Ole Lund
Morten Nielsen
Claus Lundegaard
Can Keşmir
Søren Brunak

Immunological Bioinformatics

A Bradford Book
The MIT Press
Cambridge, Massachusetts
London, England

Chapter 6

Prediction of Cytotoxic T Cell (MHC Class I) epitopes

Cytotoxic T lymphocytes (CTLs) recognize foreign peptides presented on other cells in the body and help to destroy infected or malignant cells. The peptides are presented by the class I major histocompatibility complex (MHC), and the actual binding of the peptide to the MHC is the single most selective event in the antigen presentation process. The process also includes processing (cleavage) of proteins and translocation of peptides from the cytosol into the endoplasmic reticulum. These latter steps, however, only filter out approximately 4/5 of all potential 9-mer peptides whereas a particular MHC class I allele only binds 1/200 potential peptides [Yewdell and Bennink, 1999].

The class I MHCs (also called class I Human Leukocyte Antigens or HLAs) are encoded by 3 different loci on the genome called A, B and C. Each of the genes are highly polymorphic and for each loci hundreds of different alleles exists. The HLAs are thus highly diverse, and each allele binds a very specific set of peptides. All the different alleles can be divided into at least 9 supertypes, where the alleles within each supertype exhibit roughly the same peptide specificity [Sette and Sidney, 1999, Lund et al., 2004]. The concept of HLA supertypes has great implications for the use of bioinformatical prediction algorithms in the search for novel vaccine candidates. The HLA allele space is very large, and reliable identification of potential epitope candidates would be an immense task if all alleles were to be included in the search. However, many HLA alleles share a large fraction of their peptide binding repertoire, and it is often possible to find promiscuous peptides, which bind to a series of HLA alleles. This allows the search to be limited to a manageable set of alleles. A detailed description of the different HLA supertypes is given in Chapter 13.

6.1 Background and historical overview of methods for peptide MHC binding prediction

A number of methods for predicting the binding of peptides to MHC molecules have been developed (reviewed by Schirle et al. [2001]) since the first motif methods were presented [Rothbard and Taylor, 1988, Sette et al., 1989b].

The majority of peptides binding to the HLA complex have a length of 8-10 amino acids. For nonamer peptides, positions 2 and 9 are very important for the binding to most class I HLAs, and these positions are referred to as anchor positions [Rammensee et al., 1999]. For some alleles the binding motif further have auxiliary anchor positions. Peptides binding to the A*0101 allele thus have position 2, 3, and 9 as anchors [Kubo et al., 1994, Kondo et al., 1997, Rammensee et al., 1999].

The importance of the anchor positions for peptide binding, and the allele specific amino acid preference at the anchor positions was first described by Falk et al. [1991]. The discovery of such allele specific motifs lead to the development of the first reasonable accurate algorithms [Pamer et al., 1991, Rotzschke et al., 1991]. In these prediction tools, it is assumed that the amino acids at each position along the peptide sequence contribute with a given binding energy, which can independently be added up to yield the overall binding energy of the peptide [Parker et al., 1994, Meister et al., 1995, Stryhn et al., 1996]. Similar types of approaches are used by the EpiMatrix method [Schafer et al., 1998], the BIMAS method [Parker et al., 1994] and the SYFPEITHI method [Rammensee et al., 1999]. An example of an peptide binding to a HLA molecule can be seen in Figure 6.1.

These methods can not take into account correlated effects where the binding affinity of a given amino acid at one position is influenced by amino acids at other positions in the peptide. Two adjacent amino acids may for example compete for the space in a pocket in the MHC molecule. Artificial neural networks (ANN) are ideally suited to take such correlations into account.

Several prediction methods have been made publicly available including weight matrix methods such as BIMAS [Parker et al., 1994] and SYFPEITHI [Rammensee et al., 1999], weight matrices with optimized position specific weighting [Yu et al., 2002] and ANNs [Brusic et al., 1994, Adams and Koziol, 1995]. Recently we have developed a comprehensive HLA/peptide binding prediction server including allele specific weight matrix predictions for more than 120 HLA alleles, as well as artificial neural networks and weight matrix predictions for 12 alleles representing 12 distinct HLA supertypes. This NetMHC (NetMHC2.0) server is available at www.cbs.dtu.dk/services/NetMHC. A more comprehensive list of servers can be found in Chapter 12.

Detailed predictions of peptide binding have been made by dividing bind-

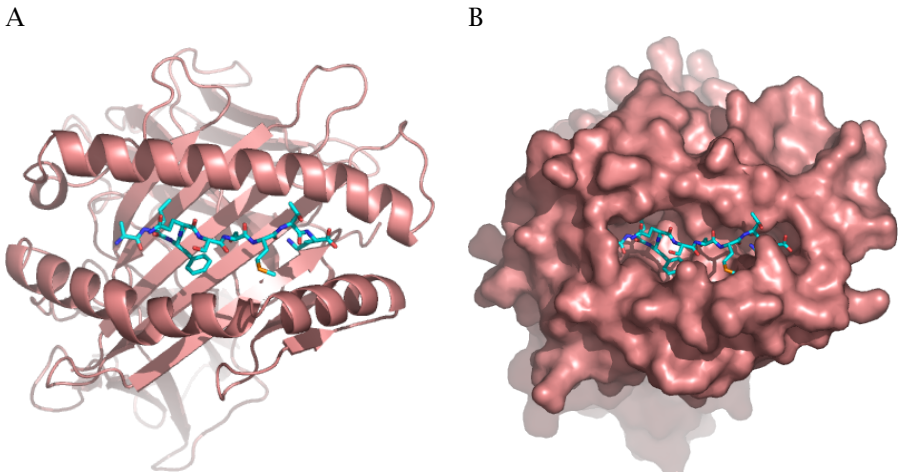

Figure 6.1: Example of structure of peptide binding to MHC I. A) Cartoon representation of MHC I showing that the peptide is binding on a "floor" made by a β-sheet, and restricted on each side by two α-helices. The bound peptide is shown as a sticks-model. B) MHC molecule shown as the molecular surface. It can be seen that the binding is more close than it appears from the cartoon model. The figure is based on the PDB (www.rcsb.org/pdb) entry 1q94. The figure was kindly provided by Anne Mølgaard.

ing affinities into classes of affinity ranges, it has been found that the different classes are associated with different binding sequence motifs [Adams and Koziol, 1995]. Neural networks have also been trained to predict MHC binding using different affinity thresholds [Gulukota et al., 1997]. Mamitsuka trained the transition and emission probabilities of a fully connected hidden Markov model using a steepest descent algorithm so as to minimize the differences between the predicted and target probabilities for each peptide [Mamitsuka, 1998].

Other prediction algorithms have been developed to predict not only if a peptide binds, but also the actual affinity of the binding [Marshall et al., 1995, Stryhn et al., 1996, Rognan et al., 1999, Doytchinova and Flower, 2001, Buus et al., 2003, Nielsen et al., 2003]. For affinity predictions, ANNs in general outperform the simpler methods [Gulukota et al., 1997, Nielsen et al., 2003], however generally ANNs need a large number of examples in the training [Yu et al., 2002] to achieve accurate predictions.

Buus et al. [2003] have demonstrated that neural networks trained to perform quantitative predictions of peptide MHC binding are superior to conventional classification neural networks trained to predict binding vs. non-binding. Nielsen et al. [2003] have further demonstrated that neural network

methods perform significantly better than linear method in predicting high affinity peptides.

