Utilizing Domain Knowledge in Neural Network Models for Peptide-Allele Binding Prediction

Vasileios Megalooikonomou[1], Despina Kontos[1], Nicholas DeClaris[2,3], Pedro Cano[4]

[1] Department of Computer and Information Sciences, Temple University, Philadelphia, USA
[2] Department of Pathology, University of Maryland at Baltimore, Baltimore, MD
[3] Department of Electrical and Computer Engineering, University of Maryland at College Park, MD
[4] Department of Laboratory Medicine, University of Texas, MD Anderson Cancer Center, TX
vasilis@temple.edu, dkontos@temple.edu, declaris@eng.umd.edu, pcano@mdanderson.org

***Abstract*- We developed Radial Basis Function Neural Networks (RBFNN) for allele-peptide binding prediction. We explored utilizing prior domain knowledge in order to optimize the prediction. We investigated the effect of encoding of inputs of the RBFNN considering chemical properties of amino acids, detecting motifs in alleles and reducing the dimensionality based on common motifs discovered. We also explored a number of parameters such as the data set size, unknown-binding data generation, model architecture and training algorithms. Our approach improved the prediction accuracy of peptide-allele binding reaching up to 90% for our best models.**

I. INTRODUCTION

The immune system is composed of many interdependent cell types, organs, and tissues that jointly protect the body from infections (bacterial, parasitic, fungal, or viral) and from the growth of tumor cells. This convoluted structure is considered to be the second most complex body system in humans, after the brain. One of the key players in regulating immune response are the T-cells. In order for these to be activated, a very critical stimulus is required, which is generated from peptides bound to Major Histocompatibility Complex (MHC) Class I molecules. The human MHC is known as the Human Leukocyte Antigen (HLA). The peptides are molecules formed by the linking (in a defined order) of amino acids. An allele is any of the possible DNA codings of the same gene. Binding of peptides to HLA alleles is necessary for immune reaction, but there is a specific limited number of peptides that can bind to a certain HLA molecule. The core for determining the matching in the chain of amino acids of a peptide is 9 amino acids long. Our immune system has to discriminate between self and non-self peptides in order to regulate the immune system responses appropriately.

Predicting this binding is very important in understanding immunity, more specifically designing candidates for vaccines and immunotherapeutic drugs, predicting the success of a transplant, studying cancer growth, even understanding the immunological deficits that follow HIV infection. The importance of computational analysis in this field is increasing with recent advances in both the fields of biology and computer science, especially with the advances of clinical immunology and the accumulation of experimental data.

In this study we address several issues encountered in knowledge discovery and prediction in allele-peptide binding databases. We address the question of how to utilize prior knowledge on amino acid chemical properties in order to optimize allele-peptide binding prediction. We employ the modular neural network (MNN) model based on Radial Basis Functions (RBFNN) which provides the ability to apply different criteria in the modules and easily adapt the network to the domain problem by utilizing relevant domain knowledge. We also explore approaches for generating training datasets with balanced distributions within contrasting classes in order to efficiently train the classifier models and avoid overfitting due to training bias. Furthermore, we consider clustering of alleles, detection of motifs and reduction of dimensionality to further improve prediction accuracy.

II. BACKGROUND

A. Related work on peptide-allele binding

Predicting MHC class I to peptide binding is a field where computational methods have been applied in order to investigate certain biological patterns [1-2]. The techniques that have been mostly used for this purpose are Artificial Neural Networks (ANN), Hidden Markov Models (HMMs), Support Vector Machines (SVM) and methods for detecting binding motifs.

A review of how ANNs have been used in these immunology reaction predictions can be found in [3]. Usually, a three layer feed-forward ANN is used that predicts binding peptides for a specific MHC molecule. The inputs of the network correspond to a binary vector representing the peptide amino acids. Each amino acid is encoded by a 20-bit number having "1" at the bit representing the particular amino acid and "0" at the rest of the bits. Hence, a 180 input ANN with one output can be trained using binder and non-binder peptides (represented by 9 amino acids) in the dataset. More applications of ANNs in allele-peptide binding prediction can be found in [4-7].

HMMs have been utilized to model the profiles of peptide binders and accordingly predict the success of a peptide to MHC molecule binding [8]. SVMs have also been recently used for small datasets [9]. Binding motifs, a method already

employed in molecular biology for general sequence alignment and homology searches in databases, have been employed as well for MHC to peptide binding prediction [8].

Many of these approaches have focused on specific MHC molecules and constructed prediction models corresponding only to those. We are interested in more general predictors that are trained on a group of MHC molecules (alleles) and can predict the success of allele to peptide binding in general, when such a pair is being presented. This could be particularly useful for transplant immunology. Furthermore, these previous approaches are based on amino acid sequence pattern matching. Amino acid sequences reveal information about the structure and function of a protein. We are interested in predictors that detect and utilize binding properties based on the structure and chemical function of the amino acids.

