MHC Class I Epitope Binding Prediction Trained on Small Data Sets

Claus Lundegaard[1], Morten Nielsen[1], Kasper Lamberth[2], Peder Worning[1], Christina Sylvester-Hvid[2], Søren Buus[2], Søren Brunak[1], and Ole Lund[1]

[1] Center for Biological Sequence Analysis, BioCentrum, Technical University of Denmark. Building 208, DK-2800 Lyngby, Denmark.
Tel: (+45) 45 25 24 26, Fax: (+45) 45 83 15 95
lunde@cbs.dtu.dk
[2] Department of Experimental Immunology, Institute of Medical Microbiology and Immunology, University of Copenhagen. Denmark

Abstract. The identification of potential T-cell epitopes is important for development of new human or vetenary vaccines, both considering single protein/subunit vaccines, and for epitope/peptide vaccines as such. The highly diverse MHC class I alleles bind very different peptides, and accurate binding prediction methods exist only for alleles were the binding pattern have been deduced from peptide motifs. Using empirical knowledge of important anchor positions within the binding peptides dramatically reduces the number of peptides needed for reliable predictions. We here present a general method for predicting peptides binding to specific MHC class I alleles. The method combines advanced automatic scoring matrix generation with empirical position specific differential anchor weighting. The method leads to predictions with a comparable or higher accuracy than other established prediction servers, even in situations where only very limited data are available for training.

1 Introduction

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 a larger antigen presentation process. The process also includes processing (cleavage) of proteins and transportation into the endoplasmic reticulum. These two 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 allele space of human class I MHCs (also called class I Human Leukocyte Antigens or HLAs) is 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). 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.

G. Nicosia et al. (Eds.): ICARIS 2004, LNCS 3239, pp. 217–225, 2004.

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). Several prediction methods have been proposed for discrimination between binders and non-binders, such as data derived weight matrices, including the publicly available BIMAS (Parker et al. 1994) and SYFPEITHI (Rammensee et al. 1999), weight matrices with optimized position weighting (Yu et al. 2002) and ANNs (Brusic et al. 1994; Adams and Koziol 1995). 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), and for affinity predictions ANNs outperform the simpler methods (Gulukota et al. 1997; Nielsen et al. 2003) but generally ANNs needs many examples in the training (Yu et al. 2002). Predictions in general tends to be more precise when more examples have been included in the training (Yu et al. 2002), but experimental data on peptides binding to HLA complexes are published in large numbers for just a few alleles. Here we will investigate the possibility to get reliable predictions even when data is limited. It have earlier been shown that a position weighted matrix were slightly better for A*0201 predictions than an unweighted matrix (Yu et al. 2002), but not to which extent such a weighting will influence the number of data needed to generate acceptable predictors.

2 Materials and Methods

Data sets: All 9-mer peptides assigned as binding to HLA class I alleles HLA-A*0101 (82 peptides), HLA-A*0201(450 peptides), HLA-A*0301 (63 peptides), HLAA* 1101 (77 peptides), and HLA-B*0702 (69 peptides) was extracted from the databases SYFPEITHI (Rammensee et al. 1999) and MHCPEP (Brusic et al. 1998). These peptides are in the following referred to as training sets. As evaluation sets we used peptides for which the affinities for the selected alleles had been measured (K. Lamberth, unpublished) using the ELISA method described by Sylvester-Hvid et al. (Sylvester-Hvid et al. 2002). The binders/non binders ratios were as follows using a threshold for binders of 500 nM: A*0101 (37/284), A*0201 (20/197), A*0301 (6/211), A*1101 (9/208). Furthermore peptides relevant to SARS (Sylvester-Hvid et al. in press) were used to evaluate matrices trained on peptides binding to the A*0101, A*0201, A*0301, A*1101, B*0702, B*1501 and B*5801 alleles. There was no peptide overlap (i.e., no identical peptides) between the training data and the evaluation sets. As independent evaluation sets for the alleles A*0101 and A*0201 we further used peptides extracted from the MHCBN 3.1 database (Bhasin et al. 2003) excluding peptides all ready present in the training sets. The same evaluation sets were also used to evaluate the prediction accuracy of both the BIMAS (Parker et al. 1994) and SYFPEITHI (Rammensee et al. 1999) prediction methods. SYFPEITHI predictions were performed using the web server (http://syfpeithi.bmiheidelberg. com), and BIMAS predictions were performed as described at the web server, using matrices downloaded from the web site
(http://bimas.cit.nih.gov/cgibin/ molbio/hla_coefficient_viewing_page).