Technical University of Denmark

Comparing methods for MHC peptide binding using PSSM, ANN, and SMM

27625 Algorithms in bioinformatics

Hand in date: 20-01-2016

Written by:
Kamilla Kjærgaard Jensen, s112819
Marie Louise Jespersen, s152153
Patrick Zecchin, s150701

DTU Systems Biology
Department of Systems Biology

Abstract

Predicting binding peptides to major histocompatibility complex (MHC) class I molecules is important for addressing many immunological questions and for vaccine development. Finding a good prediction tool is therefore fundamental.

The aim of this report was to investigate the performance of three different prediction methods, namely two matrix-based methods, Position Specific Scoring Matrix (PSSM) and Stabilisation Scoring Matrix (SMM), and the Artificial Neural Network (ANN). The investigation was performed with special emphasis on the impact of the size of the dataset on the prediction. This was done using a five-fold cross-validation to assure the correct performance measurements. The results presented here show that the ANN is the most successful predictor both on small and large dataset followed by PSSM and lastly SMM.

Introduction

In higher organisms, major histocompatibility complex (MHC) class I molecules (from here on referred to as MHC molecules) are present on nearly all cell surfaces [1]. The function of these MHC molecules is to present peptides derived from proteins found within the cell to the cytotoxic T lymphocytes of the immune system (Figure 1), thereby allowing the immune system to detect infections or cancerous cells. Both self and nonself peptides are presented, but only peptides derived from nonself-proteins, such as viruses or bacteria, trigger an immune response leading to the destruction of the infected cell [1]. MHC bound peptides that trigger an immune response are termed T-cell epitopes. Identifying such epitopes is very important in the development and evaluation of peptide-based vaccines [2]. Many factors influence whether or not a peptide is a T-cell epitope, but one of the most important requirement is the peptides' ability to bind to the MHC molecule with a high affinity. Fortunately, binding affinity is relatively manageable to characterize experimentally and to model computationally, since the binding affinity depends on the amino acid sequence of the peptide. One issue that complicates this process is that there are many different MHC molecules, each with a distinct peptide binding specificity.

Over the years, many methods have been designed to predict possible T-cell epitopes. In this study three different prediction methods are investigated, an Artificial Neural Network (ANN) algorithm [3] and two matrix based prediction methods: Position Specific Scoring Matrix (PSSM) [4] and Stabilization Matrix Method (SMM) [5], [6].

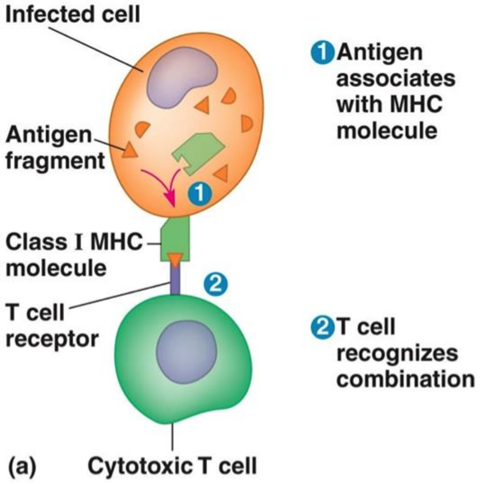

Figure 1: Overview of the interaction between a MHC class I molecule and a cytotoxic T lymphocytes. Adapted from http://drjosephtm.blogspot.dk/.

The first matrix-based method is the PSSM [4]. When making the PSSM the values are calculated from known binding peptides, including both the estimated and the background frequencies of the different amino acids in sequences. The formula used for calculating the weights is

$$W_{ia} = 2 \cdot \frac{log\left(\frac{p_{ia}}{q_a}\right)}{\log(2)}, \tag{1}$$

Here p_{ia} is the estimated frequency of amino acid a at position i in the motif and q_a is the background frequency of amino acid a. p_{ia} is calculated as

$$p_{ia} = \frac{\alpha \cdot f_{ia} + \beta \cdot g_{ia}}{\alpha + \beta}, \tag{2}$$

where α is the number of sequences minus one, β is an input parameter for the method, f_{ia} is the observed frequency of amino acid a in position i and g_{ia} is the pseudo count calculated as

$$g_{ia} = \sum_b f_{ib} \cdot q(a|b), \tag{3}$$

in order to consider previous knowledge of amino acid substitutions. Here f_{ib} is the observed frequency of amino acid b and $q(a|b)$ is the BLOSUM probability for substituting a to b.

The weight on pseudo counts is incorporated through a parameter termed β, cf. (2). The greater β, the greater weight on pseudo counts. When predicting peptide binding with this model the output is an arbitrary score for the peptide affinity, thus, it is not possible to compute a prediction error and a mean squared error (MSE) [7].

The second matrix-based method is the SMM [6]. In this method as well, the output is a matrix of weights for every position in the considered motif. The input is a vector corresponding to the sequence of the peptide, which can either be sparse or BLOSUM encoded [8]. At first the weights are randomly initialised, subsequently, the output is calculated by summing over the product of the input I and the weights w for each position i in the sequence

$$O = \sum_i I_i \cdot w_i \,. \tag{4}$$

The method then aims to optimize an error between the output and the target value. For updating the error, different numerical optimization methods can be used. In this study, gradient descent and Monte Carlo have been chosen. When using gradient descent the error function is

$$E_{per\ target} = \frac{1}{2} \cdot (O - t)^2 + \frac{\lambda}{N} \cdot \sum_i w_i^2 \,. \tag{5}$$

In this equation the error is calculated per target. The output value O is calculated as shown in (4), t is the measured IC50, λ is an input parameter, N is the number of sequences, and w is the weight. For Monte Carlo optimisation the global error is calculated as

$$E = \frac{1}{2} \cdot \sum_i (O_i - t_i)^2 + \lambda \cdot \sum_i w_i^2 \,. \tag{6}$$

Additionally, these two error functions include a penalty term, in which the sum of the squared weights identifies the model complexity. An input parameter λ decides how much influence this penalty term has on the overall error. The derivative of the error function for gradient descent is

$$\frac{\partial E}{\partial w_i} = (O - t) \cdot I_i + \frac{2 \cdot \lambda}{N} \cdot w_i \,, \tag{7}$$

and it is used to update the weights with the following formula;

$$\Delta w_i = -\epsilon \cdot \frac{\partial E}{\partial w_i} \,, \tag{8}$$

where ε is step length for the gradient descent algorithm. The values of the final weights will compose the matrix, which is the output of the method.