A central issue in the development of bioinformatical prediction algorithms is the number of training examples needed to achieve reliable predictions. A stated above ANNs in general need a large number of training data in order to achieve a predictive performance beyond that of the simpler methods. Hidden Markov models (or weight matrices) on the other hand, can be trained to a very accurate performance on small data sets by use of the techniques described in Chapter 4. Common for both artificial networks and hidden Markov models is that both methods rely on the availability of peptides known to bind a given MHC complex. For many alleles such data is not available or available only in very limit numbers, and for these alleles another approaches have to be taken. The number of MHC/peptide complexes solved by X-ray crystallography is growing. Based on such structural information, a MHC/peptide binding potential can be derived. Such an approach has been taken by [Altuvia et al., 1995, Schueler-Furman et al., 2000, Doytchinova and Flower, 2001] where peptide binding is predicted by either free energy calculations or threading. In situations where no peptide motif information exist, these energy-based algorithms are highly valuable.

In this chapter, we will demonstrate how bioinformatical methods can be applied to derive prediction methods for HLA/peptide binding. In the first part, we describe how accurate prediction methods can be derived in situations when very limited training data is available. The second part shows how highly reliable prediction methods can be constructed using combination of many neural networks trained with different sequence encoding schemes.

6.2 MHC class I epitope binding prediction trained on small data sets

The highly diverse MHC class I alleles bind very different peptides, and accurate binding prediction methods exist only for alleles were the binding pattern has been deduced from peptide motifs. Predictions in general tend to be more precise when more examples are included in the training [Yu et al., 2002], but experimental data on peptides binding to HLA complexes are published in large numbers for only a few alleles.

It has earlier been shown that a position specific weighted matrix where the weight on selected positions in the matrix describing binding motif is increased, performs slightly better for A*0201 predictions than an unweighted matrix [Yu et al., 2002]. A similar result was found in the example for weight matrix construction in Chapter 4, where a weight matrix was constructed from 10 HLA-A*0201 restricted peptide using the technique of sequence weighting,

pseudo count correction for low counts, and position specific weighting. This matrix was shown to share many of the features of a weight matrix trained on close to 500 HLA-A*0201 restricted peptides. It is, however, not clear from these two examples to what extent such a weighting will influence the number of data needed to generate accurate predictors.

In the following section, we will describe a method for predicting which peptides bindto given MHC class I alleles based on scoring matrices with empirical position specific anchor weighting.

6.2.1 Weight matrix training

The selected peptides can be stacked into a multiple alignment and using an ungapped HMM like approach the log-odds weight matrix was calculated as $log(p_{pa}/q_a)$, where p_{pa} is the frequency of amino acid a at position p in the alignment and q_a the background frequency of that amino acid in the Swiss-prot database [Henikoff and Henikoff, 1994]. The values for p_{pa} were estimated using the techniques of sequence weighting and pseudo count correction for low counts described in Chapter 4 [Altschul et al., 1997, Henikoff and Henikoff, 1992]. A schematic view of the procedure is outlined in Figure 6.2. To analyze how the predictive performance of a weight matrix depends on the number of training data, we varied the numbers of peptides included to calculate the weight matrix. For each number of training peptides, 200 data sets were constructed, using the Bootstrap procedure [Press et al., 1992], by randomly drawing the chosen number of peptides with replacement from the original data set of peptides.

To visualize the problem one is facing when training a prediction method on limited amounts of data, we generated sequence logos for peptides binding to the A*0201 allele using 10 and 100 peptides, respectively. From the logo constructed using 10 random A*0201 binding peptides (Figure 6.3a), it can be seen that the importance of the anchor positions 2 and 9 is not yet visible, while this feature is clearly apparent in the logo based on 100 sequences (Figure 6.3b). The amino acid preferences for the hydrophobic amino acids L and L/V at position 2 and 9 respectively is, however, present in both logos. Based on the information content visualized with the logos in Figure 6.3, a prediction method trained on very few data would very likely benefit by incorporating the prior knowledge about the differential importance of the different positions in the motif. This is naturally done by increasing the relative weight on the anchor positions. The logo of a matrix with position specific differential weighting at position 2 and 9 is shown in Figure 6.3c.

Figure 6.4 shows that weight matrix predictions can benefit from such position specific weighting. A set of weight matrices were generated for the A2

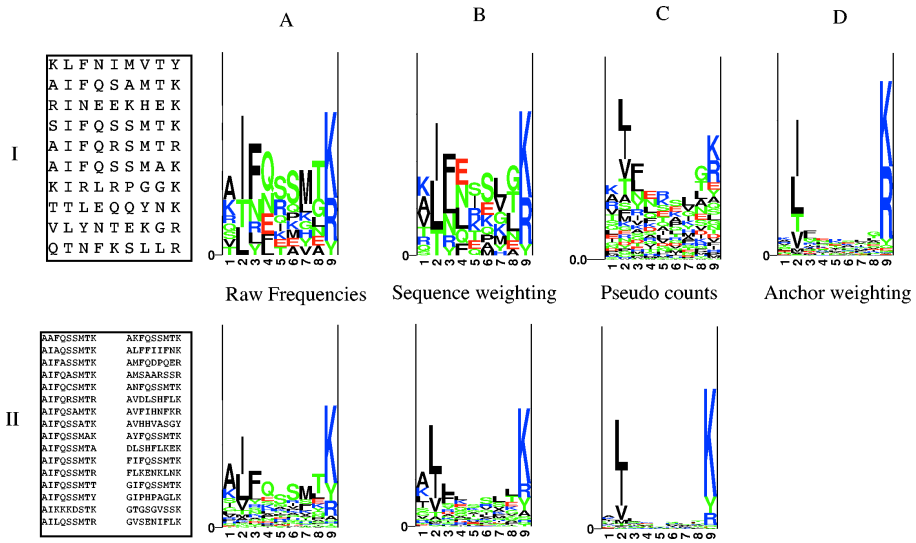

Figure 6.2: Logos showing the distribution of information content after each step in the matrix calculation using few (I) or many (II) A*0301 training peptides. The sequences used for training are shown in the box to the left in each row. The number of peptides in the two examples is 10 and 32, respectively A) The distribution of amino acids at each position. B) After sequence weighting. C) After low count correction D) Extra weight on anchor positions when few peptides are used for training. The logos were calculated as described by Hebsgaard et al. [Hebsgaard et al., 1996], and visualized using the Logo program [Schneider and Stephens, 1990]. Figure adapted from [Lundegaard et al., 2004].

allele A*0201 for a different number of training data. In the work of [Yu et al., 2002] all positions in the weight matrix were scaled differently. Here only the weights on the positions 2 and 9 and any addition position assigned as anchor in the SYFPEITHI database [Rammensee et al., 1999], were scaled (biased) by a factor of 5. The different matrices were evaluated on 217 peptides with experimentally determined affinities to the A*0201 allele (K. Lamberth, unpublished). From the figure, it is clear that when using un-biased weight matrices at least 20 training peptides is needed to get a reasonably performance (A_{ROC}>0,8, Pearson cc > 0.5), and at least 100 training examples to get values comparable to those obtained by public available prediction servers. When applying position specific weighting on the matrices the performance, on the other hand, is surprisingly high, even for matrices trained just a handful of peptides. For a number of training peptides of 20 both the public methods and the position specific weighted matrix reach similar predictive performance.