B. An overview of Radial Basis Function Neural Networks

Radial Basis Function Neural Networks (RBFNNs) are considered a subclass of Modular Neural Networks (MNNs). RBFNNs are nonlinear feed-forward networks. Their most particular characteristic is their architecture, which is based on Radial Basis Function (RBF) activation units. Typical RBFNNs are constructed by one hidden layer with RBF units and an output layer with only one linear neuron (see Fig. 1). Hence, the computational models are different for the units of each layer. In contrast to sigmoid functions, radial basis functions have radial symmetry about a center c in the n-dimensional space where n equals to the number of inputs. The spread σ indicates the selectivity of each neuron. The selectivity has an inverse relationship to the spread of the RBF. As the spread of the RBF becomes wider the selectivity of the neuron tends to become smaller.

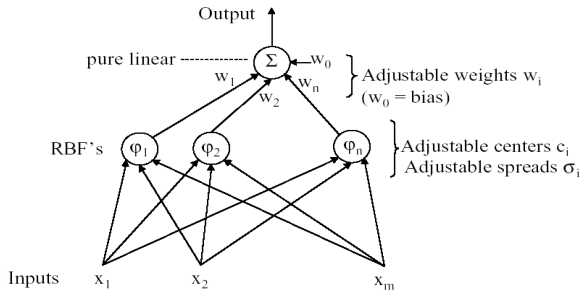


Fig. 1. A Radial Basis Function Neural Network (RBFNN)

According to the *regularization theory* [10], an optimal *regularization* RBFNN uses one hidden unit for each data point in the training set. The activation of these hidden units is defined by the *Green's* functions [10], having each training sample as center to one unit of the hidden layer. The *regularization* network is a universal approximator. Nevertheless, this is an optimization very difficult to achieve in practice due to the restrictive computational cost for learning the large number of network parameters during the training phase. To overcome these computational obstacles, the *generalized* RBFNNs implement an approximation to the *regularization* theory, by searching for a suboptimal solution in a lower dimensional space in order to determine the weights, spreads, and centers for best fit [10-13]. A range of different techniques have being employed [12-13] in order to select a subset of the available training samples that can be used as RBF unit centers c, while providing sufficient accuracy. This is achieved by selecting as unit centers c samples that are highly representative of groupings and interesting patterns in the dataset. Most of these techniques though are based on heuristics that cannot guarantee optimal subset selection and some loss in accuracy is often introduced.

The functionality of an RBFNN is based on a distance metric (usually the *Euclidean Norm*) that is computed by the RBF units among the input and the selected center of each RBF unit. At the input of each hidden unit, the distance between the unit center c and the input vector is calculated. The output of the hidden RBF unit is then formed by applying the radial basis function to this distance. In order for this functionality to be effective, the input space has to be represented (encoded) in a way that such a distance computation is meaningful and reflects significant properties of the training samples. Compared to other Neural Network architectures, RBFNNs have the advantage of not converging to *local minima* during the training phase. It has also been shown that the output of RBFNNs reflects the response of a mixture of experts, when Gaussian radial functions are selected for the hidden units.

In the particular case of predicting allele-to-peptide binding, it is very important that the allele-peptide pair input sequences are encoded in a meaningful manner, in order for the RBFNN functionality to be effective. More specifically, the aminoacid input sequences have to be encoded properly in order to reflect binding properties and chemical similarities when the distance metric is calculated in the hidden RBF units. In other words, aminoacid sequences that appear to be similar when applying a distance metric (i.e. the Euclidean distance) should be similar with respect to binding and chemical properties. This can be achieved by selecting an amino acid representation that reflects binding properties.

III. DATASET

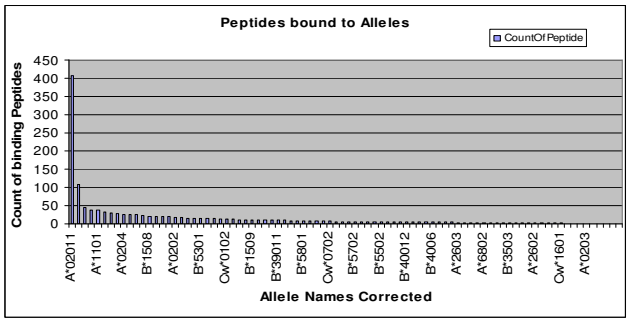
Before applying the proposed methodology, the data sets were treated in order to become appropriate to be used with the prediction models. We performed certain preprocessing steps in order to compensate for high dimensionality, noise, imprecisions and class imbalances. We also investigated techniques for generating training datasets with balanced distributions within the contrasting classes, in order to efficiently train the RFNNs and deal with training bias.

A. The original dataset

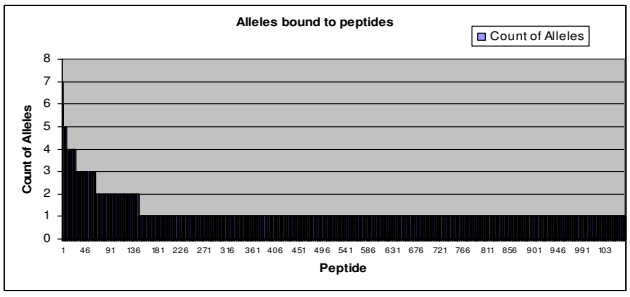
The original data set contained alleles, peptides, and known combinations of bindings showing which alleles bind to which peptides. Confirmed non-binding pairs of alleles and peptides were not available. The amino acid sequences of both the alleles and the peptides were also available. The data set consisted of 426 unique alleles and 1080 unique peptides. Of these, 86 unique alleles were known to bind to at least one specific peptide in the data set, and 1080 unique peptides were

known to bind to at least one specific allele in the data set. We considered 1318 unique allele-peptide combinations (complexes) in the data set that were known to bind.

