BIOINFORMATICS

Vol. 00 no. 00 2005
Pages 1–4

# High-Order Neural Networks and Kernel Methods for Peptide-MHC Binding Prediction

Submitted for blind review

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

**ABSTRACT**

Effective computational methods for peptide-protein binding prediction can greatly help clinical peptide vaccine search and design. However, previous computational methods fail to capture key nonlinear high-order dependencies between different amino acid positions. As a result, they often produce low-quality rankings of strong binding peptides. To solve this problem, we propose nonlinear high-order machine learning methods including high-order neural networks with deep extensions and high-order Kernel Support Vector Machines to predict peptide-MHC binding. The proposed methods improve quality of binding predictions over other prediction methods. With the proposed methods, a significant gain of up to 34% is observed on benchmark and reference peptide data sets and tasks, which demonstrates the importance of effectively modelling high-order feature interactions for predicting peptide targeting. Moreover, for the first time, our experiments show that pre-training with high-order semi-Restricted Boltzmann Machines will significantly improve the performance of feed-forward high-order neural networks, which is an important result in deep learning.

## 1 INTRODUCTION

In this paper, we propose novel machine learning methods to study a specific type of peptide-protein interaction, that is, the interaction between peptides and Major Histocompatibility Complex class I (MHC I) proteins, although our methods can be readily applicable to other types of peptide-protein interactions. Peptide-MHC I protein interactions are essential in cell-mediated immunity, regulation of immune responses, vaccine design, and transplant rejection. Therefore, effective computational methods for peptide-MHC I binding prediction will significantly reduce cost and time in clinical peptide vaccine search and design.

Previous computational approaches to predicting peptide-MHC interactions are mainly based on linear or bi-linear models, and they fail to capture key non-linear high-order dependencies between different amino acid positions. Although previous Kernel SVM and Neural Network (NetMHC) [9, 7, 3] approaches can capture nonlinear interactions between input features, they fail to model the direct strong high-order interactions between features. As a result, the quality of the peptide rankings produced by previous methods is not good enough. Producing high-quality rankings of peptide vaccine candidates is essential to the successful deployment of computational methods for vaccine design. For this purpose, we need to effectively model direct non-linear high-order feature interactions to directly capture interactions between primary (anchor) and secondary amino acid residues involved in the formation of peptide-MHC complexes.

Therefore, for the first time, we use high-order Restricted Boltzmann Machine (RBMs) to pre-train a (deep) feed-forward high-order neural network and propose high-order Kernel SVM for peptide-MHC binding prediction. With the proposed methods, a significant gain of up to 34% is observed on benchmark and reference peptide data sets and tasks.

## 2 METHODS

### 2.1 Deep Neural Network and High-Order Neural Network

Instead of using an ensemble of traditional neural networks to predict MHC class-peptide bindings as in the state-of-the-art approach NetMHC [11, 2, 9], we use deep neural networks pre-trained with RBMs and High-Order Neural Networks (HONN) pre-trained with a special type of high-order Semi-RBMs called mean-covariance RBMs (mcRBMs), capable of capturing strong high-order interactions of feature descriptors of input peptides, to produce high-quality rankings of binding peptides (T-cell epitopes). In our experiments, we use BLOSUM substitution matrix as continuous descriptors of input peptide sequences.

In our Deep Neural Network (DNN) as shown on the left panel of Fig. 1, we use Gaussian RBM to pre-train the network weights of the first layer, and we use binary RBM to pre-train the connection weights of upper layers in a greedy layer-wise fashion. In our High-Order Neural Network (HONN) as shown on the right panel of Fig. 1, we use mcRBM to pre-train the network weights of the first layer, and we optionally add upper layers if we have enough training data, and we use binary RBM to pre-train the connection weights in possibly available upper layers. In both DNN and HONN, we use a logistic unit as our final output layer, and then we use back-propagation to fine-tune the final network weights by minimizing the cross entropy between predicted binding probabilities and true binding probabilities.

