BIOINFORMATICS

Vol. 00 no. 00 2014
Pages 1–7

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

Pavel P. Kuksa[1,2,3†], Martin Renqiang Min[3,†,*], Rishabh Dugar[3,†], Mark Gerstein[4,5,6,*]

[1] Institute for Biomedical Informatics, University of Pennsylvania School of Medicine, [2] Department of Pathology and Laboratory Medicine, University of Pennsylvania School of Medicine, Philadelphia, PA 19104, USA, [3]Department of Machine Learning, NEC Laboratories America, Princeton, NJ 08540, USA, [4]Program of Computational Biology and Bioinformatics, [5]Department of Molecular Biophysics and Biochemistry and [6]Department of Computer Science, Yale University, New Haven, CT 06511, USA.

[†]These authors contributed equally.

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

ABSTRACT

Motivation: 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 possible deep extensions and high-order Kernel Support Vector Machines to predict major histocompatibility complex (MHC)-peptide binding.

Results: The proposed high-order methods improve quality of binding predictions over other prediction methods. With the proposed methods, a significant gain of up to 25-40% is observed on the benchmark and reference peptide data sets and tasks. In addition, for the first time, our experiments show that pre-training with high-order semi-Restricted Boltzmann Machines significantly improves the performance of feed-forward high-order neural networks. Moreover, our experiments show that the proposed shallow high-order neural network outperform the popular pre-trained deep neural network on most tasks, which demonstrates the effectiveness of modeling high-order feature interactions for predicting MHC-peptide binding.

Availability: Licensed software is available upon request.

Contact: renqiang@nec-labs.com, mark.gerstein@yale.edu

1 INTRODUCTION

Complex biological functions in living cells are often performed through different types of protein-protein interactions. An important class of protein-protein interactions are peptide (i.e. short chains of amino acids) mediated interactions, and they regulate important biological processes such as protein localization, endocytosis, post-translational modifications, signaling pathways, and immune responses etc. Moreover, peptide-mediated interactions play important roles in the development of several human diseases including cancer and viral infections. Due to the high medical value of peptide-protein interactions, a lot of research has been done to identify ideal peptides for therapeutic and cosmetic purposes, which renders *in silico* peptide-protein binding prediction by computational methods a highly important problem in immunomics and bioinformatics (Lundegaard *et al.*, 2011; Brusic *et al.*, 2002; Hoof *et al.*, 2009; Nielsen *et al.*, 2003).

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, transplant rejection, and vaccine design. 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) (Lundegaard *et al.*, 2011; Hoof *et al.*, 2009; Giguere *et al.*, 2013) 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.

Deep learning models such as Deep Neural Networks (DNNs) pre-trained with Restricted Boltzmann Machine (RBM) have been successfully applied to handwritten digit classification, embedding,

[*]To whom correspondence should be addressed

image recognition and many other applications (Hinton, 2010; Min *et al.*, 2010; Ranzato *et al.*, 2013). But they have never been successfully applied to peptide-protein interaction problems.

In this paper, we propose using high-order semi-Restricted Boltzmann Machine (RBMs) to pre-train a feed-forward *high-order* neural network and propose high-order Kernel Support Vector Machine (SVM) for peptide-MHC binding prediction, including identification of MHC-binding, naturally processed and presented (NPP), and immunogenic peptides (T-cell epitopes). Our proposed models achieved a significant gain of up to 25-40% over the state-of-the-art approach on benchmark and reference peptide data sets and tasks. Furthermore, our shallow high-order neural networks even outperformed popular powerful pre-trained deep neural networks that was applied to model peptide-MHC binding prediction for the first time by this work.

