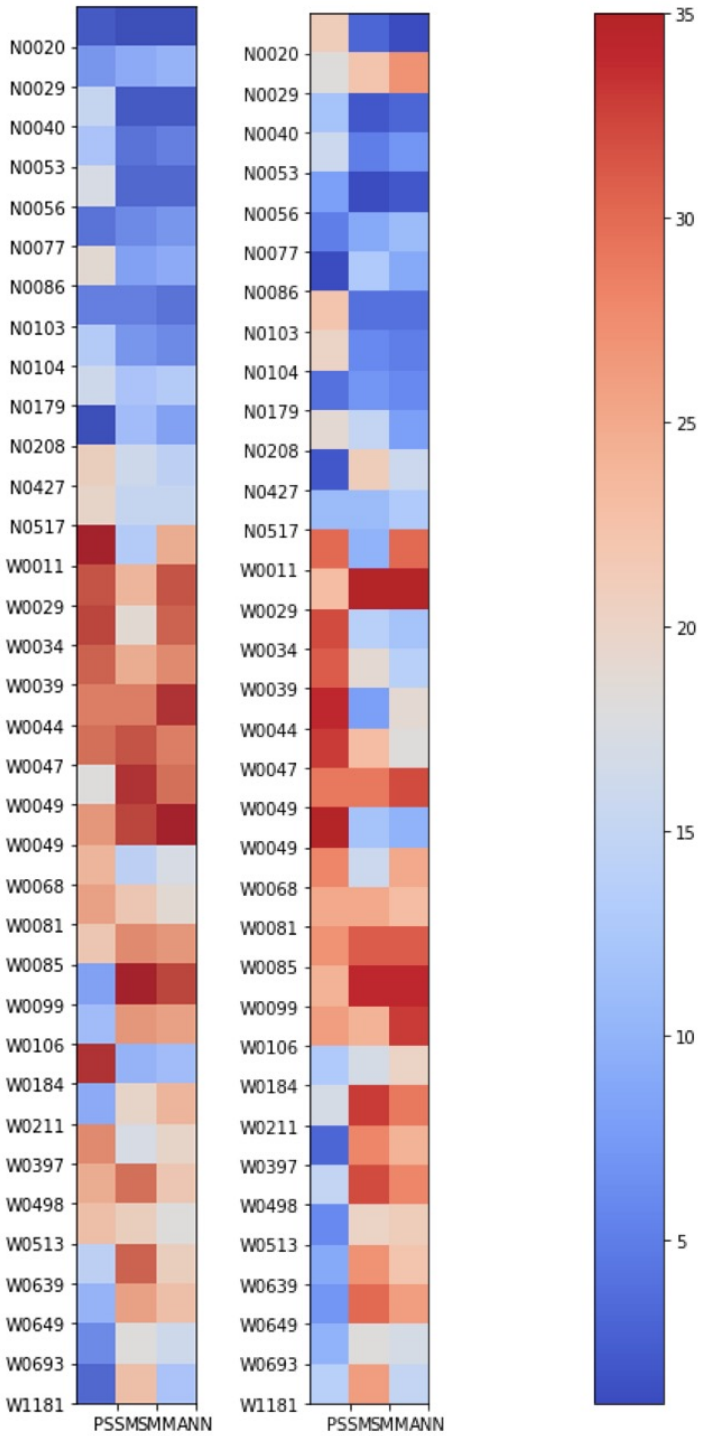


Figure 8: Rank of Accuracy (left) and RMSE (right). The method performs best for the given metric when it is represented in blue (top ranks).

4 Discussion

The outcome has proven that ANN has the highest prediction performance and is stable even through different types of evaluation. ANN is outperforming both PSSM and SMM is an overwhelming majority of datasets.

However, in comparison to the results found in other papers, many cases have shown when ranking using the AUC that SMM seems to perform poorly compared to PSSM and ANN than what has been predicted in earlier studies. The choice of encoding scheme could be one of many causes of such result as the BLOSUM 50 scheme has only been used in our experiment. Moreover, our result also showed that PSSM has been outperforming ANN. These scenarios seem very unlikely and could be caused by various reasons.