In our original dataset, the binding distribution of alleles and peptides was skewed. In the 1318 binding pairs, a few alleles were present most of the time (see Fig. 2(a)). In particular, one allele (A*02011) participated in 407 binding pairs. The peptides were more evenly distributed. Their distribution is shown in Fig. 2(b). There is only one peptide that binds to 7 alleles; most of the peptides bind to only one allele. For the prediction model we used 83 polymorphic allele aminoacid positions. These are positions below 199 since the important part of the molecule is in the first 200 amino acids and any amino acid position after 199 is considered not reliable due to incomplete sequence data for various alleles in our database.



(a)



(b)

Fig. 2. Binding statistics.

B. Dealing with the unavailability of 'negative' data

To cope with the unavailability of negative data (i.e. confirmed non-binding pairs) we used two classes, "known" and "unknown" for training and testing. For this reason we needed to generate allele-peptide combinations for which it was not known whether the allele binds to the peptide. We used a number of different ways to generate these "unknown" combinations.

1. **Unknown-StaticRandom**: We created the Cartesian product of all the 86 unique alleles that were known to bind to at least one peptide in the data set and all the 1080 unique peptides that were known to bind to at least one allele in the data set (92,880 pairs). Note that the Cartesian product lists all the combinations of peptides and alleles (binding and non-binding pairs). From this Cartesian product we removed the allele-peptide pairs that have a confirmed binding property (1318 pairs). This resulted in 91,562 actual unknown binding pairs of alleles and peptides. We then randomly sampled the resulting set of unknown binding pairs. Since the known binding pairs were 1318, we created a balanced dataset of both classes by randomly selecting 1318 unknown binding pairs out of the 91,562 available ones. This dataset is called static since the 1318 unknown pairs were selected once and were fixed after the selection (this is in contrast to the dataset in item 2 below). Hence, this dataset was built up from 1318 known binding pairs of allele-peptides (class 1) and 1318 unknown binding pairs of allele-peptides (class 2).
2. **Unknown-DynamicRandom:** In this approach we used again the complete set created by the Cartesian product of all possible combinations (pairs) of peptides and alleles available in our original dataset (92,880 pairs) after removing the ones that have a confirmed binding property (1318 pairs). This dataset was particularly employed for cross-validation experiments with ensemble classifiers. We dynamically selected training datasets by selecting "on-the-fly" the set of 1318 pairs of peptides-alleles for which the binding property was unknown. We randomly picked each time different 1318 unknown pairs. Hence, for each cross-validation experiment, we created a different dataset with the 1318 known binding peptide-allele pairs (class 1) and a set of 1318 unknown binding pairs (class 2) randomly selected each time out of the 91,562 available ones.
3. **Unknown-SimilartoKnown**: This dataset was constructed so that we have the same distribution of alleles in the generated known binding set (class 1) as in the unknown binding set (class 2), in order to take care of the skewed binding distribution of certain alleles in the known binding set (see Fig. 2). This dataset is called Unknown-SimilartoKnown since there is more similarity between the distributions of unknown and known combinations. Both of them are skewed in the same manner. The unknown data in this case were generated manually using a lengthy process. The resulting dataset consisted of 1318 binding and 1318 unknown binding pairs. More specifically, peptides were grouped according to how many alleles they bind with and an equal number of each peptide grouping was selected in the unknown dataset. In the process of constructing the test dataset, we observed that peptides tend to bind to alleles from the same family (A, B, C). Hence, in order to simulate as much as possible actual unknown (or rather non-binding) pairs we selected for each peptide alleles from contrasting classes. The distribution of the alleles in both datasets was also preserved, especially for the alleles that seem to dominate the known binding pairs (A*02011 and B*27052).

IV. METHODS

In this section we describe in detail the data representation, and the architecture of the proposed predictor models. We also present our approach for incorporating prior knowledge in order to optimize binding prediction. This is a challenging task not only because extracting domain knowledge is difficult but also because bringing it in a form that is useful for the model is even more intricate.

A form of domain knowledge in the particular application of allele-peptide binding is the chemical properties of amino acids. Nevertheless, it is not currently clear which chemical properties are associated with this binding function. We investigate how amino acid hierarchies based on chemical properties can be employed for determining optimal encoding. Our hypothesis is that incorporating such domain knowledge in the encoding scheme should increase the information content of the representation and boost the binding prediction accuracy. Following this principle, the encoding of the inputs of the neural network was performed in such a way so that amino acids that are "closer" in terms of chemical properties have encodings that are "close" to each other as well.

Having to deal with a large number of high-dimensional training points, we implement RBFNNs, which have the advantage of transforming the input space into a higher dimensional space where the problem becomes linear separable. Taking also into consideration that large encoding schemes give rise to a high dimensionality of the input space and RBFNNs cannot generalize appropriately, domain knowledge is also utilized in order to reduce the dimensionality of the input sequences. For this purpose, we apply preprocessing to the data before presenting them as inputs to the neural network. The preprocessing involves clustering and motif discovery in order to detect aminoacid positions that are redundant with respect to discovering interesting allele-peptide binding patterns.

