

Length-Independent, Signal Processing Features for Functional Classification and Prediction of Small Sequences of Lipase, Protease, and Isomerase

D. S. Warren & Kayvan Najarian

Computer Science Department, University of North Carolina at Charlotte
9201 University City Blvd, Charlotte, NC 28223, U.S.A.

E-mail: {dswarren, knajaria}@uncc.edu

ABSTRACT

We build upon previous results showing that length independent, signal processing features can characterize the functional families of small proteins. Here we introduce new features that improve the classification accuracy of our neural networks. Our sample case uses small lipase, protease, and isomerase proteins (containing between 100 and 200 amino acids). The amino acids were converted to signals by replacing them with hydrophobic, solubility, or binary encodings. Length independent features were then extracted from the signals. These features are discussed and used as input to neural networks for training and classification. These computationally cheap features can easily characterize some of the protein functional classes.

KEYWORDS

Neural Networks, Protein Sequencing, DNA Sequencing, Protein Function Prediction, Gene Function Prediction.

1. INTRODUCTION

DNA is the repository of information about much of life as we know it. Experiments such as the Human Genome Project lead to large amounts of DNA sequence data that needs to be classified. Many research efforts are underway trying to make sense of that information. DNA sequences fully determine the amino acid sequences in their expressed

proteins. The information contained in the DNA sequences contains redundancies, noise, and special signal sequences (such as terminators). DNA directly expresses RNA sequences, which can function on their own or be used as amino acid templates to create proteins. These amino acid sequences are the primary structures of the proteins. Much research has been put into determining the secondary structure given the primary structure of proteins [1]. An interesting question is whether the secondary structure is necessary to determine important properties of a protein. C.B. Anfinsen postulated that the primary sequence contains all the information needed for 3D structure (hence, biological function)[2]. In this paper, we will explore measures of the complete primary sequence as features for describing the functional class of the protein.

Proteins can be classified in many ways: solubility (albumins, histones, etc.), shape (globular or fibrous), 3-dimensional structure (helices, sheets, etc.), and biological function are examples. A natural and useful classification of proteins is according to biological function.

One broad classification of biological function is as an enzyme. Enzymes are proteins that facilitate chemical reactions and are named according to the substrate they act upon. Proteins that facilitate breaking down lipids (fats) are called lipases. A lipase enzyme needs

a lipophilic part of the protein to interact with the lipids. A lipase enzyme also needs an active site, which would be a non-contiguous sequence of amino acids (example: Ser126 His176 Asp206 form the active site for a lipase but are far apart in sequence number [3]). This information is buried within the primary structure and is not easily recognized. The buried, non-contiguous information would be hard to extract with simple measures; however, regions of lipophilicity and changes in regions can be measured. We will approach the classification problem by using measures of change along the primary sequence.

Several attributes are available within a given sequence related to composition and distribution of amino acids. An example of the distribution of amino acids being useful for biological function classification is that collagen contains a primary structure where each third residue is glycine with a preceding residue of proline or hydroxyproline [4]. Alternatively, the distribution of physical properties of amino acids allows a measure of distance to be used. With labels only, arginine is as different from lysine as it is from glycine. With physical properties, these amino acids can be ordered and their differences may even vanish (the physical property could be identical). Our features will be derived from physical properties of individual amino acids.

Classifying protein sequences according to function is not a new problem. Sequence databases have been used to train neural networks as classifiers (see reference [5] for a good review). The features used for classification included the complete primary structures and *N*-gram encodings of primary structures [6,7,8]. Complete primary structure encodings involve translating each amino acid to a unique identifier. The identifier can be a set of bit flags for example (21 bits, one for each amino acid and one for no amino acid).

Other encodings include a relative hydrophobicity scale [9] and accepted point mutation values [10]. The set of identifiers can then be presented to a neural network as either one long vector or, as in the *N*-gram method, as a set of fixed length vectors.

Other statistical methods have also been applied to the classification of protein structures [11,12]. These techniques are “nearest neighbor” type methods where the nearness is a measure of distances as changes in primary structures. These studies do not address the question of whether or not there are features other than the entire primary structure that can be used to classify proteins. This is an important point since for a given protein classification there is a large range of amino acid sequence sizes.

