Prediction of MHC Class II binding peptides incorporating bayesian transfer hierarchies

Ravikiran Janardhana
Department of Computer Science
University of North Carolina at Chapel Hill
Email: ravikirn@cs.unc.edu

Abstract

T-cells are key players in regulating a specific immune response. Activation of cytotoxic T-cells requires recognition of specific peptides bound to Major Histocompatibility Complex (MHC) class II molecules. MHC-peptide complexes are potential tools for diagnosis and treatment of pathogens and cancer, as well as for the development of peptide vaccines. Only one in 100 to 200 potential binders actually binds to a certain MHC molecule, therefore a good prediction method for MHC class II binding peptides can reduce the number of candidate binders that need to be synthesized and tested for successful design of peptide and protein based vaccines.

I. Introduction

The activation of CD4+ helper T cells is essential for the development of adaptive immunity against pathogens [1]. A critical step in CD4+ T cell activation is the recognition of epitopes presented by MHC class II molecules [2]. MHC class II molecules are heterodimers expressed on the surface of professional antigen presenting cells that bind peptide fragments derived from protein antigens. X-ray crystallographic studies demonstrated that the MHC class II epitope binding site consists of a groove and several pockets provided by a β-sheet and two α-helices. Unlike class I, the class II binding groove is open at both ends. As a result, peptides binding to class II molecules tend to be of variable length, but typically between 13 and 25 residues.

A hallmark of the MHC class II binding peptide groove is that there are four major pockets. These pockets accommodate side chains of residues 1, 4, 6, and 9 of a 9-mer core region of the binding peptide. This core region interaction largely determines binding affinity and specificity [3]. In addition, peptide residues immediately flanking the core region have been indicated to make contact with the MHC molecule outside of the binding groove, and to contribute to MHC-peptide interaction. MHC class II molecules are highly polymorphic, and this polymorphism largely corresponds with differences along the peptide binding groove. However, the binding motifs derived for MHC class II molecules are highly degenerate, and many promiscuous peptides have been identified that can bind multiple MHC class II molecules. Promiscuous peptides are a prime target for vaccine and immunotherapy and computational tools have been developed to facilitate systematic scanning for promiscuous peptides.

Computational prediction of MHC class II epitopes is of important theoretical and practical value, as experimental dentification is costly and time consuming. Many computational methods exist to predict peptide MHC binding.

Experimentally determined affinities data have formed the basis of many peptide-MHC binding prediction methods and has enabled classification of binding and nonbinding peptides. Highly sophisticated computer science algorithms such as artificial neural networks [4], HMMs [5], support vector machines (SVMs) [6] and methods derived from computational chemistry, such as QSAR analysis [7] and structure-based approaches [8] have been previous deployed to solve this problem.

Dimitrov et al. [9] and Wang et al. [10] have published detailed assessment of MHC Class II peptide binding predictors. In the above mentioned assessment, it was clear that each predictor for a specific genetic loci (including its alleles) worked independently i.e., no knowledge sharing occured between predictors although they might be predicting alleles of the same genetic loci. In this paper, I propose that the prediction of MHC Class II alleles of the same genetic loci (HLA-DR*, HLA-DQ* and HLA-DP*) are dependent on each other since these alleles are nothing but an alternative form of the same gene. I tie pairs of MHC Class II alleles together and jointly learn the prediction/classifier parameters. This approach allows for transfer learning via high dimensional bayesian transfer hierarchies [11]. My results are then compared to the state of the art support vector machine methods [6], GLMNET [12] and I show that our joint classifier either matches or outperforms the SVM and GLMNET methods consistently.

II. Method

In the proposed method, pairs of MHC Class II alleles of the same genetic locus are paired together and I train a joint classifier on both to predict the binding and non-binding peptides. The following subsections describes the optimization setup and the algorithm of my method.

