ORIGINAL PAPER

Morten Nielsen · Claus Lundegaard · Ole Lund ·
Can Keşmir

The role of the proteasome in generating cytotoxic T-cell epitopes: insights obtained from improved predictions of proteasomal cleavage

Abstract Cytotoxic T cells (CTLs) perceive the world through small peptides that are eight to ten amino acids long. These peptides (epitopes) are initially generated by the proteasome, a multi-subunit protease that is responsible for the majority of intra-cellular protein degradation. The proteasome generates the exact C-terminal of CTL epitopes, and the N-terminal with a possible extension. CTL responses may diminish if the epitopes are destroyed by the proteasomes. Therefore, the prediction of the proteasome cleavage sites is important to identify potential immunogenic regions in the proteomes of pathogenic microorganisms (or humans). We have recently shown that NetChop, a neural network-based prediction method, is the best method available at the moment to do such predictions; however, its performance is still lower than desired. Here, we use novel sequence encoding methods and show that the new version of NetChop predicts approximately 10% more of the cleavage sites correctly while lowering the number of false positives with close to 15%. With this more reliable prediction tool, we study two important questions concerning the function of the proteasome. First, we estimate the N-terminal extension of epitopes after proteasomal cleavage and find that the average extension is relatively short. However, more than 30% of the peptides have N-terminal extensions of three amino acids or more, and thus, N-terminal trimming might play an important role in the presentation of a substantial fraction of the epitopes. Second, we show that good TAP ligands have an increased chance of being cleaved by the proteasome, i.e., the specificity of TAP has evolved to fit the specificity of the proteasome. This evolutionary relationship allows for a more efficient antigen presentation.

Keywords Proteasomal cleavage · MHC class I epitope · Neural networks · Sequence encoding · Hidden Markov models · Evolution of TAP specificity · N-terminal trimming

M. Nielsen (✉) · C. Lundegaard · O. Lund · C. Keşmir
Center for Biological Sequence Analysis,
Technical University of Denmark,
BioCentrum-DTU, Building 208,
2800 Lyngby, Denmark
e-mail: mniel@cbs.dtu.dk
Tel.: +45-4525-2477
Fax: +45-4593-1585

C. Keşmir
Theoretical Biology/Bioinformatics,
Utrecht University,
Padualaan 8,
3584 CH Utrecht, The Netherlands

Introduction

The MHC class I pathway of antigen processing and presentation is highly complex and involve many steps that select the peptides to be presented on the cell surface. The most selective step is the MHC binding, where only a minor fraction of the peptide repertoire will bind to a given MHC molecule. The first step in this pathway, i.e., the cleavage by the proteasome, also shows some degree of specificity and some sites in a protein are preferentially cleaved (Eggers et al. 1995).

Previously, others and we have developed methods to predict the specificity of the mammalian proteasome (Holzhutter et al. 1999; Kesmir et al. 2002; Kuttler et al. 2000). Among these, the neural network-based method NetChop 2.0 of Kesmir et al. (2002) has recently been shown to be the most accurate of a set of publicly available cleavage prediction methods (Saxová et al. 2003). However, the current performance of NetChop is not sufficient to increase our ability to detect cytotoxic T-cell (CTL) epitopes Peters et al. 2003.

Here we present an updated version of NetChop 2.0. The new method consists of a combination of several neural networks, each trained using a different sequence-encoding scheme of the data. In this paper, we have a twofold objective. First, we will demonstrate how the predictive performance can be improved by use of different neural network-training strategies and sequence-encoding schemes. For this purpose, we shall restrict the network training to the data used in the

original NetChop publication. In this way, we can directly point out the source of improvements in the predictive performance. The second objective of this paper is to develop the best possible prediction method, and for this purpose, it is natural to extent the size of the training data sets. We evaluated the predictive performance of the new method using the benchmark described by Saxová et al. (2003), and found that using different sequence-encoding schemes allows us to achieve a performance that is significantly higher than that of NetChop 2.0. Specifically, the new method has a significant increase in the prediction sensitivity as compared to NetChop 2.0, without lowering the specificity. This means that the new method is capable of correctly identifying the cleavage sites better than NetChop 2.0. In addition to the presentation of this new method, we here address two important questions concerning the function of the proteasome: (1) How long are N-terminal extensions of peptides generated by the proteasome? (2) Can one find signs of adaptation between the specificity of TAP and the proteasome?

Material and methods

Data

We train two distinct prediction methods. The first method is trained on MHC ligands as described by Kesmir et al. (2002). C-termini of known MHC class I epitopes/ligands are assigned as proteasomal cleavage sites, whereas the positions within a ligand are assumed to be not likely cleavage sites and thus are taken as negative sites. For the negative sites, an additional filtering was made to exclude positions that carry strong characteristics of a cleavage site (see Kesmir et al. (2002) for the details of this procedure).

The second method is trained on in vitro degradation data. We use again the same data as were used by Kesmir et al. (2002): degradation of yeast enolase and bovine casein by the human constitutive proteasome. We refer to Kesmir et al. (2002) for further details and analysis on the two data sets.

For evaluation of the prediction accuracy of the two prediction methods, we perform a benchmark calculation as described by Saxová et al. (2003). The method trained with epitope data is evaluated on a set of 231 MHC class I epitopes, and the method trained with in vitro data is evaluated on digestion data of three proteins. In both cases, none of the evaluation data are included in the training data. We refer to Saxová et al. (2003) for a description of the two evaluation data sets.

Since the development of NetChop 2.0 and NetChop-20S, more data on both MHC ligands and proteasomal in vitro digests have become available. To extend our training sets, we include a set of 150 MHC ligands downloaded from the SYFPEITHI database (Rammensee et al. 1999). None of these ligands are included in the original NetChop training data set, and further, they are not part of the evaluation data set described by Saxová et al. (2003). For the in vitro digest training, we extend the training data and include

constitutive proteasome digest data of the prion protein (Tenzer et al. 2004).