2-1

AWK

2 RELATED WORK

Position-specific scoring matrix (PSSM) and matrix based methods: Authors in (Reche and Reinherz, 2007; Reche *et al.*, 2002; Nielsen *et al.*, 2004) derive PSSMs from a set of known binding peptides and use PSSM matching score as an indicator of the binding potential of the query peptide. In (Peters and Sette, 2005), the peptide binding task is solved as a matrix-vector regression problem.
Neural network based methods: Authors in (Zhang *et al.*, 2005; Brusic *et al.*, 2002) built neural networks to predict peptide binding potentials by encoding peptides and contact residues on the MHC molecules as a fixed-dimensional vector of amino-acid and contact residues. Similarly, in (Nielsen *et al.*, 2003; Buus *et al.*, 2003; Lundegaard *et al.*, 2011) authors proposed to use neural networks and committees of networks with peptide representations combining sparse, BLOSUM, and profile HMM encodings of the peptides. In (Hoof *et al.*, 2009), both the peptide sequence and MHC protein sequence are used as input to neural networks in order to enhance predictive ability for MHC alleles with limited peptide binding data.
Kernel-based methods: The work in (Salomon and Flower, 2006) uses the local alignment (LA) kernel method for predicting MHC-II-peptide binding. Authors in (Tung *et al.*, 2011) adopt weighted-degree kernels to identify immunogenic peptides. In (Liu *et al.*, 2007), authors employ support vector regression (with RBF, polynomial, etc kernels) using sparse encoding of a peptide sequence and 11-dim physicochemical amino-acid descriptors. Recent work (Giguere *et al.*, 2013) uses kernel logistic regression for MHC-II-peptide binding prediction using both peptide and MHC sequences. In (Gigure *et al.*, 2013), an SVM with kernel from (Giguere *et al.*, 2013) is used for naturally processed and presented ("eluted") peptide prediction.

2-2

VIPETAL.

3 METHODS

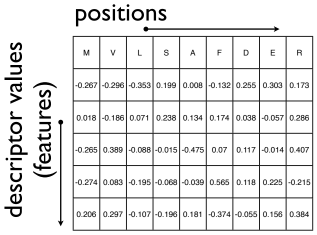
In order for the peptides to bind to a particular MHC allele (i.e., its peptide-binding groove), the sequences of the binding peptides should be approximately superimposable: contain amino-acids or strings of amino acids (k-mers) with similar physicochemical properties at approximately the same positions along the peptide chain.

It is then natural to model peptide sequences $X = x_1, x_2, \ldots, x_n$, $x_i \in \Sigma$ (i.e., sequences of amino acid residues) as a sequences of *descriptor* vectors $\mathbf{d}_1, \ldots, \mathbf{d}_n$, encoding relevant properties of amino acids observed along the peptide chain and/or MHC-peptide interaction terms.

3.1 Descriptor Sequence peptide representations

While the descriptor vectors $\mathbf{d}_i$ in general may be of unequal length, in the matrix form (equal-sized vectors $\mathbf{d}_i \in \mathcal{R}^R$) of this representation ("feature-spatial-position matrix"), the rows are indexed by features (e.g., individual amino acids, strings of amino acids, k-mers, physicochemical properties, peptide-MHC interaction features, etc), while the columns correspond to their spatial positions (coordinates). Figure 1 illustrates descriptor sequence representation of a nonamer.

In this descriptor sequence representation, each position in the peptide is described by a feature vector, with features derived from the amino acid occupying this position or from a set of amino acids (e.g., a k-mer starting at this position or a window of amino acids centered at this position) and/or amino acids present in the MHC protein molecule and interacting with the amino acids in the peptide.


positions
descriptor values (features)

M	V	L	S	A	F	D	E	R
-0.267	-0.296	-0.353	0.199	0.008	-0.132	0.255	0.303	0.173
0.018	-0.186	0.071	0.238	0.134	0.174	0.038	-0.057	0.286
-0.265	0.389	-0.088	-0.015	-0.475	0.07	0.117	-0.014	0.407
-0.274	0.083	-0.195	-0.068	-0.039	0.565	0.118	0.225	-0.215
0.206	0.297	-0.107	-0.196	0.181	-0.374	-0.055	0.156	0.384

Fig. 1: Peptide descriptor sequence representation of a nonamer 'MVLSAFDER' using 5-dim amino acid descriptors

The purpose of a descriptor is to capture relevant information (e.g., physicochemical properties) that can be used by our high-order neural networks and kernel functions to differentiate peptides into binding, non-binding, immunogenic, etc.