When using Monte Carlo optimisation the weights are initialised randomly. Hereafter, the Monte Carlo algorithm performs a random move for which new weights, a new output, and a new error are calculated using the functions shown in (4) and (6).
If the $\Delta E \leq 0$ the move is always accepted, but if $\Delta E > 0$, then the move is only accepted if a random number between zero and one is smaller than the current ΔE, according to

$$\Delta E = E_1 - E_0 \,, \tag{9}$$

$$P(accept) = min\left(1, e^{\frac{-\Delta E}{T}}\right). \tag{10}$$

Both the gradient descent and the Monte Carlo algorithms run for a given number of iterations to find the optimal solution. When predicting peptide binding with the SMM, the output is the predicted affinity for the peptide of interest [9][7].

The ANN is similar to the SMM in aiming to optimise the output to a target value [7]. Additionally, the input of this method is the sequence of the peptide and the measured affinity as well.

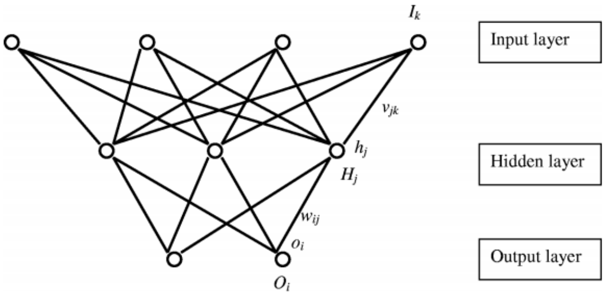

Figure 2: Schematic drawing of an ANN adapted from [4].

The ANN includes an extra layer compared to the SMM. This extra layer contains hidden neurons (Figure 2). The input to each hidden neuron is

$$h_j = \sum_k v_{jk} I_k \,, \tag{11}$$

where v_{jk} is the weight on the input k to the hidden neuron j and I_k is the value of the input neuron. The output from the hidden neuron is

$$H_j = g(h_j) \,, \tag{12}$$

where *g(h_j)* can differ, but the sigmoid function

$$g(x) = \frac{1}{1+e^{-x}} \quad , \tag{13}$$

is the most commonly used [7].

The input for the output layer is calculated as the input for the hidden layer, i.e. (11), using the output from the hidden layer and the corresponding weights. The error function

$$E = \frac{1}{2}\sum_i (O_i - t_i)^2, \tag{14}$$

is used to calculate the difference from the obtained result to the target.
If the error is greater than a threshold, the weights are updated through back propagation

$$\Delta v_{jk} = -\varepsilon \frac{\partial E}{\partial v_{jk}}. \tag{15}$$

A set of bias neurons can also be included in order to optimise the output by modifying the chosen function, *g*.

In order to avoid overfitting, early stopping of network training must be used, in which the training stops when the test set correlation decreases, and not when the training set performance decreases (example in Figure 3).

Materials and method

The data used in this report was obtained from [2]. Adjustments were made to exclusively include human MHC alleles and peptides with a length of nine amino acids. The human MHC molecules included in the dataset, as well as the number of sequences and binders are presented in Table 1. Data collecting is described in [2]. In the dataset all affinity measurements were converted with

$$1 - \frac{\log(IC50)}{\log(50000)} \ , \tag{16}$$

and these are the values used to represent affinities in this study.

Cross-validation

Prior to using the different prediction methods, the dataset for each molecule was divided into five approximately equally large groups, which were used to implement a five-fold cross-validation to generate and evaluate predictions for each of the models. From these five groups, one was left for validation of the prediction model, while the four remaining were used to train and test the model. Subsequently, the resulting model was used to predict the affinities of the peptides in the validation set. Repeating this for all five validation sets, each peptide in the original dataset was assigned a predicted score or affinity. The Pearson correlation coefficient (PCC) and Mean Squared Error (MSE) were calculated on an ensemble of the five validation sets (example of plotting in Figure 7). The results are shown in Table 1.

Optimisation

While training and testing each prediction model was optimised with regard to a parameter, *β* in PSSM, *λ* in SMM, and early stopping in ANN. The optimal parameter for each ensemble of test sets (corresponding to one validation set) was chosen to be the one with the highest PCC (example in Figure 3) and this parameter was eventually used to create a model based on the entire dataset except the validation set.

