Pacific Symposium on Biocomputing 7:77-87 (2002)

A SOFM Approach to Predicting HIV Drug Resistance

R. Brian Potter[a], Sorin Draghici
Department of Computer Science, Wayne State University, Detroit, MI 48202

The self-organizing feature map (SOFM or SOM) neural network approach has been applied to a number of life sciences problems. In this paper, we apply SOFMs in predicting the resistance of the HIV virus to Saquinavir, an approved protease inhibitor. We show that a SOFM predicts resistance to Saquinavir with reasonable success based solely on the amino acid sequence of the HIV protease mutation. The best single network provided 69% coverage and 68% accuracy. We then combine a number of networks into various majority voting schemes. All of the combinations showed improved performance over the best single network, with an average of 85% coverage and 78% accuracy. Future research objectives are suggested based on these results.

1 Introduction

1.1 Overview

The *human immunodeficiency virus (HIV-1)*, the causative agent of *acquired immune deficiency syndrome (AIDS)*, has been the subject of extensive research in recent years. A good, although somewhat dated introduction to AIDS research is provided by Watson, et. al.[1]

HIV-1 infection has been approached via many treatment pathways. One of the first was the use of *Azidothymidine (AZT)* to inhibit the synthesis of the HIV provirus in vivo. Unfortunately, the HIV virus was able to mutate in order to resist AZT, eventually overcoming its therapeutic benefits. Two other popular methods of treating the HIV virus are by attacking the reverse transcriptase responsible for synthesizing the DNA provirus from the retroviral RNA, and by inhibiting the HIV protease responsible for splicing the primary polyproteins produced by the HIV virus into the active proteins necessary for its replication. Both of these approaches also eventually fail due to mutation of the viral genome, leading to protease inhibitor resistant viral strains. Most current therapies involve combinations of drugs aimed at inhibition of both the reverse transcriptase and the protease.

Artificial neural network (ANN) based self-organizing maps were developed by Kohonen.[2] SOFM algorithms belong to the unsupervised learning, competitive network class of ANNs. An input vector is introduced to the network, after which a winning neuron is determined and the weight vectors of all neurons within a specified neighborhood of the winning neuron are updated.[3] In this

[a] Please send correspondence to this author at potterrb@ieee.org.

way, SOFMs are useful for clustering related patterns together. When patterns in the training set are labelled, clusters containing these labelled patterns can then be used to identify unknown patterns.

This laboratory has previously applied SOFM clustering to the HIV drug resistance problem.[4] Resistance to the protease inhibitor Indinavir was studied first by applying supervised learning techniques to protein structural data for various HIV protease mutants to predict Indinavir IC90 values. Only limited success was obtained, primarily due to an insufficient number of mutations with corresponding Indinavir IC90 values available from the literature with which to train the classifier. An SOFM was used to segment the same data into clusters of Indinavir-resistant mutants and non-resistant mutants based on structural features. We were able to divide all reported HIV mutants into several categories based on their 3-dimensional molecular structures and the pattern of contacts between the mutant protease and Indinavir. Our classifier shows reasonable prediction performance, being able to predict the drug resistance of previously unseen mutants with an accuracy of between 60% and 70%. We believe that this performance can be greatly improved once more data becomes available. The results support the hypothesis that structural features of the HIV protease can be used in antiviral drug treatment selection and drug design.

The goal of this research is to build a SOFM to predict the resistance of known mutations of HIV protease to Saquinavir, a protease inhibitor related to Indinavir that is also approved for use in the treatment of HIV infection. No attempt is made to understand the mechanism or reasons why certain mutation are or are not resistant to Saquinavir, only to predict such resistance based solely on the amino acid sequence of HIV protease mutants, a small number of which have reported Saquinavir IC90 values. Our hope is that this early work will ultimately enable clinicians to prescribe HIV treatments based on drug resistance predictions.

1.2 Related Work

Self-organizing maps have been used successfully in a wide variety of life science applications. Kaartinen et.al. have successfully used a SOFM to discriminate between human blood plasma lipoprotein lipids (LDL and HDL cholesterol, triglycerides) and furthermore to cluster plasma samples into different lipoprotein lipid risk profiles.[5] Makipaa et. al. have applied SOFMs to the clustering and subsequent classification of blood glucose data from insulin-dependent diabetic patients.[6] Santos-Andre and Roque da Silva combined a SOFM with a multi-layer perceptron to provide radiologists with a "second opinion" in

the diagnosis of breast cancer.[7] Christodoulou and Pattichis have developed medical diagnostic systems for the assessment of electromyographic (EMG) signals necessary for the diagnosis and monitoring of patients with neuromuscular disorders, and carotid plaques based on ultrasound images of patients with pulmonary disease. The systems were comprised of multiple SOFM classifiers whose results were combined using majority voting and SOFM-derived confidence measures.[8,9] Finally, Golub et. al. were able to distinguish between acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL) from SOFM clustering of gene expression monitoring data.[10]

2 Experimental Detail

2.1 Data Preparation

Only thirty-two patterns (HIV Protease mutants) were found in the literature with reported IC90 drug resistance values.[11] These patterns were supplemented with 910 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.