Finally, we studied the relationship between model performance improvements and the training/testing data set information content increment, by ranging the percentage of the available data used for training vs. testing. This may assist guiding the process for collection of new data so that the data collected are as informative as possible, i.e., they are in areas of the search space that are critical for the predictive model.

A. Encoding of Aminoacid Sequences

We have worked on improving the representation of the amino acid sequences as inputs to the neural network. We have used a hierarchy of the amino acids based on their chemical properties, which is illustrated in Fig. 3. Aminoacids A, V, L, I, P, F, W, and M are non-polar; G, S, T, Y, C, N, and Q are polar, not charged; D and E are polar, negatively charged; and K, R, and H are basic, positively charged. We employed tree node labeling of breadth first search in-order traversal with interleaved binary labels to construct new binary and decimal representations that preserve the chemical distances as much as possible. These representations would be more appropriate for RBFNNs since the radial units actually compute the distances between the input vector and the unit center.

Two different encodings were constructed for the amino acids: a binary and a decimal encoding. In the first encoding approach, the amino acids were coded into a binary code that was represented with 6 bits. In the second approach, a decimal encoding of the binary number corresponding to each amino acid was used representing each amino acid with a single decimal number. The different encodings that were used for the amino acids are shown in Table I. The missing value (for an amino acid) was represented as [111111] in the bit encoding and as the number 38 in the decimal encoding.

B. Basic RBFNN model that utilizes domain knowledge

In this basic model, prior knowledge is integrated through the encoding of the amino acids based on their chemical properties (i.e. amino acid hierarchy). The RBFNN has inputs equal to the total number of amino acids representing the peptides and alleles, which is equal to 92. The neurons of the hidden layer use the radial basis function for activation whereas the output neuron is linear.

In addition to the encoding of the amino acid sequences, we search the full space of RBFNN optimization parameters to select appropriate values for these parameters and make proper selections of the spread of the Radial Basis Functions (RBFs). For the experiments we performed the spread was set to 0.5 for all RBFs. The threshold on the prediction error that was used during the training phase was set to 0.01. These models were implemented using the neural network toolbox PRTools 3.1.7 (Pattern Recognition Tools) for Matlab [14].

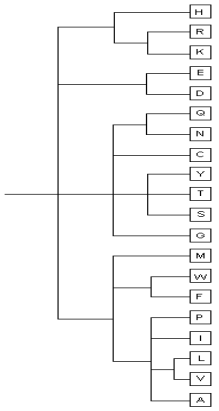


Fig.3. Hierarchical classification of amino acids based on chemical structure.

TABLE I
THE TWO ENCODINGS USED FOR THE AMINOACIDS

Amino Acids	Binary Encoding	Decimal Encoding
A	1 0 0 0 1 0	34
C	0 1 1 1 0 0	28
D	1 1 0 0 0 0	48
E	1 1 0 1 0 0	52
F	0 1 0 0 0 0	16
G	1 0 0 0 0 0	32
H	1 1 1 0 0 0	56
I	0 0 1 1 0 0	12
K	1 1 1 0 1 0	58
L	0 0 1 0 0 0	8
M	0 1 1 0 0 0	24
N	1 1 1 1 0 0	60
P	1 0 1 1 0 0	44
Q	1 1 1 1 1 0	62
R	1 1 1 0 1 1	59
S	1 0 0 1 0 0	36
T	1 0 0 1 1 0	38
V	0 0 0 0 0 0	0
W	1 0 1 0 1 0]	42
Y	1 0 1 0 0 0	44
X	1 1 1 1 1 1	38 (median)

C. Motif Discovery – Detecting sequence patterns

In order to further investigate the aminoacid sequence patterns responsible for allele-to-peptide binding we performed experiments to discover highly conserved aminoacid sequence regions (motifs). For this purpose we used TEIRESIAS [15-16] (http://cbcsrv.watson.ibm.com/Tspd.html), which is a tool for biological sequence pattern discovery and is distributed freely on the web. The main goal was to identify motifs on allele and peptide sequences. The motivation came from the high dimensionality we observed due to the large number of positions we considered for alleles and the insufficient data (initially unbalanced classes) we had to fit the neural network model. Our primary idea was to further reduce the dimensionality of the sequences by removing the positions corresponding to motifs that are present to all the sequences (i.e., reduce from the original 83 and 9 positions). Another aim was to include the positions of alleles only for specific binding motifs that are present to groups of alleles that bind to specific groups of peptides. We wanted to investigate whether specific binding peptide-allele motifs were present in the dataset. We also intended to use these groupings to reduce the number of different alleles and different peptides we considered. In particular, we planned to explore the possibility of using groups of alleles and peptides that go together to form the centers of the radial basis functions.

We detected motifs present either in all the selected allele sequences (considering only the sequence constructed by the selected and polymorphic positions) or within specific families of alleles (A, B or C). These motifs had high statistical significance and support. We used these findings to reduce the dimensionality of the allele sequences. This was achieved by removing from the input of the RBFNN the amino acid positions corresponding to motifs present to all the peptide-allele sequences (redundant information). In the case of the peptides, since the sequences included in our study consisted only of 9 selected aminoacid positions, no significant motifs were detected.