Neural networks are valuable tools, closely related to statistical tools, used for classifying data. The multilayer nature of neural networks allows them to discover non-linear higher order correlations among the data. Neural networks consist of a set of interconnected “neurons”, decision units that are activated based on their inputs. The inputs are weighted, the weights being determined during fitting of the input data. To assess the suitability of a given neural network, the input data is divided into a training set for creating the weights and a testing set for evaluating the neural network. Multiple layers can be used where the outputs from one layer are the inputs to the next. There are many variables associated with developing a neural network to model a problem. There are choices in the number of neurons and number of layers, the learning algorithm, and the activation functions of the neurons. In this problem, the incoming data has an associated desired outcome so a supervised learning algorithm is desired. Neural networks, once properly trained, are fast classifiers.

An important goal of the general field called bioinformatics is to utilize easy to obtain data, such as protein sequences, to predict relatively harder to obtain data, such as structure or biological function. The rapid decrease in cost for sequencing, DNA and protein, has created this imbalance of data. While there is no doubt that the primary sequence of a protein contains all the necessary information needed to determine the purpose (biological function within the context of a biological system) of a protein, so far there has been no clear method of extracting that information through models. We have been applying a signal model to the amino acid sequence in order to extract features for functional classification. Three protein families are considered along with the further restriction of only looking at primary structures with 100 to 200 amino acids.

The present paper is organized as follows: Section 2 describes the experimental procedures used in each classification process and defines the measures (features) used. In Section 3, the results of each classification task are given and discussed. Section 4 concludes this study with some directions for our future research.

2. EXPERIMENTAL

The protein primary structures used in this study were obtained from the Swiss Protein Sequence database [14]. This database is freely available over the World Wide Web. A keyword search found 295 lipases, 415 proteases and 426 isomerases having 100 to 200 amino acid units in length. Many of these are just fragments so we removed the fragments before extracting features in this study. This yielded 295 isomerases, 254 lipases, and 148 proteases. The input size range was chosen to use as test cases for the various features due to the large number of samples available.

The sequence data on the Swiss Protein Sequence database use the single letter per amino acid notation. The primary structures were preprocessed such that each amino acid was replaced by experimentally determined physical properties of the amino acids. These properties are described later in this section.

We used the Bayesian Regularization BackPropagation algorithm for training[15,16]. Training is the process of fitting the neural network weights and biases to the data (the training data in this case). BackPropagation refers to the method of updating the weights using a gradient descent method and updating the weights going backward (from the output toward the input) during a training cycle. As there are many variations for function minimization there are many variations of the training algorithm for neural networks. Regularization adds another parameter to the optimization process. This parameter controls the requirements for “smoothness” such that the optimization can be constrained to keep a smooth function at the expense of error or to minimize the error only. The Bayesian framework considers the neural network weights as random variables that can be updated using Bayes rule. Bayesian Regularization allowed choosing much smaller neural networks with reasonable training times and good generalization.

The data were randomly assigned to training or testing sets (roughly 70% training and 30% testing). The numbers assigned to each group for each class of data are shown in Table 4.1.

Lipase		Protease		Isomerase	
Train	Test	Train	Test	Train	Test
190	64	111	37	206	69

Table 4.1. Data Distribution for Sequences of 100 to 200 Amino Acids

The neural networks were created in MATLAB 6.0 using the Neural Network Toolbox [17]. The type of neural network used was a feedforward-backpropagation network with various numbers of hidden nodes in order to find the optimal network size. All nodes use the tansig activation function. For each classification, Isomerase, Lipase, or Protease, a neural network classifier was created and trained. The neural networks were trained to classify their chosen enzyme as +1 and the other two enzymes as -1.

The input to the neural networks were a set of 46 length independent signal processing features. The features are described first and then the encoding of the amino acids used for deriving the features.

