Machine learning approaches to modeling the physiochemical properties of small peptides

Kyle Jensen, Mark Styczynski, and Gregory Stephanopoulos
Department of Chemical Engineering, Massachusetts Institute of Technology

***Abstract*— Peptide and protein sequences are most commonly represented as a strings: a series of letters selected from the twenty character alphabet of abbreviations for the naturally occurring amino acids. Here, we experiment with representations of small peptide sequences that incorporate more physiochemical information. Specifically, we develop three different physiochemical representations for a set of roughly 700 HIV–I protease substrates. These different representations are used as input to an array of six different machine learning models which predict whether or not a given peptide is likely to be an acceptable substrate for the protease. Our results show that, in general, higher–dimensional physiochemical representations tend to have better performance than representations incorporating fewer dimensions selected on the basis of high information content. We contend that such representations are more biologically relevant than simple string–based representations and are likely to more accurately capture peptide characteristics that are functionally important.**

***Index Terms*— Machine learning, peptide, modeling, physiochemical properties**

I. Introduction

IN this manuscript we discuss the modeling of small peptide sequences. Most commonly, peptides and protein sequences are represented as a string of letters drawn from the alphabet of characters representing the twenty natural amino acids. Here, we use a more meaningful representation of amino acids and test the ability of various machine learning techniques to predict peptide function. Specifically, we develop a set of three amino acid representation schemes and test these schemes combinatorially with a set of six machine learning techniques.

A. Amino acid representations

The most common representation of small peptides are as strings of letters representing the twenty amino acids, e.g. KWRAG, which is the five residue sequence lysine, tryptophan, arginine, alanine, and glycine. Notably, both amino acid names and their corresponding abbreviations are human constructs that carry no information about the underlying physiochemical characteristics of each amino acid. That is, the string KWRAG carries little information in and of itself, without some information about what a K is and how it is different from the other amino acids. In place of such physical descriptions, previous efforts have described the similarity of amino acids based on the tendency for one amino acid to substitute for another in homologous, similarly–functioning proteins across different species [1], [2]. That is, substitutions that are observed in nature can be construed in some sense as indicating similarity between certain amino acids. While such efforts have been extremely useful for tasks such as aligning more distant protein homologs, they typically do not capture enough information to be practically useful in *de novo* design or prediction of protein activity.

Here we experiment with feature vector representations of small peptides using sets of amino acid physiochemical characteristics derived from the AAindex database [3]–[5]. The AAindex database lists 453 physiochemical parameters for each of the twenty amino acids. These parameters range from those that are very tangible and intuitive — for example, residue volume, which is AAindex parameter BIGC670101 [6] — to the abstract — for example, the normalized frequency of participation in an N-terminal beta–sheet, which is AAindex parameter CHOP780208 [7]. The parameters were culled from the scientific literature by the AAindex authors and might be considered the universe of what we, as the scientific community, know about each amino acid.

Thus, a very logical way of representing an amino acid is as a feature vector of these 453 attributes. In this sense each type of amino acid has a different feature vector of the same dimensionality. This might be considered the "maximally informative" representation of the amino acids since it incorporates an expansive set of features culled from the literature. Extending this, we could write an amino acid sequence as the concatenation of these vectors. That is, a three residue peptide could be represented as a $3 * 453 = 1359$ feature vector. Intuitively, this representation retains more information than the string representation. Further, we would imagine that the physiochemical representation would be more useful for modeling the function of a peptide sequence, such as its propensity to fold in a certain manner or to react with a certain enzyme.

The representation of amino acids has received some previous attention in the literature. For example, Atchley *et. al.* [8] use the physiochemical parameters from the AAindex to create a low–dimensional projection of the characteristics of each of the twenty natural amino acids. Further, they used this low–dimensional progression to derive metrics of similarity between the amino acids, similar to popular amino acid scoring matrices such as Blosum [1] and PAM [2].

B. HIV–I Protease

In this work we will use the HIV–I protease as a model system for demonstrating the merits of different physiochemical

K.J. has been partially supported by SMA.
Manuscript received November 10, 2005.

amino acid representations. Specifically, we show the success of different representations and different machine learning methods at modeling substrate specificity of the protease.

The HIV–1 protease is a proteolytic enzyme encoded by the HIV genome [9]. The protease plays a critical role in viral replication and the development of viral structure [10]. The protease recognizes specific eight–residue sequences in its substrates (see Figures 1 and 2). The protease's natural targets are subsequences of other HIV genes which must be cleaved for the virus to successfully replicate. Accordingly, small molecule inhibitors of the protease are a common therapy for HIV/AIDS [11].