Table 1 summarizes the different data sets.

Methods

Sequence encoding

We train neural networks using combinations of three distinct sequence-encoding schemes: (1) conventional sparse encoding, (2) Blosum encoding, and (3) hidden Markov model encoding. In the sparse encoding, an amino acid is represented as a 20-digit binary number, with 1 at one position and zeros at the remaining 19 (Baldi and Brunak 2001). The Blosum encoding scheme makes use of the BLOSUM50 matrix (Henikoff and Henikoff 1992), which is a measure of how similar/dissimilar amino acids are. An amino acid is encoded as a vector of 20 BLOSUM50 scores, where each score corresponds to the penalty of replacing that particular amino acid with other 20 amino acids. This encoding helps the neural network to generalize; for instance, if a cleavage site with a leucine residue is presented to the neural network, the neural network parameters corresponding to similar and dissimilar amino acids are adjusted in such a way that the neural network appears to have seen cleavage sites with isoleucine, valine, etc. This sequence-encoding scheme has previously been shown effective when training neural networks for prediction of T-cell epitopes and protein secondary structure (Nielsen et al. 2003; Thorne et al. 1996). Finally, for the hidden Markov model encoding, we construct an un-gapped hidden Markov model (or a weight matrix) describing the proteasomal cleavage motif, using the Gibbs sampler approach described by Nielsen et al. (2004). The weight matrix is constructed from all positive cleavage sites in the training data sets using sequence weighting and pseudo-count correction for low count. Using the hidden Markov model, we encode the peptide sequence as the match scores for the different positions in the cleavage motif (Nielsen et al. 2003). These match scores are given in addition

Table 1 The different data sets used for training and evaluation of the cleavage prediction methods

Data sets[a]	N	$N_{cleavage}$
Train MHC ligands	3,870	746
Extended MHC ligands	5,245	896
Evaluation MHC ligands	231	231
Train in vitro	643	164
Extended train in vitro	853	226
Evaluation in vitro	236	61

(N is the total number of data points in the different sets, $N_{cleavage}$ is the total number of cleavage sites)
Train MHC ligands, *Extended MHC ligands*, and *Evaluation MHC ligands* refer to the training, extended training and evaluation data sets extracted from the SYFPEITHI database, respectively. *Train in vitro*, *Extended train in vitro*, and *Evaluation in vitro* refer to the in vitro degradation data sets

to sparse/Blosum sequence coding. This type of sequence encoding we refer to as hidden Markov model encoding.

Neural network training

Conventional feed-forward networks with back-propagation are used in this study. A network consists of three layers—an input layer, a hidden layer with 2–22 neurons, and a single neuron output layer. For sparse and Blosum sequence encoding, the input layer has $20 \times L$ neurons, where L is the window size. When combining with the hidden Markov model encoding, the input layer has an extra L neurons, and the total number of neurons in the input layer thus is $(20+1) \times L$. Following previous optimizations made for NetChop, we use a window size of 17 to train with MHC ligands, and a window size of 7 for the in vitro data (Kesmir et al. 2002).

We train the neural networks using a fivefold cross-validation. The data are randomly split into five orthogonal sets. In one round of network training, four of the sets are combined into the training set, and the fifth set is used as a test set to decide when to terminate the training. Five rounds of training were performed so that all sets serve as a test set. The network training is terminated when the error on the test set is minimal (similar performance was obtained when stopping the training on maximum Pearson correlation on the test set data). Since the ratio of cleavage to non-cleavage sites is rather small in our data sets, we train the networks in a balanced manner, i.e., cleavage and non-cleavage sites are presented to the network with an equal frequency (Baldi and Brunak 2001; Nielsen et al. 2003).

For each five train and test data sets, we train a series of neural networks, varying the number of hidden neurons between 2 and 22, and select the network with the lowest test set error. When applying the networks to predict cleavage sites in an independent data set, the prediction of cleavage of the central amino acid in the sequence window is calculated as the simple average over the five individual neural network predictions. For the combined method using both sparse and Blosum encoding in combination with hidden Markov models, the final combined prediction score is taken as the average of the two individual predictions. The in vitro cleavage data are of very limited size, and to limit the risk of over-training the network by stopping the training on test sets of very limited size, we select an optimal value for the training cycle number and number of hidden neurons from the fivefold test performance, and repeat the training on the complete data set, using these optimal values. Using this approach the final network is trained 20 cycles, using ten hidden neurons.

Performance measurements

To evaluate the cleavage predictions is a difficult task, because negative cleavage sites (i.e., the sites in a protein that is not used for proteasomal cleavage) are only available from in vitro degradation data. Thus, when MHC ligands or T-cell epitopes are used as test sets, one needs to define the negative sites. It is erroneous to assign every internal site within an epitope as a negative site, because many T-cell epitopes with possible internal cleavage sites are found (see, e.g., Goldberg et al. 2002). We here adopt the assumptions of Saxová et al. (2003). The C-terminal of the epitopes naturally gives the positive set. We assume that none of the internal cleavages of an epitope can be more likely than the C-terminal cleavage. These requirements impose the following classification scheme:

$$TP : P_C > T \qquad FN : P_C < T$$
$$TN : P_I < \max(P_C, T) \quad FP : P_I > \max(P_C, T),$$

where P_C is the C-terminal cleavage prediction score, P_I is the maximal internal cleavage prediction score, T is the threshold value classifying predictions into cleavage and non-cleavage sites, and $\max(x,y)$ defines the maximum value of x and y. TP is true positives, TN true negatives, FP false positives and FN false negatives. Below we will use the following performance measures: Sensitivity$=TP/(TP+FN)$, Specificity$=TN/(TN+FP)$, and CC is the Matthews correlation coefficient. For the neural network predictions, T is set to 0.5.