Assume that the characteristics of a given protein sequence is expressed as a vector x , where x_i represents the code assigned to the amino acid in location i of the sequence. Also, let $d = \Delta x$ represent the vector of variations in x , i.e. $d_i = x_i - x_{i-1}$. Moreover, define g as the vector of variations in d , i.e. $g = \Delta d$. Then use the above definitions for x , d and g to define the following fundamental statistical parameters:

$$S_0 = \sqrt{(\sum x_i^2)/n}$$

$$S_1 = \sqrt{(\sum d_i^2)/(n-1)}$$

$$S_2 = \sqrt{(\sum g_i^2)/(n-2)}$$

Where n is the number of amino acids in a sequence. Now, define Complexity and Mobility factors as:

$$\text{Complexity} = \sqrt{(S_2^2/S_1^2) - (S_1^2/S_0^2)}$$

$$\text{Mobility} = S_1/S_0$$

As can be seen, Mobility addresses the first order variations of the signal, while Complexity deals with the first and the second order variations.

The final measure that is used in this research is the fractal dimension. Fractal dimension describes the self similarity of a signal by scaling and shifting of itself. Several methods of estimating the fractal dimension of a sequence were reported in the signal processing literature. Among all these techniques, the Higuchi algorithm [14] is known to be the most accurate and efficient method of estimating the fractal dimension, and as a result, is used in this research. From a time series with N points a set of k subseries are obtained each with a different step size or interval size (where $m = 1, 2, 3, \dots, k$):

$$X_k^m : X(m), X(m+k), X(m+2k), \dots, X(m + [\frac{N-m}{k}] * k)$$

The length of the curve X_k^m is:

$$\frac{\left(\sum_{i=1}^{[\frac{N-m}{k}]} |X(m+ik) - X(m+(i-1)*k)| \right) * \frac{N-1}{[\frac{N-m}{k}] * k}}{k}$$

Plotting the curve lengths versus $\log_2(k)$ gives an estimate for the fractal dimension as the negative slope of this plot. This is only an estimate and our sequences are much shorter than the sequences used to justify Higuchi's algorithm. With this caveat in mind, we only use these numbers for comparison among sequences so we empirically test the usefulness of this measure to see if the errors are biased.

The first set of 40 features measures the complexity and mobility of each amino acid. The amino acid sequence is converted to 0's and 1's for each amino acid (20 amino acids yield 20 binary sequences with 2 measures each for a total of 40 measures). An example encoding for Alanine, A, is shown below:

Amino Acid Sequence: AVWCEVWCEAA
 Binary Sequence: 10000000011

The second set of 3 features is complexity, mobility, and fractal dimension for each

sequence as represented by the amino acid hydrophobicity value[18]. This value is a measure of the free amino acids dislike for an aqueous environment. Each amino acid is replaced by its experimental hydrophobicity value to create a new sequence. The features are calculated off of the new sequence.

The third set of 3 features is similar to the above 3 features except that the sequence used is the experimental solubility values from another source[19].

3. RESULTS

A comparison of various encodings and their resulting features are shown in Figures 1-6. These show plots of the average measure found with tick marks at the plus and minus one standard deviation of the measure. This shows the relative range and tightness of the measures. Each plot compares a single encoding and single measure for each of the three classes of protein studied.

The first two figures show fractal dimension calculated with the solubility and hydrophobicity encodings. One immediate observation is that the hydrophobicity values are much tighter than the solubility values (note that the range of Y values is not the same for all figures). Another observation is that the calculated fractal dimensions are unreasonable since they are over 2.0 (they should be between 1.0 and 2.0). The error in fractal dimension could be due to the short sequences. However, removing the fractal dimension values from the set of features did not improve the neural network classifiers so they were kept.