A *real-valued* descriptor of an amino acid is a quantitative descriptor encoding (1) relevant properties of amino acids such as their physicochemical properties and substitution probabilities by other amino acids, and/or (2) interaction features (such as binding energy) between the amino acids in the peptide and those in the MHC molecule. An example of the real-valued descriptor sequence representation of a peptide using 5-dim physicochemical amino acid descriptors is given in Figure 1.

3.2 Deep Neural Network and High-Order Neural Network

Given the matrix-form descriptor representation of each peptide based on BLOSUM substitution matrix as illustrated above, we concatenate all the columns of the matrix into a long vector as input feature vector to our neural networks. In this representation, a 9-mer peptide is represented by a 180-dimensional continuous vector, with each amino acid represented by its corresponding 20-dimensional substitution probabilities. Instead of using an ensemble of traditional neural networks to predict MHC class-peptide bindings as in the state-of-the-art approach NetMHC (Nielsen *et al.*, 2003; Buus *et al.*, 2003; Lundegaard *et al.*, 2011), we use High-Order Neural Networks (HONN) pre-trained with a special type of high-order Semi-Restricted Boltzmann Machines (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). The pre-training strategy has been widely adopted for training a popular powerful model called Deep Neural Networks (DNN) (Hinton *et al.*, 2006; Bengio, 2009).

DNN has attracted world-wide attention in the machine learning community recently. In this paper, for the first time, we apply DNN to predict peptide-MHC binding, and we compare its performance to our proposed HONN. DNN is shown on the left panel of Fig. 1. We use Gaussian RBM to pre-train the network weights of its first layer, and we use binary RBM to pre-train the connection weights of upper layers in a greedy layer-wise fashion. Our proposed High-Order Neural Network (HONN) is shown on the right panel of Fig. 1. We use mcRBM to pre-train the network weights of it first layer, and we optionally add upper layers, 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

2

FIG2

2-3

WHAT'S THIS

3-1

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 σ_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)(Hinton, 2002) 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,

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

use v, h def

3-2

where ϵ 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 (Hinton, 2010) tried to use hidden units to gate the pairwise interaction between the visible units, leading to the following energy function,

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

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,

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

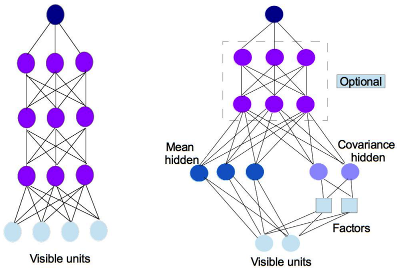
In the above equation, each hidden unit modulates the interaction between each pair of input features leading to a large number of parameters in w_{ijk} to be learned. To reduce this complexity, we can factorize the weight w_{ijk} as follows (Hinton, 2010),

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

The energy function can now be written as

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

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.



EXPL

3-3

Fig. 2: The structure of DNN (left) and HONN (right).

During pre-training, 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 (Ranzato *et al.*, 2013) with 20 leap-frog steps. During the fine-tuning of both DNN and HONN, we used gradient descent with mini-batch size 15, learning rate 0.01, and momentum 0.9. The structures of both DNN and HONN are decided based on performance on validation sets. Although HONN can be easily extended to have many upper layers to form a deep architecture, HONN without deep extensions works best in all our experiments, which is probably due to the limited training data we have.

3.3 High-order Kernel Models

The sequence of the descriptors corresponding to the peptide $X = x_1, x_2, \ldots, x_{|X|}$, $x_i \in \Sigma$ (as in, e.g., Fig 1) 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.