Figure 6.5 shows that the position specific weighting approach is also appli-

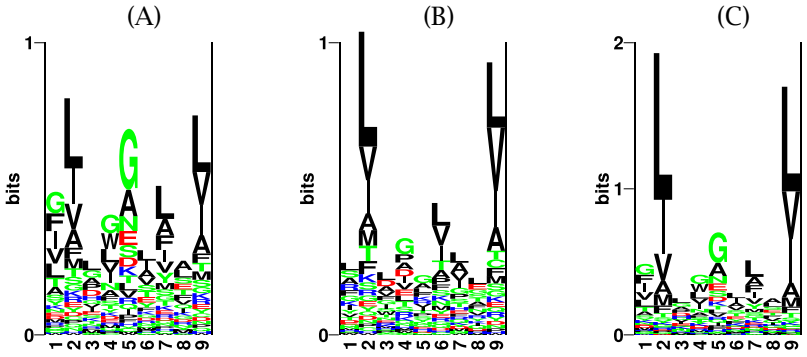

Figure 6.3: Sequence logos generated by 10 (a+c) and 100 (b) randomly chosen A*0201 binding peptides. The logos are constructed using the techniques of sequence weighting and pseudo count correction for low counts. In (c) the method of position specific differential weighting of the positions 2 and 9 is applied with a weight of 3. Figure adapted from [Lundegaard et al., 2004].

cable to other HLA alleles The position weighting strategy was applied to train matrices with peptides belonging to the A*0101, A*0301, A*1101, and B*0702 alleles. Note that position 3 is an additional anchor position in the A*0101 allele and that this position thus was also biased for this allele. For each of the 4 alleles, a series of weight matrices were trained by varying the number of training examples. On Figure 6.5 it can be seen how the weight matrix predictive performance varies as a function of the number of training examples for each of the 4 alleles. For each allele is show the performance of an unweighted matrix, a weight matrix with position specific weighting of the anchor positions of the bind motif as assigned in the SYFPEITHI database, as well as the two public method of BIMAS [Parker et al., 1994] and SYFPEITHI [Rammensee et al., 1999]. SYFPEITHI predictions were performed using the web server (http://syfpeithi.bmi-heidelberg.com), and BIMAS predictions were performed as described at the web server, using matrices downloaded from the web site (http://bimas.cit.nih.gov/cgi-bin/molbio/hla_coefficient_viewing_page). In all cases reliable predictions were obtainable with matrices trained on as few as 5 training examples. For all alleles the the performance of the position specific weighted matrix is comparable to that of the public methods when 20 training examples are available.

Table 6.1 shows the prediction accuracy for predictors for different alleles on SARS derived peptides [Sylvester-Hvid et al., 2004] and on peptides from evaluation sets obtained from the MHCBN 3.1 database [Bhasin et al., 2003]. In the evaluation the predictive performance in terms of the A_{ROC} value was calculated for each of the evaluation sets using position specific weight matri-

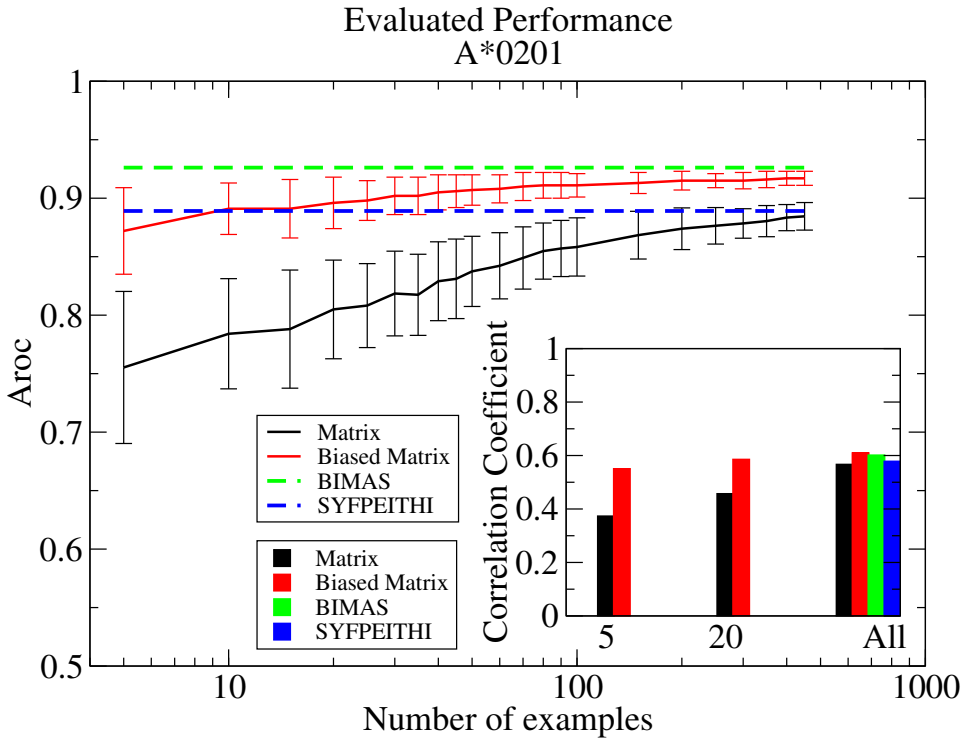

Figure 6.4: Curves of the A_{ROC} value (major graph) and the Pearson correlation coefficient (inserted graph) plotted against the number of training examples randomly selected from the total pool of peptides. Each value is the simple average of 200 independent calculations with the indication of one standard deviation. The matrices were generated and evaluated with peptides binding to the alleles A*0201. The score for a given peptide is calculated as the sum of the scores at each position. Training examples was extracted from the databases SYFPEITHI [Rammensee et al., 1999] and MHCPEP [Brusic et al., 1998a]. As evaluation sets, we used peptides for which the affinities for the selected alleles had been measured using the ELISA method described by Sylvester-Hvid et al. [2002] (K. Lamberth, unpublished) using a threshold for binders of 500 nM. Predictions were made for the corresponding evaluation set by each of the 200 matrices of each train set size, and the predictive performance was measured in terms of both the linear (Pearson) correlation coefficient between the prediction output and log-transformed measured affinities [Buus et al., 2003] and the area under a Receiver Operating Characteristic (ROC) curve, the A_{ROC} value [Swets, 1988]. The final predictive performance is given as the simple average of the 200 values. Figure adapted from [Lundegaard et al., 2004].

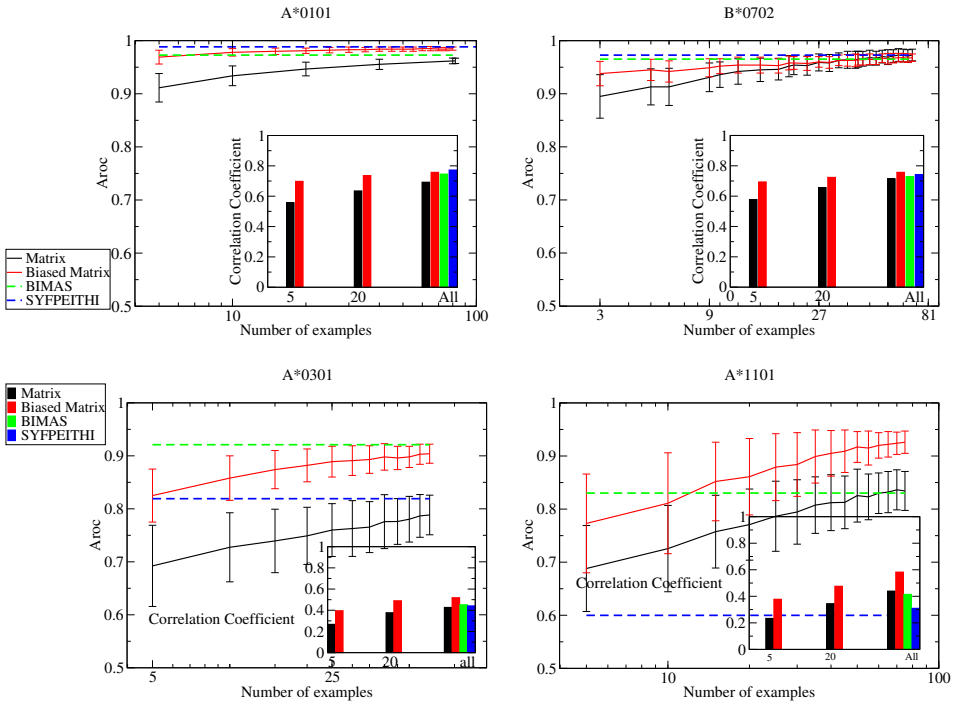

Figure 6.5: Curves of the A_{ROC} value (major graph) and the Pearson correlation coefficient (inserted graph) plotted against the number of training examples randomly selected from the total pool of peptides. Each value is the simple average of 200 independent calculations with the indication of one standard deviation. The matrices were generated and evaluated with peptides binding to the alleles A*0101, A*0301, A*1101 and B*0702. Note that the SYFPEITHI A*1101 predictions were generated using the A03 predictor. Figure adapted from [Lundegaard et al., 2004].

ces trained on different number of data, an un-weighted matrix trained on all available data, and the two public methods BIMAS and SYFPEITHI.