D. Advanced RBFNN models that utilize prior knowledge

In these advanced RBFNN models we used again the amino acid encodings based on the hierarchy of their chemical properties. In addition, we used motif discovery and dimensionality reduction. To implement these advanced models we used the Radial Basis Function Toolbox [17] which allows for advance customization of the network parameters and architecture. All the acceptable models we identified used one hidden layer of neurons.

The network model included 433 hidden Gaussian Radial Basis function units with centers selected by the toolbox with a forward selection approach [11]. We also used a regularization parameter λ equal to 10^{-10} in order to penalize large weights during the training process. The error estimation for training was preformed using the generalized cross validation criterion [11] with a threshold of 1/1000 and wait equal to 5 in order to achieve convergence that generalizes well (avoid over-fitting). We performed an exhaustive search through a range of acceptable exponents for the Gaussian functions as well as a range of acceptable radii for the radial units. Trial experiments showed that the best scaling parameter for the specific dataset was equal to 20. Our model had inputs equal to the total number of amino acids used for representing the peptides and alleles, which due to the dimensionality reduction introduced by the allele motif discovered by TEIRESIAS, was equal to 74. One output linear neuron was used to indicate the output label for the test data. A threshold equal to zero was selected as a cutoff value for discriminating between class labels.

V. Results

In this section we discuss the experimental results obtained by incorporating domain knowledge into the models in the form of coding the amino acids before they are given as inputs to the NN models and where motif discovery and dimensionality reduction was also involved. Considering the model performance improvement we searched the full space of parameters (number of neurons, percentage of data set used for training, spread of Radial Basis Functions, etc.) to select appropriate values for these parameters. We employed known cross-validation techniques to prevent over-fitting of the models. In order to provide a basis for comparison, we initially present experiments using the RBFNN that does not utilize prior knowledge.

We considered both classes (known vs. unknown) having equal prior probabilities and we employed an RBFNN with one hidden layer and with a number of neurons that we varied. We performed an exhaustive search of the values of parameters, aiming into discovering the ones that optimize the performance of the RBFNN models. The initial approach that we followed is to add nodes successively until the prediction error is minimized. We also experimented with using a wide range of different subsets of the total available dataset for training the models (from 50% up to 75%), in order to understand the relationship between model performance improvements and the training/testing data set information content increment. In the following paragraphs we present the best experimental results reporting the specific experimental settings, such as training set size and number of neurons, for which these results were obtained.

A. Experiments using the basic RBFNN model without incorporating prior knowledge

To provide a basis for comparison for the evaluation of our approach we first performed cross-validation experiments with the Unknown-StaticRandom and the Unknown-Dynamic Random datasets without incorporating any prior knowledge. For the Unknown-DynamicRandom dataset we used an ensemble of 10 basic RBFNNs classifiers in order to compensate for the variability introduced by having a different random dataset of true unknown binding pairs in the training-testing trial. The amino acids were not coded into any binary form in order to be used as inputs to the classifier model. The PRTools 3.1.7 toolbox for Matlab [14] that we used to implement these initial models has the capability of dealing with letters as inputs.

For cross-validation the set was first randomly permutated, a percentage of samples was left out, the classifier was trained and these samples were used for estimating the instability and the error. The spread and the centers of the Radial Basis Units were selected automatically by the training algorithm. The training of the RBFNNs stopped when the error threshold reached 0.02 for the sum-squared error on the training set. The reported accuracy is the average over the accuracies of all 10 classifiers. Table II illustrates these results.

TABLE II
CLASSIFICATION ACCURACY OBTAINED USING AN ENSEMBLE OF RBFNN MODELS THAT DO NOT UTILIZE ANY PRIOR KNOWLEDGE

Classification Accuracy				
	Unknown-StaticRandom		Unknown-DynamicRandom	
Percent of Data used as training set	60%	65%	60%	65%
Number of Neurons				
5	0.67	0.70	0.68	0.63
10	0.76	0.73	0.70	0.66
15	0.77	0.73	0.71	0.67
20	0.77	0.74	0.71	0.67
25	0.77	0.74	0.72	0.67
30	0.77	0.74	0.72	0.68
35	0.78	0.74	0.72	0.67
40	0.77	0.75	0.72	0.68
45	0.76	0.74	0.72	0.68
50	0.78	0.74	0.72	0.68
55	0.78	0.76	0.72	0.68
60	0.77	0.75	0.72	0.68
65	0.78	0.74	0.72	0.69
70	0.78	0.75	0.72	0.69

B. Experiments with the basic RBFNN model using the encoding that incorporates chemical properties

In this set of experiments we used again the basic RBFNN model implemented with the PRTools v. 3.1.7 Pattern Recognition toolbox for Matlab [15]. Both the binary and the decimal encoding, based on the chemical properties of the amino acids (i.e. amino acid hierarchy), were used in these experiments. We used the Unknown-StaticRandom dataset for training and testing. We performed cross-validation experiments varying the percentage of samples used for training. The spread and the centers of the Radial Basis Units were selected automatically by the training algorithm, applying a sum-squared error threshold equal to 0.02 to converge training. We report the best classification accuracies obtained for each case and the specific experimental settings (i.e. number of neurons and training set size). Table III illustrates the classification accuracy obtained when using the binary encoding based on chemical properties to represent the amino acids as inputs to the classifiers. The performance of this RBFNN model was not very robust. The standard deviation in this case ranged between 0.01-0.5. This is due to the large dimensionality of the input space introduced by the binary encoding. This imposes a difficulty in generalizing the RBFNN well during the training process. Table IV illustrates the classification accuracy obtained when using the decimal encoding based on chemical properties to represent the amino acids as inputs to the classifiers. The classification accuracy improved in this case. Moreover, the models obtained in this case were much more robust. The standard deviation in this case ranged between 0.01-0.03.