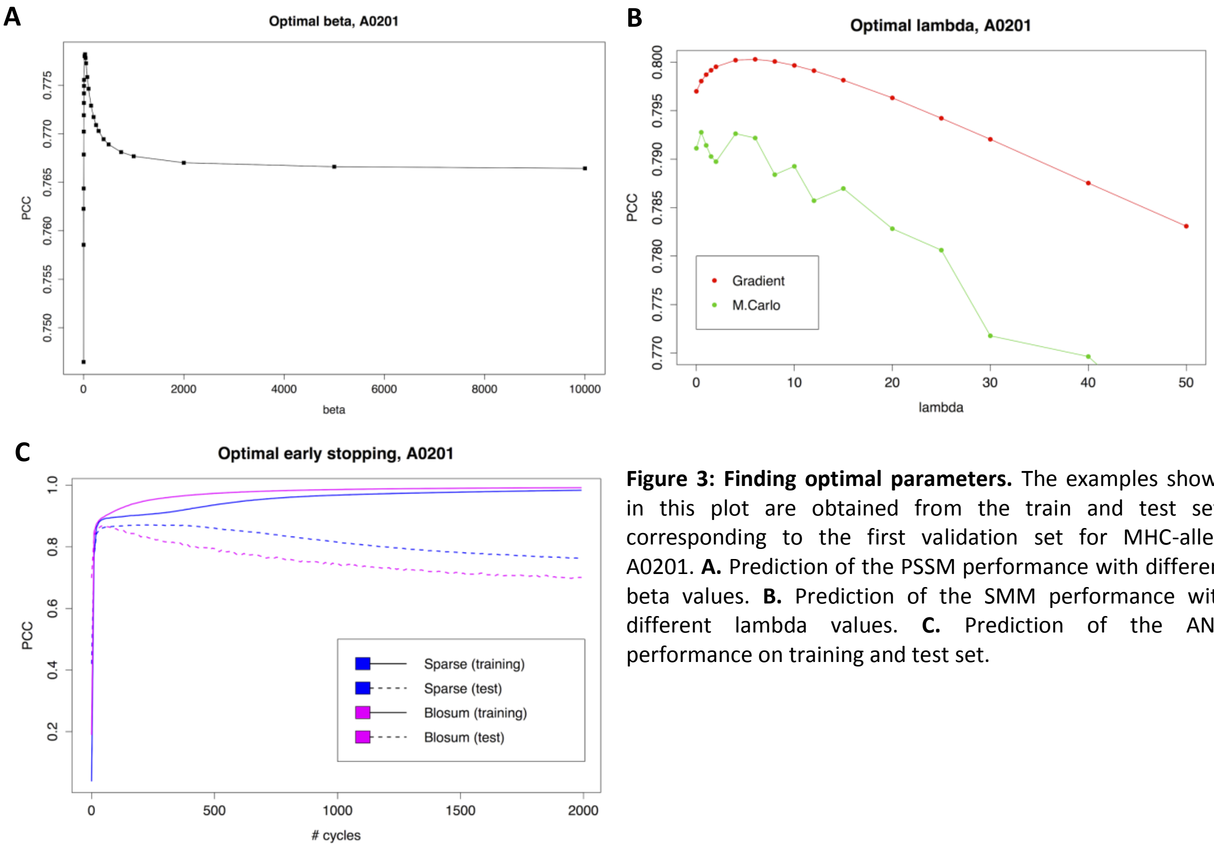

Figure 3: Finding optimal parameters. The examples shown in this plot are obtained from the train and test sets corresponding to the first validation set for MHC-allele A0201. **A.** Prediction of the PSSM performance with different beta values. **B.** Prediction of the SMM performance with different lambda values. **C.** Prediction of the ANN performance on training and test set.

The optimal λ and early stopping could also have been calculated using MSE as the parameter to be optimised, but PCC was chosen, with regard to using the same evaluation for the three different models.

The algorithm

For generating the results obtained in this study a general algorithm able to complete model development and cross-validation was created. A flowchart of the algorithm is presented in Figure 4.

PSSM

In the training sets used for PSSM non-binding peptides were removed. Here the cutoff IC50 = 500 nM was used to classify peptides into binders (IC50 > 500 nM) or non-binders (IC50 < 500) [10]. Both binding and non-binding peptides were included in the test and validation sets.

In order to construct the PSSM's for the different MHC binding peptides two programs were used. The first program "pep2mat" was used for creating a weight matrix based on the data from the training set, using both pseudo counts and sequence weightings. The second program "pep2score" scored the peptides from the test or validation set using the newly created PSSM.

