BIOINFORMATICS

Vol. 19 no. 1 2003
Pages 98–107

Predicting HIV drug resistance with neural networks

Sorin Drăghici* and R. Brian Potter

Department of Computer Science, 431 State Hall, Wayne State University, Detroit, MI 48202, USA

Received on January 25, 2002; revised on May 16, 2002; June 28, 2002; accepted on July 3, 2002

ABSTRACT

Motivation: Drug resistance is a very important factor influencing the failure of current HIV therapies. The ability to predict the drug resistance of HIV protease mutants may be useful in developing more effective and longer lasting treatment regimens.

Methods: The HIV resistance is predicted to two current protease inhibitors, Indinavir and Saquinavir. The problem was approached from two perspectives. First, a predictor was constructed based on the structural features of the HIV protease–drug inhibitor complex. A particular structure was represented by its list of contacts between the inhibitor and the protease. Next, a classifier was constructed based on the sequence data of various drug resistant mutants. In both cases, self-organizing maps were first used to extract the important features and cluster the patterns in an unsupervised manner. This was followed by subsequent labelling based on the known patterns in the training set.

Results: The prediction performance of the classifiers was measured by cross-validation. The classifier using the structure information correctly classified previously unseen mutants with an accuracy of between 60 and 70%. Several architectures were tested on the more abundant sequence data. The best single classifier provided an accuracy of 68% and a coverage of 69%. Multiple networks were then combined into various majority voting schemes. The best combination yielded an average of 85% coverage and 78% accuracy on previously unseen data. This is more than two times better than the 33% accuracy expected from a random classifier.

Contact: Sorin Drăghici; sod@cs.wayne.edu.

INTRODUCTION

Drug resistance is probably the most important factor influencing the failure of present HIV therapies. The emergence of anti-retroviral drug resistance is not unexpected, as drug resistance had been reported for other viruses such as herpes simplex, varicella-zoster, cytomegalovirus, influenza A and rhinovirus. However, the drug resistance problem is far more important in the case of the HIV virus because of the dramatic final outcome of HIV-related illnesses.

One of the main mechanisms behind drug action and drug resistance is believed to be specific molecular recognition in the binding pocket. This study included both mutations directly related to the binding pocket as well as mutations distant from it, as long as such mutations were shown in the literature to be associated with drug resistance.

First, the mechanisms are examined that allow the HIV virus to develop drug resistance to the FDA-approved protease inhibitor Indinavir. The structural changes that characterize drug resistant protease mutants are studied in order to understand the effect that various structural changes have upon drug resistance. The results suggest that the drug resistance phenomenon is associated with a loss of contacts between the drug and the target viral enzyme. A further observation is that different point mutations may lead to similar structural changes in the active site.

This study also investigated the resistance of HIV protease mutants to Saquinavir (another FDA approved protease inhibitor). No attempt is made to understand the mechanism or reasons why certain mutations are or are not resistant to Saquinavir, but rather to predict such resistance based solely on the amino acid sequence of HIV protease mutants. A small number of these mutants have reported Saquinavir IC90 values, which were used to classify the resistance of the mutants tested.

Background

The HIV protease is an aspartyl protease which has only 99 residues. Retroviral proteases have a dimeric structure which is composed of two identical monomers (see Figure 1). The three dimensional structure of the HIV protease contains primarily β-sheet, turn and extended structural elements. The structure is unusual in that the dimer has only one active site. Each monomer contributes one of the two aspartyl residues within the Asp–Thr–Gly sequences of the active site. The HIV

*To whom correspondence should be addressed.

protease active site is located in a cleft into which the polypeptide to be cleaved is positioned. Substrate or inhibitor binding to the protease induces a large conformational change. The flaps of the protease move as much as 15 Å when the ligand is bound. In protease inhibitor design, the scissile bond of the substrate is often replaced by stable isosteres such as statine or a reduced amide. A network of hydrogen bonds and Van der Waals contacts is formed between the protease and the inhibitor.

During HIV replication, a single DNA molecule is synthesized, which is integrated into the host genome. Polyprotein precursors are cleaved by the HIV protease. This mechanism suggests that the replication of the virus can be stopped or drastically reduced if the activity of the protease is blocked. This has led to the development of a class of drugs known as protease inhibitors. The protease inhibitors bind to the protease in the active site and block its activity. Figure 1 presents the structure of an inhibitor-protease complex. Another enzyme that has an essential role in the viral replication is the reverse transcriptase (RT). Modern combination therapies attack the virus with a combination of one protease inhibitor and two RT inhibitors. The work presented here has concentrated on the study of the protease inhibitors and of the mutants that occur during the treatment with such inhibitors.

Once a treatment failure has been detected, the usual measure is to change the treatment and attack the virus with a different combination of drugs. There are two major problems here. First, the number of FDA-approved drugs is limited, and therefore the number of effective combinations of drugs is also limited. It is conceivable that a viral quasi-species may become resistant to all known drugs, thus rendering the treatment ineffective. A second problem is that of cross-resistance, which further reduces the number of effective combination therapies.

Previous Work

Trying to relate the structure of the virus to drug resistance is only a particular sub-problem of understanding the more general relationship between the structure and function of the HIV virus. Various papers studying the HIV protease have focused on the flaps and dimer-interface flexibility (Ishima *et al.*, 1999), molecular surface analysis (Pattabiraman *et al.*, 1999) and the auto-processing of the HIV-1 protease (Louis *et al.*, 1999). The drug resistance problem was also studied in the context of various mutations (Mahalingam *et al.*, 1999; Xie *et al.*, 1999). There have been attempts to classify and predict drug resistance based on genotypic information. One of the most successful attempts was realized at UC Irvine (Brown *et al.*, 1999; Lathrop *et al.*, 1999).