For the in vitro data set, we have clear assignments of positive and negative cleavage sites, and no additional assumptions are necessary. We use the following performance measures: CC is the Matthews correlation coefficient, PCC is the Pearson correlation coefficient, and AROC is the area under the relative operating characteristic (ROC) curve (Swets 1988). The two latter measures are non-parametric, and hence not biased by the threshold selection implicit in the CC measure.

To address the question of whether the difference in predictive performance between two different prediction methods is statistically significant, we perform a bootstrap experiment (Press et al. 1992). In the bootstrap experiment, we generate a series of data set replica by randomly drawing n data points with replacement from the original data set, where n is the size of the original data set. For each data set, we evaluate the predictive performance of two methods. The P-value for the hypothesis that method M1 performs better than method M2 is then estimated from the simple ratio #(M1<M2)/N, where #(M1<M2) is the number of experiments where method M2 outperforms method M1, and N the number of bootstrap replica. A P-value less than 0.05 will indicate that method M1 significantly outperforms method M2.

Results and discussion

Before describing the results from the benchmark calculations comparing the predictive performance of the different methods, we first give some general comments regarding the network training with different types of sequence encoding. When training a neural network using sparse or Blosum encoding combined with a hidden Markov model,

the network training becomes significantly shorter compared to the training of a network using sparse or Blosum encoding alone. The average training cycle number for sparse- or Blosum-encoded networks is approximately 300, where the corresponding number for networks trained including hidden Markov sequence encoding is approximately 50. This difference is expected, however, since the hidden Markov model encoding provides the network with a rough description of the linear information content in the cleavage motif. On the other hand, the number of training cycles, the average number of hidden neurons and the predictive performance between networks trained using Blosum and sparse sequence encoding is very similar. Even though the Blosum encoding implicitly provides the neural network with relevant information about the chemical similarities among the different amino acids, this additional information does not lead to a significant change in neither network training nor predictive performance.

In Fig. 1, we give the benchmark calculation comparing the predictive performance of FragPredict, PAProC, NetChop 2.0, and two of the neural network methods developed in this work (Comb, NetChop 3.0) trained on epitope data, using different types of sequence and hidden Markov model sequence-encoding schemes. To see the effect of training strategy only, Comb was trained on the NetChop 2.0 data set. At the time of writing the manuscript, we were not able to access the MAPPP prediction server, and the results for FragPredict were therefore taken from Saxová et al. (2003). For PAProc, we report the prediction performance for both the older version (PAProcI) and the updated version (PAProcII). From Fig. 1, it is clear that the neural network-based methods have a predictive performance superior to that of both the FragPredict and the two PAProC methods. Note also that in this benchmark
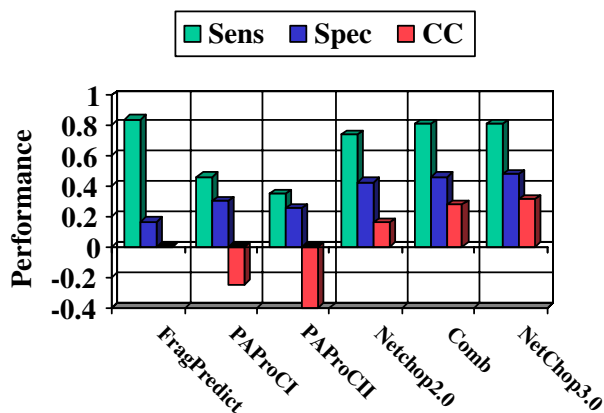

Fig. 1 Benchmark calculation of cleavage predictions evaluated on 231 MHC ligands. The performance values for FragPredict were taken from Saxová et al. (2003). *PAProCI* refers to the old version of PAProC, and *PAProCII* refers to the updated version of PAProC, predicting cleavage of the immunoproteasome. *NetChop 2.0* is a sparse-encoded single neural network (Kesmir et al. 2002), *Comb* refers to a combination of two neural network ensembles trained using sparse and Blosum encoding in combination with the hidden Markov model, respectively, and *NetChop 3.0* is a neural network ensemble as the Comb method trained on an extended data set. The performance measures were calculated as described in the text. *Sens* Sensitivity, *Spec* specificity, *CC* the Mathews correlation coefficient

calculation the updated PAProCII method had a lower performance than PAProCI. Performing the bootstrap experiment on the Matthews correlation coefficient values, we found that the combined method (Comb) had a performance that is significantly higher than that of NetChop 2.0 ($P<0.001$). There are several reasons for the increase in predictive performance. First of all, the network training strategy differs between the NetChop 2.0 and the new network methods. We here performed a fivefold cross-validated training, in which each network training was stopped when the test set error was minimal. This strategy led to an ensemble of five networks, each with an individual prediction bias. The NetChop 2.0 method was trained to optimize the Matthews correlation coefficient. In the training of NetChop 2.0, the fivefold training was performed to estimate optimal parameter settings and the final NetChop 2.0 network was a single network trained on all data using these optimal parameter settings. When we compared the performance of a fivefold cross-validated sparse-encoded network to that of NetChop 2.0, we found that the fivefold trained sparse-encoded network outperformed NetChop2.0 with a P-value of 0.03 (results not shown). Use of a network ensemble is the only difference between NetChop 2.0 and the fivefold sparse-encoded network, and it is thus clear that the strategy of generating a network ensemble led to a higher predictive performance. The second difference between NetChop 2.0 and the Comb method presented here was the use of different sequence-encoding schemes. Comparing the predictive performance of the sparse-encoded and the combined neural network, it is clear that Blosum encoding and the information coming from the hidden Markov model led further to an increase in performance.