A. Optimization Setup

The optimization problem for logistic regression with a fused lasso penalty for a single predictor is given as

$$\underset{\mathbf{w}}{\text{minimize}} \sum_i \log\left\{1+\exp\left\{-y_i(\mathbf{x}_i\mathbf{w})\right\}\right\} + \lambda\left\|\mathbf{w}\right\|_1 + \mu\left\|\mathbf{D}\mathbf{w}\right\|_1. \tag{1}$$

where,

$$\mathbf{x},\mathbf{y},\mathbf{w}\in\mathbf{R}^n, \text{ and } \mathbf{D}=\begin{bmatrix} 1 & 0 & \cdots & 0 & -1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & -1 \end{bmatrix}$$

Since, we want to build a joint classifier for a pair of predictors and want to impose a constraint on the weights of both of their features, the optimization problem in (1) is modified as below:-

$$\begin{aligned}\underset{\mathbf{w}^1,\mathbf{w}^2}{\text{minimize}} &\sum_i \log\left\{1+\exp\left\{-y_i^1(\mathbf{x}_i^1\mathbf{w}^1)\right\}\right\} + \sum_i \log\left\{1+\exp\left\{-y_i^2(\mathbf{x}_i^2\mathbf{w}^2)\right\}\right\} \\ &+ \lambda_1\left\|\mathbf{w}^1\right\|_1 + \lambda_2\left\|\mathbf{w}^2\right\|_1 + \mu\left\|\mathbf{D}\mathbf{w}\right\|_1.\end{aligned} \tag{2}$$

where,

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}^1 & \mathbf{w}^2 \end{bmatrix}^\top$$

Equation (2) can be further simplified by packing variables $\mathbf{y}^1$ and $\mathbf{y}^2$ into $\mathbf{y}$, $\mathbf{x}^1$ and $\mathbf{x}^2$ into $\mathbf{x}$ and suitably modifying $\mathbf{D}$ as below:-

$$\underset{\mathbf{w}}{\text{minimize}} \sum_i \log\left\{1 + \exp\left\{-y_i(\mathbf{x}_i\mathbf{w})\right\}\right\} + \mu \left\|\mathbf{D}\mathbf{w}\right\|_1 . \tag{3}$$

where,

$$\mathbf{D} = \begin{bmatrix} \mu & 0 & \cdots & 0 & -\mu & 0 & \cdots & 0 \\ 0 & \mu & \cdots & 0 & 0 & -\mu & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mu & 0 & 0 & \cdots & -\mu \\ \lambda_1 & 0 & \cdots & 0 & \lambda_2 & 0 & \cdots & 0 \\ 0 & \lambda_1 & \cdots & 0 & 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_1 & 0 & 0 & \cdots & \lambda_2 \end{bmatrix}, \mathbf{y} = \begin{bmatrix} \mathbf{y}^1 \\ \mathbf{y}^2 \end{bmatrix}, \mathbf{x} = \begin{bmatrix} \mathbf{x}^1 & 0 \\ 0 & \mathbf{x}^2 \end{bmatrix} \text{ and } \mathbf{w} = \begin{bmatrix} \mathbf{w}^1 \\ \mathbf{w}^2 \end{bmatrix}$$

Writing out the augmented lagrangian for (3),

$$\begin{aligned} AL(\mathbf{w}, \mathbf{z}^0, \mathbf{z}^1, \mathbf{u}^0, \mathbf{u}^1) = &\sum_i \log\{1 + \exp\{-y_i z_i^0\}\} + \mu \left\|\mathbf{z}^1\right\|_1 + \\ &\sum_i u_i^0(z_i^0 - \mathbf{x}_i\mathbf{w}) + \mathbf{u}^1(\mathbf{z}^1 - \mathbf{D}\mathbf{w}) + \\ &\sum_i \frac{\rho}{2}\left\|z_i^0 - \mathbf{x}_i\mathbf{w}\right\|_2^2 + \frac{\rho}{2}\left\|\mathbf{z}^1 - \mathbf{D}\mathbf{w}\right\|_2^2 \end{aligned} \tag{4}$$

We can solve the above augmented lagrangian using alternating direction method of multipliers (ADMM) [15]. The ADMM algorithm iterates following updates,

$$\begin{aligned} \mathbf{w}^k &= \underset{\mathbf{w}}{\text{argmin}}\, AL(\mathbf{w}, \mathbf{z}^{0,k-1}, \mathbf{z}^{1,k-1}, \mathbf{u}^{0,k-1}, \mathbf{u}^{1,k-1}) \\ \mathbf{z}^{0,k} &= \underset{\mathbf{w}}{\text{argmin}}\, AL(\mathbf{w}^k, \mathbf{z}^0, \mathbf{z}^{1,k-1}, \mathbf{u}^{0,k-1}, \mathbf{u}^{1,k-1}) \\ \mathbf{z}^{1,k} &= \underset{\mathbf{w}}{\text{argmin}}\, AL(\mathbf{w}^k, \mathbf{z}^{0,k}, \mathbf{z}^1, \mathbf{u}^{0,k-1}, \mathbf{u}^{1,k-1}) \\ \mathbf{u}^{0,k} &= \mathbf{u}^{0,k-1} + \rho(\mathbf{z}^{0,k} - \mathbf{w}^k) \\ \mathbf{u}^{1,k} &= \mathbf{u}^{1,k-1} + \rho(\mathbf{z}^{0,k} - \mathbf{D}\mathbf{w}^k) \end{aligned} \tag{5}$$