TABLE III
CLASSIFICATION ACCURACY OBTAINED USING RBFNN MODELS THAT UTILIZE PRIOR KNOWLEDGE BASED ON AMINOACID CHEMICAL PROPERTIES (BINARY ENCODING)

Classification Accuracy						
Percent of Data used as training set	75%	70%	65%	60%	55%	50%
Number of Neurons						
5	0.61	0.55	0.64	0.59	0.54	0.48
10	0.61	0.54	0.64	0.59	0.51	0.47
15	0.61	0.54	0.65	0.60	0.54	0.49
20	0.59	0.53	0.64	0.59	0.54	0.48
25	0.61	0.54	0.61	0.60	0.54	0.48
30	0.62	0.54	0.64	0.60	0.55	0.50
35	0.61	0.52	0.63	0.59	0.53	0.49
40	0.60	0.53	0.64	0.59	0.54	0.50
45	0.61	0.54	0.64	0.59	0.54	0.48
50	0.62	0.53	0.64	0.58	0.54	0.50
55	0.61	0.52	0.65	0.59	0.53	0.49
60	0.61	0.54	0.65	0.59	0.54	0.48
65	0.60	0.53	0.64	0.59	0.55	0.49
70	0.60	0.53	0.64	0.60	0.54	0.50
75	0.59	0.53	0.64	0.59	0.54	0.47
80	0.61	0.52	0.63	0.60	0.54	0.50
85	0.60	0.54	0.63	0.60	0.55	0.47
90	0.61	0.53	0.63	0.60	0.53	0.48
95	0.62	0.55	0.65	0.59	0.55	0.50

TABLE IV
CLASSIFICATION ACCURACY OBTAINED USING RBFNN MODELS THAT UTILIZE PRIOR KNOWLEDGE BASED ON AMINOACID CHEMICAL PROPERTIES (DECIMAL ENCODING)

Classification Accuracy						
Percent of Data used as training set	75%	70%	65%	60%	55%	50%
Number of Neurons						
5	0.60	0.67	0.70	0.70	0.67	0.55
10	0.60	0.69	0.76	0.73	0.68	0.65
15	0.67	0.70	0.77	0.73	0.71	0.66
20	0.67	0.70	0.77	0.74	0.70	0.67
25	0.67	0.71	0.77	0.74	0.71	0.68
30	0.68	0.71	0.77	0.74	0.71	0.67
35	0.68	0.72	0.78	0.74	0.71	0.68
40	0.68	0.72	0.77	0.75	0.71	0.68
45	0.68	0.71	0.76	0.74	0.71	0.68
50	0.68	0.72	0.78	0.74	0.72	0.68
55	0.68	0.71	0.78	0.76	0.70	0.68
60	0.68	0.71	0.77	0.75	0.71	0.69
65	0.68	0.72	0.78	0.74	0.71	0.68
70	0.69	0.72	0.78	0.75	0.63	0.69
75	0.69	0.72	0.79	0.75	0.72	0.69
80	0.69	0.73	0.78	0.75	0.72	0.69
85	0.69	0.72	0.78	0.75	0.72	0.69
90	0.69	0.72	0.79	0.75	0.71	0.59
95	0.69	0.73	0.78	0.75	0.63	0.69

C. Experiments with the advanced RBFNN models that incorporate prior knowledge using dimensionality reduction and motif discovery.

To implement these advanced RBFNN models we used the Radial Basis Function Toolbox [11-12, 17] which allows for advance customization of the network parameters and architecture. The decimal encoding of amino acids, based on their chemical properties, was used. For these experiments we used the Unknown-SimilartoKnown Dataset. The data were split into a 65% training and 35% testing sets, resulting in 1714 (857 binding + 857 unknown) pairs and 922 (461 binding + 461 unknown) pairs respectively. Class labels were assigned as 1 for the known binding pairs and -1 for the unknown pairs. Our network included 433 hidden Gaussian Radial Basis function units with centers selected by the toolbox with a forward selection approach. We also used a regularization parameter lambda equal to 10^{-10} in order to penalize large weights during the training process. The error estimation for training was preformed using the generalized cross validation criterion (gcv) with a threshold of 1/1000 and wait equal to 5 in order to achieve convergence that generalizes well (avoid over-fitting). We experimented for a range of acceptable exponents for the Gaussian functions as well as a range of acceptable radii for the radial units. Trial experiments showed that the best scaling parameter for the specific dataset was equal to 20.

