Modeling epitope affinity to MHC molecules using multiple SVM kernels

Peter DeFord

March 18, 2017

1 Introduction

Activation of cytotoxic T cells in the immune system requires the presentation of epitopes on the cell surface by the major histocompatibility complex (MHC) receptors[1]. These epitopes provide the basis for the activated T cells to specifically target and kill infected or foreign cells.

Methods for predicting which peptides would bind to MHC receptors have been around since at least 1988[1, 2]. This opened the way for engineering specific epitopes that when introduced to the immune system would serve as a vaccine[1]. This approach demonstrates promise for traditional vaccine applications, as well as vaccines targeting disorders such as certain families of cancer.

The current state-of-the-art method for making these kinds of predictions is known as NetMHC[3]. As the name implies, it is based on artificial neural networks (ANNs), which have been demonstrated to flexibly fit any function without needing it to be explicitly defined. One of the limitations of ANNs, however, is that they are difficult to interpret. This limitation is true for other new methods currently under development such as MHCflurry[4].

I propose to use linear methods, such as support vector machines (SVMs), which are easily interpretable. SVMs are able to fit a larger number of functions than some other linear methods due to the kernel trick and the intrinsic ability to fit data in higher-dimensional spaces without actually transforming the data. In the method EnhancerFinder, multiple SVM kernels are successfully combined to take advantage of each data types's respective characteristics[5].

In order to compensate for the trade off between interpretability and performance, I propose to investigate additional encoding schemes for the

peptides in order to capture the relevant features in a lower dimensional space.

Objective The objective of this project will be to evaluate the ability of simple linear models to approach the performance of Neural Network based approaches, and to identify which features contribute most to this performance.

2 Methods

Data The data will be obtained from the *immune epitope database*[6] as is done in the MHCflurry paper[4].

Encoding My initial strategies towards encoding the feature data will be as follows:

- 20 length vector with a binary indicator for the presence of a given amino acid
- 20 by k length vector with a positional indicator for each amino acid, where k is the length of the peptide to be considered.
- Binary positional indicators representing biochemical features of each amino acid, such as charge, size, resonance, and functional groups.
- Blosum matrix values as used in NetMHC[3]. In short, for each amino acid in the peptide, the corresponding column of substitution scores from the Blosum 50 matrix is appended to the feature vector.

Algorithms My focus will be on using SVMs to make my predictions. Instead of simply creating one high dimensional feature space, I will use a multiple kernel SVM approach, such as that implemented in EnhancerFinder[5]. For each of the encoding strategies I will use cross validation to select the best kernel to represent that feature.

Determining feature importance will be accomplished several ways. First, I will examine the model parameters to identify which features are contributing the most. Secondly, I will use principal component analysis to reduce the dimensionality of the feature space. The loadings on this reduced space

should give insight into the role of each feature. Finally, I will use a leave-one-out strategy, seeing how the model performance suffers from the exclusion of each feature independently. Those with the largest impacts will be the most important.

Scoring metrics For early cross validation stages, the area under the receiver operator characteristic curve (auROC) will be used to evaluate the ability of the classifiers to discriminate between low affinity and high affinity interactions. The final model's performance will be assayed in this way (auROC), as well as computing the correlation between predicted affinities and true affinities, similar to the metrics used by Rubinsteyn, *et al*[4].

References

[1] Markus Schirle, Toni Weinschenk, and Stefan Stevanovic. Combining computer algorithms with experimental approaches permits the rapid and accurate identification of t cell epitopes from defined antigens. *Journal of Immunological Methods*, 257(12):1 – 16, 2001.

[2] J.B. Rothbard and W.R. Taylor. A sequence pattern common to t cell epitopes. *EMBO Journal*, 7(1):93–100, 1988.

[3] Morten Nielsen, Clause Lundegaard, Peder Worning, Sanne Lise Laue-Moller, Kasper Lamberth, Soren Buus, Soren Brunak, and Ole Lund. Reliable prediction of t-cell epitopes using neural networks with novel sequence representations. *Protein Science*, 12:1007–1017, 2003.

[4] Alex Rubinsteyn, Timothy O'Donenell, Nandita Damaraju, and Jeffrey Hammerbacher. Predicting peptide-mhc binding affinities with imputed training data. *bioRxiv*, June 2016.

[5] Genevieve D. Erwin, Nir Oksenberg, Rebecca M. Truty, Dennis Kostka, Karl K. Murphy, Nadav Ahituv, Katherine S. Pollard, and John A. Capra. Integrating diverse datasets improves developmental enhancer prediction. *PLOS Computational Biology*, 10(6):1–20, 06 2014.

[6] N Salimi, W Fleri, B Peters, and A Sette. The immune epitope database: a historical retrospective of the first decade. *Immunology*, 137:117–123, September 2012.