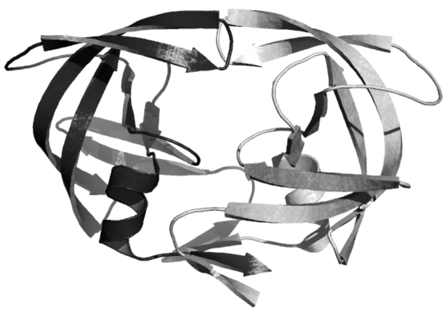

Fig. 1. Structure of the HIV–I protease, derived from the Protein Data Bank (PDB) [12] entry 7HVP [13]. Over one hundred other structures of the protease have been solved since the first in 1989 and are available from the PDB's website. The protein is a dimer of two 99 amino acid chains. The regions of the protein at the top of the figure, the "flaps," open up and accept a substrate protein, closing behind it. Two aspartate residues in the active site, aided by the presence of water, act to cleave the substrate.

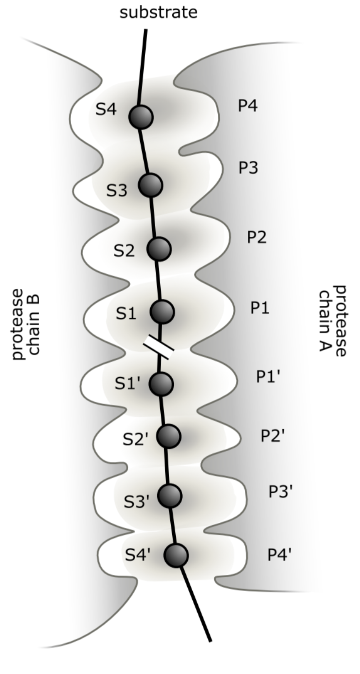

Fig. 2. Schematic of the HIV–I protease active site. The active site comprises eight binding pockets (P1–P4 and P1'–P4') into which eight residues from the target protein fall. The target protein is cleaved between the S1 and S1' residues. One half of the catalytic unit is made up by chain A of the protease and the other by chain B (see Figure 1).

In addition to the handful of sites that the protease cleaves to facilitate viral development, it can cleave a number of other "non–natural" substrates [14]. These substrates have been the focus of intense experimental study [15]–[18]. In a recent manuscript, You *et. al.* collected a comprehensive set of 700+ eight–residue substrates that have been tested for cleavability by the HIV–I protease [19]. In addition, You *et. al.* developed a series of models for the protease's substrate selectivity that, in general, outperform previous computational models [20]–[23], which relied on a much smaller dataset [24].

II. Methods

A. Amino acid representations and input data set

A set of 746 eight–residue peptides were generously provided by You *et. al.* [19], each with a class: cleavable by the HIV–I protease or not cleavable. In addition, the complete set of 453 physiochemical parameters for each of the 20 naturally occurring amino acids was downloaded from the AAindex database (release 7.0, July 2005).

From these 453 parameters, we removed redundant parameters for which the magnitude of the correlation coefficient with another parameter was greater than 0.80. The remaining 155 independent parameters were kept. Using these parameters, we made three different projections of the 746 experimentally tested protease substrates as detailed below.

1) Full physiochemical projection: In this projection each eight–residue peptide was represented as a 1241–dimensional feature vector: 8 residues with 155 physiochemical features per residue plus the class — cleaved or not cleaved. Of our three representations, this one retains the most information about the peptides.

2) Feature–selected physiochemical projection: Using the "FULL" projection (above) we performed a feature selection routine to select only those features that are most correlated to the class. (Throughout this manuscript, all modeling and feature selection were performed using the Waikato Environment for Knowledge Analysis, or WEKA [25]). Briefly, we evaluated the worth of a subset of features by considering the individual predictive ability of each feature with respect to the cleaved/uncleaved class, along with the degree of redundancy between the features. Using this method, we created a 54–dimensional projection of the peptide substrates (53 features plus the class).

Analysis of this lower–dimensional projection revealed that the features of the outer residues (S4, S4') are relatively unimportant, whereas the central residues (S1, S1') are quite important in determining cleavability. For the S1 position, seven parameters were chosen:

- FASG760102: Melting point [26];
- FAUJ880105: Minimum width of the side chain [27];
- PALJ810111: Normalized frequency of beta–sheet in alpha+beta class [28];
- PRAM900101: Hydrophobicity [29];
- ROBB760107: Information measure for extended without H-bond [30];
- KOEP990101: Alpha–helix propensity derived from designed sequences [31]; and
- MITS020101: Amphiphilicity index [32].