We applied motif identification techniques to reduce the dimensionality of the data (i.e., the number of allele positions considered). For motif identification we used TEIRESIAS as explained in the Section IV.C. More specifically, in the case of the alleles, we applied TEIRESIAS using the chemical equivalence classes of amino acids provided by the software, removing patterns that overlap. TEIRESIAS requires the setting of several parameters for the detection of motifs, such as the minimum literals in a motif, the maximum window spanned by a motif, and the minimum support of a motif. The particular values selected for these parameters were as follows: L=3 (minimum literals in a motif), W=17 (maximum window spanned by a motif) and K=60 (minimum support). We detected motifs present either in all the selected allele sequences (considering only the sequence constructed by the selected and polymorphic positions) or within specific families of alleles (A, B or C). These motifs had a high statistical significance and support (e.g., Log-Probability < -30, Occurrences = Sequences = 82).

In the experiments reported here, we selected a motif present in all the selected allele sequences. By removing this motif from all the peptide-allele sequences the dimensioality of the dataset was reduced to 74, since the motif sequence was 18 aminoacids long. We performed an extensive search for identifying the RBFNN parameters that optimize performance. The classification accuracies obtained for there various experimental settings are illustrated in Tables V and VI.

TABLE V
CLASSIFICATION ACCURACY USING RBFNN MODELS THAT INCORPORATE PRIOR KNOWLEDGE BASED ON AMINOACID CHEMICAL PROPERTIES (DECIMAL ENCODING) AND DIMENSIONALITY REDUCTION BASED ON MOTIF DISCOVERY

Classification Accuracy											
Radial Basis Function Radius	**Radial Basis Function Exponent**										
	1.2	**1.4**	**1.6**	**1.8**	**2.0**	**2.2**	**2.4**	**2.6**	**2.8**	**3.0**	**3.2**
1.2	86.44	87.31	87.42	87.96	87.96	88.07	88.18	87.96	88.39	88.50	88.50
1.4	86.88	86.66	87.09	87.53	87.64	87.64	87.85	88.18	88.50	88.18	88.07
1.6	80.15	86.66	86.88	87.20	87.53	87.74	87.64	87.96	87.42	87.53	87.42
1.8	78.63	86.01	86.77	86.77	87.09	87.31	87.42	87.53	87.74	87.42	87.09
2.0	76.25	82.43	86.23	86.55	87.09	86.66	86.98	86.66	86.23	86.66	85.47
2.2	76.90	78.20	85.57	86.01	86.23	87.09	87.53	87.09	87.31	86.88	86.23
2.4	75.92	77.77	85.36	86.23	86.55	85.25	86.01	86.88	86.55	85.90	85.90
2.6	75.38	77.87	85.36	85.03	85.03	85.68	85.03	85.90	85.47	86.23	83.51
2.8	75.27	77.01	81.45	84.16	84.71	85.90	85.68	83.95	84.92	85.57	85.36
3.0	74.73	76.25	78.20	84.38	84.06	85.47	84.60	85.68	85.25	83.84	82.86
3.2	74.19	76.03	78.09	83.95	84.71	85.14	85.03	83.51	85.25	84.27	84.82

TABLE VI
CLASSIFICATION ACCURACY USING RBFNN MODELS THAT INCORPORATE PRIOR KNOWLEDGE BASED ON AMINOACID CHEMICAL PROPERTIES (DECIMAL ENCODING) AND DIMENSIONALITY REDUCTION BASED ON MOTIF DISCOVERY WITH FURTHER TUNING OF THE RADIAL BASIS EXPONENT AND RADIUS

Classification Accuracy											
Radial Basis Function Exponent	**Radial Basis Function Radius**										
	2.8	**2.9**	**3.0**	**3.1**	**3.2**	**3.3**	**3.4**	**3.5**	**3.6**	**3.7**	**3.8**
1.1	88.07	88.07	88.18	88.29	88.18	88.29	88.39	88.29	89.39	89.39	89.29
1.15	87.96	88.29	90.29	88.39	90.07	88.39	88.29	88.39	88.07	90.39	90.61
1.2	88.39	88.39	88.50	88.29	88.50	88.50	88.72	88.50	88.61	88.61	88.50

VI. Discussion

The results obtained using the selected Radial Basis Function Neural Network models were very promising. Incorporating prior knowledge on amino acid chemical properties into the models definitely optimized performance. We clearly demonstrated that the proposed encoding of the amino acids represents more meaningfully the distance between the amino acids (according to various metrics) and can increase the prediction accuracy. However, decimal encoding was more successful in increasing the performance than binary encoding. This was expected since the number of inputs of the NN model was reduced when using the decimal encoding and this had an effect on its ability to be trained and generalize more easily.

The best performance obtained was for training sets consisting of 40% to 70% of the full data set although the variation in performance for all experiments performed was small. Our attempt to generate "unknown binding" peptide-allele pairs following the distribution of known binding data was quite successful. The dataset "Unknown-SimilartoKnown" demonstrated very reliable accuracy in the neighborhood of 90%. Our more robust model is the one that uses the specific range of parameters illustrated in Table VII. This RBFNN model utilizes domain knowledge, motifs and uses the Unknown-SimilartoKnown dataset in which the "unknown" class has similar distribution to the "known" class. Obtaining actual non-bind data still seems to be very critical in generating even more robust predictive models. Motif discovery which we used as dimensionality reduction improved performance significantly. There is still significant work that can be performed in this area.

