Epitope Classification Using Support Vector Machines

Bornika Ghosh and Andrew Parker

12th May 2009

Goal for Milestone:

We wanted to design an SVM to classify epitopes (peptide sequences) as binders or non-binders for an MHC class II allele.

Data:

We chose MHC class II allele DRB1*0401 from the MHCBN database. The set of binders for the above allele consists of 867 binders and 211 non-binders. In the next two weeks we plan to test epitope binding for more alleles from MHCBN.

Algorithm:

The algorithm to classify epitopes as binders and non-binders consists of two stages. The first is to create fixed length feature vectors for the variable length epitopes. The feature vectors are constructed to represent physicochemical properties of the peptide sequences or epitopes in the data set. Each of the feature vectors is 21 elements long [1]. For determining the various physicochemical properties, we divided the 20 amino acids into 3 groups based upon them being hydrophobic, neutral and polar in nature. We then defined 3 descriptors, composition (C), transition (T) and distribution (D), to describe the global composition of the epitopes. The descriptor C was computed as a vector of 3 real numbers, each corresponding to the fraction of amino acids for each of the above 3 groups, found in the epitope. So C1 gave the fraction of amino acids from group one, C2 from group 2 and C3 from group 3 in the epitope. The descriptor T characterizes the fraction of frequency with which amino acids from a group are followed and preceded by amino acids from a different group. T1 gives the fraction of transitions from class 1 to class 2, T2 gives the fraction of transitions from class 1 to class 3, and T3 gives the fraction of transitions from class 2 to class 3. The third descriptor D, is the chain length up to which, the first, 25%, 50%, 75% and 100% of amino acid of each class are found in the peptide sequence. We have D1, D2 and D3 for the 3 groups mentioned above and each of these, have 5 elements for first, 25%, 50%, 75% and 100%. Hence the feature vector consists of 3 + 3 + (3*5) features, making it a 21 element long vector. We derive the feature vector for every epitope in the dataset, and this forms the input to the next stage of the algorithm. In the second stage we compute the support vectors for the training set of feature vectors. We chose 50 binding epitopes and 50 non-binding epitopes from the list of epitopes of the two types. We tried to learn first a linear SVM model for the feature vectors. We used

MATLAB function quadprog for it for the dual problem. The MATLAB function quadprog did not converge. We next tried to learn the SVM parameters, by using the algorithm given by Varma and Ray [2,3], which generalizes Multiple Kernel Learning. We used code that was kindly provided by Manik Varma, to learn weights on individual kernels associated with every feature in the high-dimensional feature space. The algorithm works in two loops, first the kernels are learnt by optimizing the weights associated with them, and then based upon the set of the learnt kernels, the SVM parameters are learnt, and this is continued till convergence. We tried two different kernel options for the GMKL algorithm. We tried Product of the RBF kernels associated with the features and the Product of exponential kernels of pre-computed distance matrices associated with the features. These two different kernel options were mentioned in the code provided by Manik Varma. The code also gives the option of choosing between MATLAB function quadprog or LIBSVM for learning the SVM parameters. We have used the MATLAB quadprog option. We intend to use LIBSVM next for learning the SVM parameters, because quadprog does not always converge, and does not give good results.

Results so far:

We used 50 binders and 50 non-binders from the list of all epitopes for our chosen allele. We had to choose a small number of epitopes for the training set, because quadprog was not converging for bigger training set sizes. We got a set of 85 support vectors. The number of support vectors is very high, which may be due to use of MATLAB quadprog for solving the dual problem, or maybe the features in the higher dimensional feature space are not good discriminators. We constructed a test set which had 300 feature vectors consisting of epitopes from each class. The misclassification rate was around 40%, which is quite high. The reason behind this could be the lack of good support vectors. We intend to use LIBSVM next for learning the SVM parameters, which might improve the results we have got so far. We also plan to test our algorithm on other alleles. In our future work, we would want to run cross-validation type of experiments. The sizes of the datasets for the two different classes are sometimes imbalanced, and cross-validation shall allow us to classify the epitopes more appropriately.

REFERENCE:

[1] Prediction of MHC binding peptides of flexible lengths from sequence-derived structural and physicochemical properties. J. Cui, L. Y. Han, H. H. Lin, H. L. Zhang, Z. Q. Tang, C.J. Zheng, Z.W. Cao, Y.Z. Chen. Molecular Immunology. (2007)

[2] More Generality in Efficient Multiple Kernel Learning. Manik Varma, Bodla Rakesh Babu. Appearing in Proceedings of the 26th International Conference on Machine Learning. (2009)

[3] Learning the Discriminative Power-Invariance Trade-Off. Manik Varma, Debajyoti Ray. (2007)