where $k-1$ and k denote previous and current iteration respectively.

Now, we derive the updates for $\mathbf{w}^k, \mathbf{z}^{0,k}, \mathbf{z}^{1,k}$

1) Deriving update for $\mathbf{w}^k$: Eliminating terms that are constants with respect to $\mathbf{w}$ and completing the squares we obtain,

$$\mathbf{w}^k = \underset{\mathbf{w}}{\operatorname{argmin}} \frac{\rho}{2} \left\| \begin{bmatrix} \mathbf{x} \\ \mathbf{D} \end{bmatrix} \mathbf{w} - \begin{bmatrix} \mathbf{z}^{0,k-1} + \frac{1}{\rho}\mathbf{u}^{0,k-1} \\ \mathbf{z}^{1,k-1} + \frac{1}{\rho}\mathbf{u}^{1,k-1} \end{bmatrix} \right\|_2^2 \tag{6}$$

Thus, an update for $\mathbf{w}$ requires solving the above system of linear equations.

2) Deriving update for $\mathbf{z}^{0,k}$:

$$z_i^{0,k} = \underset{z_i^0}{\operatorname{argmin}} \log\{1 + exp\{-y_i z_i^0\}\} + \frac{\rho}{2} \left\| z_i^0 - (\mathbf{x}_i \mathbf{w}^k - \frac{1}{\rho} u_i^{0,k-1}) \right\|_2^2 \tag{7}$$

Since $\mathbf{z}^0$ is scalar, the objective is univariate. We can solve the above unconstrained optimization using Newton algorithm with backtracking and the weak Wolfe condition for backtracking termination [16].

3) Deriving update for $\mathbf{z}^{1,k}$:

$$z_i^{1,k} = \underset{z_i^1}{\operatorname{argmin}} \frac{\mu}{\rho} \left\| z_i^1 \right\|_1 + \frac{1}{2} \left\| z_i^1 - \mathbf{D}\mathbf{w}^k + \frac{1}{\rho} u_i^{1,k-1} \right\|_2^2 \tag{8}$$

The above optimization can be solved in a closed form

$$z_i^{1,k} = S(\mathbf{D}\mathbf{w}^k - \frac{1}{\rho} u_i^{1,k-1}, \frac{\mu}{\rho}) \tag{9}$$

where $\mathbf{S}$ is the Shrink Threshold operator defined as below,

$$S(x, \lambda) = sign(x)\ max(|x| - \lambda, 0) \tag{10}$$

The updates for the dual variables $\mathbf{u}^{0,k}$ and $\mathbf{u}^{1,k}$ are already listed in equation (5).

B. Algorithm

Putting all of the updates listed in Section II-A together, we can compute '$\mathbf{w}$' using the steps listed in algorithm 1. In this algorithm, due to the vast feature space, we will iterate only for a maximum of 100 iterations instead until convergence.

III. Experimental Results

The following subsections explain the dataset, features and the results of my method.

A. Dataset

I have used the MHC Class II dataset publicly made available by Wang et al [10] at [13]. For the purposes of my experiments, I have considered 15-mers of HLA-DRB1* MHC Class-II type and filtered out the rest. The detailed split of the dataset is shown in Table I. For each MHC Class-II type, I have used 70% of the data for training and 30% for testing. Within each training and testing set, 75% of the peptides are binding and 25% are non-binding peptides.