The analysis confirms that a weight matrix with position specific weighting of the anchor position trained on 20 peptide examples achieves a predictive performance comparable to that of BIMAS and SYFPEITHI. In many cases the performance for a biased matrix trained on only 5 peptide examples is comparable to that of the two public methods.

In summary we have shown that the empirical knowledge of important anchor positions within the binding motif dramatically reduces the number of peptides needed for reliable predictions. The method leads to predictions with a comparable or higher accuracy than other established prediction servers,

Allele	Matrix all peptides	Biased matrix 5 peptides	Biased matrix 20 peptides	Biased matrix all peptides	BIMAS	Syfpeithi
A*0101[1]	1.000	0.992 ± 0.026	1.000 ± 0.002	1.000	1.000	1.000
A*0201[1]	0.925	0.803 ± 0.024	0.830 ± 0.017	0.871	0.907	0.864
A*0101[2]	0.963	0.986 ± 0.011	0.992 ± 0.004	0.997	0.951	0.987
A*0201[2]	0.992	0.973 ± 0.015	0.978 ± 0.006	0.984	0.979	0.970
A*0301[2]	0.912	0.885 ± 0.072	0.873 ± 0.028	0.877	0.857	0.829
A*1101[2]	0.937	0.914 ± 0.038	0.948 ± 0.018	0.968	0.950	0.830[3]
B*0702[2]	0.983	0.972 ± 0.013	0.977 ± 0.009	0.985	0.990	0.990
B*1501[2,4]	0.928	0.932 ± 0.039	N.A.	0.955	0.893	N.A.
B*5801[2,4]	0.892	0.959 ± 0.008	N.A.	0.959	0.994	N.A.

Table 6.1: Evaluation of 200 matrices made by selecting 5 or 20 peptides respectively, by the Bootstrap method, or a single matrix generated by all available different peptides from MHCPEP and SYFPEITHI databases. The performance was measured in terms of the A_{ROC} value. Evaluation was performed with peptides extracted from the MHCBN 3.1[1] database, and SARS[2] relevant peptides. [3] Predictions were made using the A03 predictor. [4] The Bootstrapping procedure was not used due to the small total number of peptides available. Instead all possible combinations of the available peptides were used to estimate the standard deviation. Table adapted from [Lundegaard et al., 2004].

even in situations where only very limited data are available for training.

6.3 Prediction of CTL epitopes using neural network methods

Having described how accurate weight matrix based method can be derived when very limited training data is available, we now turn the focus to situations where the training set is large. In such situations neural networks would be the choice of method.

Neural network methods for predicting whether or not a peptide binds MHC molecules have earlier been developed [Brusic et al., 1994, Buus et al., 2003]. In this section we will describe how prediction of MHC I binding peptides may be improved using methods that combine several neural networks each derived using different sequence encoding schemes.

Brusic et al. use a conventional sparse (orthogonal) encoding of the 20 amino acid alphabet as well as 6 and 9 letter reduced alphabets [Brusic et al., 1994]. The conventional sparse encoding of the amino acids ignores their chemical similarities. We shall use a combination of several sequence encoding strategies in order to take these similarities in to account, explicitly. The different encoding schemes are defined in terms of Blosum matrices and hidden Markov models in addition to the conventional sparse encoding. The input to the neural network can consist of a combination of sparse encoding, Blosum encoding and input derived from hidden Markov models. We will show that this can lead to a performance superior to neural networks derived using a single sequence encoding-scheme, especially for the high affinity binding peptides.

We start the section by demonstrating that peptides binding to the HLA-A*0204 molecule display signal of higher order sequence correlations, next we train a series of neural network using different sequence encoding schemes, and demonstrate how the combination of many such diverse networks improves the prediction accuracy. In the last part of the section we apply the neural network algorithm to perform a genome-wide search for potential CTL epitopes in the genome of HCV.

6.3.1 Data

Two sets of data were used to derive the prediction method. One set was used to train and test the neural networks, and consists of 528 nine-mer amino acids peptides for which the binding affinity to the HLA class I molecule A*0204 have been measured by the method described by Buus et al. [Buus et al., 1995]. This data set is hereafter referred to as the Buus data set. The second data set was used to train the hidden Markov model. This data set was constructed from sequences downloaded from the Syfpeithi database [Rammensee et al., 1995]. All sequences from the database were downloaded and clustered into the nine super-types (A1, A2, A3, A24, B7, B27, B44, B58, and B62) and 3 outlier types (A29, B8 and B46) described by Sette and Sidney [Sette and Sidney, 1999]. The sequences in the A2 super-type cluster were aligned manually and trimmed into 211 unique nine amino acid long peptides. This data set is hereafter referred to as the Rammensee data set.

6.3.2 Mutual information

One important difference between linear prediction methods like 1^{st} order hidden Markov models and non-linear prediction methods like neural networks with hidden layers is their capability to integrate higher order sequence correlation into the prediction score. A measure of the degree of higher order sequence correlations in a set of aligned amino acid sequences can be obtained by calculating the mutual information matrix. For the case of peptide nine-mers, this is a 9x9 matrix where each matrix element is as decribed in Chapter 4 calculated using the formula

$$M_{ij} = \sum_{a} \sum_{b} P_{ij}(ab) \log \frac{P_{ij}(ab)}{P_i(a) P_j(b)} \tag{6.1}$$

In this example the summation is over the 20 letters in the conventional amino acid alphabet and i, j refer to positions in the peptide. $P_{ij}(ab)$ is the probability of mutually finding the amino acid a at position i and amino acid b at position j. $P_i(a)$ is the probabilities of finding the amino acid a at position