The aspects novel to this study include the use of structural information and the use of neural processing (SOFMs) in the context of HIV drug resistance.

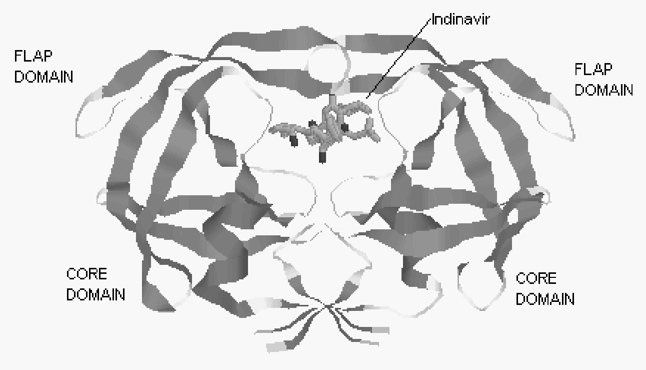

Fig. 1. The structure of the HIV protease in complex with Indinavir. The protease has a dimeric structure composed of two identical monomers. The upper part of the structure has two flaps that close when the protease binds to a substrate. When bound to a protease inhibitor such as Indinavir, the protease is unable to function normally. The drug is the small structure in the center of the protease.

METHODS

Structure-based Data Mining

This work is based on the mutations reported in (Schinazi *et al.*, 1999) and (Winters *et al.*, 1998). The input data is represented by mutations reported with respect to the HIV consensus B wild type. A data processing pipeline was set up as follows.

1. Construct mutant genotypes and produce 3D structures using Modeler
2. Use Ligplot to analyze the 3D structures and produce a list of contacts between the mutant proteases and protease inhibitor
3. Preprocess the contact information (input reduction, normalization)
4. Construct and train a self-organizing map to categorize mutant resistance to the protease inhibitor as high, medium, or low
5. Test the network and analyze its performance

The goal was to learn the relationship between some structural features of the mutant HIV protease (e.g. contacts with the inhibitor) and the corresponding drug resistance as characterized by IC90[†]. Two primary difficulties confronted this attempt. The first difficulty is that different mutants have different structures, and therefore they

[†] The IC90 is the result of a 'phenotypic test', and represents the amount of drug necessary to reduce viral replication by 90%. The ratio between the IC90 value of a specific mutant and the IC90 value of the wild type (the virus in its non-mutated form) is the fold resistance.

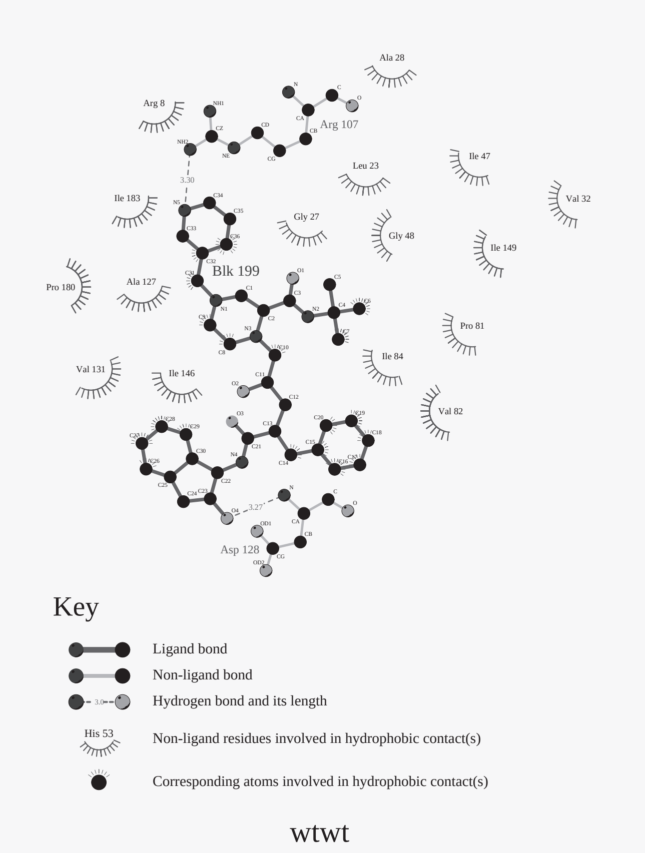

Fig. 2. An example Ligplot output showing the contacts between a protease mutant and inhibitor. In this drug-inhibitor complex there are two hydrogen bonds and 16 non-bonding interactions.

have different contact points with the drug. The number of atomic contacts between the protease and the inhibitor is typically around 30. However, the particular set of contacts is different from one mutant to another. In order to be able to construct a training set having patterns with a fixed length, a master list of contacts was compiled including all contacts that appear in at least one mutant. Thus, the input pattern corresponding to one mutant has non-zero values for the contacts that are actually present and zero values for the missing contacts. This list included 173 contacts. The second difficulty was related to the reduced number of patterns. Unfortunately, the phenotypic tests are relatively difficult and expensive. Therefore, the number of IC90 values available is rather limited.