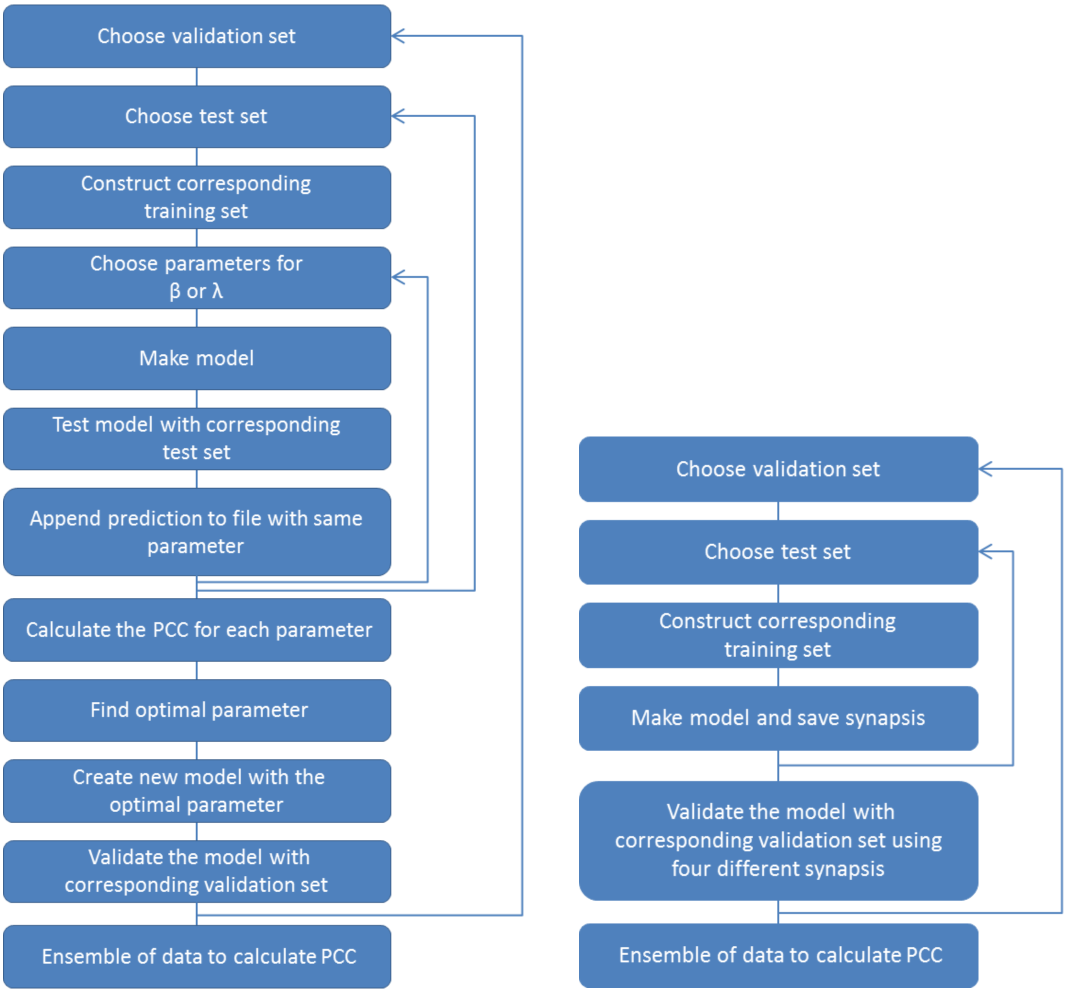
Choose validation set
Choose test set
Construct corresponding training set
Choose parameters for β or λ
Make model
Test model with corresponding test set
Append prediction to file with same parameter
Calculate the PCC for each parameter
Find optimal parameter
Create new model with the optimal parameter
Validate the model with corresponding validation set
Ensemble of data to calculate PCC
Choose validation set
Choose test set
Construct corresponding training set
Make model and save synapsis
Validate the model with corresponding validation set using four different synapsis
Ensemble of data to calculate PCC

Figure 4: Flowchart of the algorithm used for model development, parameter optimisation, and five-fold cross-validation.

SMM

In the SMM method three different programs were used, “smm”, “smm_mc”, and “pep2score”. The “smm” program calculated the stabilisation matrix using gradient descent optimisation. The “smm_mc” worked similarly except for using Monte Carlo optimisation instead of gradient descent. The inputs for the SMM algorithm were a string containing the peptide sequence, which was converted to sparse encoding (0.05, 0.9), and the measured affinity of the peptide. “pep2score” was used as previously to calculate the peptide affinity based on the matrix.

ANN

In the ANN a string with the peptide sequence and the measured affinity were used as input, as well. The program “seq2inp” converted this input to the sparse encoded vector. The program “nnforward” was used to do the forward iterations of the network, and “nnbackprop” did the back propagations. The back propagations did early stopping at the point where the test set prediction is the best. From one training set four ANNs were generated, the prediction was then performed on the corresponding validation set four times and the results were collected into an ensemble prior to calculating PCC or MSE.

Results

To compare the performance for each of the three prediction methods the PCC and MSE between the measured and the predicted affinities were calculated. The result of this comparison can be seen in Table 1. Based on these results the performance of the different prediction methods, depending on the size of the datasets, was evaluated.