3) PCA projection of physiochemical properties: Using the full, 155–dimensional representation of each of the 20 naturally occurring amino acids, we performed principal component analysis (PCA) to find linear combinations of features that capture the variation between different kinds of amino acids. More formally, PCA, or the Karhunen–Loève transform, is a linear transformation by which the 20 data points in a 155–dimensional space are projected onto a new coordinate system. The system is chosen such that the greatest variance is captured by the first axis, or the first "principal component." Successive principal components (axes) capture progressively less variance. Each component is a linear combination of some of the initial features; given appropriate uniform normalization, the weight of each feature in a given component indicates the relative importance of that feature in defining the component.

Using PCA, we derived 16 principal components that capture 95% of the variance in the amino acids, with the first PC capturing 30% of the variance. The set of 746 peptide 8–mers were projected into a reduced 129–dimensional space: 8 concatenated 16–dimensional residues plus the class of the peptide.

B. Model creation and classification

For each of the three peptide representations detailed above, we tested the ability of six machine learning techniques to classify the peptides as either cleaved or uncleaved. Each of these models is described below. For each model, we evaluated the performance using 10x10 cross–validation (see Conclusion): for each of ten runs, 10% of the peptide dataset was withheld for testing a classifier trained by the remaining 90% of the peptides. The sensitivity and specificity of each classifier's predictions for all ten of its cross–validation runs can then be combined to determine the percentage of correctly classified peptides. This value is used to quantify the classifier's overall accuracy and facilitates pairwise comparison of models and representation schemes.