Initially, the genotype of the mutants was constructed using the mutations and the known genotype of the wild type. The 3D molecular structure of these mutants was then modeled using the Modeler package (Sali and Blundell, 1993; Sali *et al.*, 1995). Modeler is able to construct the molecular structure of a protein given its genetic sequence and a reference genetic sequence whose molecular structure is known (e.g. from x-ray crystallography). Each sequence was modeled together with the protease inhibitor Indinavir. The errors introduced in the modelling process increase proportionately to the genotypic difference between the sequence to be modeled and the known sequence. In this case, the genotypic differences are minimal (each mutant contains only a small number of point mutations) and therefore such errors are expected to be small.

The next step in the processing pipeline was to analyze the structures constructed by Modeler. This analysis was performed using Ligplot (Wallace *et al.*, 1995). Ligplot is able to analyze a molecular structure given in Protein Data Bank (PDB) format and calculate the distances between specific atoms belonging to the drug and specific atoms belonging to the mutant protease. A sample output of Ligplot for a protease bound with Indinavir is presented in Figure 2.

The structures produced by Modeler were then analyzed in order to find the contacts between the inhibitor and the protease. Two types of contacts were considered: hydrogen bonds (HB) and non-bonding interactions (NBI). Then for each mutant–inhibitor structure, the pattern of HB and NBI contacts was determined as follows. For each mutant–inhibitor complex, lists of residues involved in the contacts were constructed. A master list of residues was then compiled by taking the union of the individual lists across all cases available. The HB and NBI contacts for each residue were considered separately. That is, if a residue has both HB and NBI contacts, that residue will be represented in the feature vector by two values representing the number of HBs and NBIs, respectively. In order to further reduce the number of dimensions, all contacts that exhibited no differences from the wild type across the entire set of mutants were eliminated. The feature vectors constructed in this manner contained information only about those 22 contacts that differed from the wild type in at least one mutant.

Weights of 1.0 and 2.0, were used for NBIs and HBs, respectively, in order to reflect the fact that a hydrogen bond is stronger than a non-bonding interaction.[‡] The 22-element feature vectors describing each mutant were then normalized to a length of 1. This normalization does not reduce the amount of structural information because each contact is represented in a different direction. Thus, each vector has a direction in the *n*-dimensional vector space

[‡] Initially, weights of 0.2 and 2.0 were used for NBIs and HBs, respectively. Although these weights reflect more accurately the relative strength of the two types of bonds, the clustering almost completely ignored the NBIs. Since the Indinavir–wild type complex has only two hydrogen bonds, and most mutants lose one such bond, the result was a clustering that did not discriminate between various mutants. To compensate, the weights were changed as described.

Table 1. Resistance values of HIV Protease mutants to Indinavir, pattern set for all available data. The fold resistance was calculated as a ratio between the IC90 value of the mutant and the IC90 value of the wild type. All mutations were obtained from Winters *et al.* (1998), except as noted. Resistance values were not available for 14 out of the total of 38 mutants. (NR = No Resistance reported. Patterns 1–6, 52, and 53 are from Schinazi *et al.* (1999). Patterns 7–21 are also from Schinazi *et al.* (1999), but are not included because they are redundant with patterns from Winters *et al.* (1998))

ID	Mutation	IC_{90}(uM)	Fold resistance
-	Wild Type	0.12	1
1	L10I	4.08	34
2	G48V	9.12	76
3	I54V	NR	NR
4	G73S	NR	NR
5	V82A	NR	NR
6	I84V	2.28	19
22	L10I K14R N37D M46I F53L A71V G73S V77I L90M	1.57	13.08
23	L10I E35D M36I R41K I62V L63P A71V G73S I84V L90M I93L	9.14	76.17
24	L10I I15V M36I G48V I54V I62V V82A	0.66	5.5
25	L10I I15V M36I G48V I54V I62V	0.57	4.75
26	K14R I15V N37D F53L A71V G73S L90M	1.09	9.08
27	K14E M36V G48V L63P A71V T74S V82A	0.65	5.42
28	I15V R41K L63P A71T G73S L90M	0.72	6.0
29	G48V L63P T74A	0.35	2.92
30	K20I M36I L63P A71T G73S L90M	0.43	3.58
31	L10I E35D R41K I62V L63P A71V G73S I84V L90M I93L	0.28	2.33
32	K14R R41K L63P V77I L90M I93L	0.41	3.42
33	L10I K20M L63P A71T V77I L90M I93L	0.32	2.67
34	N37D R57K D60E L63P A71V G73S L90M I93L	0.22	1.83
35	I15V D30N E35D M36I R41K L63P	0.08	0.67
36	L63P T74S L90M	NR	NR
37	L63P L90M	NR	NR
38	K14R R41K L63P V77I I93L	0.16	1.33
39	L10V I62V G73S L90M	0.05	0.42
40	L63P T74A V77I	NR	NR
41	L63P L90M	0.07	0.58
42	N37D L63P A71V G73S L90M I93L	0.15	1.25
43	L10I L63P A71T V77I I93L	0.11	0.92
44	I15V E35D R41K L63P	0.08	0.67
45	K14R/K L63P I93I/L	NR	NR
46	K14E L63P A71V	NR	NR
47	I15V	NR	NR
48	L63P	0.02	0.17
49	L10I L63T A71T	NR	NR
50	L63P A71V L90M	NR	NR
51	L63A	NR	NR
52	G48V I54V L90M	NR	NR
53	G48V I84V L90M	NR	NR

given by all possible contacts. Normalizing is a simple rescaling of the length of the vector. The direction of the vector (containing the information about the contacts) remains the same even if the length of the vector is brought to 1. This normalization is also crucial for the Kohonen map used subsequently. If this is not done, the clustering will group together big and small vectors, whereas we are interested in grouping together vectors going in similar directions.