Next, we trained neural networks, using the extended epitope data set, and a training strategy similar to that outlined for the Comp method described above (referred to as NetChop 3.0 from now on). The additional data available for training led to a better identification of non-cleavage sites: the specificity for the NetChop 3.0 and Comb methods was 0.48, and 0.46, respectively. The sensitivity was 0.81 for both methods, and the Mathews correlation was 0.31, and 0.29 for the NetChop 3.0 and Comb method, respectively. Compared to the NetChop 2.0 method, the NetChop 3.0 method thus had a close to 10% increase in sensitivity and a 15% increase in specificity. However, the gain was only marginally significant. Performing a bootstrap analysis, we found that the gain in sensitivity and Mathew's correlation was significant with P-values of 0.1 and 0.2, respectively.

Next, we compared the prediction accuracy of the combined neural network trained on in vitro data to that of FragPredict, PAProCI, PAProCII, and NetChop-20S. The results of this benchmark calculation are shown in Fig. 2.

The combined neural network method (NetChop-20S 3.0) had higher predictive performance than the other methods in the benchmark. The improvement was not as significant as in the case of NetChop 2.0 (compare Fig. 1 with Fig. 2); however, a direct comparison was difficult, since we here looked at other performance measures. A
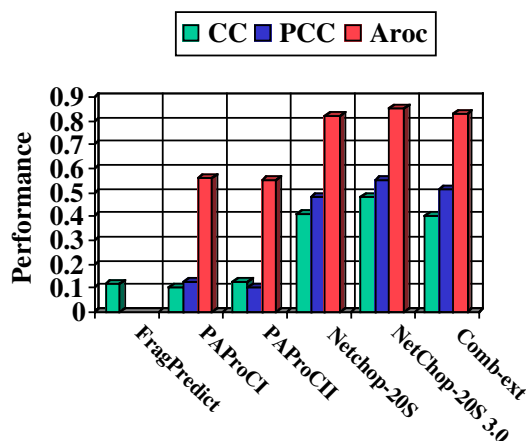
Fig. 2 Benchmark calculation of cleavage predictions evaluated on in vitro cleavage data. The performance values for FragPredict were taken from Saxová et al. (2003). *Comb-ext* is a neural network ensemble trained as NetChop-20S 3.0 on an extended data set (i.e., including the degradation of the prion protein). The performance measures are: *CC* Mathews correlation, *PCC* Pearson correlation, AROC area under the relative operating characteristic (ROC) curve. Note that the CC value for NetChop-20S reported here differs from the value given by Saxová et al. (2003)
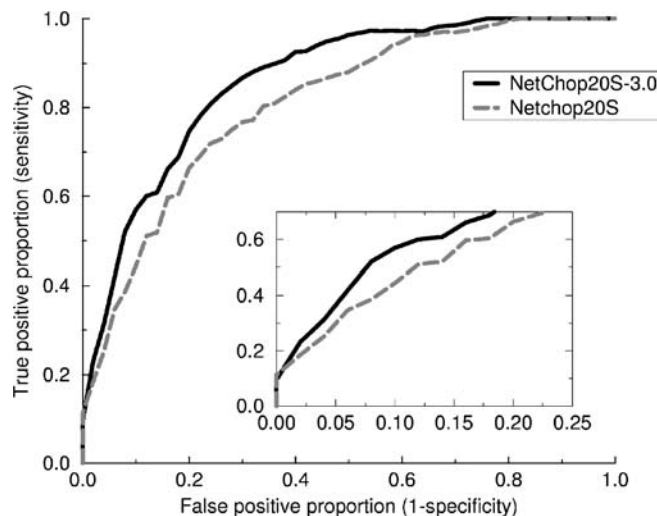

Fig. 3 ROC curves comparing the predictive performance of the combined neural network method NetChop-20S 3.0 and NetChop-20S. The curves were calculated using the bootstrap method described in the text, and were averaged over 1,000 bootstrap replications. The corresponding AROC values were 0.85 and 0.81 for the NetChop-20S 3.0 and Netchop-20S methods, respectively. The *inset to the graph* shows the high specificity part of the ROC curves in detail

detailed comparison of the predictive performance of the different methods and their statistical significance is given in Appendix A. One result in Fig. 2 is striking: the significant drop in predictive performance when including the prion in vitro digest data (Tenzer et al. 2004) in the network training (given as Comb-ext in the figure). A bootstrap calculation comparing the predictive performance of the NetChop-20S 3.0 and Comb-ext methods demonstrated that the Comb-ext method performed significantly worse than the NetChop-20S 3.0 method ($P<0.001$). There might be several reasons for this at-first odd result. The prion protein has a highly unusual amino acids composition, with close to 20% glycines. In general, glycines are poorly cleaved by the proteasome, e.g., in enolase protein, only 5% of glycines are found to be cleaved (Toes et al. 2001). However, in prion protein 27% of glycines are cleaved (Tenzer et al. 2004). It is very likely that such atypical data will pollute rather than enrich the network training. Moreover, the prion protein contains repeat regions, which can again bias the network training. We hence chose not to include the prion data in the network training.

To be able to analyze more in detail the performance differences between the NetChop-20S 3.0 and the NetChop 20S, we made ROC curve plots of the two methods (see Fig. 3). One important difference between the predictive performance of the combined method compared to that of NetChop-20Swas the large increase in sensitivity or correctly predicted cleavage sites proportion, at low value of the false positive proportion. At a false-positive proportion value of 0.1, the sensitivity of the NetChop-20S method was 0.43, corresponding to a correct identification of only 26 of the 61 cleavage sites. For the combined method, the corresponding sensitivity value was 0.57, and the number of correctly identified cleavage sites thus 35. A similar behavior was found for the combined method on epitope

data (see Fig. 1). Thus, the combination of many neural networks trained on different types and combinations of sequence encodings led to more accurate prediction algorithms. These results are in agreement with our previous work, where we have shown that the combined approach improves the prediction accuracy of MHC binding (Nielsen et al. 2003).