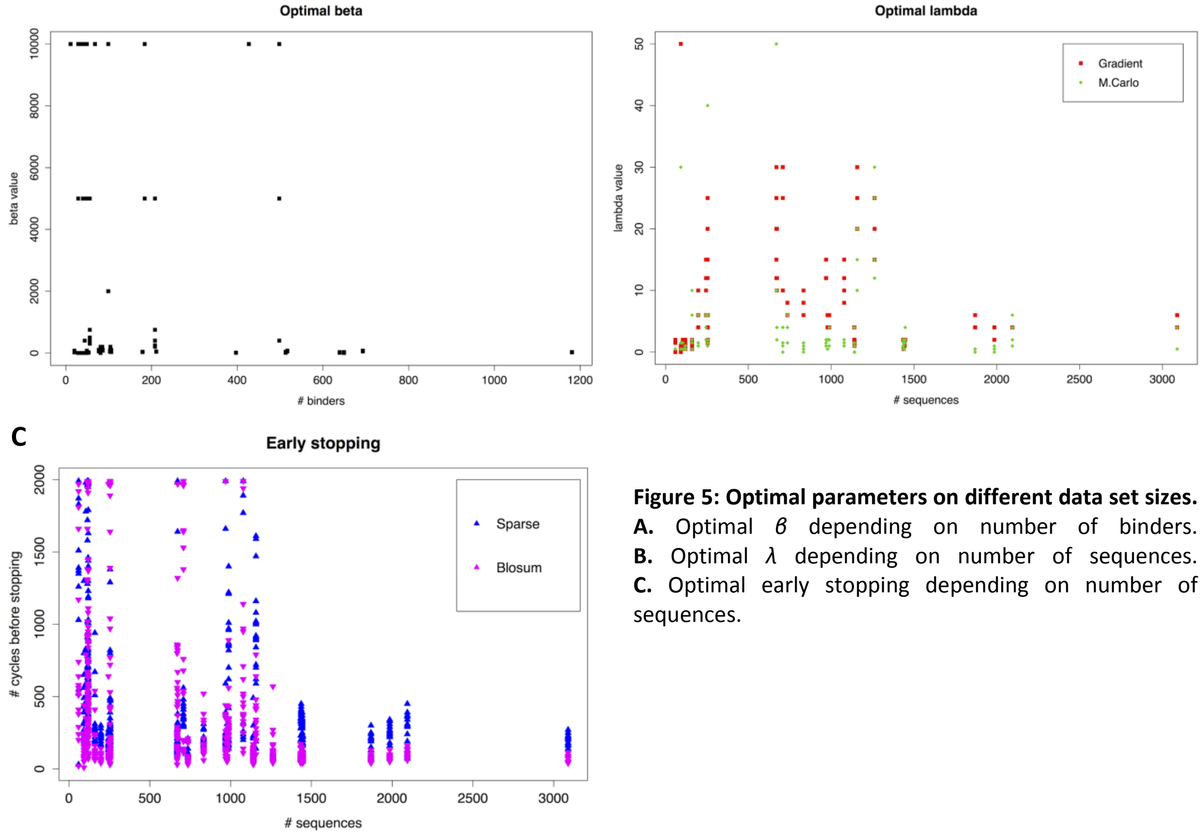

Figure 5: Optimal parameters on different data set sizes. **A.** Optimal *β* depending on number of binders. **B.** Optimal *λ* depending on number of sequences. **C.** Optimal early stopping depending on number of sequences.

As seen in Figure 5, the optimal *β* value when the number of binders is greater than 600 is zero. In other cases it can have different values, especially around 0, 5,000, and 10,000. This is more observed when the number of binders is below 200.

The optimal *λ* for different numbers of sequences presents a similar pattern, with low values when the number of peptides exceeds 1,500. Moreover, the gradient descent method usually demands a higher *λ* value than the Monte Carlo algorithm does.

Concerning the ANN, molecules with a higher number of available sequences correspond to a lower number of cycles before stopping; whereas, fewer sequences correlate with more spread early stopping values. Overall it is found that ANNs with a BLOSUM encoded input reach their performance peak with a lower number of cycles.

Generally, the ANN outperforms the other models. 26 MHC molecules had the best prediction using the ANN model with BLOSUM encoding; nine molecules had a better performance when using ANN with sparse encoding.

In some cases the performance was particularly off with a PCC below 0.1. Using PSSM this was true for two molecules (B4002, B4403), whereas, for SMM this was the case for four molecules (A2301, B4002, B4403, B5701) regardless of using gradient descent or Monte Carlo. For both models one of these molecules even had a negative correlation (B4002 for PSMM and B4403 for SMM using Monte Carlo optimisation).

Generally, the PSSM performs with an average PCC of 0.52; however the variation is more than a third of that. The average performance of the SMM is 0.41 (for gradient descent) and 0.37 (for Monte Carlo). The standard deviations for the SMM models are almost half of their PCC's. The ANN performs with a PCC of 0.72 (sparse) and 0.75 (BLOSUM) together with low standard deviations of 0.10 (sparse) and 0.09 (BLOSUM). These results stratified by the number of sequences are visualised in Figure 6.

Table 1: Overview of the dataset and results. Showing the MHC allele, the number of peptides tested for binding, the number of binding peptides, and the result from the three different methods used in this study given as either PCC or MSE.

MHC allele	# Peptides	# Binders	PSSM	SMM				ANN			
				Gradient descent		Monte Carlo		Sparse encoding		BLOSUM encoding	
			PCC	PCC	MSE	PCC	MSE	PCC	MSE	PCC	MSE
A0101	1157	103	0.50	0.58	0.02	0.5	0.03	0.83	0.01	0.84	0.01
A0201	3089	1181	0.76	0.71	0.05	0.66	0.07	0.87	0.02	0.87	0.02
A0202	1447	649	0.51	0.57	0.08	0.51	0.10	0.8	0.03	0.80	0.03
A0203	1443	639	0.69	0.56	0.08	0.51	0.10	0.82	0.03	0.83	0.03
A0206	1437	513	0.65	0.53	0.07	0.48	0.09	0.82	0.03	0.81	0.03
A0301	2094	517	0.64	0.64	0.04	0.53	0.06	0.81	0.02	0.81	0.02
A1101	1985	693	0.72	0.70	0.05	0.57	0.08	0.85	0.02	0.85	0.02
A2301	104	49	0.43	0.06	0.15	0.04	0.20	0.58	0.06	0.77	0.04
A2402	197	99	0.57	0.40	0.12	0.45	0.17	0.60	0.05	0.57	0.05
A2403	254	29	0.60	0.32	0.06	0.34	0.08	0.61	0.03	0.68	0.03
A2601	672	53	0.54	0.39	0.03	0.41	0.03	0.66	0.01	0.74	0.01
A2902	160	68	0.69	0.47	0.1	0.45	0.13	0.77	0.04	0.82	0.03
A3001	669	77	0.56	0.53	0.04	0.54	0.05	0.72	0.02	0.74	0.02
A3002	92	29	0.40	0.22	0.12	0.19	0.16	0.42	0.06	0.41	0.06
A3101	1869	427	0.67	0.6	0.05	0.51	0.07	0.81	0.02	0.81	0.02
A3301	1140	184	0.60	0.53	0.04	0.38	0.06	0.71	0.03	0.73	0.03
A6801	1141	498	0.55	0.41	0.09	0.35	0.12	0.78	0.03	0.78	0.03
A6802	1434	397	0.59	0.53	0.06	0.45	0.08	0.77	0.03	0.76	0.03
A6901	833	86	0.42	0.29	0.03	0.20	0.04	0.57	0.02	0.67	0.02
B0702	1262	208	0.73	0.67	0.03	0.62	0.05	0.82	0.02	0.83	0.02
B0801	708	20	0.41	0.28	0.01	0.24	0.02	0.53	0.01	0.64	0.01
B1501	978	179	0.61	0.44	0.03	0.43	0.05	0.75	0.01	0.77	0.01
B1801	118	47	0.23	0.23	0.05	0.19	0.07	0.73	0.02	0.76	0.02
B2705	969	56	0.53	0.43	0.02	0.35	0.02	0.73	0.01	0.76	0.01
B3501	736	211	0.57	0.45	0.07	0.41	0.10	0.68	0.04	0.71	0.03
B4001	1078	40	0.37	0.37	0.01	0.27	0.02	0.67	0.01	0.69	0.01
B4002	118	39	-0.01	0.04	0.06	0.01	0.08	0.68	0.02	0.79	0.02
B4402	119	44	0.27	0.17	0.04	0.14	0.05	0.66	0.02	0.62	0.02
B4403	119	34	0.01	0.03	0.05	-0.01	0.07	0.79	0.01	0.69	0.02
B4501	114	49	0.66	0.47	0.04	0.42	0.05	0.76	0.01	0.74	0.01
B5101	244	85	0.59	0.44	0.09	0.43	0.11	0.71	0.05	0.73	0.04
B5301	254	106	0.60	0.45	0.12	0.44	0.14	0.76	0.05	0.74	0.05
B5401	255	81	0.51	0.41	0.10	0.39	0.12	0.77	0.04	0.83	0.03
B5701	59	11	0.28	0. 03	0.04	0.01	0.04	0.56	0.03	0.72	0.02
B5801	988	104	0.59	0.44	0.03	0.41	0.04	0.82	0.01	0.86	0.01
Average			**0.52**	**0.41**	**0.06**	**0.37**	**0.08**	**0.72**	**0.03**	**0.75**	**0.02**
Standard deviation			**0.18**	**0.19**	**0.03**	**0.18**	**0.04**	**0.10**	**0.01**	**0.09**	**0.01**