The set of 38 available patterns (Table 1) (Winters *et al.*, 1998; Schinazi *et al.*, 1999) was then divided into a training set of 31 patterns and a validation set of seven patterns. The training set included approximately 75% of the patterns with known resistance (17 patterns) and 14 patterns without resistance values for a total of 31 training patterns. The validation set contained only patterns with known resistance, and was used to test the prediction abilities of the system.

Finally, a Kohonen network (Kohonen, 1982) was trained with the patterns obtained above. In the classical self-organizing feature map (SOFM) used here, all inputs are connected to all neurons. When a pattern is presented, the excitation of each unit is proportional to the dot product between the input vector and the weight vector. The unit with the weight vector closest to the input vector will have the largest excitation and will be declared the winner. The training involves changing the weights of the winner and of its neighbors in such a way that their weight vectors become more similar to the current input pattern. Training a Kohonen network is typically accomplished by gradually decreasing the learning rate and the radius of the neighborhood with each iteration. The training stops when the learning rate becomes zero. The experiments were performed with a learning rate between 0.6 and 0.9 and the learning rate was linearly decreased to zero over 10–50 training cycles.

In the trained SOFM, each neuron is a cluster representing the prototype of those patterns for which that neuron was declared the winner. In applications in which there are many more patterns than units (e.g. SOFM of microarray data where there are thousands of genes and tens or hundreds of neurons) many patterns fall into each cluster with adjacent clusters containing similar patterns. Here, a pattern will activate entire groups of neurons with the winner having the strongest activation. An example of the response of a trained map with patterns from six different clusters is shown in Figure 3. These figures were obtained with SNNS (Zell *et al.*, 1991c,a,b). These images were generated from individual mutants, showing different patterns of neuron activation that may be clustered to deduce similarities in resistance. Table 2 presents the distribution of the patterns in the 15 clusters formed by the SOFM. Lighter shades represent higher activation, while

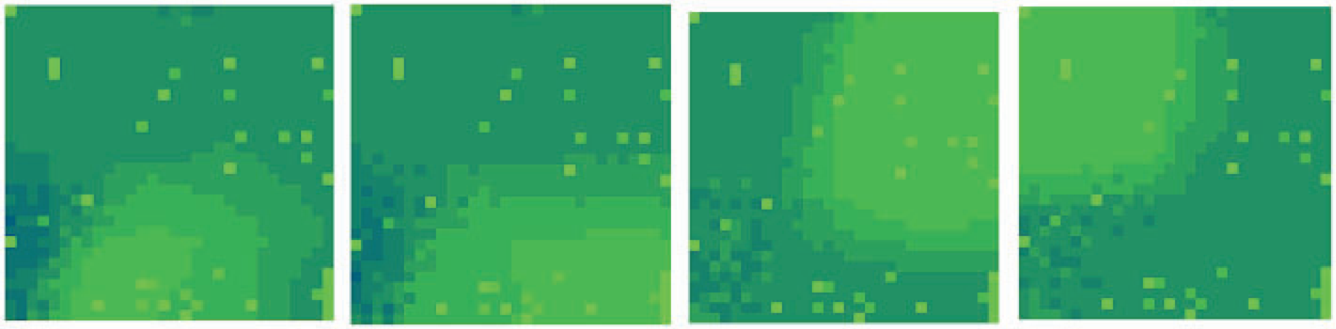

Fig. 3. Examples of responses of a trained 20 × 20 Kohonen map when patterns from four different clusters were used as inputs. Left to right: patterns from clusters 1 and 2 in the high resistance meta-class followed by a pattern from the medium and a pattern from the low resistance meta-classes. The figures were produced using the Stuttgart Neural Network Simulator (SNNS).

Table 2. The results of the self-organization process. The table shows the distribution of the patterns in different clusters

Cluster ID	Mutants
1	3, 6
2	1, 4, 5
3	22, 43, 45, 47
4	23
5	24, 27
6	25, 52
7	28, 30
8	31
9	33, 38
10	34, 42
11	36, 49, 50
12	37, 39, 40
13	44, 46
14	51
15	53

darker shades represent inactive units.[§]

Apparently, 15 clusters may seem too many for only 17 labelled data points. This may be interpreted as putting every mutant with a known resistance in its own cluster. There are two aspects that may be discussed here. First, the training was done with all available 38 patterns (labelled or not) patterns and the clustering was based exclusively on structural similarity. Second, the number of clusters was determined by the data itself. There were 20 × 20 = 400 neurons available so we could have ended up with 31 patterns in 31 clusters (and 369 clusters to spare). However, we ended up with only 15 clusters, with one cluster of four patterns, three clusters of three patterns, etc. This shows that there was indeed some structure in the data which was subsequently proven by a statistically significant improvement in the class prediction ($p = 0.056$ for the worst case). We also tried other architectures. Those architecture with sufficient neurons performed similarly to the 20 × 20 one. When we tried to force the network to use fewer clusters (e.g. by using a 2 × 2 or 3 × 3 architectures, the generalization results were poor. This suggests that while there are some useful features in the data that a large architecture can pick up and use, there are also features that differ between patterns. In other words, the data is not consistent enough to be modeled well by only a few clusters.