Algorithm 1 Fused ADMM Optimization Solver

INPUT: $\mathbf{y}$, $\mathbf{X}$, $\mathbf{D}$, λ_1 (lambda1), λ_2 (lambda2), μ (mu)

rho $\leftarrow 1$

N $\leftarrow$ length(y); p $\leftarrow$ size(X,2)

e $\leftarrow$ size(D,1); w $\leftarrow$ zeros(p,1)

z0 $\leftarrow$ zeros(N,1); u0 $\leftarrow$ zeros(N,1);

z1 $\leftarrow$ zeros(e,1); u1 $\leftarrow$ zeros(e,1);

MAXIT $\leftarrow$ 100;

n $\leftarrow$ size(X, 2);

for $it \leftarrow 1 \rightarrow MAXIT$ **do**

% computeAL method computes the augmented lagrangian value listed in equation (4)

$before \leftarrow computeAL(y, X, D, rho, w, z0, z1, u0, u1)$;

$w \leftarrow updatew(X, D, z0, z1, u0, u1, rho)$;

$after \leftarrow computeAL(y, X, D, rho, w, z0, z1, u0, u1)$;

$assert(after < before + 1e - 6)$;

$before \leftarrow after$;

for $i \leftarrow 1 \rightarrow length(y)$ **do**

% updatez0i calls a function which uses newton method with weak wolfe condition

$z0(i) \leftarrow updatez0i(y(i), X(i,:), w, u0(i), rho)$;

end for

$after \leftarrow computeAL(y, X, D, rho, w, z0, z1, u0, u1)$;

$assert(after < before + 1e - 6)$;

$before \leftarrow after$;

$Dw \leftarrow D * w$;

for $i \leftarrow 1 \rightarrow size(D, 1)$ **do**

$z1(i) \leftarrow updatez1i(Dw(i), u1(i), mu, rho)$;

end for

$after \leftarrow computeAL(y, X, D, rho, w, z0, z1, u0, u1)$;

$assert(after < before + 1e - 6)$;

$u0 \leftarrow u0 + rho * (z0 - X * w)$;

$u1 \leftarrow u1 + rho * (z1 - D * w)$;

end for

OUTPUT: w

TABLE I
MHC CLASS II DATASET

MHC Class II type	Total no. of peptides	Binding Peptides	Non-binding Peptides
HLA-DRB1*0301	502	368	134
HLA-DRB1*0401	512	389	123
HLA-DRB1*0404	449	370	79
HLA-DRB1*0405	457	375	82
HLA-DRB1*0701	505	402	103
HLA-DRB1*0901	412	304	108
HLA-DRB1*1101	520	400	120
HLA-DRB1*1302	289	202	87
HLA-DRB1*1501	520	388	132

B. Features

I use sparse encoding [14] for encoding the amino acid sequence. Each amino acid of the peptide sequence is mapped in a sparse orthonormal vector space by a 20-bit vector with 19 bits set to zero and one bit set to one. Therefore a peptide, which is a sequence of M consecutive amino acid letters can be represented by the concatenation of M x 20 features. In my experimental setup, I use 15-mers of HLA-DRB1* MHC Class-II type, hence each peptide can be represented by 15 x 20 = 300 feature length vector of 0s and 1s. These features alone were not sufficient to achieve a comparitive performance, hence, in addition to these I use interaction features dependent on pairs of positions. In a 15-mer, there are 210 pairs (15 x 14) of positions (x, y) where $1 \leq x \leq 15, 1 \leq y \leq 15$ and $x \neq y$. Each pair of positions can be occupied by any pair of 20 amino acids, thus, the total number of interaction features will be 84000 (15 x 14 x 20 x 20), in which one bit is set for a particular pair presence and 0 for absence. Hence, we have a total of 84300 (300 + 84000) features per peptide represented by a series of 0s and 1s.

C. Results

Table II specifies the precision, recall, f1-score and accuracy for the pairs of HLA-DRB1* MHC Class-II type alleles computed with my method (Fused ADMM method), SVM and GLMNET. My method learns the weights of features on the pair of alleles while SVM and GLMNET work independently on each of the pairs.