1) Decision tree model: Decision trees are simple, intuitive classification schemes that use a series of questions (decisions) to place a sample in a class with low error rate. More specifically, a decision tree is a structure in which the internal branches represent conditions, such as "hydrophobicity index at S3 > 0.52". Following these conditions leads to the leaves of the tree, which are classifications indicating whether the peptide is cleaved or not. Here, we use a particular variant of the decision tree, a C4.5 decision tree [33], which is notable for not being prone to overfitting of input data. An example decision tree from our experiments is shown in Figure 3.

```
CHOP780207_S2′ <= 0.41765
|   FAUJ880105_S1 <= 0.57778
|   |   FASG760102_S1 <= 0.27711: uncleaved (32.0/1.0)
|   |   FASG760102_S1 > 0.27711
|   |   |   QIAN880122_S4′ <= 0.81022
|   |   |   |   PRAM900101_S1 <= 0.27463
|   |   |   |   |   MEEJ810102_S4 <= 0.33702
|   |   |   |   |   |   RACS820112_S2 <= 0.58621
|   |   |   |   |   |   |   ZIMJ680101_S1′ <= 0.52117
|   |   |   |   |   |   |   |   PRAM820101_S2′ <= 0.43367
|   |   |   |   |   |   |   |   |   ROSM880103_S3′ <= 0.23077: cleaved (2.0)
|   |   |   |   |   |   |   |   |   ROSM880103_S3′ > 0.23077
|   |   |   |   |   |   |   |   |   |   CHOP780207_S4 <= 0.21176: cleaved (2.0)
|   |   |   |   |   |   |   |   |   |   CHOP780207_S4 > 0.21176: uncleaved (11.0/1.0)
|   |   |   |   |   |   |   |   PRAM820101_S2′ > 0.43367
|   |   |   |   |   |   |   |   |   RADA880105_S2 <= 0.75274
|   |   |   |   |   |   |   |   |   |   PRAM900101_S1 <= 0.06866: cleaved (10.0/2.0)
|   |   |   |   |   |   |   |   |   |   PRAM900101_S1 > 0.06866: uncleaved (4.0)
|   |   |   |   |   |   |   |   |   RADA880105_S2 > 0.75274
|   |   |   |   |   |   |   |   |   |   QIAN880137_S3′ <= 0.5124: cleaved (69.0/3.0)
|   |   |   |   |   |   |   |   |   |   QIAN880137_S3′ > 0.5124
|   |   |   |   |   |   |   |   |   |   |   RACS820112_S2 <= 0.43103: cleaved (2.0)
|   |   |   |   |   |   |   |   |   |   |   RACS820112_S2 > 0.43103: uncleaved (4.0/1.0)
|   |   |   |   |   |   |   ZIMJ680101_S1′ > 0.52117: cleaved (248.0/7.0)
|   |   |   |   |   |   RACS820112_S2 > 0.58621
|   |   |   |   |   |   |   RACS820103_S4 <= 0.43007
|   |   |   |   |   |   |   |   CHAM830104_S3′ <= 0
|   |   |   |   |   |   |   |   |   RADA880105_S2 <= 0.75274: uncleaved (5.0/1.0)
|   |   |   |   |   |   |   |   |   RADA880105_S2 > 0.75274: cleaved (2.0)
|   |   |   |   |   |   |   |   CHAM830104_S3′ > 0: cleaved (11.0)
|   |   |   |   |   |   |   RACS820103_S4 > 0.43007: uncleaved (6.0)
|   |   |   |   |   MEEJ810102_S4 > 0.33702
|   |   |   |   |   |   GARJ730101_S4′ <= 0.01426: uncleaved (9.0)
|   |   |   |   |   |   GARJ730101_S4′ > 0.01426
|   |   |   |   |   |   |   CHAM830104_S3′ <= 0
|   |   |   |   |   |   |   |   QIAN880102_S4 <= 0.57143: uncleaved (7.0/1.0)
|   |   |   |   |   |   |   |   QIAN880102_S4 > 0.57143: cleaved (3.0)
|   |   |   |   |   |   |   CHAM830104_S3′ > 0: cleaved (9.0)
|   |   |   |   PRAM900101_S1 > 0.27463
|   |   |   |   |   GEIM800106_S1′ <= 0.94
|   |   |   |   |   |   RACS820102_S3 <= 0.81522
|   |   |   |   |   |   |   FAUJ880108_S2′ <= 0.4375: uncleaved (31.0/1.0)
|   |   |   |   |   |   |   FAUJ880108_S2′ > 0.4375: cleaved (4.0/1.0)
|   |   |   |   |   |   RACS820102_S3 > 0.81522: cleaved (6.0)
|   |   |   |   |   GEIM800106_S1′ > 0.94: cleaved (9.0)
|   |   |   QIAN880122_S4′ > 0.81022
|   |   |   |   MITS020101_S1 <= 0.35354
|   |   |   |   |   ZIMJ680101_S1′ <= 0.82085: uncleaved (20.0)
|   |   |   |   |   ZIMJ680101_S1′ > 0.82085
|   |   |   |   |   |   RACS820102_S3 <= 0.3587: uncleaved (4.0)
|   |   |   |   |   |   RACS820102_S3 > 0.3587: cleaved (5.0)
|   |   |   |   MITS020101_S1 > 0.35354: cleaved (2.0)
|   FAUJ880105_S1 > 0.57778
|   |   QIAN880137_S3′ <= 0: cleaved (3.0)
|   |   QIAN880137_S3′ > 0: uncleaved (37.0/1.0)
CHOP780207_S2′ > 0.41765
|   ZIMJ680101_S1′ <= 0.58306: uncleaved (145.0/2.0)
|   ZIMJ680101_S1′ > 0.58306
|   |   PRAM900101_S1 <= 0.27463
|   |   |   FAUJ880105_S1 <= 0.57778
|   |   |   |   FAUJ880105_S1 <= 0: uncleaved (2.0)
|   |   |   |   FAUJ880105_S1 > 0
|   |   |   |   |   RACS820103_S3 <= 0.72378
|   |   |   |   |   |   WILM950104_S2 <= 0.44834: uncleaved (5.0)
|   |   |   |   |   |   WILM950104_S2 > 0.44834
|   |   |   |   |   |   |   PRAM820101_S2′ <= 0.77041: cleaved (8.0)
|   |   |   |   |   |   |   PRAM820101_S2′ > 0.77041: uncleaved (4.0/1.0)
|   |   |   |   |   RACS820103_S3 > 0.72378: cleaved (9.0)
|   |   |   FAUJ880105_S1 > 0.57778: uncleaved (4.0)
|   |   PRAM900101_S1 > 0.27463: uncleaved (12.0)
```

Fig. 3. The decision tree calculated for the CFS, a 54–dimensional representation of the 8–mer peptides. The branch points are in the form PARAMETER_RESIDUE. For example, CHOP780207_S2′ represents the AAindex parameter CHOP780207 (normalized frequency of participation in a C–terminal non–helical region) at the S2' residue. Values for all AAindex parameters are normalized to 1 across all amino acids. The tree shows various questions about a peptide that, when followed, lead to a set of conclusions. For example, if a given peptide has CHOP780207_S2 <= 0.41765 and FAUJ880105_S1 > 0.57778 and QIAN880137_S3 > 0 then the peptide is classified as uncleaved. As shown in the table, 37 of the 746 known peptides are correctly classified by this scheme and only one is incorrectly classified.