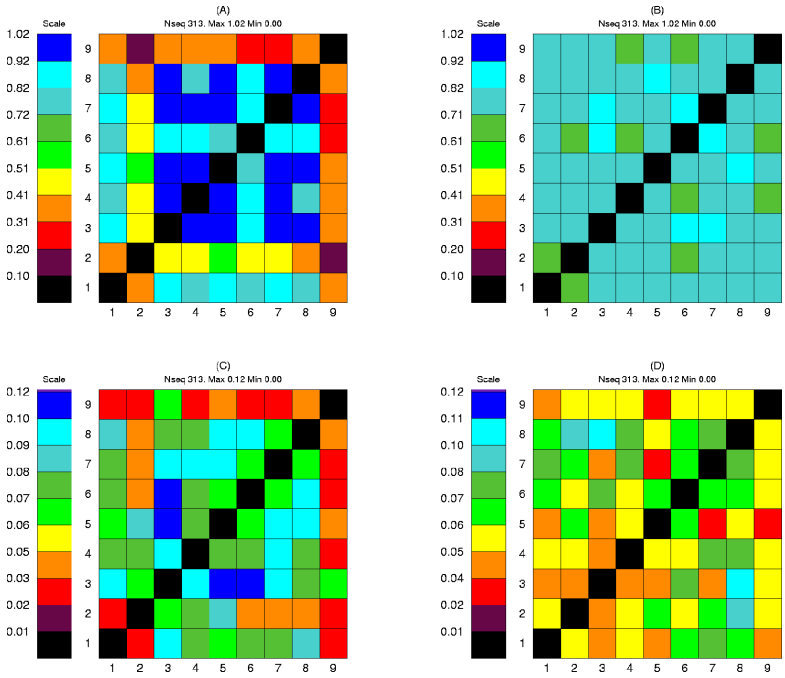

Figure 6.6: Mutual information matrices calculated for two different data sets. The left panel shows the mutual information matrix calculated for a data set consisting of 313 peptides derived from the Rammensee data set combined with peptides from the Buus data set with a binding affinity stronger than 500 nM. The right panel shows the mutual information matrix calculated for a set of 313 random peptides extracted from the Mycobaterium tuberculosis genome. In the upper row the mutual information plot is calculated using the conventional 20-letter amino acid alphabet. In the lower row the calculation is repeated using the 6-letter amino acid alphabet defined in the text. Figure adapted from [Nielsen et al., 2003].

i irrespectively of the content at the other positions, and likewise for $P_j(b)$. A positive value in the mutual information matrix indicates that prior knowledge of the amino acid content at position i will provide information about the amino acid content at position j. The statistical reliability of a mutual information calculation relies crucially on the size of the corresponding data set. In the mutual information calculation one seeks to estimate 400 amino acid pair frequencies at each position in the matrix. Such estimates are naturally associated with large uncertainties when dealing with small data sets. Figure 6.6 shows the mutual information matrix calculated for two different sets of nine-mer alignments.

The first data set was constructed to obtain the largest possible positive

set, by combining peptides from the Rammensee data set with the peptides from the Buus dataset that have a measured binding affinity stronger than 500 nM. This set contains 313 unique sequences. The second data set was constructed as a negative set by extracting 313 unique random peptides from the Mycobaterium tuberculosis genome. The mutual information content is calculated using the conventional 20 amino acid alphabet. The figure demonstrates a signal of mutual information between the seven non-anchor residues positions (1, 3, 4, 5, 6, 7 and 8) in the data set defined by peptides that bind to the HLA molecule. It is worth remarking that the mutual information content between any of the two anchor positions (2 and 9) and all other amino acids is substantially lower than the mutual information content between any two non-anchor positions. The significance of the mutual information content calculations can be improved by applying a suitable reduced sequence alphabet in the calculations Brusic et al. [1994]. In Figure 6.6(c) and (d) we show the mutual information matrices for the two data sets described above calculated using a reduced 6-letter alphabet derived from the side-chain surface area defined as A=GAS, B=CTDV, C=P, D=NLIQMEH, E=KFRY and F=W. Here the syntax A=GAS means that amino acids G, A and S all are encoded by the letter A. The matrices in Figure 6.6(c) and (d) display a similar behavior to the plots in Figure 6.6(a) and (b), however with the difference that the signal of mutual information in the data set derived from low and non-binding peptides has been substantially reduced compared to that of the data set defined by HLA-A2 binding peptides.

6.3.3 Combination of more than one neural network prediction

We combine the output from the two networks trained using sparse and Blosum sequence encoding, respectively, in a simple manner, as a weighted sum of the two. To select the weight that corresponds to the optimal performance, we plot the sensitivity/PPV as well as the ROC (relative operating characteristic) curves [Swets, 1988] for a series of weighted sum combinations of the two network outputs. The sensitivity is defined as the ratio TP/AP. Here TP (true positives) is the number of data points for which both the predicted score is above a given prediction threshold value and the measured binding affinity is above a given classification threshold value. AP (actual positives) is the total number of data points that have a measured binding affinity above the affinity threshold value. The PPV is defined as the ratio TP/PP. Here PP (predicted positives) is the total number of predictions with score above the prediction threshold value. The PPV is a measure of the reliability of the prediction method. The ROC curves are closely related to the sensitivity/PPV curves. However, with the important difference that one of the axis in the ROC curve is the false positive proportion FP/AN (actual negatives) and not the true positive to pre-

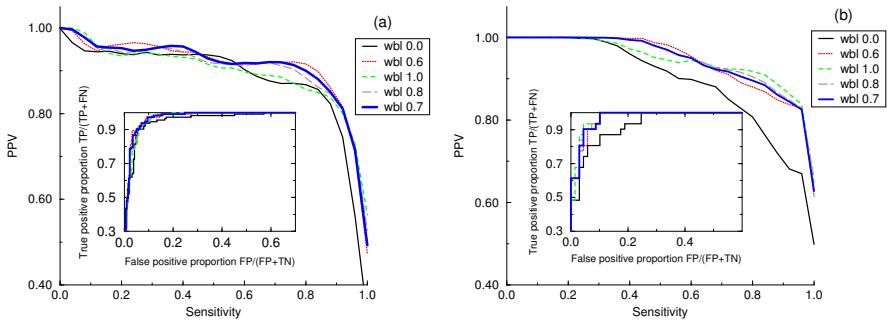

Figure 6.7: (a) Sensitivity/PPV plot calculated using a classification binding affinity of 500 nM for a series of linear combinations of the two neural network methods corresponding to Blosum50 and sparse sequence encoding, respectively. The curves were calculated by use of the Bootstrap method [Press et al., 1992] using 500 data set realizations. (a) 428 peptides in the test/train data set, (b) 100 peptides in the evaluation set. In the upper graph we determine the optimal performance to be the thick blue curve, corresponding to an combination of the two neural network methods with 70% weight on the Blosum50 encoded prediction and 30% weight on the sparse encoded prediction. This set of weights also results in close to optimal performance in lower graph. Insert to the graphs show the corresponding ROC curves. Figure adapted from [Nielsen et al., 2003].

dicted postive ratio (the PPV). The area under the ROC curve (A_{ROC}) provides an estimate of the accuracy of the prediction method. A random method will have a value of $A_{ROC} = 0.5$. $A_{ROC} > 0.8$ indicates that the method has moderate accuracy and $A_{ROC} = 1$ that the prediction method is perfect [Swets, 1988]. In a sensitivity/PPV plot, the curve for the perfect method is the one where the area under the curve is unity. The curves are estimated using the Bootstrap method [Press et al., 1992]. N data sets were constructed by randomly drawing M data points with replacement from the original data set of M peptides. For each of the N data sets a sensitivity/PPV curve and a ROC curve was calculated and the curves displayed in Figure 6.7 are derived from the mean of these N sensitivity/PPV and ROC curve realizations.

In Figure 6.7, the sensitivity/PPV curves for the 428 peptides in the train and test set and the 100 peptides in the evaluation set are shown for a measured binding affinity threshold value equal to 0.426, corresponding to a binding affinity of 500nM. In the insert to the figures the corresponding ROC curves are shown. From the figure, it is clear that both the sparse and the Blosum encoded neural networks have a performance that is inferior to any combination of the two. In Figure 6.7(a) the optimal combination is found to have a weight on the Blosum encoded network close to 0.7 and a weight on the sparse

encoded network close to 0.3, respectively. This set of weights for the combination of the two neural network predictions is also, in Figure 6.6(b), seen to improve to the prediction accuracy for the 100 peptides in the evaluation set. This is, however, less obvious, due to the small number of binding peptides in the evaluation set. The evaluation set contains 31 peptides with binding affinity stronger than 500 nM.

The Pearson correlation coefficient between the predicted and the measured binding affinities for the sparse encoded, the Blosum encoded and the combined neural network method on the peptides in the train/test set is found to be 0,849, 0.887 and 0.895, respectively. For the peptides in the evaluation set the corresponding values are found to be 0.866, 0.926, and 0.928 respectively.