The pre-training module mcRBM of HONN extends traditional Gaussian RBM to model both mean and explicit pairwise interactions of input feature values, and it has two sets of hidden units, mean hidden units modeling the mean of input features and covariance hidden units gating pairwise interactions between input features. If the gating hidden units are binary, they act as binary switches controlling the pairwise interactions between input features.

In the following, we will first review traditional Gaussian RBMs. The energy function of Gaussian RBM is,

$$E(v,h) = -\sum_{i,j} \frac{v_i}{\sigma_i} h_j w_{ij} - \sum_i \frac{(v_i - a_i)^2}{2{\sigma_i}^2} - \sum_j b_j h_j, \qquad (1)$$

where $i$ indexes visible units such as peptide sequence features, $j$ indexes hidden units, $w_{ij}$ is the network connection weight between visible feature $i$ and hidden unit $j$, $b_j$ is the bias of hidden unit $j$, and $a_i$ and $\sigma_i$ are, respectively, the bias and variance of visible feature $i$. For simplicity, we assume the variance of the visible units to be 1. We use Contrastive Divergence (CD) to learn the network connection weights, which approximately maximizes the log-likelihood of input data. The CD updates

for the weights can be written as follows,

$$w_{ij} = \epsilon(< v_i h_j >_{data} - < v_i h_j >_T), \quad (2)$$

where $\epsilon$ is the learning rate, $< \cdot >_{data}$ denotes the expectation with respect to data distribution, and $< \cdot >_T$ denotes the expectation with respect to the $T$-step Gibbs Sampling samples from the model distribution. Binary RBM takes a similar energy function to that of Gaussian RBM except that both visible units and hidden units are binary. As a result, the conditional probability distributions of binary RBM take the form of sigmoid functions.

Gaussian RBMs are very difficult to train using binary hidden units. This is because unlike binary data, continuous valued data lie in a much larger space. One obvious problem with the Gaussian RBM is that given the hidden units, the visible units are assumed to be conditionally independent, meaning it tries to reconstruct the visible units independently without using the abundant covariance information present in all datasets. The knowledge of the covariance information reduces the complexity of the input space where the visible units could lie, thereby helping RBMs to model the continuous distribution more efficiently. Covariance RBM [5] tried to use hidden units to gate the pairwise interaction between the visible units, leading to the following energy function,

$$E(v,h) = \frac{1}{2}\sum_{i,j,k} v_i v_j h_k w_{ijk} - \sum_i a_i v_i - \sum_k b_k h_k \quad (3)$$

To take advantage of both the Gaussian RBM (which models the mean) and the covariance RBM, the resulting model called mean-covariance RBM (mcRBM) uses an energy function that includes both the energy terms,

$$\begin{aligned} E(v,h^g,h^m) &= \frac{1}{2}\sum_{i,j,k} v_i v_j {h_k}^g w_{ijk} - \sum_i a_i v_i - \sum_k b_k {h_k}^g \\ &\quad - \sum_{ij} v_i {h_j}^m w_{ij} - \sum_k c_k {h_k}^m \end{aligned} \quad (4)$$

In the above equation, each hidden unit modulates the interaction between each pair of input features leading to a large number of parameters in $w_{ijk}$ to be learned. To reduce this complexity, we can factorize the weight $w_{ijk}$ as follows [6],

$$w_{ijk} = \sum_f C_{if} C_{jf} P_{kf} \quad (5)$$

The energy function can now be written as

$$\begin{aligned} E(v,h^g,h^m) &= \frac{1}{2}\sum_f \left(\sum_i v_i C_{if}\right)^2 \left(\sum_k h_k P_{kf}\right) - \sum_i a_i v_i \\ &\quad - \sum_k b_k {h_k}^g - \sum_{ij} v_i {h_j}^m w_{ij} - \sum_k c_k {h_k}^m \end{aligned} \quad (6)$$

Using this energy function, we can again derive the conditional probabilities of hidden units given visible units, as well the respective gradients for training the network. The structure of this factorized mcRBM is shown on the bottom of the right panel of Fig. 1, the hidden units on the left model mean and those on the right model covariance. We used CD to learn the factorized weights in mcRBM as in Gaussian RBM, and we used Hybrid Monte Carlo sampling to generate the negative samples as in [13].