2) Logistic regression model: A logistic regression is just a non–linear transformation of a linear regression. In this model, each independent variable (the different dimensions of our various projections) are regressed to the class (cleaved or not cleaved). Here we use a variant of logistic regression that leads to automated feature selection and is described elsewhere [34].

3) Bayesian network model: Bayesian network models use directed acyclic graphs to model the joint probability distribution of each class over all input features. That is, the model captures conditional dependencies between the features with regards to how they impact the final classification of each sample. Bayesian networks can be used to find causality relationships, one of many features that make these models particularly well–suited to many applications in computational biology (see, for example, [35]–[37]). The method uses a Bayesian scoring metric that ranks multiple models based on their ability to explain data with the simplest possible method. The Bayesian metric is a function of the probability of the model being correct given a set of observed data; this is, in turn, correlated to the model's prior probability and its physical likelihood. For a more detailed explanation of Bayesian networks, see Witten and Frank [25] or Heckerman [38].

4) Naive Bayes model: The naive Bayes model, or “Idiot’s” Bayes model [39], is a simple machine learning scheme that assumes *naively* that each feature has an independent effect on the classification of each sample [40]. In the case of the HIV–I protease substrates, this means that the physiochemical characteristics of the S1 residue contribute to the cleavability of the peptide in a way that is independent of the other residues: S1’, S2, etc. The resulting network dependencies are less complex than one might otherwise obtain from a Bayesian network model but are frequently useful, particularly for unwieldy datasets or problems with physical characteristics that may warrant the assumption of conditional independence of features.

5) Support vector machine model with linear basis function: The support vector machine (SVM) is a machine learning technique posed as a quadratic programming (QP) problem [41]. The formulation can best be conceptualized by considering the problem of classifying two linearly separable groups of points. The first step is to define the “convex hull” of each group, which is the smallest–area convex polygon that completely contains a group. The SVM approach looks for the best linear classifier (single straight line) between the two groups of points, defined as either the line that bisects the two closest points on each convex hull or the two parallel planes tangent to each convex hull that are furthest apart. These alternative definitions provide two alternative formulations of a convex QP problem; notably, they both reduce to the same problem. (A rigorous mathematical treatment of these qualitative explanations can be found elsewhere [42], [43].) Tried and true methods for solving QP problems can then be used to (relatively quickly) determine the best classifier. This method can be expanded to allow for linearly inseparable cases by altering the optimization problem to account for a weighted cost of misclassification when training the model. There is evidence in the literature that an SVM approach to defining the best classifier is less susceptible to overfitting and generalization error [44]–[46].

6) Support vector machine model with radial basis function: The above description of an SVM, despite accounting for the possibility of inseparability, does not address the need for non–linear classifiers. For instance, if the members of one class fall within a well–defined circle and the non–members fall outside of the circle, the above method will perform extremely poorly because it will try to form just one plane to separate the groups [41]. Rather than attempting to fit higher–order curves, it is easier to project the input attributes into a higher–dimensional space in which the groups are (approximately) linearly separable. The higher–dimensional spaces can be characteristic of any desired classifier (e.g., nonlinear terms generated by multiplying attributes or squaring attributes). The same method for computing the best linear classifier is then used. The result is mapped back into attribute space of the appropriate dimensions and constitutes a non–linear classifier. Though one may expect such a process to be prohibitively expensive for data with many attributes, there exists a computational shortcut using “kernel functions” to avoid calculating all possible higher–dimensional feature values. In this work, the basis function for the kernel gives us the ability to detect optimal classifiers that are based upon training points’ radius from some center point (as in the above example).

III. Conclusion

Our results show that the full, 1241–dimensional representation performed the best, followed by the PCA representation and, finally, the representation made via feature selection. (See Figure 4 and Table III & IV. In these tables “FULL” is the full physiochemical, 1241–dimensional representation; “CFS” is the feature–selected, 55–dimensional representation; and “PCA” is the 129–dimensional representation created using principal component analysis.)