From Figure 6 it is found that the ANN outperforms the other models, no matter the dataset size. The three different groups, corresponding to the number of sequences in the dataset, show the same order in performance with the SMM and especially the Monte Carlo version underperforming the other models.

The ANN has a quite high performance level in all three groups, whereas, the PSSM and SMM are more sensitive to the number of available sequences. Thus, the models that performs badly on the small data sets benefit more from an increasing number of sequences. The SMM increases its average correlation with the dataset size, however, no reduction in the MSE when moving from the medium to the large datasets is found.

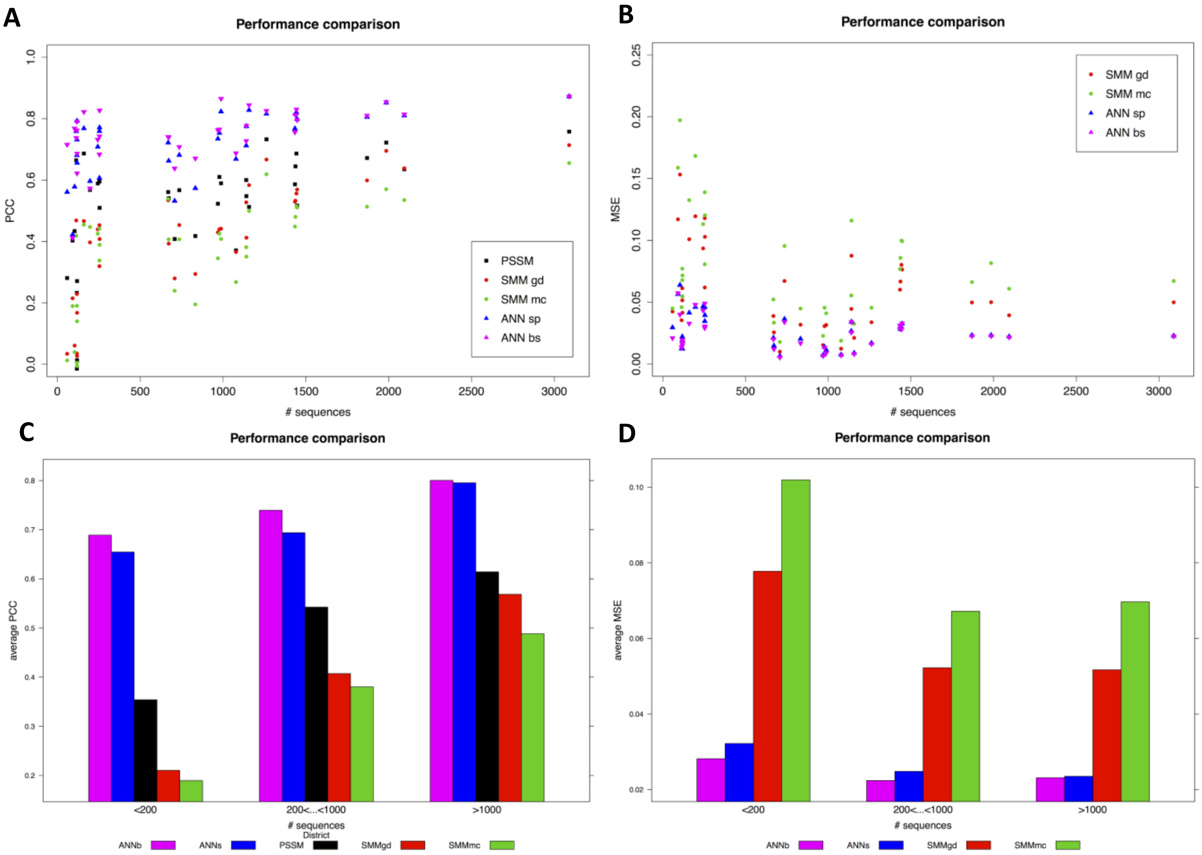

Figure 6: Comparison of the prediction performance of the methods. Prediction performance calculated as **A.** PCC or **B.** MSE, shown on the number of sequences. Average prediction performance of each method calculated as **C.** PCC or **D.** MSE, grouped according to the number of sequences.