3.4 High-order kernel functions on peptide descriptor sequence representations

In the following we define kernel functions for peptides based on peptide descriptor sequence representations (such as in Fig. 1). The proposed kernel functions for peptide sequences X and Y have the following general form:

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

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. While $k_{\mathbf{d}}$ measures similarity between descriptors, the context/position kernel $k_{\mathbf{p}}$ measures similarity of the of the descriptor context (e.g., position, spatial distribution of amino acids, etc). 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 $k(i,j|\alpha,\beta)=\dfrac{1}{|i-j|^\alpha}+\beta=exp(-\alpha\log(|i-j|))+\beta$

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 $\mathbf{d}_i=(d_1^i,d_2^i,\ldots,d_R^i)$ and $\mathbf{d}_j=(d_1^j,d_2^j,\ldots,d_R^j)$ 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 $k_{\mathbf{p}}(\mathbf{p}_a,\mathbf{p}_b)$.

4 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*. The datasets used for MHC-I binding prediction task are listed in Table 1.
2. *Naturally processed ("eluted") peptide prediction*. We use recently compiled benchmark data from the 2nd Machine Learning in Immunology competition (MLI-II). Table 2 provides details of this dataset.
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, and HLA-A*2402. The choice of these alleles is motivated by the target population group (Japanese) in our research lab. The application of our method to other alleles or peptide lengths would be straightforward.

4.1 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) (Vita *et al.*, 2010). The datasets are labeled with `IEDB` suffix in Table 1.

Table 1. Peptide-MHC binary datasets (binding/non-binding)

Dataset	#peptides	#binders	#non-binders
A0201-IEDB	8471	3939	4532
A0201-Japanese	281	106	175
A0206-IEDB	1820	951	869
A0206-Japanese	278	97	181
A2402-IEDB	2011	890	1121
A2402-Japanese	405	176	229

Table 2. Naturally-processed (NP) peptide datasets

Dataset	#peptides	#eluted	#non-eluted
A0201-MLI-II	8225	971	7254
A0201-MLI-II-EvalSet	492	63	429

For testing, we use the experimental data from our lab for each allele. These datasets are denoted with `'Japanese'` suffix in Table 1. The training `'IEDB'` datasets and the test `'Japanese'` datasets are completely disjoint.

4.2 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 $f(P_1),\ldots,f(P_N)$, the discounted cumulative gain (DCG_N) is defined as a sum of individual peptide relevance scores (experimentally determined 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 (i.e., normalized DCG=1 when the predictor orders peptides according to their actual binding strength).

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., area under ROC curve) that measure whether "binders" are ranked higher than "non-binders" *irrespectively* of the actual peptide binding strength). This measure is popular for assessing performance of the document retrieval systems (e.g., Web search engines) as it is maximized if the most relevant documents appear at the top of search results, but it has not been used to differentiate performance of the MHC binding predictors. In the case of the peptide-MHC prediction, the nDCG is maximized if peptides are placed (according to the predictor output) in the ideal order: from the strongest binders to the weakest/non-binders. We emphasize that the two methods with the same AUC scores, may differ significantly with respect to their nDCG scores: even with the equally good separation between "binders" and "non-binders" for the two methods, the method that correctly ranks stronger binders higher than weaker binder will have a higher nDCG score.

5 RESULTS

We first present results for MHC-I binding prediction on benchmark datasets and experimental data from our lab (Sec. 5.1). We show next results on predicting peptides naturally processed by the MHC pathway (Sec. 5.2). Finally, we show results for predicting promising T-cell epitopes for clinical development (Sec. 5.3). The following AUC and nDCG scores are shown in %.

5.1 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., (Lundegaard *et al.*, 2011; Zhang *et al.*, 2009; Gigure *et al.*, 2013)).

We first use `'Japanese'` data sets to test our methods. Results are shown in Tables 3, 5, 7 for target alleles on `Japanese` test datasets. Corresponding ROC curves are shown in Figure 3f (top row). We also plot $nDCG@n$ curves in Fig. 3f (bottom row), where $nDCG@n$ is $nDCG$ up to nth peptide in the sorted output (i.e., nDCG of the top-n predicted peptides).