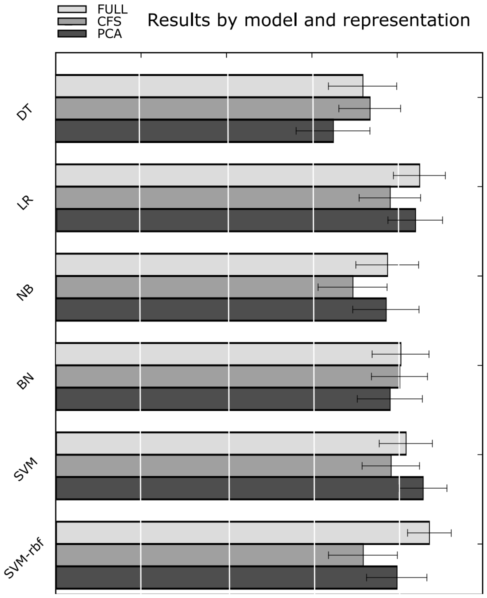

Fig. 4. Classification results for all amino acid representations and model types. The three different amino acid representations are shown in shades of gray: “FULL” is the full physiochemical, 1241–dimensional representation; “CFS” is the feature–selected, 55–dimensional representation; and “PCA” is the 129–dimensional representation using created using princple component analysis (see text). Error bars show the standard deviation over the 10x10 cross–validation test (100 samples per representation/model combination with a total of 1800 tests.) The best performing model was the SVM with radial basis function (SVM–rbf in the figure) with the full 1241–dimensional feature vector representing each eight–residue sequence. Averaged over all representations, the logistic regression model is best (see Table I). The poorest performing model is the decision tree (DT) with the 129–dimensional feature vector created using the PCA projections created as described in the text. In general the full 1241–dimensional representation performed the best, followed by the PCA representation and finally the CFS representation, which was created by a feature selection process.

Of the models tested, results show that logistic regression is the best, followed by (linear basis function) SVMs and Bayesian networks (See Figure 4 and Table I & II.) The single best model/representation combination was the SVM model with radial basis function (SVM–rbf) and the FULL representation. It is worth noting that though this single combination was the best, the radial basis function SVM itself did not perform consistently well. Though this may not have been expected, it is definitely reasonable per the “No Free

TABLE I

MODEL COMPARISON

	DT	LR	NB	BN	SVM	SVM–rbf
DT	-	2	1	3	2	2
LR	0	-	0	0	0	0
NB	0	3	-	1	2	1
BN	0	1	0	-	1	1
SVM	0	0	0	0	-	1
SVM–rbf	0	2	0	1	2	-

Each i, j entry represents the number of representations, out of three, for which the i model performed *worse* than the j model. Here "worse" means that the model had a statistically significant lower performance, based on a two–tailed t–test at the 0.05 confidence level.

TABLE II

MODEL RANKING

total wins	total losses	model
8	0	LR
7	1	SVM
5	3	BN
5	5	SVM–rbf
1	7	NB
0	10	DT

Each row shows, for each model, how many other model/representation pairs that model (with any representation) "wins" against. (Thus, the max of the sum of the columns in any row is $18 - 3 = 15$; however, ties are not shown.) Here "win/loss" means that the model had a statistically significant higher/lower performance, based on a two–tailed t–test at the 0.05 confidence level.

Lunch" theorem: no single machine–learning method should be expected to perform the best in all cases [47].

In general, these results suggest that higher–dimensional physiochemical representations tend to have better performance than representations incorporating fewer dimensions selected on the basis of high information content. As such, it seems that as long as the training set is a reasonable size, more accurate classifiers can be constructed by keeping as many significant input attributes as possible. Though methods like principal components analysis help to reduce computational complexity for unwieldy datasets, it is better to avoid feature selection until a supervised method (like the models tested in this work) can determine which features are most important in classifying samples.

TABLE III

REPRESENTATION COMPARISON

	FULL	CFS	PCA
FULL	-	0	1
CFS	3	-	4
PCA	2	1	-

Each i, j entry represents the number of models, out of six, for which the i representation performed *worse* than the j representation. Here "worse" means that the representation had a statistically significant lower performance, based on a two–tailed t–test at the 0.05 confidence level.

TABLE IV

REPRESENTATION RANKING

5	1	FULL
5	3	PCA
1	7	CFS

Each row shows, for each representation, how many other model/representation pairs that representation (with any model) "wins" against. (Thus, the max of the sum of the columns in any row is $18 - 6 = 12$; however, ties are not shown.) Here "win/loss" means that the representation had a statistically significant higher/lower performance, based on a two–tailed t–test at the 0.05 confidence level.

REFERENCES

[1] S. Henikoff and J. G. Henikoff, "Amino acid substitution matrices from protein blocks," *Proc Natl Acad Sci U S A*, vol. 89, pp. 10915–10919, Nov 1992.

[2] M. O. Dayhoff, R. M. Schwartz, and B. C. Orcutt, "A model of evolutionary change in proteins," in *Atlas of Protein Structure* (M. O. Dayhoff, ed.), vol. 5(Suppl. 3), pp. 345–352, Silver Spring, Md.: National Biomedical Reasearch Foundataion, 1978.