## 2.2 High-order Kernel Models

The sequence of the descriptors corresponding to the peptide $X = x_1, x_2, \ldots, x_{|X|}, x_i \in \Sigma$ can be modeled as an *attributed set* of descriptors corresponding to different positions (or groups of positions) in the peptide and amino acids or strings of amino acids occupying these positions:

$$X_A = \{(\mathbf{p}_i, \mathbf{d}_i)\}_{i=1}^n$$

where $\mathbf{p}_i$ is the coordinate (position) or a set (vector) of coordinates and $\mathbf{d}_i$ is the descriptor vector associated with the $\mathbf{p}_i$, with $n$ indicating the cardinality of the attributed set description $X_A$ of peptide $X$. The cardinality of the description $X_A$ corresponds to the length of the peptide (i.e., the number of positions) or to in general to the number of unique descriptors in the descriptor sequence representation. A unified descriptor sequence representation of the peptides as a sequence of descriptor vectors is used to derive attributed set descriptions $X_A$.

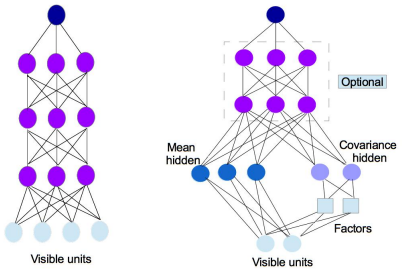


Fig. 1: The structure of DNN (left) and HONN (right).

## 2.3 High-order kernel functions on peptide descriptor sequence representations

In the following we define kernel functions for peptides based on peptide descriptor sequence representations. The proposed kernel functions for peptide sequences $X$ and $Y$ have the following general form:

$$\begin{aligned} K(X,Y) &= K(M(X), M(Y)) = K(X_A, Y_A) \\ &= \sum_{i_X}\sum_{j_Y} k_{\mathbf{p}}(\mathbf{p}^X_{i_Y}, \mathbf{p}^Y_{j_Y}) k_{\mathbf{d}}(\mathbf{d}^X_{i_X}, \mathbf{d}^Y_{j_Y}) \end{aligned} \quad (7)$$

where $M(\cdot)$ is a descriptor sequence (e.g., spatial feature matrix) representation of a peptide, $X_A(Y_A)$ is an attributed set corresponding to $M(X)$ $(M(Y))$, $k_{\mathbf{d}}(\cdot,\cdot)$, $k_{\mathbf{p}}(\cdot,\cdot)$, are kernel functions on descriptors and context/positions, respectively, and $i_X$, $i_Y$ index elements of the attributed sets $X_A$, $Y_A$. A number of kernel functions for descriptor sequence (e.g., matrix) forms $M(\cdot)$ is described below.

Using real-valued descriptors (e.g., vectors of physicochemical attributes), with RBF or polynomial kernel function on descriptors, the $k_{\mathbf{d}}(\mathbf{d}_\alpha, \mathbf{d}_\beta)$ is defined as

$$\exp(-\gamma_{\mathbf{d}} ||\mathbf{d}_\alpha - \mathbf{d}_\beta||)$$

where $\gamma_{\mathbf{d}}$ is an appropriately chosen weight parameter, or

$$(\langle \mathbf{d}_\alpha, \mathbf{d}_\beta \rangle + c)^p$$

where $p$ is the degree (interaction order) parameter and $c$ is a parameter controlling contribution of lower order terms.