N-terminal extension of epitopes

The C-terminal of CTL epitopes are generated precisely by the proteasome and no further trimming is needed (Cascio et al. 2001). An exact N-terminal cleavage is, however, less essential, since a precursor peptide may be trimmed at the N-terminal by other peptidases in the cytosol (Levy et al. 2002; Reits et al. 2003) and after TAP transport into the endoplasmic reticulum by the aminopeptidase associated with antigen processing while it binds to the MHC class I molecule (Saric et al. 2002; Serwold et al. 2002; York et al. 2002). To investigate the length distribution of N-terminal extensions of CTL epitopes, we located the "host" protein in the Swiss-Prot database (Bairoch and Apweiler 2000) for all CTL epitopes in the Saxová et al. (2003) benchmark data set. Searching the protein sequence, we estimated the N-terminal extension as the distance to the nearest cleavage site at the reported N-terminal side of the epitope (i.e., we did not normalize the natural epitope length to define the extension). The output from the neural network is related to the probability of a site being cleaved. The cleavage is, however, a stochastic process, and not all potential cleavage sites are used in a given digest (Nussbaum et al. 1998). To take this stochasticity into account, we estimated the transformation from network output to the probability of

being cleaved in a given digest in two steps. The output from the neural network is a score between 0 and 1, where a value close to one indicates strong preference for cleavage, and vice versa for values close to 0. First, for all residues in the in vitro digest data set from the Saxová et al. (2003) that are predicted to be preferred cleavage sites (cleavage scores between 0.8 and 1.0), we calculated the fraction of the residue that were actually cleaved in the digest. We found that in 50% of the cases for the NetChop 3.0 predictions and 60% of the cases in NetChop-20S 3.0 predictions, a predicted cleavage site was also observed during one digest. In other words, this means that a likely cleavage site will be used by the proteasome only in approximately every second digest. Thus, a scaling factor of 0.5 for the NetChop 3.0 and 0.6 for NetChop-20S 3.0 allowed us to correctly model the stochastic nature of the proteasome. Second, we used 100 simulated digestions to estimate the average N-terminal extension of an epitope. Figure 4 shows the N-terminal extension distributions found using the two prediction methods of NetChop 3.0 and NetChop-20S 3.0 and the above approach to model stochasticity.

Since the NetChop 3.0 network was trained on epitope data, and hence might have predicted a combined specificity of proteasome, TAP and MHC direct comparison of the two methods was difficult. However, the results given in Fig. 4 suggest that a significant proportion of the epitopes have substantial N-terminal extensions. For the NetChop 3.0 method, we found that close to 45% of the epitopes have N-terminal extensions of five amino acids or more, and for the NetChop-20S 3.0 method more than 30% of the epitopes have N-terminal extension of three amino acids or more. It is clear that details of these estimates
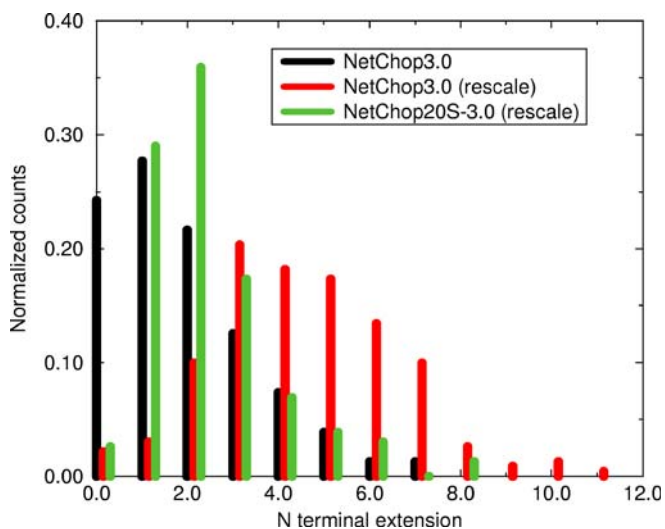

Fig. 4 Distribution of N-terminal extensions for the 231 epitopes in the Saxová et al. (2003) benchmark data set. The N-terminal extension was calculated as the distance to the nearest cleavage site at the N-terminal side of the epitope. The stochastic nature of the proteasomal cleavage was estimated from the network output score as described in the text. The *red and green bars* show the N-terminal extensions predicted using the NetChop 3.0 and NetChop-20S 3.0 methods and correcting the predictions for stochasticity. The *black bars* show the N-terminal extensions predicted using the raw NetChop 3.0 output

depend strongly on the rescaling used to transform the neural network output into cleavage probabilities. For comparison, we in Fig. 4 include a histogram for N-terminal extensions calculated using the raw NetChop 3.0 output scores as the cleavage probability. Even with this estimate, a substantial fraction of the epitopes (more than 25%) have an N-terminal extension of three amino acids or more. The general conclusion is thus clear: even though some epitopes would be generated by the proteasome precisely at the N-terminal, the majority of epitopes are generated with a N-terminal extension, indicating that N-terminal trimming plays an important role in effective antigen presentation. It is important to note that this analysis is restricted to only the epitopes that are generated by the proteasomes. Other cytosolic peptidases, like TPIII (Reits et al. 2003, 2004) are also involved in the generation of CTL epitopes. However, at the moment we have no prediction tools available to estimate how large N-terminal extensions would be for these epitopes.

An evolutionary adaptation of the proteasome and TAP specificities?

We have recently shown that the specificity of human MHC molecules have evolved to fit the specificity of the immunoproteasome (Kesmir et al. 2003). Thus, good MHC ligands also have a high probability to be generated by the proteasome. To add TAP into this evolutionary relation would increase the efficacy of the antigen processing and the presentation even further.