The mutants were classified based on their fold resistance (the ratio between IC90 or IC50 values). The first class included the wild type and mutants with no resistance or very low resistance (less than five-fold resistance). The second class included the mutants with low resistance (between five- and ten-fold resistance) and the third class included the mutants with high resistance (more than ten-fold). The most resistant mutant in the study had an approximate 76-fold resistance to Indinavir and was also cross-resistant to Saquinavir and Nelfinavir.

In the next stage, the clusters were grouped into five meta-classes. The first meta-class included all clusters that could clearly be associated to high resistance. The second meta-class included the clusters that could be associated with low resistance. The third meta-class included clusters containing only patterns with very low resistance or no resistance at all. Clusters that included patterns from various groups formed the fourth meta-class (inconclusive) and the clusters that include only patterns for which there were no reported resistance values form the fifth meta-class (unknown). The meta-classes are presented in Table 3.

[§] The units appearing as active by themselves are units ‘stuck’ at that value (note their pattern is the same in all images). However, their presence does not affect the classification which is done based on the winning neuron only (the one in the center of the each active neighborhood).

Table 3. Organization of clusters into meta-classes. (NR = No Resistance reported)

Meta-class	Clusters	Patterns	Pattern fold resistances
high resistance	1, 2, 4	3, 6, 1, 4, 5, 23	NR, 19, 34, NR, NR, 76.17
low resistance	5, 7	24, 27, 28, 30	5.5, 5.42, 6.0, 3.58
no resistance	6, 8, 9, 10	34, 42, 33, 38, 31, 25, 52	1.83, 1.25, 2.67, 1.33, 2.33, 4.75, NR
inconclusive	3	22, 43, 45, 47	13.08, 0.92, NR, NR
unknown	12	37, 39, 40	NR, NR, NR

The mixed approach to unsupervised learning in which clusters are formed followed by supervised learning in which clusters are labelled using known patterns, is often referred to as semi-supervised clustering (Demiriz and Embrechts, 1999; Cohn *et al.*, 2000; Basu *et al.*, 2002).

Sequence-based Data Mining

This second approach used the amino acid sequence of the HIV protease mutants to predict their resistance to Saquinavir. Only 32 HIV protease mutants with reported IC90 drug resistance values for Saquinavir were found in the literature (Winters *et al.*, 1998). These patterns were supplemented with 811 reported HIV protease mutants obtained from the Los Alamos National Laboratory HIV Sequence Database (http://hiv-web.lanl.gov/), along with the wild type HIV protease sequence. The method used is as follows:

1. Preprocess the sequence information (input reduction, normalization)
2. Construct and train a set of SOFM networks while systematically varying the network parameters, excluding one known pattern each time
3. Assess network performance using leave-one-out cross validation.

The goal of this research was not to predict the exact IC90 resistance of a particular mutant, but rather to be able to classify a mutant as having high, medium, or low resistance to Saquinavir. Low resistance for a mutant was defined as having less than a five-fold resistance as compared to the resistance of the wild type. High resistance was defined as greater than ten-fold resistance. Using these cutoffs, 12 of the 32 patterns with known IC90 values were classified as having low resistance, three with medium resistance (between five- and ten-fold resistance), and the remaining patterns classified as exhibiting high resistance. The actual range of resistance values was from 0.33-fold to 269-fold (see Table 4).

In the preprocessing step, the amino acid sequences of the mutants were converted into normalized numeric patterns suitable for a neural network. Patterns that matched the wild type at a particular residue were assigned a value of zero for that feature (residue). Residues that differ from the wild type are ordered by frequency of occurrence. They were then assigned a value between 0 and 1 based on dividing $(0, 1]$ into n equal increments, where n is the number of different mutations from the wild type for that residue. Each pattern was then normalized to a length of 1. Two residues were unchanged in all mutants and the wild type, and were eliminated as inputs, producing input vectors with 97 inputs.

Thirty-six network architectures were considered. Each network architecture was trained 32 times (one for each leave-one-out pattern to be tested), for a total of 1152 trials. See Table 5 for a complete listing of the networks. To summarize, networks with output matrices of 12×12, 10×10, 8×8, 6×6, 5×5, 4×4, and 3×3 were trained using initial learning rates of 0.9–0.5 and initial neighborhoods corresponding to the dimensionality of their output matrix (e.g. an initial neighborhood of 12 for the 12×12 matrix). Training was accomplished as described for structure-based data mining; the learning rate was decreased during each iteration and training stopped when the learning rate reached zero. All networks except one trained using ten iterations. The 10×10 matrix was also trained using 50 iterations, an initial learning rate of 0.7, and an initial neighborhood of 10. The results of this test were then compared to the same conditions and 10 iterations to see if increasing the number of iterations would improve the performance of the network. This experiment actually performed worse than at 10 iterations. This showed that a longer training does not translated in better performance.