Kernel functions $k_{\mathbf{p}}(\cdot,\cdot)$ on position sets $\mathbf{p}_i$ and $\mathbf{p}_j$ are defined as a set kernel

$$k_{\mathbf{p}}(\mathbf{p}_i, \mathbf{p}_j) = \sum_{i \in \mathbf{p}_i} \sum_{j \in \mathbf{p}_j} k(i,j|\alpha,\beta)$$

where

$$k(i,j|\alpha,\beta) = \frac{1}{|i-j|^\alpha} + \beta = exp(-\alpha \log(|i-j|)) + \beta$$

is a kernel function on pairs of position coordinates $(i,j)$.

The position set kernel function above assigns weights to interactions between positions $(i,j)$ according to $k(i,j|\alpha,\beta)$.

The descriptor kernel function (e.g., RBF or polynomial) between two descriptors $\mathbf{d}_i = (d^i_1, d^i_2, \ldots, d^i_R)$ and $\mathbf{d}_j = (d^j_1, d^j_2, \ldots, d^j_R)$

induces high-order (i.e. products-of-features) interaction features (such as $d_{i_1} d_{i_2} \ldots d_{i_p}$ for polynomial of degree $p$) between positions / attributes.

The proposed kernel function (Eq. 7) captures high-order interactions between amino acids / positions by considering essentially all possible products of features encoded in descriptors $\mathbf{d}$ of two or more positions. The feature map corresponding to this kernel is composed of individual feature maps capturing interactions between particular combinations of the positions. The interaction maps between different positions $\mathbf{p}_a$ and $\mathbf{p}_b$ are weighted by the position/context kernel function $k_{\mathbf{P}}(\mathbf{p}_a, \mathbf{p}_b)$.

## 3 DATA

In order to assess the performance of our high-order methods, we tested our methods on three prediction tasks: 1. MHC-I binding prediction; 2. Naturally processed ("eluted") peptide prediction. We use recently compiled benchmark data from the 2nd Machine Learning in Immunology competition (MLI-II). 3. T-cell epitope prediction. We use data of known T-cell epitopes to test ability of the methods in predicting promising candidates for clinical development.

For all of the tasks, we focused on the 9-mer peptides. For MHC-I binding prediction, we threshold at a standard value $IC50 = 500$ to separate binding peptides ($IC50 < 500$) and non-binding ($IC50 > 500$) peptides and focus on three alleles (HLA-A*0201, HLA-A*0206, HLA-A*2402). The choice of these alleles is motivated by the target population group (Japanese) in our research (due to the 4-page space constraint, we only show results for HLA-A*2402, and the results for the other two alleles have a similar trend). The application of our method to other alleles or peptide lengths would be straightforward.

### 3.1 Training and testing protocol

For MHC-I binding prediction, we train our models for each allele on the publicly available data from the Immune Epitope Database and Analysis Resource (IEDB). For testing, we use the experimental data from our lab for each allele. These datasets are denoted with `'Japanese'` suffix. The training `'IEDB'` datasets and the test `'Japanese'` datasets are completely disjoint.

### 3.2 Evaluation metrics

To assess performance, we use two sets of metrics, classical binary metrics and non-binary relevance metrics.

*Binary performance metrics*. We used (1) Area under ROC curve (AUC); (2) area under ROC curve up to first $n$ false positives (ROC-$n$).

*Non-binary relevance/quality metrics*. While classical binary performance metrics use binary relevance (i.e. "1"=relevant, "0"=non-relevant), to take into account more "precise" relevance measure, i.e. the binding strength of the peptides, we use *normalized discounted cumulative gain* (nDCG), a classical *non-binary* (graded) relevance metric.

Given a list of peptides $p_1, \ldots, p_N$ ordered by the output scores of the predictor, the discounted cumulative gain ($DCG_N$) is defined as a sum of individual peptide relevance scores (binding strength) $q_1, q_2, \ldots, q_n$ discounted by the log of their position $i$ in the list:

$$DCG_N = \sum_{i=1}^{N} \frac{2^{q_i} - 1}{\log(i+1)}$$

The normalized $DCG_N$ is defined as a ratio between DCG of the method and an ideal DCG $iDCG_N$ (i.e., DCG of an ideal ordering of peptides from the highest degree of binding affinity to the lowest binding affinity):

$$nDCG_N = \frac{DCG_N}{iDCG_N}$$

The normalized $DCG_N$ value is then ranges between 0 and 1, with $nDCG_N = 1$ corresponding to the ideal value.