The neural network training and testing is next repeated using the full data set in a five-fold cross-validation. The combined method, hereafter referred to as comb-I, is defined using the weights on the Blosum and the sparse encoded neural networks, respectively, estimated above.

6.3.4 Integration of data from the Rammensee database in the neural network training

In Figure 6.8(b), we show the performance of the hidden Markov model evaluated on the 528 peptides in the Buus data set. The figure displays a reasonable correlation between the hidden Markov model score and the measured binding affinity. This correlation demonstrates that the sequences in the Rammensee data set contain valuable information and that the neural network training could benefit from an integration of the Rammensee sequence data into the training data set. It is however not obvious how such an integration should be done. The Rammensee data are binary in nature. They describe that a given peptide does bind to the HLA molecule but not the strength of the binding. The data in the Buus data set on the other hand are continuous in that each peptide is associated with a binding affinity. It turns out that a fruitful procedure for integrating the Rammensee data into the neural network training is to use the output scores generated by the hidden Markov model as additional input to the neural network. The hidden Markov model is trained on the peptides in the Rammensee data set. The model is nine residues long, and the scores used as input to the neural network are the nine scores obtained when aligning a nine-mer peptide to the model. Two neural networks each with 189 input neurons (180 for sequence encoding and 9 to encode the scores from the hidden Markov model) are trained in a five-fold manner as described above using the hidden Markov model scores combined with the sparse or Blosum sequence encoding in the input layer, respectively. In the final combined method, the

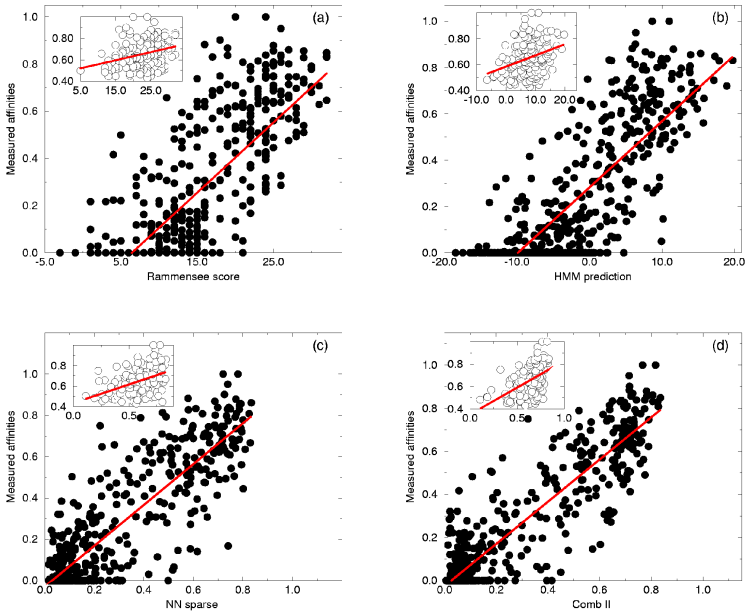

Figure 6.8: Scatter plot of the predicted score vs. the measured binding affinity for the 528 peptides in the Buus data set. The figure shows the performance for four different prediction methods. The insert to each figure shows an enlargement of the part of the plot that corresponds to a binding affinity stronger than 500 nM. (a) Rammensee matrix method, (b) Hidden Markov model trained on sequences in the Rammensee data set, (c) Neural network trained with sparse sequence encoding, and (d) The comb-II neural network method. The straight line fit to the data in (c) and (d) have slope and intercept of 0.989, -0.029 and 0.979, -0.027, respectively. Figure adapted from [Nielsen et al., 2003].

prediction value is calculated as the simple average with equal weight of the sparse and Blosum encoded neural network predictions, respectively.

This method is hereafter referred to as comb-II and is the one used in the HCV genome predictions described below.

6.3.5 Neural network methods compared to hidden Markov model methods and the matrix method by Rammensee

In Table 6.2, we give the test performance measured in terms of the Pearson correlation coefficient for the 528 peptides in the Buus data set for six different prediction methods: One method is the matrix method by Rammensee

Method	Pearson (all)	Pearson (500 nM)	Pearson (50 nM)
Rammensee	0.761 ± 0.016	0.296 ± 0.073	0.066 ± 0.116
HMM	0.804 ± 0.014	0.332 ± 0.061	0.142 ± 0.096
NN_{Sparse}	0.877 ± 0.011	0.438 ± 0.065	0.345 ± 0.090
NN_{Bl50}	0.899 ± 0.010	0.498 ± 0.064	0.382 ± 0.099
Comb-I	0.906 ± 0.009	0.508 ± 0.063	0.392 ± 0.092
Comb-II	0.912 ± 0.009	0.508 ± 0.054	0.420 ± 0.080

Table 6.2: The Pearson correlation coefficient between the predicted score and the measured binding affinity for the 528 peptides in the Buus data set. The six methods in the table are Rammensee: Score matrix method by H. G. Rammensee, HMM: Hidden Markov model trained on sequence data in the Rammensee data set, NN_{Sparse}: Neural network with sparse sequence encoding, NN_{BL50}: Neural network with Blosum50 sequence encoding, Comb-I: Combination of neural network trained using sparse and Blosum50 sequence encoding, respectively, and Comb-II: Combination of neural network trained using sparse, Blosum50 and hidden Markov model sequence encoding, respectively. The numbers given in the table are calculated using the Bootstrap method [Press et al., 1992] with 500 data set realizations. The correlation values are estimated as average values over the 500 data set realizations and the error-bars the associated standard deviations. Table adapted from [Nielsen et al., 2003].

et al. [1999], the second the hidden Markov model trained on the Rammensee data set, and the other four are neural networks methods trained using sparse and Blosum sequence encoding, the linear combination of the two, and the linear combination including input from the hidden Markov model, respectively. For the matrix method by Rammensee and the hidden Markov method, we calculate the Pearson correlation between the raw output scores and the logarithmicly transformed measured binding affinities even though this might not be what optimally relates the prediction score to the measured binding affinity.

From the results shown, it is clear that the neural network methods all have a higher predictive performance compared to both the method by Rammensee and the hidden Markov model. The difference in predictive performance between the neural network and the Rammensee and the hidden Markov model methods is most significant for data sets defined by peptides with a binding affinity stronger than 50 nM, thus indicating that the signal of higher order sequence correlation is most strongly present in peptides that bind strongly to the HLA A2 molecule. The same conclusion can be drawn from the data displayed in Figure 6.8. Here the test performance for the 528 peptides is shown as a scatter plot of the prediction score versus the measured binding affinity for four of the six methods above. Again it is clear that the neural network methods in general and the combined methods in particular have a higher predictive performance than both the Rammensee and the hidden Markov model methods. The least square straight line fit to the data shown in Figure 6.8 (c)

and (d) also validates the quality and accuracy of the neural network predictions. In the two plots the straight line fits have a slope and intercept of 0.989, -0.029 and 0.979, -0.027, respectively, thus demonstrating the strength of the neural network trained on quantitative date in providing a direct relationship between the neural network output and the measured binding affinity.