Once each network was trained, the pattern with known resistance that was not used during the training was run through the network. If the pattern was assigned to a ’mixed’ cluster or to one with no label, we concluded that the classifier could not predict the resistance for that pattern. Otherwise, a predicted resistance would be assigned based on the label of the cluster in which the pattern was placed. A false positive (FP) is defined as a mutant that was classified as being more resistant than it actually was. For instance a false positive condition exists if the mutant’s IC90 value would place the mutant in the low-resistance category (i.e. the IC90 of the mutant is less than five-fold more resistant to Saquinavir than the wild type) and the network assigns to that mutant a label of

Table 4. HIV Protease mutants resistant to Saquinavir. The fold resistance was calculated as a ratio between the IC90 value of the mutant and the IC90 value of the wild type. All mutations were obtained from Winters *et al.* (1998), except as noted

Mutation	IC_{90}(uM)	Fold resistance
Wild Type	0.03	1
L10I K14R N37D M46I F53L A71V G73S V77I L90M	8.08	269
L10I E35D M36I R41K I62V L63P A71V G73S I84V L90M I93L	6.00	200
L10I I15V M36I G48V I54V I62V V82A	1.18	39
L10I I15V M36I G48V I54V I62V	0.92	30.67
K14R I15V N37D F53L A71V G73S L90M	0.58	19
K14E M36V G48V L63P A71V T74S V82A	0.58	19
I15V R41K L63P A71T G73S L90M	0.37	12
G48V L63P T74A	0.80	27
K20I M36I L63P A71T G73S L90M	0.42	14
L10I E35D R41K I62V L63P A71V G73S I84V L90M I93L	0.34	13
K14R R41K L63P V77I L90M I93L	0.21	7
L10I K20M L63P A71T V77I L90M I93L	0.20	7
N37D R57K D60E L63P A71V G73S L90M I93L	0.20	7
I15V D30N E35D M36I R41K L63P	0.03	1
L63P T74S L90M	0.09	3
L63P L90M	0.08	3
K14R R41K L63P V77I I93L	0.07	2
L10V I62V G73S L90M	0.07	2
L63P T74A V77I	0.07	2
L63P L90M	0.06	2
N37D L63P A71V G73S L90M I93L	0.06	2
L10I L63P A71T V77I I93L	0.06	2
I15V E35D R41K L63P	0.06	2
K14R/K L63P I93I/L	0.06	2
K14E L63P A71V	0.06	2
I15V	0.04	1
L63P	0.05	2
L10I L63T A71T	0.02	1
L63P A71V L90M	0.02	1
L63A	0.01	0.33
G48V I54V L90M (Schinazi *et al.*, 1999)	1.50	50
G48V I84V L90M (Schinazi *et al.*, 1999)	0.90	30

medium or high resistance. Conversely, if a mutant is more resistant than the label assigned by the network, a false negative (FN) condition exists.

For each network, the 32 test patterns are identified as correctly classified, FP, FN, or not classified (if they are assigned to a mixed or unlabelled cluster). Then the coverage and accuracy of the network is calculated. Coverage is defined as the ratio of test patterns that were classified (i.e. assigned to a labelled cluster) to total test patterns. Accuracy is defined as the ratio of patterns that were *correctly* classified to the total number patterns classified. Both are expressed as percentages. A third number calculated for each network is the network's score:

$$\text{Score} = \text{Coverage}*\text{Accuracy}*100$$

This score allows us to compare networks based on a single number. Obviously, there are other ways one may calculate a score that weights the contribution of coverage and accuracy differently. Here, they are treated as equal contributions to the overall score of the network (although the results are ordered by coverage prior to selecting the network with the best accuracy).

DISCUSSION

The classical machine learning approach that was followed for both structure and sequence-based data mining can be summarized as follows: (1) use unsupervised learning with as many patterns as possible, some of which (few) are labelled; (2) once clusters have been obtained, use the few known patterns to label the clusters; (3) use known patterns and the now labelled clusters to predict the output for known patterns not used during the training (i.e. cross-validate). This was done with both types of data. For the structural data, the unsupervised learning was done on 31 patterns. For the sequence data, this was done on 843 patterns.

Structure-based Data Mining

The classification abilities were assessed in several ways. First of all, training abilities can be estimated by calculating the number of patterns with known resistance values that cannot be classified at the end of the training phase. This occurs when patterns with different resistances fall into the same cluster. For the pattern set presented in Figure 3, the only cluster of this type is cluster 3 which contains two known patterns with different resistance characteristics (high and low). Therefore, the device learned $15/17$ or 88% of the patterns.

As usual in machine learning, the performance is reported on cross-validation data only. For the leave-one-out cross validation used here (Breiman *et al.*, 1984), the training is repeated *n* times, every time leaving out one pattern and subsequently predicting its output. Using this procedure, the predictor was able to estimate accurately the resistance of unseen patterns in six out of ten cases. In one of the remaining cases, the prediction was no resistance (<five-fold) when in fact the pattern belonged to the low resistance category (between five- and ten-fold). However, a closer examination showed that the pattern had a fold resistance of 5.5 which is on the border between no resistance and low resistance. Taking this into consideration, the accuracy of the predictor was estimated to be between 60% and 70%. The predictor would not classify in 12 out of 24 cases.

To summarize the prediction based on structural features, the leave-one-out cross validation produced a prediction rate of 45% and an accuracy of at least 60% (six correct predictions out of ten predictions). The probability of guessing the correct class for any given pattern

Table 5. Summary of Results. Values listed for learning rate and neighborhood are initial values