For further investigating the influence of the dataset size the predicted binding scores or affinities against the measured binding affinities for the largest data set (A0201) and the smallest one (B5701) were plotted.

From the examples in Figure 7, it is seen that the large dataset has a better prediction with all models than the small one. The performance is the best using ANN for both A0201 and B5701, in both cases with the BLOSUM encoding (Table 1).

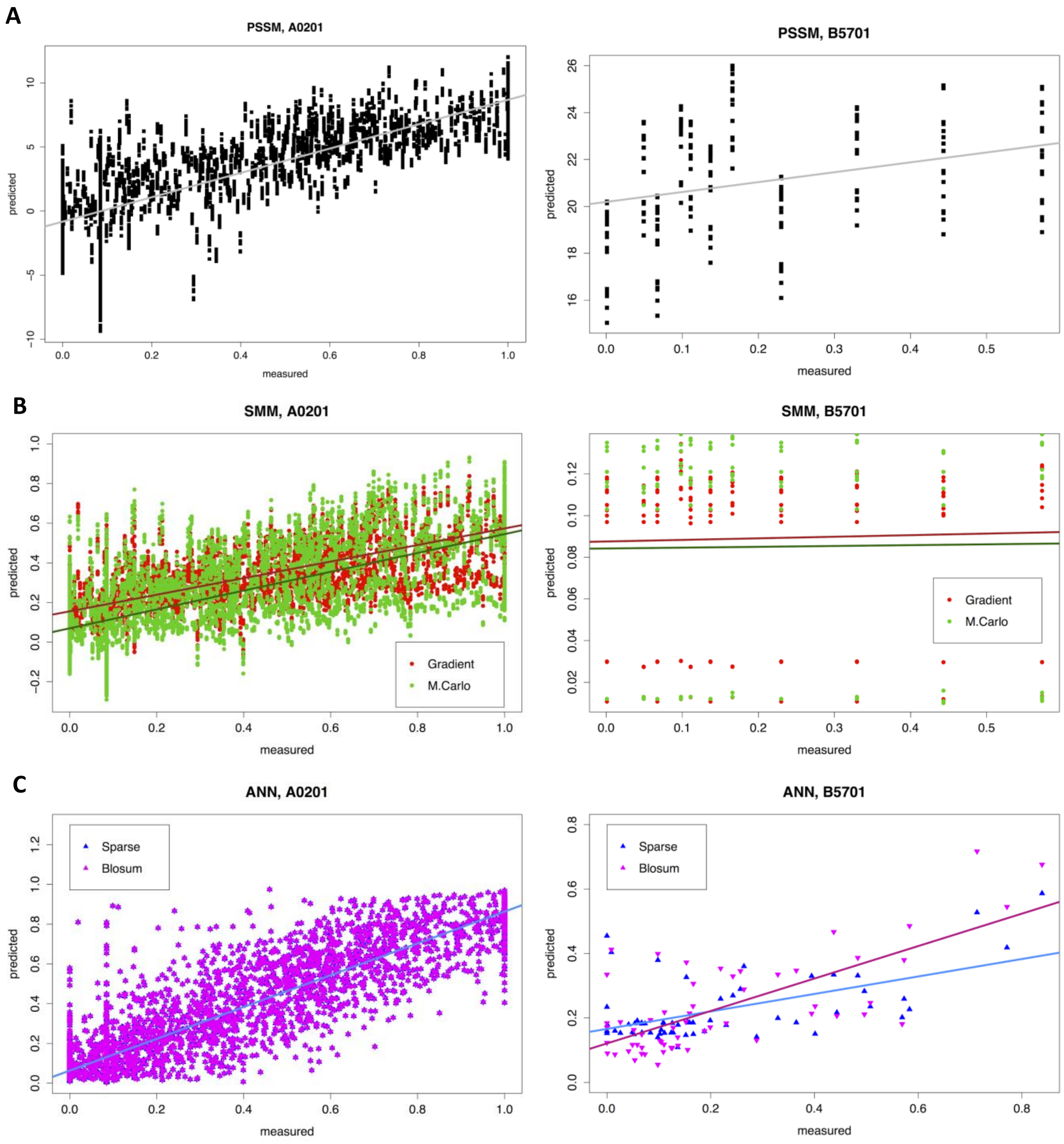

Figure 7: Comparison of the predicted and the measured values for peptides in the A0201 and the B5701 datasets. Using **A.** PSSM **B.** SMM **C.** ANN. In the ANN A0201 plot, the two regression lines are on top of each other, thus only one is visible.

Discussion

The small *β* value for the molecules with many binders were expected, as the pseudo count is not an important parameter to take into account when there is a lot of information to make the PSSM from (2). Likewise, large *β* values were expected for the smaller datasets. However, the low *β* values for the small data sets were unexpected. This could be a result of sequence variance in the small datasets representing different binding motifs, thus, the PSSM could be efficiently constructed without considering the pseudo count. The problem with unequal sequence variance among the datasets could have been prevented by removing similar sequences before making the models, e.g. by using the Hobohm algorithm [11]. In this way the number of sequences would represent the number of different peptides rather than the total dataset size. Another explanation could be that the binding motifs for these MHC molecules are very restricted, thus, even a small number of binding sequences could represent all the binding peptides possible.