In Figure 6.9, we show the sensitivity/PPV curves calculated for the data in the 528 peptide-set using the four different neural network methods as well as the method by Rammensee and the hidden Markov model method. All curves are estimated using the Bootstrap method described above. The upper graph shows the sensitivity/PPV curves for the six methods calculated for a classification threshold corresponding to 500 nM, and the lower graph the sensitivity/PPV curves for a classification threshold corresponding to 50 nM. In the insert to the graphs is shown the corresponding ROC curves for the six methods. In the labels to the curves in the insert, we give the estimated ROC areas [Swets, 1988]. In both graphs, it is clear that the combined neural methods have a performance superior to that of the other four methods. All four neural network methods and in particular the two combined methods have a performance that is substantially higher than that of the Rammensee method. The ranking of the six methods obtained using the ROC area method is identical to the ranking estimated using the Pearson correlation measure given in Table 6.2. Using a students t-test to compare the mean error of prediction (predicted binding affinity - measured binding affinity) between the comb-II method and the two neural network methods trained with a single sequence encoding, we find that the p-values are less than 10^{-4} and 0.005 for sparse and Blosum sequence encoding, respectively. The individual schemes for ranking the different methods thus all confirm that the combination of several neural network methods trained with different sequence representation has a performance superior to any neural network trained with a single sequence representation. Figure 6.9 further demonstrates that the integration of the data from the Rammensee database in the training of the neural networks, in terms of the hidden Markov model input data, increases the reliability of the combined neural network method substantially. For an affinity threshold of 500 nM the plot shows that at a PPV of 0.975 the combined neural network method comb-II has a sensitivity of 0.54, where the combined neural network method comb-I, that does not include HMM data, has a sensitivity of only 0.22. In Figure 6.9(a) the largest sensitivity gap between the combined neural method Comb-II and the method of Rammensee is found at a PPV equal to 0.7 corresponding to a difference of 0.38 in sensitivity or a difference in the number of true positive predictions of 29 of a total of the 76 high binding peptides in the data set. In Figure 6.9(b) the largest sensitivity gap between the two methods is found at a PPV equal to 0.88 corresponding to a difference of 0.37 in sensitivity or difference in the number of true positive predictions of 54 of a total of the 144

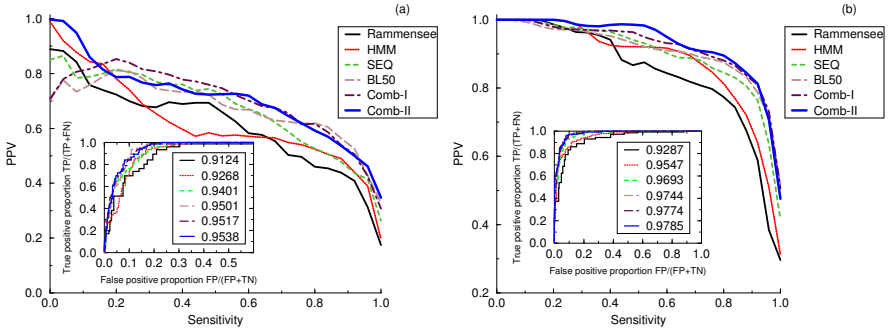

Figure 6.9: Sensitivity/PPV curves calculated from the 528-peptide data set. Six methods are shown in the graphs: Rammensee: Matrix method by Rammensee et al. [1999], HMM: Hidden Markov Model trained on data from the Rammensee database, SEQ: Neural network with sparse sequence encoding, Bl50: neural network with Blosum50 sequence encoding, comb-I: combination of neural network trained with sparse and Blosum50 sequence encoding, respectively, and comb-II: combination of neural network with sparse, Blosum50 and hidden Markov model sequence encoding. The upper graph (a) shows the curves for a classification affinity threshold of 50 nM. The lower graph (b) shows the curves corresponding to a classification affinity threshold of 500 nM. The sensitivity/PPV curves were calculated as described in Figure 6.8 using 528 data set realizations. In the insert to the graphs is shown the ROC curves defined in the text. The value given with the label to each of the curves in the insert is the area under the ROC curve. Figure adapted from [Nielsen et al., 2003].

intermediate binding peptides in the data set.

Both the method by Rammensee and the hidden Markov model are linear methods derived from binary affinity data. Neural networks can, on the other hand, both train on data with continuous binding affinities and, if it contains a hidden layer, include higher order sequence correlations in the output score. To estimate the importance of the ability of the neural network to train on continuous data and the importance of integration of higher order sequence correlations in the prediction score, we transformed the Buus data set into binary data by assigning peptides with a measured binding affinity stronger than 500 nM an output value of 0.9, and all other peptides a value of 0.1. In a five-fold cross-validation of a neural network using sparse sequence encoding the test performance on the 528 peptides in the Buus data set was found to be 0.838 $\pm$ 0.013 and 0.856 $\pm$ 0.013 for networks trained without and with a hidden layer, respectively. These numbers should be compared to the 0.877 $\pm$ 0.011 obtained for a neural network with a hidden layer trained and tested in a similar manner using continuous affinity data. The result hence confirms the importance of both training the prediction method on data with continuous

binding affinities and ability of the neural network method to integrate higher order sequence correlation in the prediction score.

6.3.6 HCV genome predictions

We use the prediction method (comb-II) to predict the location of potential CTL epitopes in the genome of Hepatitis C virus (HCV) (Genbank entry: NC 001433). The genome was downloaded from Genbank [Benson et al., 2002].

The HCV genome is relatively small. It contains 9.413 basepairs, and a coding region that translates into a number of 3,002 nine-mer peptides. Using the comb-II method to predict the binding affinity for all possible nine-mers in the genome, we find a number of 54 strong binding peptides (affinity stronger than 50 nM) and 177 intermediate binding peptides (affinity stronger than 500 nM). Figure 6.10 shows an atlas representation of the spatial distribution of predicted epitopes for the HCV genome. The atlas shows the location of the annotated proteins, the predicted binding affinity, the location of predicted high and intermediate binding peptides, as well as the estimate amino acid sequence variability mapped on to the DNA sequence of the gemone. A detailed analysis of the location of the predicted epitopes in the HCV genome demonstrates that the genome contains regions of high epitope concentration, as well as large regions where epitopes basically are absent. Most striking is the total absence of both strong and intermediate binding peptides in the N-terminal part of the structural E2 (1476-2564) domain of the genome. This domain contains the hyper-variable sequence region located in the N-terminal of E2, and one could speculate that the absence of epitopes in the region might be related to viral escape from the host immune system by means of sequence mutations [Cooper et al., 1999]. Further, we observe that epitopes are most abundant in the non-structural domain NS2 (2565-3407), and in the C-terminal of the structural E2 domain.

6.3.7 Rational vaccine design. Identification of potential CTL epitopes in the SARS genome

The use of reliable prediction tools for MHC binding is a critical step in the process of rational vaccine design and development of diagnostical tools. Here we give an example of how prediction of CTL epitopes in combination with high throughput immunology effectively can guide the identification of CTL epitopes.

The outbreak of the SARS epidemic 2002/2003 clearly demonstrated how vulnerable humans are to the merging of novel viral diseases. In seven month the SARS (Severe acute respiratory syndrome) infected more than 8400 patient

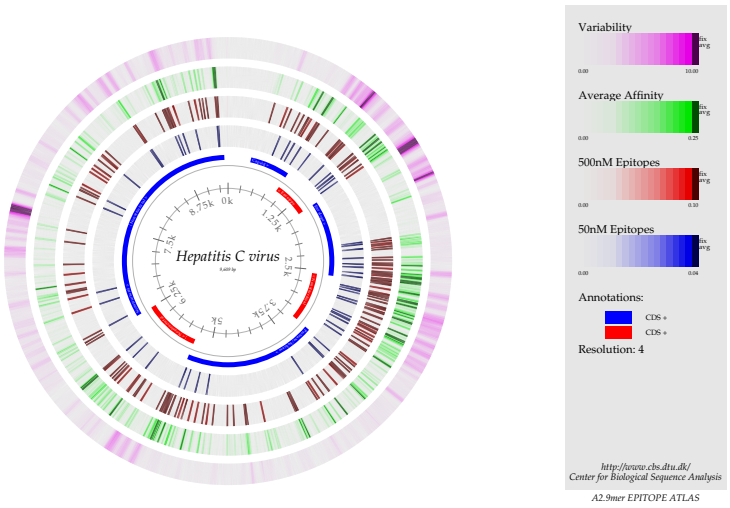

Figure 6.10: Epitope atlas for the Hepatitis C virus. The inner thin (blue) circle shows the location of annotated proteins, the broader (blue) circle show the location of high binding peptides, the red circle shows the location of intermediate binding peptides, the green circle the predicted binding affinity value, and the purple circle the sequence variability. The atlas is plotted using the "Genewiz" program by HH Staerfelt.

in over 30 counties world wide, and caused more than 800 deaths. The rapid spread of the disease and the high mortality made the need for rapid development of diagnostical tools and vaccines become of highest priority.