As evident from the AUC and ROC-n results in the tables and ROC plots, our method achieves significant improvements in separating "binders" vs "non-binders". For example, for A2402 allele ROC-n=10 score increases from 66.88 for NetMHC to 77.76 for HONN and hkSVM. Similar improvements are observed on A0201 allele data where ROC-n=10 score improves from 26.61 for NetMHC to 35.59 with HONN and hkSVM.

At the same time, the results in terms of nDCG quality scores suggest significant increase in ranking quality (Tables 4,6, and 8). 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. For instance, for the A0201 allele $nDCG@n$ scores improve from 60.98, 63.50 achieved by NetMHC to 65.94, 70.61 using our HONN method for $n = 20$ and $n = 30$ respectively.

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 supplementary material (Supplementary Tables S2-S7). All the results of DNN and HONN reported in the main paper are based on pre-training and fine-tuning.

Using a combination of network and kernel models 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).

Table 3. Comparison of AUC test scores on A0201-Japanese data

method	AUC	ROC-10	ROC-20	ROC-30	ROC-50
hkSVM	**79.60**	32.71	**50.59**	**63.67**	**77.56**
DNN	77.23	30.34	47.03	60.11	74.95
HONN	77.26	33.39	48.14	60.11	74.98
hkSVM+HONN	**79.11**	**35.59**	**50.51**	62.99	**77.02**
NetMHC	76.90	26.61	46.02	58.87	74.47

Table 4. A0201-Japanese data. Relevance/ranking quality (nDCG).

method	nDCG@10	nDCG@20	nDCG@30	nDCG@50	nDCG
hkSVM	60.69	61.75	66.78	74.11	85.01
DNN	63.89	65.59	70.12	74.57	86.33
HONN	63.93	**65.94**	70.61	75.55	86.46
hkSVM+HONN	**65.69**	65.12	**71.49**	**76.46**	**86.98**
NetMHC	59.48	60.98	63.50	72.68	83.94

Table 5. Comparison of AUC test scores on A0206-Japanese data

method	AUC	ROC-10	ROC-20	ROC-30
hkSVM	**86.23**	**54.84**	72.58	78.68
DNN	80.24	52.42	64.02	71.31
HONN	84.41	49.7	69.7	77.78
hkSVM+HONN	**86.24**	**54.24**	**73.33**	**80.2**
NetMHC	83.93	50.91	67.42	76.77

Table 6. A0206-Japanese data. Relevance/ranking assessment (nDCG)