As for the SMM, when analysing the observations of the optimal λ values, it seems that the smaller datasets need a larger weight on the penalty term in (5) and (6). Furthermore, when creating the SMM using the Monte Carlo algorithm, a lower λ is generally needed, thus, when using gradient descent it is necessary to have a larger weight on the penalty term. Hence, large data sets and Monte Carlo optimisation allow for more complex models than small datasets and gradient descent.

Concerning the ANN, the point where the overfitting starts could also be interpreted as a point where the ANN has learned to optimize prediction. This is the point to stop learning, i.e. early stopping. The higher amount of cycles before early stopping in the small data sets was expected, because the information content in these is lower. Thus, the ANN uses more iterations in order to predict the correct affinity.
The ANN trained with an input of sparse encoding does generally uses a larger amount of cycles to learn. Again, telling that the information content using BLOSUM encoding is greater than with sparse encoding.

The MHC molecules for which prediction of peptide binding were extremely low using PSSM or SMM all had quite small datasets. This could indicate that the models are not able to be developed to predict binding from these datasets. For ANN, no patterns on the size of the datasets in the groups of predictions that worked best with either BLOSUM or sparse encoding were found.

The ANN outperformance was expected, considering the complexity of this tool compared to the two matrix-based methods and previous results [2]. The underperformance of SMM was however not expected. As the SMM updates the weights considering a target value, and the PSSM only computes the matrix once, it was expected that the SMM would have predicted better than the simpler PSSM.
At first the SMM models were created using sparse encoding with zeros and ones (data not shown), but this was even worse than the results shown in Table 1. The performance was increased approximately 10% with the 0.05 and 0.9 sparse encoding. Thus, the SMM performance might be increased further if the encoding used as input was changed to BLOSUM.
The reason why the Monte Carlo generally performs worse than the gradient descent based SMM could be that the number of iterations for the first method was too low. Because the Monte Carlo SMM only updates the weights once per iteration while the gradient descent based model does this for each peptide in each iteration, i.e. equation (8) and (10).
Additionally, in the other matrix-based method, the PSSM, sequence weighting was included. As this again makes the algorithm distinguish similar sequences from each other, it could increase the performance of the SMM models.

Conclusion

In conclusion, it was found that the ANN was the best predictor for all peptides, with an average PCC of 0.72-0.75. The binding of the peptides to some MHC molecules was predicted better when encoded with the BLOSUM, whereas others had the best prediction with the sparse encoding. In addition, it was found that the PSSM method performs better than the SMM. However, it could be interesting to see how the SMM based on the Monte Carlo method would perform if the number of iterations were increased, if sequence weighting was implemented, and if the sequence encoding was BLOSUM. Furthermore, it could also be interesting to see how the SMM and ANN would perform after an additional optimisation step of the ε-parameter.

References

[1] J. Neefjes, M. L. Jongsma, P. Paul, and O. Bakke, “Towards a systems understanding of MHC class I and MHC class II antigen presentation,” *Nat Rev Immunol*, vol. 11, no. 12, pp. 823–36, 2011.

[2] B. Peters, H. H. Bui, S. Frankild, M. Nielsen, C. Lundegaard, E. Kostem, D. Basch, K. Lamberth, M. Harndahl, W. Fleri, S. S. Wilson, J. Sidney, O. Lund, S. Buus, and A. Sette, “A community resource benchmarking predictions of peptide binding to MHC-I molecules,” *PLoS Comput. Biol.*, vol. 2, no. 6, pp. 0574–0584, 2006.

[3] M. Nielsen, C. Lundegaard, P. Worning, S. L. Lauemøller, K. Lamberth, S. Buus, S. Brunak, and O. Lund, “Reliable prediction of T-cell epitopes using neural networks with novel sequence representations,” *Protein Sci.*, vol. 12, no. 5, pp. 1007–1017, 2003.

[4] K. Nishida, M. C. Frith, and K. Nakai, “Pseudocounts for transcription factor binding sites,” *Nucleic Acids Res.*, vol. 37, no. 3, pp. 939–944, 2009.

[5] B. Peters and A. Sette, “Generating quantitative models describing the sequence specificity of biological processes with the stabilized matrix method.,” *BMC Bioinformatics*, vol. 6, p. 132, 2005.

[6] M. Nielsen, C. Lundegaard, and O. Lund, “Prediction of MHC class II binding affinity using SMM-align, a novel stabilization matrix alignment method,” *BMC Bioinformatics*, vol. 8, no. 1, p. 238, 2007.

[7] O. Lund, M. Nielsen, C. Lundegaard, C. Ke¸smir, and S. Brunak, *Immunological bioinformatics*. 2005.

[8] S. Henikoff and J. G. Henikoff, “Amino acid substitution matrices from protein blocks.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 89, no. 22, pp. 10915–10919, 1992.

[9] B. Peters, W. Tong, J. Sidney, A. Sette, and Z. Weng, “Examining the independent binding assumption for binding of peptide epitopes to MHC-I molecules,” *Bioinformatics*, vol. 19, no. 14, pp. 1765–1772, 2003.

[10] a Sette, a Vitiello, B. Reherman, P. Fowler, R. Nayersina, W. M. Kast, C. J. Melief, C. Oseroff, L. Yuan, J. Ruppert, J. Sidney, M. F. del Guercio, S. Southwood, R. T. Kubo, R. W. Chesnut, H. M. Grey, and F. V Chisari, “The relationship between class I binding affinity and immunogenicity of potential cytotoxic T cell epitopes.,” *J. Immunol.*, vol. 153, no. 6, pp. 5586–5592, 1994.

[11] U. Hobohm, M. Scharf, R. Schneider, and C. Sander, “Selection of representative protein data sets.,” *Protein Sci.*, vol. 1, no. 3, pp. 409–417, 1992.