To investigate the footprints of a possible specificity adaptation between TAP and the proteasome, we first analyzed the available experimental data on the specificity of these molecules. In Fig. 5, we show the preference of immuno- and constitutive proteasomes at P1 position (the P1 position used by the proteasome would become the C-terminal position of TAP ligands) in the form of a sequence logo based on the data generated by Toes et al. (2001). A sequence logo (Schneider and Stephens 1990) is a clear visualization of (1) to what extent a position in a sequence is conserved (given by the height of a bar, the information content) and (2) which amino acids are most frequently found at a particular position (the height of each amino acid in the logo is proportional to the frequency of occurrence in that position). To analyze the specificity of C-terminal of TAP ligands, TAP-binding peptides were downloaded from the AntiJen database (http://www.jenner.ac.uk/AntiJen/). This database contains a set of close to 350 unique peptides with known TAP-binding affinity. Only 63 of these peptides are natural ligands with a host protein in the SwissProt database and bind to TAP with an efficient affinity (i.e., affinity <100,000 nM). The C-terminal logo of these good TAP ligands resembles the immunoproteasome more than the constitutive proteasome (see Fig. 5, given as TAP-Ct). The acidic residues D and E, especially, are frequently used by the constitutive proteasome, but hardly by the immunoproteasome and TAP. Finally, we used the extended MHC ligand data set used to train
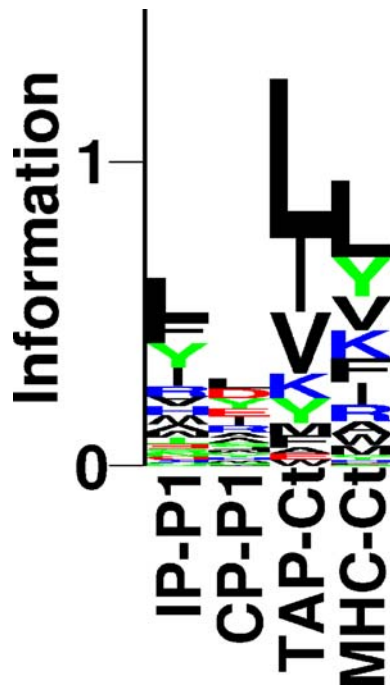
Fig. 5 Sequence logo of experimental data available on the specificity of the immuno- and constitutive proteasome (taken from Toes et al. 2001), TAP, and MHC molecules. Sixty-three good TAP ligands from the AntiJen database (http://www.jenner.ac.uk/AntiJen/) were used to generate this logo, while MHC logo was generated by using 896 MHC ligands in our extended MHC ligand data set (see Table 1). These ligands are restricted to more than 50 human molecules, and therefore the logo given here resembles the specificity of MHC molecules at the population level. Amino acids are color-coded according to their physicochemical characteristics. *Green* neutral and polar, *blue* basic, *red* acidic, *black* neutral and hydrophobic

NetChop 3.0 to analyze the preference of class I MHC molecules at C-terminal of their ligands. This specificity (given in Fig. 5 as MHC-Ct) is again very similar to TAP and immunoproteasome specificities. These results suggest that all main players of MHC class I antigen processing and presentation pathway have been through an adaptation process to optimize CTL epitope generation.

To quantify the similarity between TAP and proteasome specificity, we predict C-terminal cleavage for the above-mentioned TAP ligands. We have shown that our predictor trained on epitope data predicts the C-terminal cleavage of epitopes most correctly (Kesmir et al. 2002). However, these networks might have learned the specificity of the TAP molecule, since TAP-binding motifs could be embedded in the epitope data set. Therefore, we use the NetChop-20S 3.0 predictor to circumvent a possible bias in the predictions. We predict the average C-terminal cleavage score of these "natural TAP ligands" and compare that to the average cleavage score calculated in the set of natural TAP ligands by shuffling the amino acids in the ligands. For the TAP ligands, the average cleavage score is 0.607 ± 0.216, and for the shuffled ligand set the value is 0.427 ± 0.260. The average cleavage score in TAP ligand data set is significantly higher than for the shuffled ligands ($P<0.001$ in a student's t-test for significantly different means; Press

et al. 1992). Thus, the TAP-binding motif, especially the preference of C-terminal, allows for a significantly higher chance of being cleaved by the proteasome. Note that our NetChop-20S predictor is trained on the constitutive proteasome specificity. Since the immunoproteasome specificity is much more adapted to TAP specificity (see Fig. 5), we expect that good TAP ligands be generated by the immunoproteasome more frequently than our estimate here.

The TAP ligand data set we use contains many MHC ligands. Therefore, both analysis presented here might be showing actually the adaptation between MHC molecules and the proteasome. To remove this bias, we predict cleavage of 500,000 9-meric peptides selected randomly from proteins in the Swiss-Prot database. In Fig. 6, we plot the average proteasomal cleavage score for each of the 20 amino acids in this large peptide data set by the NetChop-20S 3.0 method versus the TAP preference score on the C-terminal, which is adapted from a method developed by Peters et al. 2003 to predict peptide-binding affinity to human TAP molecules. A high proteasomal cleavage score indicates a high chance of cleavage, and a low (negative) TAP score indicates a high chance of TAP transport. The TAP preference score and proteasomal cleavage score is significantly correlated (Kendall's $\tau=-0.44$, $P=0.007$; Press et al. 1992). This correlation indicates that the TAP specificity to some degree is adapted so that the peptides generated by the proteasome are transported efficiently to the endoplasmic reticulum. While the correlation between the two scores is not perfect, hardly any amino acids are placed in the lower left part of the plot (only K is marginally present in this part of the plot). This part of the plot contains amino acids that are favored by TAP for transport but disfavored by proteasome for cleavage. The lower right part of the plot
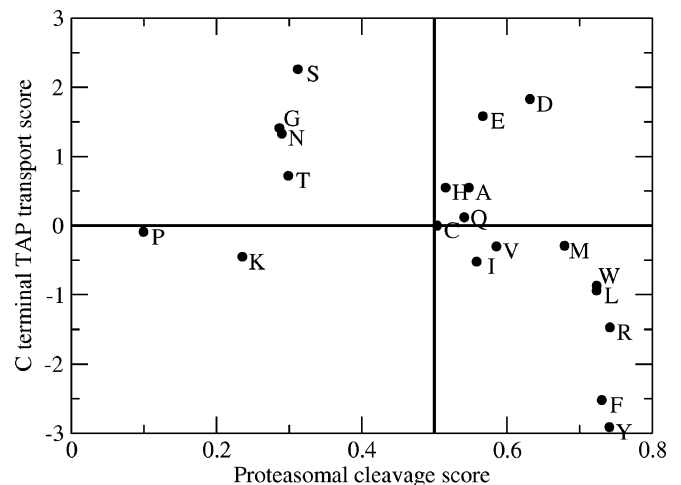