Netprep, a command line Java program, was written to convert the amino acid sequence of a protein or peptide segment (a string of alpha characters) into normalized numeric patterns suitable for input to a neural network. The input to Netprep is a file containing one peptide sequence per line, with each residue separated by a comma. The first pattern in the file is the wild type. For each residue, all of the patterns are compared to the wild type. Patterns that match the wild type at that residue are assigned a value of zero. Residues that differ from the wild type are ordered by frequency of occurrence. They are 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. For instance, if the wild type is V, and there are four mutations across all of the input patterns, say N, L, I, and A, N may be assigned a value of 1, L a value of .75, I a value of .5, and A a value of .25. Once these numeric assignments are made, each pattern is normalized and written to an output file.

The researcher may optionally specify at runtime a percentage of the patterns to withhold from training. All the patterns are processed as described above, after which the appropriate number of patterns to be withheld are randomly selected and output to a separate holdout file. The remaining patterns are used as input to the neural network. For the research described in this paper, ten percent of the 911 unclassified patterns were withheld. The 32

patterns with resistance values were all used, as described in the next section.

We were interested not in predicting the resistance of a particular mutant, but rather in classifying a mutant as having high, medium, or low resistance to saquinavir. We defined low resistance for a mutant as having less than a four-fold resistance to saquinavir as compared to the resistance of the wild type. High resistance was defined as greater than ten-fold resistance to saquinavir as compared to the resistance of the wild type. Having defined these cutoffs, twelve of the 32 patterns with IC90 values were classified as having low resistance, three with medium resistance (between 4- and 10-fold resistance), and the remaining patterns classified as exhibiting high resistance. The actual range of resistance values was from 0.33-fold to 269.33-fold (see Table 1[b]).

2.2 Training

A leave-one-out cross-validation strategy was used due to the scarcity of classified patterns. Thirty-one of the 32 patterns with resistance values were added to 800+ patterns remaining after holdout on the data set obtained from Los Alamos. The patterns with resistance values allowed us to identify clusters of mutants as high, medium or low resistance to saquinavir. Clusters with conflicting assignments were classified as 'mixed', and those with no assignment were classified as 'none'.

In all, 36 networks were trained a total of 32 times (one for each leave-one-out pattern to be tested), for a total of 1152 runs. See Table 3 for a complete listing of the networks. To summarize, networks with output matrices of 12x12, 10x10, 8x8, 6x6, 5x5, 4x4, and 3x3 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 12x12 matrix). All networks except one trained using 10 iterations. The 10x10 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.

3 Results and Discussion

3.1 Single Network Performance

Once each network was trained, the lone test pattern was run through the network. If the pattern was assigned to a 'mixed' cluster or to one with no

[b] All mutations were obtained from Winters, et. al.[11], 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	269.33
L10I E35D M36I R41K I62V L63P – A71V G73S I84V L90M I93L	6.00	200
L10I I15V M36I G48V I54V I62V V82A	1.18	39.33
L10I I15V M36I G48V I54V I62V	0.92	30.67
K14R I15V N37D F53L A71V G73S L90M	0.58	19.33
K14E M36V G48V L63P A71V T74S V82A	0.58	19.33
I15V R41K L63P A71T G73S L90M	0.37	12.33
G48V L63P T74A	0.80	26.67
K20I M36I L63P A71T G73S L90M	0.42	14
L10I E35D R41K I62V L63P A71V – G73S I84V L90M I93L	0.34	12.67
K14R R41K L63P V77I L90M I93L	0.21	7
L10I K20M L63P A71T V77I L90M I93L	0.20	6.67
N37D R57K D60E L63P A71V G73S – L90M I93L	0.20	6.67
I15V D30N E35D M36I R41K L63P	0.03	1
L63P T74S L90M	0.09	3
L63P L90M	0.08	2.67
K14R R41K L63P V77I I93L	0.07	2.33
L10V I62V G73S L90M	0.07	2.33
L63P T74A V77I	0.07	2.33
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	1.33
L63P	0.05	1.67
L10I L63T A71T	0.02	0.67
L63P A71V L90M	0.02	0.67
L63A	0.01	0.33
G48V I54V L90M[12]	1.50	50
G48V I84V L90M[12]	0.90	30