We find this measure (nDCG) to be more indicative of the prediction performance of the MHC-I binding prediction method as it directly assesses whether the predictor ranks stronger binders higher than weaker binders (as opposed to binary measures (e.g., ROC) that measure whether "binders" are ranked higher than "non-binders" *irrespectively* of the actual peptide binding strength).

**Table 1.** Comparison of AUC test scores on A2402-Japanese data

| method | AUC | ROC-5 | ROC-10 | ROC-30 |
|---|---|---|---|---|
| hkSVM | 90.59 | 68.8 | 75.92 | 86.93 |
| DNN | 89.1 | 63.52 | 70.96 | 84.75 |
| HONN | 86.29 | 54.88 | 65.04 | 81.17 |
| hkSVM+HONN | **91.07** | **72.16** | **77.76** | **87.55** |
| NetMHC | 88.88 | 53.76 | 66.88 | 84.48 |

**Table 2.** A2402-Japanese data. Relevance/ranking assessment (nDCG)

| method | nDCG@10 | nDCG@30 | nDCG |
|---|---|---|---|
| hKSVM | 53.77 | 64.33 | 86.68 |
| DNN | 51.07 | 56.88 | 84.36 |
| HONN | 57.36 | 60.82 | 85.20 |
| hkSVM+HONN | **60.41** | **69.59** | 87.35 |
| NetMHC | 55.98 | 68.76 | 87.57 |

## 4 RESULTS

We first present results for MHC-I binding prediction on benchmark datasets and experimental data from our lab. We show next results on predicting peptides naturally processed by the MHC pathway. Finally, we show results for predicting promising T-cell epitopes for clinical development.

### 4.1 MHC-I binding prediction

We train a deep neural network (DNN), a high-order semi-RBM (HONN), and a high-order kernel SVM (hkSVM) on `IEDB` data. In our experiments, we use BLOSUM substitution matrix as continuous descriptors of input peptide sequences.

We compare with the popular NetMHC method that has been shown to yield state-of-the-art accuracy for MHC-I binding prediction with respect to other best published methods (see, e.g., [9, 19, 4]).

We first use `'Japanese'` data sets to test our methods. Results are shown in Table 1 and 2 for the target allele.

Our method ranks peptides by their actual binding strength significantly better than other methods. We observe that strong binders are placed much higher in the classification results compared to the state-of-the-art NetMHC method. We note that for both HONN and DNN the pre-training is critical to achieve good performance. The performance comparisons of DNN and HONN with and without pre-training are in the supplemental material. All the results of DNN and HONN reported in the main paper are based on pre-training and fine-tuning.

Using the simple average score of HONN and hkSVM further improves peptide-MHC recognition as evident by the increase in both area under ROC curve scores (improved "binder" vs "non-binder" separation) and nDCG metric quality scores (improved ranking of peptides by binding strength).

We note that unlike the previous approaches that utilized quantitative binding information during training, no quantitative information regarding actual binding strength was used to train our models. However, even with only *binary* train data, our models correctly order peptides according to their binding strength. This can be attributed to explicit high-order interaction modeling by our method that allows to capture intrinsic binding strength information. Nevertheless, our models can easily use quantitative train data (e.g., IC50) to further improve our results.

### 4.2 Eluted peptide prediction

We test ability of our methods on a difficult task that aims at predicting whether a peptide is naturally processed by the MHC pathway ("eluted").

This is a very important task as only a fraction of binding peptides (see "MHC-I binding task") constitute a set of peptides that are processed to the surface of a cell and may serve as epitopes. Eluted peptide prediction thus aims at verifying whether a peptide not only binds to a given MHC molecule, but that it is also naturally processed by MHC pathway.