Fig. 6 Evolutionary relationship between the TAP and proteasome specificities. The average cleavage score (using NetChop-20S 3.0) was calculated for a set of 500,000 peptides 9-meric randomly selected from proteins the in Swiss-Prot database. The C-terminal TAP transport score was adapted from the TAP-binding predictor developed by Peters et al. (2003). The C-terminal TAP transport score was plotted as a function of the average proteasome cleavage score for each of the 20 amino acids. The *lines in the plot* give a schematic separation into regions in favor or in disfavor for TAP binding and proteasomal cleavage

contains amino acids that favor both proteasomal cleavage and TAP transport, and the upper left corner of the plot amino acids that disfavor both proteasomal cleavage and TAP transport. The amino acids occurring in these two parts of the figure to a large extent overlap with the amino acids preferences earlier identified for the proteasome and TAP (Kesmir et al. 2002; van Endert 1996). The cleavage predictor used here is trained on the constitutive proteasome specificity, which has a preference for cleavage after D, E (see Fig. 5). This preference is not shared by the immunoproteasome, and one would thus in this analysis expect an even stronger correlation between the TAP and proteasomal specificities when the immunoproteasome is considered.

Concluding remarks

Even though NetChop 2.0 has been shown to have the highest performance of predicting proteasomal cleavage sites (Saxová et al. 2003), there is plenty of room for improvement. Peters et al. (2003) have recently shown that integrating NetChop-20S predictions with MHC-binding predictions does not improve the overall performance of predicting CTL epitopes. The use of NetChop-20S for such a task is not the optimal choice, as this predictor is based on solely constitutive proteasome specificity (Kesmir et al. 2002). However, even integrating NetChop 2.0 with MHC-binding predictions does not improve "epitope prediction" ability drastically (Børgersen et al., unpublished results). We here attempt to increase the quality of proteasome cleavage predictions by using different training strategies and sequence-encoding schemes. We believe that the new version of NetChop (Netchop 3.0) with significant increase in both sensitivity and specificity can be used as a filter to improve CTL-epitope predictions.

High quality predictions of the proteasome cleavage can be used to achieve a more quantitative picture of the MHC class I antigen processing and presentation pathway. Based on estimates coming from average turnover of proteins in a cell, Yewdell and colleagues argue that the efficiency of antigen processing is low, meaning that most of the potential MHC ligands are destroyed by the proteasome (Yewdell 2001). If this is true, one bottleneck of the antigen presentation lies in the generation by the proteasome, and the presentation levels of the peptides can depend heavily on their cleavage probabilities. N-terminal trimming has also been suggested as a major process that can destroy CTL epitopes (Stoltze et al. 2000). Our analysis on estimation of N-terminal trimming indicates that a large fraction of the epitopes would not be destroyed by N-terminal trimming. On the contrary, they would need a trimming of three amino acids or more. However, for other epitopes N-terminal trimming can be very destructive, because these epitopes hardly need a trimming, and when exposed to proteases in the cytoplasm and the endoplasmic reticulum can become too short to be MHC class I ligands. Thus, N-terminal trimming plays an important (positive or negative) role in shaping the fate of CTL epitopes.

Obviously, there has been a large selection pressure on the adaptive immune system to increase the seemingly inefficient way of antigen processing. In this paper, we provide evidence that the specificity of TAP and the proteasome are adapted to each other. It is very hard to prove who is adapting whom in co-evolutionary systems. However, we know that the constitutive proteasome is probably the oldest molecule in MHC class I processing pathway. Thus, one can speculate that TAP has evolved to adapt the specificity of the proteasome, because it seems that TAP has a preference even for fragments digested by the constitutive proteasome. This specificity adaptation increases the efficiency of the translocation of the proteasome products, which are more likely CTL epitopes than random peptides (Kesmir et al. 2003). Also, we find a large overlap between the amino acids that are the favored/disfavored for proteasomal cleavage and the amino acid preference at the C-terminal end of the TAP transporter-binding motif. An observation that further indicates an evolutionary relationship between the two specificities.

Appendix A

The AROC values for the NetChop 20S and the NetChop 20S-3.0 are 0.81 and 0.85, respectively, and the CC and PCC are 0.41, 0.48 and 0.48, 0.55, respectively. To estimate the statistical significance of the difference in predictive performance between two methods, we performed the bootstrap experiment as described above. We found that the NetChop 20S–3.0 method has a performance that is significantly higher that that of NetChop 20S in terms of both the PCC, and the AROC values ($P<0.05$). The performance difference between the two methods in terms of the Mathews CC is not significant ($P>0.3$). That the performance increase is least significant in a fixed cut-off classification measure (Matthews CC) is not surprising, as the NetChop 20S method was trained to have an explicit classification bias around 0.5. This is not the case for any of the new neural networks.