In most of the experiments, increasing the quantity of the data available for training increased the model performance. The performance also depended on the complexity (i.e, number of neurons) of the models used. In general, increasing the number of neurons improved the performance. More experimental work is needed to determine a more quantifiable relationship.

Further understanding the relationship between model performance improvements and the training/testing data set information content increment may assist guiding the process for collection of new data so that the data collected are as informative as possible, i.e., they are in areas of the search space that are critical for the predictive model.

VII. Conclusions

We developed Radial Basis Function Neural Network (RBFNN) models based on Radial Basis Function Units and systematically studied them for the problem of allele-peptide binding prediction. We were able to achieve a prediction accuracy of 90%. In our experiments we varied a number of parameters such as the data set size, level of missing data, unknown-bind data generation algorithm, level of domain knowledge, model architecture, and training algorithms. A main contribution is in the methodology used for incorporating existing domain knowledge into these prediction models. In particular, the encoding of inputs of the RBFNN considering the chemical properties of amino acids, the discovery of motifs in alleles, the clustering of alleles and peptides and the dimensionality reduction based on common motifs discovered improved the prediction accuracy of our models. Incorporation of additional domain knowledge is expected to improve further the prediction accuracy. In addition, we expect that performance will improve by increasing the size and quality of the data set, and by acquiring even a small set of real non-bind ("negative") data.

Acknowledgement

The authors would like to thank Dr. A.G. Hatzigeorgiou for providing helpful suggestions. This work was supported in part by the National Science Foundation under grant IIS-0237921 and the National Institutes of Health under grant 5R01GM063345-02. All agencies specifically disclaim responsibility for any analyses, interpretations, or conclusions.

References

[1] V. Brusic, N. Petrovsky, G. Zhang, V. Bajic, "Prediction of promiscuous peptides that bind HLA class I molecules", *Immunology and Cell Biology*, 80: 280-285, 2002.

[2] S. Buus, "Description and prediction of peptide-MHC binding: the 'human MHC project'". *Current Opinion in Immunology*, 11: 209-213, 1999.

[3] V.Brusic, J.Zeleznikow, "Artificial Neural Networks in Immunology", in Proceedings of the *1999 International Joint Conference on Neural Networks IJCNN'99*.

[4] V. Brusic, G. Rudy, G. Honeyman, J. Hammer, L. Harrison, "Prediction of MHC class II-binding peptides using an evolutionary algorithm and artificial neural network", *Bioinformatics*, 14, 121-130, 1998.

[5] V. Brusic, G. Rudy, L.C. Harrison, "Prediction of MHC binding peptides using artificial neural networks" In: R Stonier and XH Yu (eds), *Complex Systems: Mechanism of Adaptation*. IOS Press/Ohmsha, pp. 253-260, 1994.

[6] H.P. Adams, J.A. Koziol, "Prediction of binding to MHC class I molecules", *Journal of Immunological methods*,185, 181-190, 1995.

[7] K. Gulukota, J. Sidney, A. Sette, C. DeLisi, "Two complementary methods for predicting peptides binding major histocompatibility complex molecules" *Journal of Molecular Biology*, 26, 1258-1267, 1997.

[8] H. Mamitsuka, "Predicting peptides that bind to MHC molecules using supervised learning of hidden Markov models", *Proteins* 33, 460-474, 1998.

[9] P. Dönnes, A. Elofsson, " Prediction of MHC class I binding peptides, using SVMHC", *BMC Bioinformatics* 3:25, 2002.

[10] S. Haykin, *Neural Networks: A comprehensive foundation*. Upper Saddle River, NJ, USA: Prentice Hall, 1999.

[11] M.J.L. Orr. Introduction to radial basis function networks. Technical report, Institute for Adaptive and Neural Computation, Division of Informatics, Centre for Cognitive Science, University of Edinburgh, Scotland, April 1996. www.anc.ed.ac.uk/~mjo/papers/intro.ps.

[12] M.J.L. Orr, "Regularization in the selection of radial basis function centers", *Neural Computation*, 7(3) 606-623, 1995.

[13] M. Orr, J. Hallam, K. Takezawa, A. Murray, S. Ninomiya, M. Oide, T. Leonard, "Combining regression trees and radial basis function networks", Int. J. Neural Syst., 10(6) 453-65, 2000.

[14] R.P.W. Duin, A Matlab Toolbox for Pattern Recognition, Delft University of Technology, The Netherlands, 2002, www.ph.tn.tudelft.nl/prtools/.

[15] I. Rigoutsos, A. Floratos, "Combinatorial Pattern Discovery in Biological Sequences: the TEIRESIAS Algorithm", *Bioinformatics,* 14(1), 1998.

[16] I. Rigoutsos, A. Floratos. "Motif Discovery Without Alignment Or Enumeration", *Proceedings 2nd Annual ACM International Conference on Computational Molecular Biology (RECOMB '98),* New York, NY. March 1998.

[17] Matlab Functions for RBF Networks, www.anc.ed.ac.uk/~mjo/rbf.html.