Output	Learn Rate	Nbrhood	Iterations	Coverage	Accuracy	Score
12 × 12	0.9	12	10	50%	75%	38
12 × 12	0.8	12	10	41%	62%	25
12 × 12	0.7	12	10	47%	60%	28
12 × 12	0.6	12	10	28%	56%	16
12 × 12	0.5	12	10	38%	42%	16
10 × 10	0.9	10	10	53%	76%	40
10 × 10	0.8	10	10	31%	60%	19
10 × 10	0.7	10	10	41%	62%	25
10 × 10	0.7	10	50	28%	44%	12
10 × 10	0.6	10	10	53%	71%	38
10 × 10	0.5	10	10	44%	50%	22
8 × 8	0.9	8	10	53%	65%	34
8 × 8	0.8	8	10	41%	62%	25
8 × 8	0.7	8	10	38%	58%	22
8 × 8	0.6	8	10	69%	68%	47
8 × 8	0.5	8	10	31%	100%	31
6 × 6	0.9	6	10	31%	80%	25
6 × 6	0.8	6	10	31%	80%	25
6 × 6	0.7	6	10	41%	85%	35
6 × 6	0.6	6	10	41%	62%	25
6 × 6	0.5	6	10	41%	85%	35
5 × 5	0.9	5	10	25%	88%	22
5 × 5	0.8	5	10	22%	86%	19
5 × 5	0.7	5	10	9%	33%	3
5 × 5	0.6	5	10	19%	100%	19
5 × 5	0.5	5	10	25%	75%	19
4 × 4	0.9	4	10	9%	100%	9
4 × 4	0.8	4	10	9%	100%	9
4 × 4	0.7	4	10	6%	100%	6
4 × 4	0.6	4	10	6%	100%	6
4 × 4	0.5	4	10	13%	75%	10
3 × 3	0.9	3	10	0%	N/A%	0
3 × 3	0.8	3	10	0%	N/A%	0
3 × 3	0.7	3	10	0%	N/A%	0
3 × 3	0.6	3	10	0%	N/A%	0
3 × 3	0.5	3	10	3%	100%	3

Table 6. Average performance of networks by size of output matrix

Output Matrix	Coverage	Accuracy	Score
12 × 12	41%	59%	25
10 × 10	44%	64%	29
8 × 8	46%	71%	32
6 × 6	37%	78%	29
5 × 5	20%	76%	16
4 × 4	9%	95%	8
3 × 3	1%	100%	1

is 33% as there are three resistance categories (high, medium, and low). According to a binomial distribution (correct/incorrect, with a 33% probability of being correct and a 67% probability of being incorrect), the probability of four mistakes or fewer in ten trials just by chance (p-value) is 0.0766 which is significant at 10% confidence level. This suggests that although other factors may be involved, there is indeed a connection between structural features and resistance (as represented by the pattern of contacts between the protease and the inhibitor).

Sequence-based Data Mining

The results of the sequence-based predictions are summarized in Table 5. The network with the best overall performance and also the best coverage was the 8 × 8 output matrix with an initial learning rate of 0.6. The most accurate network was the 8 × 8 output matrix with an initial learning rate of 0.5. This network produced 100% accuracy, but provided only 31% coverage. Note that there are other networks which produced 100% accuracy, but all of these networks exhibited very poor coverage (less than 10%) and were rejected from serious consideration.

Overall, it was observed (see Table 6) that the networks with 8 × 8 output matrices performed best (average score of 32) and also provided the best coverage (average of 46%). Networks with 12 × 12, 10 × 10, 6 × 6 and 5 × 5 output matrices also performed reasonably well. The networks with smaller output matrices had very high accuracy, but their coverage was quite poor (again, less than 10%). It was also observed that increasing the number of iterations during training did not improve network performance, but actually degraded performance for the test case (10 × 10 output matrix, 0.7 initial learning rate, 50 iterations).

The performance of the best single network provided 68% accuracy (15 correct predictions) at a coverage of 69%[¶] (22 patterns for which the network attempted classification). Using a binomial distribution as described previously, the probability that a three-class classifier would correctly make seven mistakes or fewer in 22 trials is 0.0007%. Hence, the results are highly significant.

One way to further improve performance is to make use of multiple networks at once using a majority voting scheme. In majority voting, the results of presenting a pattern to a number of networks is tallied, and the majority classification is taken as correct. In situations where one or more networks fail to classify the pattern (e.g. the pattern is assigned to a mixed or unlabelled cluster), only the outputs of the networks that successfully classify the pattern are used. In the case of a tie (there were none for the schemes that were explored), the lowest drug resistance classification was to be selected. That is, the risk of trying a drug treatment that did not work was

[¶] As mentioned earlier, the most accurate network had 100% success for those patterns that it was able to classify, but provided only marginal coverage at 31%. Certainly better coverage is desired for as critical an application as predicting HIV drug resistance. As such, this low-coverage network was not compared to random performance, as it would overstate the general prediction capabilities of this approach.

Table 7. Comparison of scores for various majority voting schemes. The best single network was the 8 × 8 output matrix, 0.6 initial learning rate, initial neighborhood of eight, ten iterations; the most accurate single network was the 8×8 output matrix, 0.5 initial learning rate, initial neighborhood of eight, ten iterations

Voting Scheme	Coverage	Accuracy	Score
Majority of 6 Most Accurate	84%	85%	71
Majority of Best + 3 Most Accurate	88%	79%	70
Majority of 4 Best Score	84%	70%	59
Best Single Network	69%	68%	47
Most Accurate Single Network	31%	100%	31

considered to be less critical than the risk of missing a potentially effective drug treatment.

Three schemes were tested and compared to the best single network and the most accurate single network. The first scheme was a combination of the six most accurate networks: 8 × 8—0.5, 6 × 6—0.7, 6 × 6—0.5, 5 × 5—0.9, 5 × 5—0.8, and 5 × 5—0.6 (the number after the dash is the initial learning rate). The second scheme combined the best single network with the three most accurate networks: 6 × 6—0.7, 6 × 6—0.5, and 8 × 8—0.5. Again, those networks with 100% accuracy but very low coverage (the networks with 4 × 4 and 3 × 3 output matrices) were ignored. The final scheme combines the results of the four networks with the best overall scores: 8 × 8—0.6, 10 × 10—0.9, 10 × 10—0.6, and 12 × 12—0.9.