References

Bairoch A, Apweiler R (2000) The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. Nucleic Acids Res 28:45–48

Baldi P, Brunak S (2001) Bioinformatics: the machine learning approach, 2nd edn. MIT Press, Cambridge

Cascio P, Hilton C, Kisselev AF, Rock KL, Goldberg AL (2001) 26S Proteasomes and immunoproteasomes produce mainly N-extended versions of an antigenic peptide. EMBO J 20:2357–2366

Eggers M, Boes-Fabian B, Ruppert T, Kloetzel PM, Koszinowski UH (1995) The cleavage preference of the proteasome governs the yield of antigenic peptides. J Exp Med 182:1865–1870

Goldberg AL, Cascio P, Saric T, Rock KL (2002) The importance of the proteasome and subsequent proteolytic steps in the generation of antigenic peptides. Mol Immunol 39:147–164

Henikoff S, Henikoff JG (1992) Amino acid substitution matrices from protein blocks. Proc Natl Acad Sci USA 89:10915–10919

Holzhutter HG, Frommel C, Kloetzel PM (1999) A theoretical approach towards the identification of cleavage-determining amino acid motifs of the 20 S proteasome. J Mol Biol 286: 1251–1265

Kesmir C, Nussbaum AK, Schild H, Detours V, Brunak S (2002) Prediction of proteasome cleavage motifs by neural networks. Protein Eng 15:287–296

Kesmir C, Noort VV, Boer RJD, Hogeweg P (2003) Bioinformatic analysis of functional differences between the immunoproteasome and the constitutive proteasome. Immunogenetics 55: 437–449

Kuttler C, Nussbaum AK, Dick TP, Rammensee HG, Schild H, Hadeler KP (2000) An algorithm for the prediction of proteasomal cleavages. J Mol Biol 298:417–429

Levy F, Burri L, Morel S, Peitrequin AL, Levy N, Bachi A, Hellman U, Van den Eynde BJ, Servis C (2002) The final N-terminal trimming of a subaminoterminal proline-containing HLA class I-restricted antigenic peptide in the cytosol is mediated by two peptidases. J Immunol 169:4161–4171

Nielsen M, Lundegaard C, Worning P, Lauemoller SL, Lamberth K, Buus S, Brunak S, Lund O (2003) Reliable prediction of T-cell epitopes using neural networks with novel sequence representations. Protein Sci 12:1007–1017

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

Nussbaum AK, Dick TP, Keilholz W, Schirle M, Stevanovic S, Dietz K, Heinemeyer W, Groll M, Wolf DH, Huber R, Rammensee HG, Schild H (1998) Cleavage motifs of the yeast 20S proteasome β subunits deduced from digests of enolase 1. Proc Natl Acad Sci USA 95:12504–12509

Peters B, Bulik S, Tampe R, Endert PMV, Holzhutter HG (2003) Identifying MHC class I epitopes by predicting the TAP transport efficiency of epitope precursors. J Immunol 171:1741–1749

Press WH, Flannery BP, Teukolsky SA, Vetterling WT (1992) Numerical recipes in C: the art of scientific computing. Cambridge University Press, Cambridge

Rammensee H, Bachmann J, Emmerich NP, Bachor OA, Stevanovic S (1999) SYFPEITHI: database for MHC ligands and peptide motifs. Immunogenetics 50:213–219

Reits E, Griekspoor A, Neijssen J, Groothuis T, Jalink K, Veelen PV, Janssen H, Calafat J, Drijfhout JW, Neefjes J (2003) Peptide diffusion, protection, and degradation in nuclear and cytoplasmic compartments before antigen presentation by MHC class I. Immunity 18:97–108

Reits E, Neijssen J, Herberts C, Benckhuijsen W, Janssen L, Drijfhout JW, Neefjes J (2004) A major role for TPPII in trimming proteasomal degradation products for MHC class I antigen presentation. Immunity 20:495–506

Saric T, Chang SC, Hattori A, York IA, Markant S, Rock KL, Tsujimoto M, Goldberg AL (2002) An IFN-gamma-induced aminopeptidase in the ER, ERAP1, trims precursors to MHC class I-presented peptides. Nat Immunol 3:1169–1176

Saxová P, Buus S, Brunak S, Kesmir C (2003) Predicting proteasomal cleavage sites: a comparison of available methods. Int Immunol 15:781–787

Schneider TD, Stephens RM (1990) Sequence logos: a new way to display consensus sequences. Nucleic Acids Res 18:6097–6100

Serwold T, Gonzalez F, Kim J, Jacob R, Shastri N (2002) ERAAP customizes peptides for MHC class I molecules in the endoplasmic reticulum. Nature 419:480–483

Stoltze L, Nussbaum AK, Sijts A, Emmerich NP, Kloetzel PM, Schild H (2000) The function of the proteasome system in MHC class I antigen processing. Immunol Today 21:317–319

Swets JA (1988) Measuring the accuracy of diagnostic systems. Science 240:1285–1293

Tenzer S, Stoltze L, Schonfisch B, Dengjel J, Muller M, Stevanovic S, Rammensee HG, Schild H (2004) Quantitative analysis of prion-protein degradation by constitutive and immuno-20S proteasomes indicates differences correlated with disease susceptibility. J Immunol 172:1083–1091

Thorne JL, Goldman N, Jones DT (1996) Combining protein evolution and secondary structure. Mol Biol Evol 13:666–673

Toes RE, Nussbaum AK, Degermann S, Schirle M, Emmerich NP, Kraft M, Laplace C, Zwinderman A, Dick TP, Muller J, Schonfisch B, Schmid C, Fehling HJ, Stevanovic S, Rammensee HG, Schild H (2001) Discrete cleavage motifs of constitutive and immunoproteasomes revealed by quantitative analysis of cleavage products. J Exp Med 194:1–12

van Endert PM (1996) Peptide selection for presentation by HLA class I: a role for the human transporter associated with antigen processing? Immunol Res 15:265–279

Yewdell JW (2001) Not such a dismal science: the economics of protein synthesis, folding, degradation and antigen processing. Trends Cell Biol 11:294–297

York IA, Chang SC, Saric T, Keys JA, Favreau JM, Goldberg AL, Rock KL (2002) The ER aminopeptidase ERAP1 enhances or limits antigen presentation by trimming epitopes to 8–9 residues. Nat Immunol 3:1177–1184