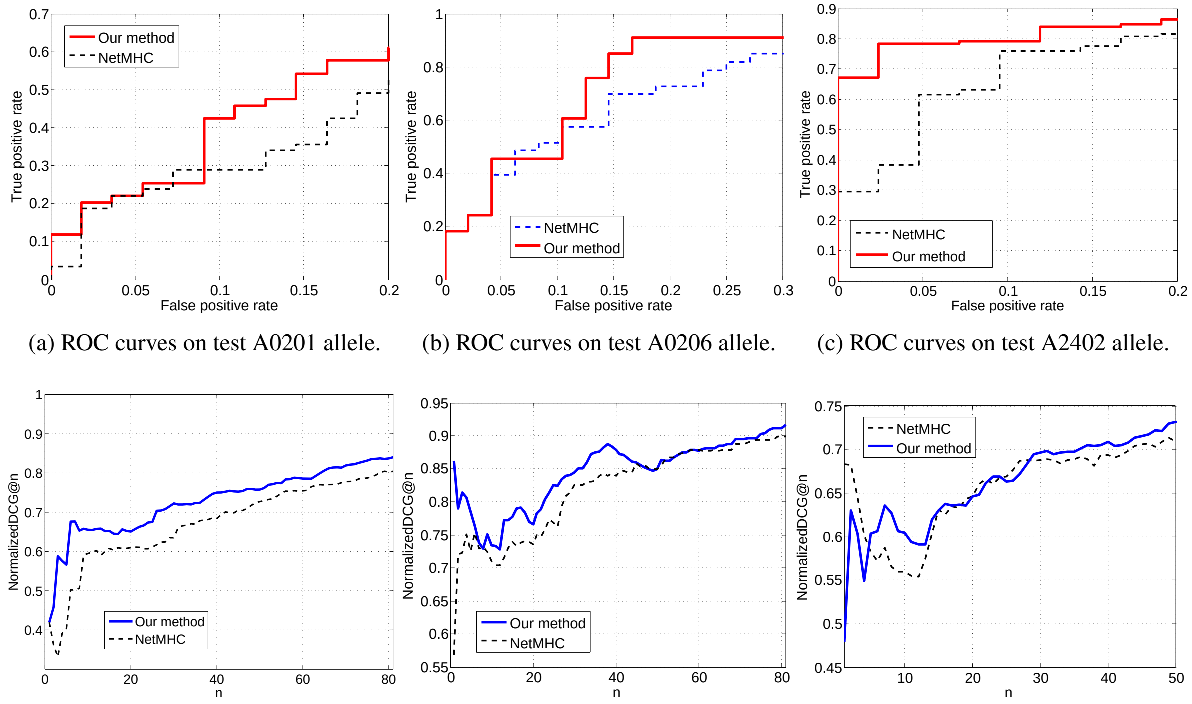
method	nDCG@10	nDCG@20	nDCG@30	nDCG
hkSVM	76.52	74.64	82.49	91.43
DNN	77.50	**82.21**	81.72	91.74
HONN	75.39	78.06	79.92	90.80
hkSVM+HONN	**80.2**	76.98	**83.75**	**91.75**
NetMHC	70.97	73.60	82.57	89.88

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 (i.e., only with binding (B) vs non-binding (NB) information), 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 ING data (e.g., IC50) to further improve our results.

S-1

To visualize the learned weights of HONN, we used 8 mean hidden units, 1 covariance hidden unit, and 1 factor unit to train HONN on the training data of A2402. We obtained AUC score 86.02 and nDCG score 85.01 that are slightly worse than the ones in Table 7 and 8. In Fig. 4, the factorized rank-1 interaction weight vector with absolute values greater than 0.1 is shown in the top, and the weight matrix connecting input features and mean hidden units with absolute values greater than 0.02 is shown at the bottom. This figure clearly shows that position 2, 8, 9, and the interaction between middle position and position 9 are very important for predicting 9-mer peptide binding, which has experimental support from the crystal structure of the interaction complex (Cole *et al.*, 2006).

WHAT MEAN INTUIT [illegible]



(a) ROC curves on test A0201 allele.

(b) ROC curves on test A0206 allele.

(c) ROC curves on test A2402 allele.

(d) Normalized DCG curves on test A0201 allele.

(e) Normalized DCG curves on test A0206 allele.

(f) Normalized DCG curves on test A2402 allele.

Fig. 3: ROC curves (top row) and normalized discounted cumulative gain (nDCG) curves (bottom row).

Table 7. 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 8. 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

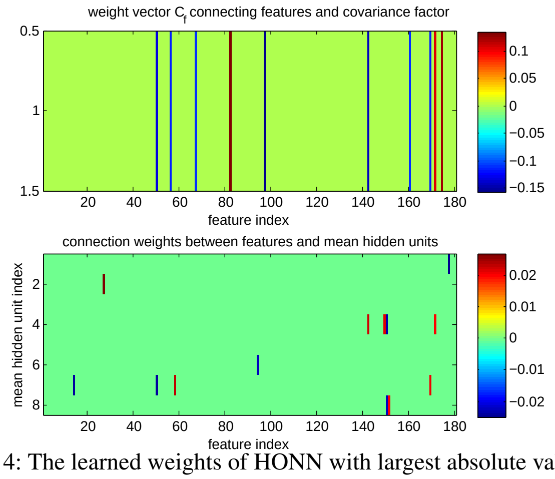


Fig. 4: The learned weights of HONN with largest absolute values.