IV. CONCLUSION

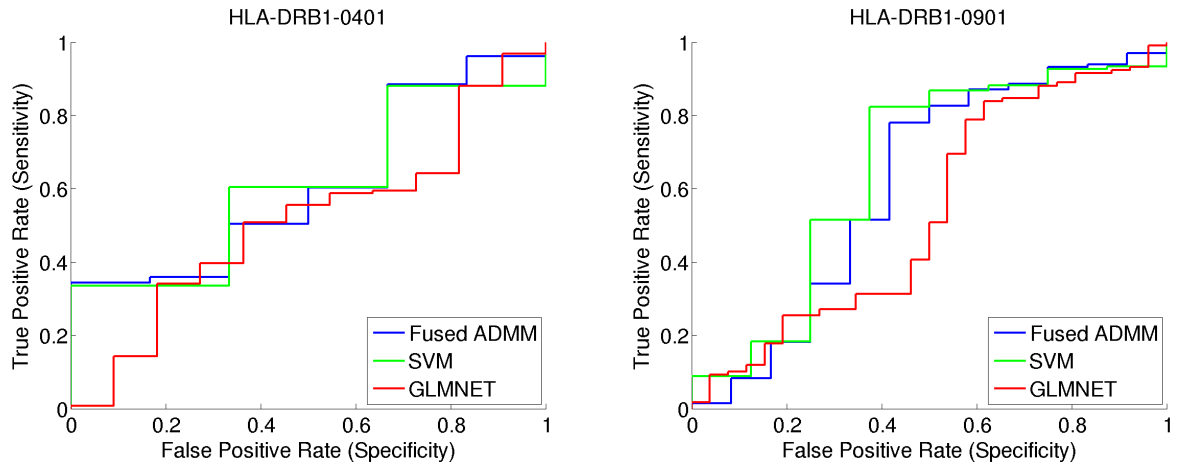
Section text here.

V. FUTURE WORK

Section text here.

VI. CONCLUSION

The conclusion goes here.



(a) (b)

Fig. 1. Prediction results of Fused ADMM, SVM and GLMNet methods for the pair (a) HLA DRB1*0401 and (b) HLA DR1*0901 are shown in the ROC curve. The curves were generated by plotting the true positive rate (y-axis) against the false positive rate (x-axis).

TABLE II

RESULTS OF BINDING AND NON-BINDING PEPTIDE PREDICTION

MHC Class-II Allele Pair	Precision			Recall			F1 Score			Accuracy		
	ADMM	SVM	GLM	ADMM	SVM	GLM	ADMM	SVM	GLM	ADMM	SVM	GLM
DRB1-0301	**0.71**	**0.71**	0.68	**0.90**	0.89	0.86	**0.80**	**0.80**	0.76	**0.70**	**0.70**	0.62
DRB1-0401	**0.73**	**0.73**	0.72	0.90	**0.91**	0.89	0.80	**0.81**	0.80	0.71	**0.72**	0.68
DRB1-0401	**0.73**	**0.73**	0.72	**0.92**	**0.92**	0.91	**0.82**	0.81	0.81	**0.72**	**0.72**	0.68
DRB1-0404	**0.77**	0.75	0.76	**0.92**	0.90	0.90	**0.84**	0.82	0.83	**0.76**	0.74	0.73
DRB1-0401	**0.73**	**0.73**	0.72	**0.92**	**0.92**	0.91	**0.82**	0.81	0.81	**0.72**	**0.72**	0.68
DRB1-0405	**0.83**	**0.83**	**0.83**	**0.92**	**0.92**	0.86	**0.87**	**0.87**	0.84	**0.83**	**0.83**	0.79
DRB1-0401	**0.73**	**0.73**	0.72	**0.92**	**0.92**	0.91	**0.82**	0.81	0.81	**0.72**	**0.72**	0.68
DRB1-0701	0.78	0.78	**0.80**	**0.92**	0.91	**0.92**	0.84	0.84	**0.85**	0.78	0.78	**0.79**
DRB1-0401	**0.73**	**0.73**	0.72	**0.92**	**0.92**	0.91	**0.82**	0.81	0.81	**0.72**	**0.72**	0.68
DRB1-0901	**0.72**	0.71	**0.72**	0.90	**0.92**	0.85	**0.80**	**0.80**	0.78	**0.71**	**0.71**	0.67
DRB1-0404	**0.77**	0.75	0.76	**0.92**	0.90	0.90	**0.84**	0.82	0.83	**0.76**	0.74	0.73
DRB1-0405	**0.83**	**0.83**	**0.83**	**0.92**	**0.92**	0.86	**0.87**	**0.87**	0.84	**0.83**	**0.83**	0.79
DRB1-0701	0.79	0.78	**0.80**	0.91	**0.92**	**0.92**	0.84	0.84	**0.85**	0.78	0.78	**0.79**
DRB1-0901	**0.73**	0.71	0.72	0.90	**0.92**	0.85	**0.80**	**0.80**	0.78	**0.72**	0.71	0.67
DRB1-0401	**0.73**	**0.73**	0.72	**0.92**	**0.92**	0.91	**0.82**	0.81	0.81	**0.72**	**0.72**	0.68
DRB1-1501	**0.72**	0.70	0.71	**0.92**	**0.92**	0.89	**0.80**	0.79	0.79	**0.72**	0.70	0.67
DRB1-1101	**0.79**	0.78	0.78	**0.92**	0.91	0.90	**0.85**	0.84	0.84	**0.80**	0.78	0.77
DRB1-1501	**0.72**	0.70	0.71	**0.92**	**0.92**	0.89	**0.80**	0.79	0.79	**0.72**	0.70	0.67