To train our models, we used the data provided by MLI-II (2012 Machine Learning in Immunology competition) `http://bio.dfci.harvard.edu/DFRMLI/HTML/natural.php`.

We directly train our models to recognize naturally processed peptides, using eluted peptides as a positive set, and all other peptides (non-binders + non-eluted binders) as a negative set. We then test our models on the data composed of non-eluted binding peptides, non-binding peptides, and naturally processed ("eluted") peptides. We compare our approach with the popular NetMHC method, which was used as a benchmark in the competition, as well as the recently introduced MHC-NP [4] method that yielded state-of-the-art accuracy for naturally processed (NP) peptide prediction.

Table 3 shows results of naturally processed peptide prediction on the test set in terms of AUC, ROC-$n$, and F1 scores. Our approach significantly outperforms both NetMHC method and the MHC-NP [4] method.

**Table 3.** Eluted peptide prediction (MLI-II competition). Comparison of test AUC scores.

| method | AUC | ROC-10 | ROC-20 | ROC-30 | ROC-50 |
|---|---|---|---|---|---|
| hkSVM | **94.75** | **53.65** | 65.71 | 71.48 | 77.46 |
| HONN | 93.17 | 49.21 | 58.20 | 64.13 | 72.73 |
| DNN | 91.80 | 30.48 | 41.11 | 51.32 | 62.92 |
| hkSVM + HONN | **94.96** | **53.65** | **68.25** | **74.39** | **79.59** |
| NetMHC | 92.26 | 10.63 | 28.33 | 40.21 | 54.32 |
| MHC-NP† | 88.06 | - | - | - | - |

†quoted from [4]

**Table 4.** Prediction of WT1-derived binding peptides

| | NetMHC-rank | hkSVM+HONN-rank |
|---|---|---|
| | A2402 allele | |
| W10 | 41 | 2 |
| W302 | 7 | 4 |

### 4.3 Epitope prediction

We demonstrate ability of the method to predict promising peptides for clinical development using as an example WT1-derived strong binding peptides W10 and W302, discovered by NEC-Kochi Univ. We compare the performance of our method and NetMHC by "predicting" in a retrospective way these T-cell epitopes from WT1 antigen. Peptides (441 9-mers) that are part of WT1 antigen are ranked by the output scores of NetMHC and our method. The order of the W10 and W302 peptides in the output of the two prediction methods is given in Table 4. As evident from the table, our method ranks these peptides higher than NetMHC method.

## 5 DISCUSSION AND FUTURE WORK

In this paper, we propose using nonlinear high-order machine learning methods including Deep Neural Network, High-Order Neural network with possible deep extensions, and High-Order Kernel SVM for peptide-MHC I protein binding prediction. Experimental results on both public and private evaluation datasets according to both binary and non-binary performance metrics (AUC and nDCG) clearly demonstrate the advantages of our methods over the state-of-the-art approach NetMHC, which suggests the importance of modeling nonlinear high-order feature interactions across different amino acid positions of peptides. Our results are even more encouraging considering that our models were only trained on a subset of the binary binding datasets used by NetMHC and NetMHC was also trained on private quantitative binding datasets.

In the future, we will use available quantitative binding datasets to refine our Deep Neural Network and High-Order Neural Network model, and we will incorporate the descriptors of structural contacting amino acids on MHC proteins into current feature descriptors. The addition of peptide binding strength and structural information will potentially further improve the performance of our current models.

## REFERENCES


[1]V. Brusic, N. Petrovsky, G. Zhang, and V. B. Bajic. Prediction of promiscuous peptides that bind HLA class I molecules. *Immunol Cell Biol*, 80(3):280–5, 2002.

[2]S. Buus, S. Lauemøller, P. Worning, C. Kesmir, T. Frimurer, S. Corbet, A. Fomsgaard, J. Hilden, A. Holm, and S. Brunak. Sensitive quantitative predictions of peptide-mhc binding by a 'query by committee' artificial neural network approach. *Tissue Antigens*, 62(5):378–384, 2003.