It has been shown that the performance of a combiner (e.g. a majority voting scheme) is never worse than the average of the individual classifiers, but not necessarily better than the best classifier (Perrone, 1995). In this case, all of the majority voting schemes outperformed the single best network (see Table 7). The average coverage across the three voting schemes was 85%, the average accuracy of the three was 78%, and the average score was 67. This represents a marked improvement over the single best network (69, 68, and 47%, respectively). The best majority voting scheme predicted 23 of 27 patterns, again compared to the 33% accuracy expected from a random classifier. The probability of four mistakes or fewer in 27 trials just by chance is of the order of 10^{-8}.

CONCLUSIONS

This report presented two approaches to predicting HIV drug resistance. The first such approach used structural data and achieved an accuracy of between 60% and 70% measured using cross-validation. The second approach used sequence data and explored several network architectures. The single best classifier found yielded 69% coverage and 68% accuracy. Several architectures were then combined using a majority voting scheme. The best combined architecture yielded a coverage of 84% and an accuracy of 85%. All majority voting schemes used outperformed the single best network. The probability that such performance should happen by chance was less than 0.1 for the results of the structure-based data mining, and negligible for the sequence-based data mining.

These results demonstrate that drug resistance can be predicted from either the structural features or protein sequence of the HIV protease.

ACKNOWLEDGEMENTS

The authors would gratefully like to acknowledge Lonnie Cumberland for the software that he developed to assist with preprocessing and data normalization, as well as his input into the early versions of this manuscript.

REFERENCES

Basu,S., Banerjee,A. and Mooney,R. (2002) Semi-supervised clustering by seeding. In *Proceedings of the 19th International Conference on Machine Learning*, (ICML-2002), Sydney, Australia.

Breiman,L., Friedman,J.H., Olsen,R.A. and Stone,C.J. (1984) *Classification and Regression Trees*. Wadsworth and Brooks.

Brown,A.J. *et al.* (1999) Sequence clusters in human immunodeficiency virus type 1 reverse transcriptase are associated with subsequent virological response to antiretroviral therapy. *J. Infect. Dis.*, **180**, 1043–1049.

Cohn,D., Caruana,R. and McCallum,A. (2000) Semi-supervised clustering with user feedback. In *Proceedings of the American Association for Artificial Intelligence*, (AAAI 2000), Austin, Texas.

Demiriz,A. and Embrechts,M. (1999) Semi-supervised clustering using genetic algorithms. *Artificial Neural Networks in Engineering*, (ANNIE'99).

Ishima,R. *et al.* (1999) Flap opening and dimer-interface flexibility in the free and inhibitor-bound HIV protease, and their implications for function. *Structure Fold Design*, **7**, 1047–1055.

Kohonen,T. (1982) Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, **43**, 59–69.

Lathrop,R.*et al.* (1999) Knowledge-based avoidance of drug-resistant Hiv mutants. *AI Magazine*.

Louis,J.M. *et al.* (1999) Autoprocessing of HIV-1 protease is tightly coupled to protein folding. *Nat. Struct. Biol.*, **6**, 868–875.

Mahalingam,B. *et al.* (1999) Structural and kinetic analysis of drug resistant mutants of HIV-1 protease. *Eur. J. Biochem.*, **263**, 238–245.

Pattabiraman,N. (1999) Occluded molecular surface analysis of ligand-macromolecule contacts: Application to HIV-1 protease-inhibitor complexes. *J. Med. Chem.*, **42**, 3821–3834.

Perrone,M.P. (1999) Averaging/modular techniques for neural networks. In Arbib,M.A. (ed.), *The Handbook of Brain Theory and Neural Networks*. MIT Press, Cambridge, Massachusetts, pp. 126–129.

Sali,A. and Blundell,T. (1993) Comparative protein modelling by satisfaction of spatial restraints. *J. Mol. Biol.*, **234**, 779–815.

Sali,A. *et al.* (1995) Evaluation of comparative protein modeling by MODELLER. *Proteins*, **23**, 318–326.

Schinazi,R.F. *et al.* (1999) Mutations in retroviral genes associated with drug resistance: 1999-2000 update. *International Antiviral News*, **7**, 46–69.

Wallace,A.C. *et al.* (1995) LIGPLOT: A program to generate schematic diagrams of protein-ligand interactions. *Protein Eng.*, **8**, 127–134.

Winters,M.A. *et al.* (1998) Human immunodeficiency virus type 1 protease genotypes and in vitro protease inhibitor susceptibilities of isolates from individuals who were switched to other protease inhibitors after long-term saquinavir treatment. *J. Virol.*, **72**, 5303–5306.

Xie,D. *et al.* (1999) Drug resistance mutations can effect dimer stability of HIV-1 protease at neutral pH. *Protein Sci.*, **8**, 1702–1707.

Zell,A., Mache,N., Sommer,T. and Korb,T. (1991a) Design of the SNNS neural network simulator. In *Proceedings of OGAI-91, Seventh Austrian Conference on Artificial Intelligence*.

Zell,A., Mache,N., Sommer,T. and Korb,T. (1991b) Recent developments of the SNNS neural network simulator. In *SPIE Conference on Applications of Artificial Neural Networks*.

Zell,A., Mache,N., Sommer,T. and Korb,T. (1991c) The SNNS neural network simulator. In *DAGM-91*.