At the height of the SARS epidemic in the spring of 2003, we performed a complete genome-wide scan covering all (at that time known) 9 HLA supertypes (covering > 99genome contains close to 10,000 unique 9mer peptides. To identify potential CTL epitopes we applied the method of artificial neural networks and weight matrices. For each HLA supertype, we selected the top 15 candidates for test in biochemical binding assays. From the 10,000 peptides we thus select 135 for biochemical validation. The biochemical validation consists of a binding experiment, where the binding affinity between the MHC molecule and the selected peptide is measured in an ELISA experiment [Sylvester-Hvid et al., 2002]. Following this approach, we identify more than 100 potential vaccine candidates, and rapidly identified more than 100 potential SARS CTL epitopes [Sylvester-Hvid et al., 2004]. In Figure 6.11, we show a graphical representation of the predicted CTL epitopes in the SARS genome for the 9 supertypes. Also included in the figure is the sequence variability in the

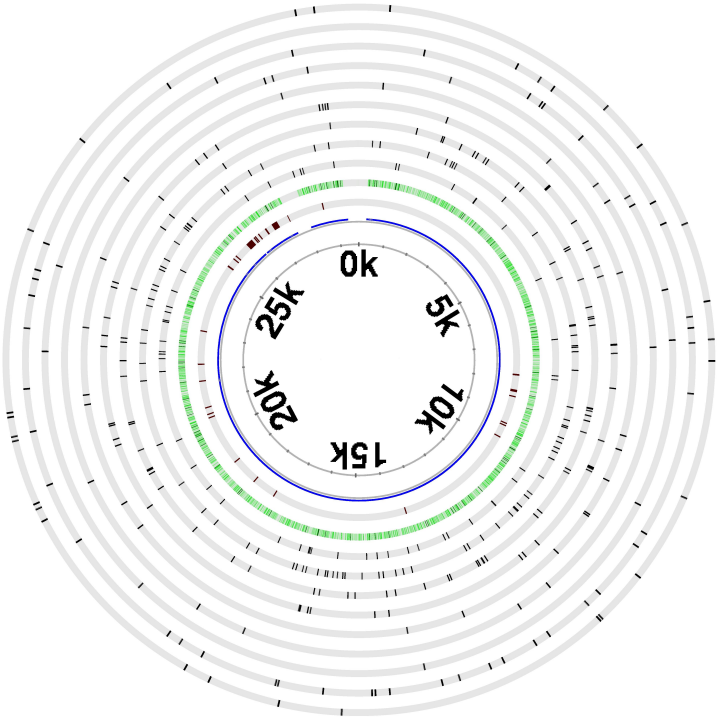

Figure 6.11: Circular Epitope map of the linear genome of SARS coronavirus. From the center and out: indexed RNA, Translated regions (blue), Observed sequence variation (brown), Predicted proteasomal cleavage (green), Predicted A1 epitopes, Predicted A*0204 epitopes, Predicted A*1101 epitopes, Predicted A24 epitopes, Predicted B7 epitopes, Predicted B27 epitopes, Predicted B44 epitopes, Predicted B58 epitopes, Predicted B62 epitopes. Figure adapted from [Sylvester-Hvid et al., 2004].

SARS genome and the predicted proteasomal cleavage. At the conclusion of this study, the SARS epidemic was over and we have been unable to get access to patients and test our putative epitopes.

6.4 Summary

When trained on very limited positive examples, matrices and other prediction methods do not contain sufficient information to distinguish between impor-

tant and less important positions in the binding motif. Empirical knowledge on position in the motif that are known to be most informative therefore can often guide the predictions if the relative weight on these position is increased. Applying this approach it is possible to obtain reliable predictions of MHC class I binding peptides, even when the allele in question is poorly investigated and few binding examples exist.

When more data is available, artificial neural network methods can be trained to predict MHC/peptide binding with a high reliability. Neural networks can take higher order sequence correlations into account when predicting the peptide MHC binding, and the analysis shown in the section of the mutual information in peptides that bind HLA-A2 revealed correlations between the amino acids located between the anchor positions. Neural networks with hidden units can take such correlations into account, but simpler methods such as neural networks without hidden units, matrix methods, and first order hidden Markov models cannot.

Here we have described a method for predicting the binding affinity of peptides to the HLA-A2 molecule which is is a combination of a series of neural networks that as input take a peptide sequence as well as the scores of the sequence to a hidden Markov model (HMM) trained to recognize HLA-A2 binding peptides. The method combines two types of neural network encoded using a classical orthogonal sparse encoding and networks where the peptide sequence is encoded as the Blosum50 scores to the 20 different amino acids. It is this ability to integrate higher order sequence correlations in to the prediction score combined with the use of several neural networks derived from different sequence encoding schemes and the fact that neural networks can be trained on data with continuous binding affinities that allows the neural network method to achieve a high reliability.

The combined approach leads to an improved performance over simpler neural network approaches. We also show that the use of the Blosum50 matrix to encode the peptide sequence leads to an increased performance over the classical orthogonal sparse encoding. The Blosum sequence encoding is beneficial for the neural network training especially in situations where data is limited. The Blosum encoding helps the neural network to generalize, so that the parameters in the network corresponding to similar and dissimilar amino acids are adjusted simultaneously for each sequence example.

A detailed comparison of the derived neural network method to that of linear methods as the matrix method by Rammensee and first order hidden Markov model has been carried out. The predictive performance was measured in terms of both the Pearson correlation coefficient and in terms of sensitivity/PPV and ROC curve plots. For all measures it was demonstrated that the neural network methods in general and the combined neural network method in particular have a predictive performance superior to that of the

linear methods.

Alternative ways to make MHC binding predictions when no or a few data are available is to use free energy calculations [Rognan et al., 1999] or threading approaches [Altuvia et al., 1995, Schueler-Furman et al., 2000]. These types of methods may be optimal when no peptides are known to bind a given MHC molecule. Also this approach may give information that is complementary to what can be obtained from the sequence alone and one possible way to improve the predictive accuracy could be to combine predictions based on sequence with predictions based on structure.

As new alleles constantly are being discovered, in humans as in animals, it is often important to be able to quickly assign these a general motif of binding peptides, e.g., for transplantation purposes or veterinary vaccination programs. Also for future rational vaccine design, it will be of great value to be able to scan for T cell epitopes as broadly as possible. For this purpose the weight matrix method trained with position specific weighting gives a major advantage as only very few binders have to be identified to be able to deduce a reliable peptide binding motif.

As an example of the use of bioinformatical prediction tools to guide the process of rational vaccine design, we perform a genome-wide scan for potential CTL epitopes in the genomes of HCV and SARS using the neural network and weight matrix methods. For the HCV genome the analysis demonstrated that the genome contains regions of high epitope concentration, as well as large regions where epitopes basically are absent. In combination with high-throughput immunology the genome-wide search for potential CTL epitopes in the SARS genome, gave an illustration of how reliable bioinformatical prediction tools effectively can be integrated in the process of vaccine design and diagnostics.