[3] S. Kawashima, H. Ogata, and M. Kanehisa, "AAindex: Amino Acid Index Database," *Nucleic Acids Res*, vol. 27, pp. 368–9, Jan 1999.

[4] K. Tomii and M. Kanehisa, "Analysis of amino acid indices and mutation matrices for sequence comparison and structure prediction of proteins," *Protein Eng*, vol. 9, pp. 27–36, Jan 1996.

[5] K. Nakai, A. Kidera, and M. Kanehisa, "Cluster analysis of amino acid indices for prediction of protein structure and function," *Protein Eng*, vol. 2, pp. 93–100, Jul 1988.

[6] C. C. Bigelow, "On the average hydrophobicity of proteins and the relation between it and protein structure," *J Theor Biol*, vol. 16, pp. 187–211, Aug 1967.

[7] P. Y. Chou and G. D. Fasman, "Prediction of the secondary structure of proteins from their amino acid sequence," *Adv Enzymol Relat Areas Mol Biol*, vol. 47, pp. 45–148, 1978.

[8] W. R. Atchley, J. Zhao, A. D. Fernandes, and T. Drüke, "Solving the protein sequence metric problem," *Proc Natl Acad Sci U S A*, vol. 102, pp. 6395–400, May 2005.

[9] A. Brik and C. Wong, "HIV-1 protease: mechanism and drug discovery," *Org Biomol Chem*, vol. 1, pp. 5–14, Jan 2003.

[10] W. Wang and P. A. Kollman, "Computational study of protein specificity: the molecular basis of HIV-1 protease drug resistance," *Proc Natl Acad Sci U S A*, vol. 98, pp. 14937–42, Dec 2001.

[11] D. Boden and M. Markowitz, "Resistance to human immunodeficiency virus type 1 protease inhibitors," *Antimicrob Agents Chemother*, vol. 42, pp. 2775–2783, Nov 1998.

[12] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The Protein Data Bank," *Nucleic Acids Res*, vol. 28, pp. 235–242, Feb 2000.

[13] A. L. Swain, M. M. Miller, J. Green, D. H. Rich, J. Schneider, S. B. Kent, and A. Wlodawer, "X-ray crystallographic structure of a complex between a synthetic protease of human immunodeficiency virus 1 and a substrate-based hydroxyethylamine inhibitor," *Proc Natl Acad Sci U S A*, vol. 87, pp. 8805–9, Nov 1990.

[14] Z. Beck, G. Morris, and J. Elder, "Defining HIV-1 protease substrate selectivity," *Curr Drug Targets Infect Disord*, vol. 2, pp. 37–50, Mar 2002.

[15] Z. Beck, L. Hervio, P. Dawson, J. Elder, and E. Madison, "Identification of efficiently cleaved substrates for HIV-1 protease using a phage display library and use in inhibitor development," *Virology*, vol. 274, pp. 391–401, Sep 2000.

[16] P. Bagossi, T. Sperka, A. Fehér, J. Kádas, G. Zahuczky, G. Miklóssy, P. Boross, and J. Tözsér, "Amino acid preferences for a critical substrate binding subsite of retroviral proteases in type 1 cleavage sites," *J Virol*, vol. 79, pp. 4213–4218, Apr 2005.

[17] Z. Beck, Y. Lin, and J. Elder, "Molecular basis for the relative substrate specificity of human immunodeficiency virus type 1 and feline immunodeficiency virus proteases," *J Virol*, vol. 75, pp. 9458–9469, Oct 2001.

[18] J. C. Clemente, R. E. Moose, R. Hemrajani, L. R. S. Whitford, L. Govindasamy, R. Reutzel, R. McKenna, M. Agbandje-McKenna, M. M. Goodenow, and B. M. Dunn, "Comparing the accumulation of

active- and nonactive-site mutations in the HIV-1 protease," *Biochemistry*, vol. 43, pp. 12141–51, Sep 2004.

[19] L. You, D. Garwicz, and T. Rögnvaldsson, "Comprehensive bioinformatic analysis of the specificity of human immunodeficiency virus type 1 protease," *J Virol*, vol. 79, pp. 12477–86, Oct 2005.

[20] Y.-D. Cai, X.-J. Liu, X.-B. Xu, and K.-C. Chou, "Support Vector Machines for predicting HIV protease cleavage sites in protein," *J Comput Chem*, vol. 23, pp. 267–274, Jan 2002.

[21] K. C. Chou, "Prediction of human immunodeficiency virus protease cleavage sites in proteins," *Anal Biochem*, vol. 233, pp. 1–14, Jan 1996.