One reason could be due to data redundancy. While analyzing the data sets used for generating these models, it has been observed that it has a lot of biased information such as zeros and repetitive values for each peptide. To solve this problem, one could consider using data redundancy algorithms like Hobohm to sort and extract unique data from biased ones to improve the training and test sets. This could also explain the different distribution types of the data sets.

In this study, the parameters have been kept fixed and it is believed some strategies could be considered to improve the models in this study. Firstly, one could try to change the hyperparameters that would cause less possibility of overfitting like β for the PSSM, λ for SMM, or modifying the error function for ANN and SMM to see whether this can improve the prediction models in some ways. The β in PSSM is interesting as it can enhance the performance when the dataset is small and biased, hence improve the ability to predict binders in smaller data sets. λ could regulate the SMM model to find the "sweet spot" before overfitting. This process is known as grid search and is already widely used in most machine learning methods.

Another way could be to change the encoding scheme for SMM to a combination of BLOSUM 50 and sparse scheme as this seems to have the best performance according to other studies [7]. Moreover, a comparison between using Gradient descent vs. Monte Carlo could be interesting to see which one is better in optimizing the SMM model. When finding the optimal parameters and optimization procedures, one could compare these three methods to other newer prediction tools such as Easypred and NetMHC in order to improve them even further.

In this study, RMSE and AUC were used to compare the models based on the number of binders in the dataset. The size of the dataset as well as the ratio between the number of binders and the size of the dataset have not been analyzed here and could lead to future research. In addition to this, the results could be enhanced by including statistical metrics such as a Student t-test for the regression part to analyze the distribution of errors for the regression method or Mc Nemar's test if we consider only a classification work (binder versus non-binder).

To expand the work, a comparison could have been made between PSSM and SMM with a focus on how well they predict which position will be the most crucial for the peptide to bind. This would require to compare how final weight matrices are constructed and then set a new an experiment to compare them to what is found by lab experiments.

References

[1] Yohan Kim, J. Sidney, S. Buus, A. Sette, M. Nielsen, and Bjoern Peters. Dataset size and composition impact the reliability of performance benchmarks for peptide-mhc binding predictions. *BMC Bioinformatics*, 15, 2014.

[2] Marek Wieczorek, Esam T. Abualrous, Jana Sticht, Miguel Álvaro Benito, Sebastian Stolzenberg, Frank Noé, and Christian Freund. Major histocompatibility complex (mhc) class i and mhc class ii proteins: Conformational plasticity in antigen presentation. *Frontiers in Immunology*, 8:292, 2017.

[3] Bjoern Peters, Huynh-Hoa Bui, Sune Frankild, Morten Nielsen, Claus Lundegaard, Emrah Kostem, Derek Basch, Kasper Lamberth, Mikkel Harndahl, Ward Fleri, Stephen S Wilson, John Sidney, Ole Lund, Soren Buus, and Alessandro Sette. A community resource benchmarking predictions of peptide binding to mhc-i molecules. *PLOS Computational Biology*, 2(6):1–11, 06 2006.

[4] Limin Jiang, Hui Yu, Jiawei Li, Jijun Tang, Yan Guo, and Fei Guo. Predicting MHC class I binder: existing approaches and a novel recurrent neural network solution. *Briefings in Bioinformatics*, 06 2021. bbab216.

[5] Morten Nielsen. Stabilized matrix method. `http://www.cbs.dtu.dk/courses/27625.algo/presentations/SMM/SMM.pdf`. Accessed: 2021–18-06.

[6] Morten Nielsen. Performance measures. `http://www.cbs.dtu.dk/courses/27625.algo/recordings/Performance_measure.mp4`. Accessed: 2021–18-06.

[7] Morten Nielsen. Artificial neural network 1. `http://www.cbs.dtu.dk/courses/27625.algo/presentations/NN-1/NNtalk_w_answers.pdf`. Accessed: 2021–21-06.