5.2 Naturally processed (NP) 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" in Sec. 5.1) 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 *in vivo*.

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

We directly train our models to recognize naturally processed and presented 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 (Gigure *et al.*, 2013) method that yielded state-of-the-art accuracy for naturally processed (NP) peptide prediction.

Table 9 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 (Gigure *et al.*, 2013) method.

5.3 Epitope prediction

We demonstrate ability of the method to predict promising peptides for clinical development using as an example WT1-derived strong

Table 9. Naturally processed (NP) 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 (Gigure *et al.*, 2013)

Table 10. Prediction of WT1-derived epitopes

	NetMHC-rank	hkSVM+HONN-rank
A0201 allele		
W10	2	1
W302	20	2
A0206 allele		
W10	2	1
W302	8	3
A2402 allele		
W10	41	2
W302	7	4

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 (HONN and hkSVM). The order of the W10 and W302 peptides in the output (out of the 441 peptides) of the two prediction methods is given in Table 10. As evident from the table, our method ranks these peptides higher than NetMHC method.

6 DISCUSSION AND FUTURE WORK

In this paper, we propose using nonlinear high-order machine learning methods including HONN 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 directly 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 HONN model with possible deep extensions, 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.

ACKNOWLEDGMENT

We thank Dr. Keiko Udaka for providing valuable experimental datasets and validations, and we thank Dr. Hans Peter Graf for helpful discussions.

REFERENCES

Bengio, Y. (2009). Learning deep architectures for AI. *Found. Trends Mach. Learn.*, **2**(1), 1–127.

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

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

Cole, D. K., Rizkallah, P. J., Gao, F., Watson, N. I., Boulter, J. M., Bell, J. I., Sami, M., Gao, G. F., and Jakobsen, B. K. (2006). Crystal structure of HLA-A*2402 complexed with a telomerase peptide. *European Journal of Immunology*, **36**(1), 170–179.

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

Gigure, S., Drouin, A., Lacoste, A., Marchand, M., Corbeil, J., and Laviolette, F. (2013). MHC-NP: Predicting peptides naturally processed by the MHC. *Journal of Immunological Methods*, **400**, 30–36.

Hinton, G. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation*, **14**(8), 1771–800.

Hinton, G., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, **18**, 1527–1554.

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

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

Liu, W., Wan, J., Meng, X., Flower, D., and Li, T. (2007). 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.

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

Min, M. R., van der Maaten, L., Yuan, Z., Bonner, A. J., and Zhang, Z. (2010). Deep supervised t-distributed embedding. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 791–798.

Nielsen, M., Lundegaard, C., Worning, P., Lauemøller, S. L., Lamberth, K., Buus, S., Brunak, S., and Lund, O. (2003). Reliable prediction of T-cell epitopes using neural networks with novel sequence representations. *Protein Science*, **12**(5), 1007–1017.

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

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

Ranzato, M., Mnih, V., Susskind, J. M., and Hinton, G. E. (2013). Modeling natural images using gated MRFs. *IEEE Trans. Pattern Anal. Mach. Intell.*, **35**(9), 2206–2222.

Reche, P. A. and Reinherz, E. L. (2007). 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.

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

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

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

Vita, R., Zarebski, L., Greenbaum, J., Emami, H., Hoof, I., Salimi, N., Damle, R., Sette, A., and Peters, B. (2010). The immune epitope database 2.0. *Nucleic Acids Res.* `http://www.iedb.org`.

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

Zhang, H., Lundegaard, C., and Nielsen, M. (2009). Pan-specific MHC class I predictors: a benchmark of HLA class I pan-specific prediction methods. *Bioinformatics*, **25**(1), 83–89.