[22] A. Narayanan, X. Wu, and Z. R. Yang, "Mining viral protease data to extract cleavage knowledge," *Bioinformatics*, vol. 18 Suppl 1, pp. S5–13, 2002.

[23] T. Rognvaldsson and L. You, "Why neural networks should not be used for HIV-1 protease cleavage site prediction," *Bioinformatics*, vol. 20, pp. 1702–1709, Jul 2004.

[24] Y. D. Cai, H. Yu, and K. C. Chou, "Artificial neural network method for predicting HIV protease cleavage sites in protein," *J Protein Chem*, vol. 17, pp. 607–15, Oct 1998.

[25] I. H. Witten and E. Frank, *Data mining: practical machine learning tools and techniques with Java implementations*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005.

[26] G. Fasman, ed., *Physical Chemical Data*, vol. 1 of *CRC Handbook of Biochemistry and Molecular Biology*. Cleveland, Ohio: CRC Press, 1976.

[27] J. L. Fauchère, M. Charton, L. B. Kier, A. Verloop, and V. Pliska, "Amino acid side chain parameters for correlation studies in biology and pharmacology," *Int J Pept Protein Res*, vol. 32, pp. 269–78, Oct 1988.

[28] J. Palau, P. Argos, and P. Puigdomenech, "Protein secondary structure. Studies on the limits of prediction accuracy," *Int J Pept Protein Res*, vol. 19, pp. 394–401, Apr 1982.

[29] M. Prabhakaran, "The distribution of physical, chemical and conformational properties in signal and nascent peptides," *Biochem J*, vol. 269, pp. 691–6, Aug 1990.

[30] B. Robson and E. Suzuki, "Conformational properties of amino acid residues in globular proteins," *J Mol Biol*, vol. 107, pp. 327–56, Nov 1976.

[31] P. Koehl and M. Levitt, "Structure-based conformational preferences of amino acids," *Proc Natl Acad Sci U S A*, vol. 96, pp. 12524–9, Oct 1999.

[32] S. Mitaku, T. Hirokawa, and T. Tsuji, "Amphiphilicity index of polar amino acids as an aid in the characterization of amino acid preference at membrane-water interfaces," *Bioinformatics*, vol. 18, pp. 608–16, Apr 2002.

[33] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann, 1992.

[34] N. Landwehr, M. Hall, and E. Frank, *Logistic model trees*, vol. 2837 of *Lecture Notes in Artificial Intelligence*, pp. 241–252. Springer–Verlag, 2003.

[35] M. S. Scott, D. Y. Thomas, and M. T. Hallett, "Predicting subcellular localization via protein motif co-occurrence," *Genome Res*, vol. 14, pp. 1957–66, Oct 2004.

[36] A. J. Hartemink, D. K. Gifford, T. Jaakkola, and R. A. Young, "Bayesian methods for elucidating genetic regulatory networks.," *IEEE Intelligent Systems*, vol. 17, no. 2, pp. 37–43, 2002.

[37] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using bayesian networks to analyze expression data," in *4th Annual International Conference on Computational Molecular Biology (RECOMB 2000)*, pp. 127–135, Apr 2000.

[38] D. Heckerman, "A tutorial on learning with bayesian networks," 1995.

[39] D. J. Hand and K. Yu, "Idiot's bayes – not so stupid after all?," *International Statistical Review*, vol. 69, no. 3, pp. 385–399, 2001.

[40] G. H. John and P. Langley, "Estimating continuous distributions in Bayesian classifiers," in *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pp. 338–345, 2005.

[41] K. P. Bennett and C. Campbell, "Support vector machines: Hype or hallelujah?," *SIGKDD Explorations*, vol. 2, no. 2, pp. 1–13, 2000.

[42] K. Bennett and E. Bredensteiner, "Duality and gemoetry in svms," in *Proc. of 17th international Conference on Machine Learning* (P. Langley, ed.), pp. 65–72, Morgan Kaufmann, 2000.

[43] D. J. Crisp and C. J. C. Burges, "A geometric interpretation of v-svm classifiers.," in *NIPS*, pp. 244–250, 1999.

[44] N. Cristianini and J. Shawe-Taylor, *An introduction to support Vector Machines: and other kernel-based learning methods*. New York, NY, USA: Cambridge University Press, 2000.

[45] V. N. Vapnik, *Statistical learning theory*. Wiley, 1998. VAP v 98:1 1.Ex.

[46] V. N. Vapnik, *The nature of statistical learning theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995.

[47] D. H. Wolpert and W. G. Macready, "No free lunch theorems for search," Tech. Rep. SFI-TR-95-02-010, Santa Fe, NM, 1995.