[3]S. Giguere, M. Marchand, F. Laviolette, A. Drouin, and J. Corbeil. Learning a peptide-protein binding affinity predictor with kernel ridge regression. *BMC Bioinformatics*, 14(1):82, 2013.

[4]S. Gigure, A. Drouin, A. Lacoste, M. Marchand, J. Corbeil, and F. Laviolette. MHC-NP: Predicting peptides naturally processed by the MHC. *Journal of Immunological Methods*, 400401(0):30 – 36, 2013.

[5]G. E. Hinton. Learning to represent visual input. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 365(1537):177–184, 2010.

[6]G. E. Hinton. A practical guide to training restricted boltzmann machines. In G. Montavon, G. B. Orr, and K.-R. Mller, editors, *Neural Networks: Tricks of the Trade (2nd ed.)*, volume 7700 of *Lecture Notes in Computer Science*, pages 599–619. Springer, 2012.

[7]I. Hoof, B. Peters, J. Sidney, L. Pedersen, A. Sette, O. Lund, S. Buus, and M. Nielsen. Netmhcpan, a method for mhc class i binding prediction beyond humans. *Immunogenetics*, 61(1):1–13, 2009.

[8]W. Liu, J. Wan, X. Meng, D. Flower, and T. Li. In silico prediction of peptide-MHC binding affinity using SVRMHC. In D. R. Flower, editor, *Immunoinformatics*, volume 409 of *Methods in Molecular Biology*, pages 283–291. Humana Press, 2007.

[9]C. Lundegaard, O. Lund, and M. Nielsen. Prediction of epitopes using neural network based methods. *Journal of Immunological Methods*, 374(1–2):26 – 34, 2011. High-throughput methods for immunology: Machine learning and automation.

[10]M. Nielsen, C. Lundegaard, P. Worning, C. S. Hvid, K. Lamberth, S. Buus, S. Brunak, and O. Lund. Improved prediction of MHC class I and class II epitopes using a novel gibbs sampling approach. *Bioinformatics*, 20(9):1388–1397, 2004.

[11]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 Science*, 12(5):1007–1017, 2003.

[12]B. Peters and A. Sette. Generating quantitative models describing the sequence specificity of biological processes with the stabilized matrix method. *BMC Bioinformatics*, 6(1):132, 2005.

[13]M. Ranzato, V. Mnih, J. M. Susskind, and G. E. Hinton. Modeling natural images using gated mrfs. volume 35, pages 2206–2222, 2013.

[14]P. A. Reche, J.-P. Glutting, and E. L. Reinherz. Prediction of MHC class I binding peptides using profile motifs. *Human Immunology*, 63(9):701 – 709, 2002.

[15]P. A. Reche and E. L. Reinherz. Prediction of peptide-MHC binding using profiles. In D. R. Flower, editor, *Immunoinformatics*, volume 409 of *Methods in Molecular Biology*, pages 185–200. Humana Press, 2007.

[16]J. Salomon and D. Flower. Predicting class II MHC-peptide binding: a kernel based approach using similarity scores. *BMC Bioinformatics*, 7(1):501, 2006.

[17]C.-W. Tung, M. Ziehm, A. Kamper, O. Kohlbacher, and S.-Y. Ho. POPISK: T-cell reactivity prediction using support vector machines and string kernels. *BMC Bioinformatics*, 12(1):446, 2011.

[18]G. Zhang, A. M. Khan, K. N. Srinivasan, J. T. August, and V. Brusic. MULTIPRED: a computational system for prediction of promiscuous HLA binding peptides. *Nucleic Acids Research*, 33(Web-Server-Issue):172–179, 2005.

[19]H. Zhang, C. Lundegaard, and M. Nielsen. Pan-specific mhc class i predictors: a benchmark of hla class i pan-specific prediction methods. *Bioinformatics*, 25(1):83–89, 2009.