Table 1: Resistance values of HIV Protease mutants 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.

x	L	M	H
L		FP	FP
M	FN		FP
H	FN	FN	

Table 2: Truth table for determining false positives and false negatives. Actual classifications are on the left, classifications predicted by the SOFM are across the top.

label, then the pattern was not classified. Otherwise, a predicted resistance classification would be assigned based on the label of the cluster in which the pattern was placed. We defined a false positive (FP) as a mutation that was classified as being more resistant than it actually was based on its n-fold resistance value. For instance a false positive condition exists if the mutant's IC90 value as reported causes the mutant to be defined as low resistance (i.e., the IC90 of the mutant is less than four-fold more resistant to saquinavir than the wild type) and the network assigns to that mutant a label of medium or high resistance. Conversely, if a mutant is reported as more resistant than the label assigned by the network, a false negative (FN) condition exists. Table 2 summarizes this logic as a truth table.

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 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. For our purposes, both are expressed as percentages. A third number that has been calculated for each network is what we call the network's score:

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

The 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. For our purposes, we will treat them as equal contributions to the overall score of the network, although we will also discriminate by coverage before attempting to find the network with the best accuracy.

Our results are summarized in Table 3. The network with the best overall performance and also the best coverage was the 8x8 output matrix with an initial learning rate of 0.6. The most accurate network was the 8x8 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

Output	Learn Rate	Nbrhood	Iterations	Coverage	Accuracy	Score
12x12	0.9	12	10	50%	75%	38
12x12	0.8	12	10	41%	62%	25
12x12	0.7	12	10	47%	60%	28
12x12	0.6	12	10	28%	56%	16
12x12	0.5	12	10	38%	42%	16
10x10	0.9	10	10	53%	76%	40
10x10	0.8	10	10	31%	60%	19
10x10	0.7	10	10	41%	62%	25
10x10	0.7	10	50	28%	44%	12
10x10	0.6	10	10	53%	71%	38
10x10	0.5	10	10	44%	50%	22
8x8	0.9	8	10	53%	65%	34
8x8	0.8	8	10	41%	62%	25
8x8	0.7	8	10	38%	58%	22
8x8	0.6	8	10	69%	68%	47
8x8	0.5	8	10	31%	100%	31
6x6	0.9	6	10	31%	80%	25
6x6	0.8	6	10	31%	80%	25
6x6	0.7	6	10	41%	85%	35
6x6	0.6	6	10	41%	62%	25
6x6	0.5	6	10	41%	85%	35
5x5	0.9	5	10	25%	88%	22
5x5	0.8	5	10	22%	86%	19
5x5	0.7	5	10	9%	33%	3
5x5	0.6	5	10	19%	100%	19
5x5	0.5	5	10	25%	75%	19
4x4	0.9	4	10	9%	100%	9
4x4	0.8	4	10	9%	100%	9
4x4	0.7	4	10	6%	100%	6
4x4	0.6	4	10	6%	100%	6
4x4	0.5	4	10	13%	75%	10
3x3	0.9	3	10	0%	N/A%	0
3x3	0.8	3	10	0%	N/A%	0
3x3	0.7	3	10	0%	N/A%	0
3x3	0.6	3	10	0%	N/A%	0
3x3	0.5	3	10	3%	100%	3

Table 3: Summary of Results. Values listed for learning rate and neighborhood are initial values. Score = Coverage*Accuracy*100.

Output Matrix	Coverage	Accuracy	Score
12x12	41%	59%	25
10x10	44%	64%	29
8x8	46%	71%	32
6x6	37%	78%	29
5x5	20%	76%	16
4x4	9%	95%	8
3x3	1%	100%	1

Table 4: Average performance of networks by size of output matrix.

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 4) that the networks with 8x8 output matrices performed best (average score of 32) and also provided the best coverage (average of 46%). Networks with 12x12, 10x10, 6x6 and 5x5 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 (10x10 output matrix, 0.7 initial learning rate, 50 iterations).

3.2 Majority Voting Schemes

The performance of the best network allowed for better-than-random accuracy (68%) and acceptable coverage of 69%. The most accurate network had 100% success for those patterns that it was able to classify, but provided only marginal coverage at 31%. Certainly for such a critical application as predicting HIV drug resistance, we would want better performance.

One possibility 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 we explored), the lowest drug resistance classification was selected. That is, we considered the risk of trying a drug treatment that did not work to be lower than the risk of missing a potentially effective drug treatment.

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[c]	69%	68%	47
Most Accurate Single Network	31%	100%	31

Table 5: Comparison of scores for various majority voting schemes.