ACKNOWLEDGMENT

The authors would like to thank...

REFERENCES

[1] M. K. Jenkins, A. Khoruts, E. Ingulli, D. L. Mueller, S. J. McSorley, R. L. Reinhardt, A. Itano, and K. A. Pape, "In vivo activation of antigen-specific cd4 t cells," *Annual Review of Immunology*, vol. 19, no. 1, pp. 23–45, 2001.

[2] M. Rudolph, R. Stanfield, and I. Wilson, "How tcrs bind mhcs, peptides, and coreceptors," *Annual Review of Immunology*, vol. 24, no. 1, pp. 419–466, 2006.

[3] E. Y. Jones, L. Fugger, J. L. Strominger, and C. Siebold, "Mhc class ii proteins and disease: a structural perspective," *Nat Rev Immunol*, vol. 6, no. 4, pp. 271–282, Apr. 2006.

[4] K. Gulukota and C. DeLisi, "Neural network method for predicting peptides that bind major histocompatibility complex molecules," *Methods in Molecular Biology*, vol. 156, September 2000.

[5] H. Noguchi, R. Kato, T. Hanai, Y. Matsubara, H. Honda, V. Brusic, and T. Kobayashi, "Hidden markov model-based prediction of antigenic peptides that interact with mhc class ii molecules," *Journal of Bioscience and Bioengineering*, vol. 94, no. 3, pp. 264 – 270, 2002.

[6] J. Wan, W. Liu, Q. Xu, Y. Ren, D. Flower, and T. Li, "Svrmhc prediction server for mhc-binding peptides," *BMC Bioinformatics*, vol. 7, pp. 1–5, 2006.

[7] I. Doytchinova, V. Walshe, P. Borrow, and D. Flower, "Towards the chemometric dissection of peptide hla-a*0201 binding affinity: comparison of local and global qsar models," *Journal of Computer-Aided Molecular Design*, vol. 19, pp. 203–212, 2005.

[8] M. Davies, C. Hattotuwagama, D. Moss, M. Drew, and D. Flower, "Statistical deconvolution of enthalpic energetic contributions to mhc-peptide binding affinity," *BMC Structural Biology*, vol. 6, pp. 1–13, 2006.

[9] I. Dimitrov, P. Garnev, D. R. Flower, and I. Doytchinova, "Mhc class ii binding predictiona little help from a friend," *Journal of Biomedicine and Biotechnology*, 2010.

[10] P. Wang, J. Sidney, C. Dow, B. Moth, A. Sette, and B. Peters, "A systematic assessment of mhc class ii peptide binding predictions and evaluation of a consensus approach," *PLoS Computational Biology*, 2008.

[11] G. Elidan, B. Packer, G. Heitz, and D. Koller, "Convex point estimation using undirected bayesian transfer hierarchies," *CoRR*, vol. abs/1206.3252, 2008.

[12] J. H. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, vol. 33, no. 1, pp. 1–22, 2 2010.

[13] P. Wang, J. Sidney, C. Dow, B. Moth, A. Sette, and B. Peters, "Mhc class ii peptide data," http://mhcbindingpredictions.immuneepitope.org/MHCII/, 2008, online.

[14] N. Qian and T. J. Sejnowski, "Predicting the secondary structure of globular proteins using neural networks models," *J. Molecular Biology*, 1988.

[15] V. Jojic, "Admm fused lasso signal approximator," http://www.cs.unc.edu/~vjojic/comp790/notes/ADMMFLSA.pdf, 2012, online.

[16] J. Burke, "Line search methods," http://www.math.washington.edu/~burke/crs/408/notes/nlp/line.pdf, 2012, online.