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: 8x8-0.5, 6x6-0.7, 6x6-0.5, 5x5-0.9, 5x5-0.8, and 5x5-0.6 (the number after the dash is the initial learning rate). The second scheme combined the best single network with the three most accurate networks: 6x6-0.7, 6x6-0.5, and 8x8-0.5. Again, those networks with 100% accuracy but very low coverage (the networks with 4x4 and 3x3 output matrices) were ignored. Our final scheme combine the results of the four networks with the best overall scores: 8x8-0.6, 10x10-0.9, 10x10-0.6, and 12x12-0.9.

Perrone claims 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.[13] In our case, all of the majority voting schemes outperformed the single best network (see Table 5). 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 significant improvement over the single best network (69%, 68%, and 47, respectively).

4 Conclusions and Further Work

This research explored the possibility of using self-organizing feature maps to predict drug resistance in HIV-1 infected patients based only on the peptide sequence of the HIV protease mutant strain. This differs from previous work which attempted to predict drug resistance based on structural features of the HIV protease.[4] This paper shows that the single best classifier found produces acceptable results (69% coverage and 68% accuracy), but to produce a predictive system suitable for clinical use, multiple networks configured in a majority voting scheme may be necessary. The best scheme was the six most

[c] Best single network was 8x8 output matrix, 0.6 initial learning rate, initial neighborhood of 8, 10 iterations; most accurate single network was 8x8 output matrix, 0.5 initial learning rate, initial neighborhood of 8, 10 iterations

accurate networks, with coverage of 84%, accuracy of 85%, and a score of 71. All majority voting schemes outperformed the single best network.

There are many opportunities for further research on using SOFMs for predicting drug resistance. In the case of HIV drug resistance, there are additional drugs (e.g., Indinavir and Nelfinavir) and drug combinations that may be explored. The difficulty with this work and work with other HIV treatments is the lack of publicly available clinical data (IC90 values). Christodoulou and Pattichis have also incorporated the use of confidence measures for weighting individual network results in majority voting schemes[8], which may be applied to the HIV drug resistance problem. Finally, SOFMs may be applied to the treatment of other retroviral diseases such as human T-cell leukemia virus (HTLV-1) and hairy cell leukemia (HTLV-2), as well as DNA viruses such as Hepatitis-B and Herpes.

References

1. James D. Watson, Michael Golman, Jan Witkowski, and Mark Zoller. *Recombinant DNA, 2nd Ed.*, pages 485–509. Scientific American Books, New York, 1992.
2. T. Kohonen. *Self-Organization Maps.* Springer-Verlag, Berlin Heidelberg, 1995.
3. Martin T. Hagan, Howard B. Demuth, and Mark Beale. *Neural Network Design*, pages 14.10–14.16. PWS Publishing Company, Boston, 1996.
4. Sorin Draghici, Lonnie Cumberland, and Ladislau C. Kovari. Correlation of hiv protease structure with indinavir resistance: a data mining and neural network approach. In *Proceedings of SPIE 2000*, volume 4057-40, Orlando, Florida, 2000.
5. Jouni Kaartinen, Yrjo Hiltunen, P.T. Kovanen, and Mika Ala-Korpela. Application of self-organizing maps for the detection and classification of human blood plasma lipoprotein lipid profiles on the basis of 1h nmr spectroscopy data. *NMR in Biomedicine*, 11:168–176, 1998.
6. Mikko Makipaa, Pekka Heinonen, and Erkki Oja. Using the som in supporting diabetes therapy. Helsinki University of Technology, Finland, June 4-6, 1997.
7. A.C.R. Santos-Andre, T.C.S.; da Silva. A neural network made of a kohonen's som coupled to a mlp trained via backpropagation for the diagnosis of malignant breast cancer from digital mammograms. In *IJCNN '99*, volume 5, pages 3647–3650, 1999.
8. C. I. Christodoulou and C. S. Pattichis. Medical diagnostic systems using ensembles of neural sofm classifiers. In *Proceedings of ICECS '99*,

volume 1, pages 121–124, 1999.

9. C. I. Christodoulou and C. S. Pattichis. Unsupervised pattern recognition for the classification of emg signals. *Biomedical Engineering, IEEE Transactions on*, 46(2):169–178, Feb 1999.
10. T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, et al. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.
11. Mark A. Winters, Jonathan M. Schapiro, Jody Lawrence, and Thomas C. Merigan. 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. *Journal of Virology*, 72(6):5303–5306, 1998.
12. Raymond F. Schinazi, Brendan A. Larder, and John W. Mellors. Mutations in retroviral genes associated with drug resistance: 1999-2000 update. *International Antiviral News*, 7(4):46–69, 1999.
13. M. P. Perrone. Averaging/modular techniques for neural networks. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 126–129, Cambridge, Massachusetts, 1999. MIT Press.