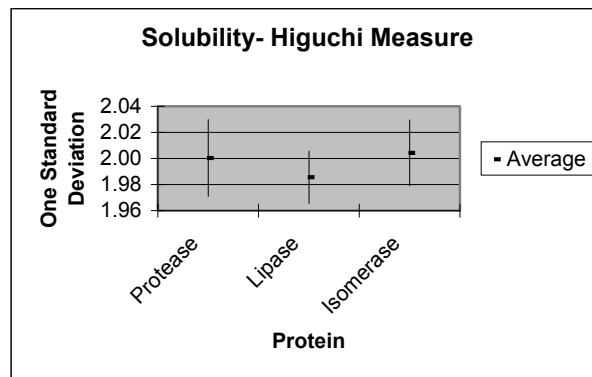


Figure 1: Plot showing one standard deviation of Higuchi fractal dimension measures for the solubility encoding of three protein classes studied.

Another observation is the amount of overlap of these measures: the Isomerase and Lipase enzymes have the smallest overlap whereas the Protease enzyme measures overlap both Isomerase and Lipase measures almost equally. This pattern is similar for both the solubility and the hydrophobicity encoding. An interpretation of the relationship of these measures is that Lipase enzymes are not as self similar, they are more linear, or they are smoother than Isomerase enzymes with respect to hydrophobicity or solubility. This pattern also indicates the difficulty of differentiating Protease enzymes from Lipase or Isomerase enzymes with this measure.

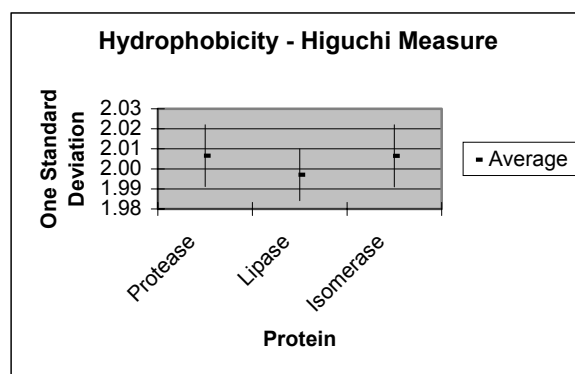


Figure 2: Plot showing one standard deviation of Higuchi fractal dimension measures for the hydrophobicity encoding of three protein classes studied.

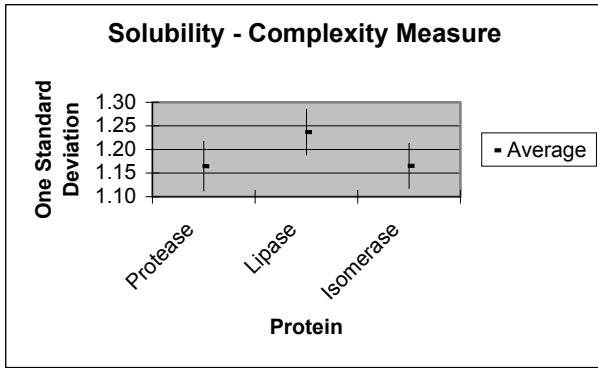


Figure 3: Plot showing one standard deviation of complexity measures for the solubility encoding of three protein classes studied.

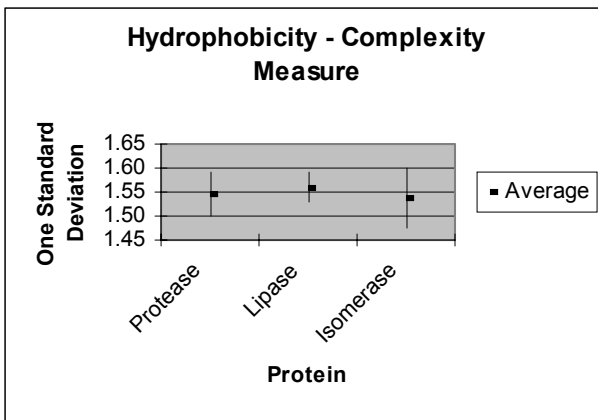


Figure 4: Plot showing one standard deviation of complexity measures for the hydrophobicity encoding of three protein classes studied.

The complexity measures shown in Figures 3 and 4 shows a large difference with the encoding used. The solubility encoding differentiates the Lipase enzyme from the Protease and Isomerase enzymes but the hydrophobicity encoding does not readily differentiate any enzyme (they all have similar averages and their first standard deviations are within the first standard deviation of the Isomerase enzyme).

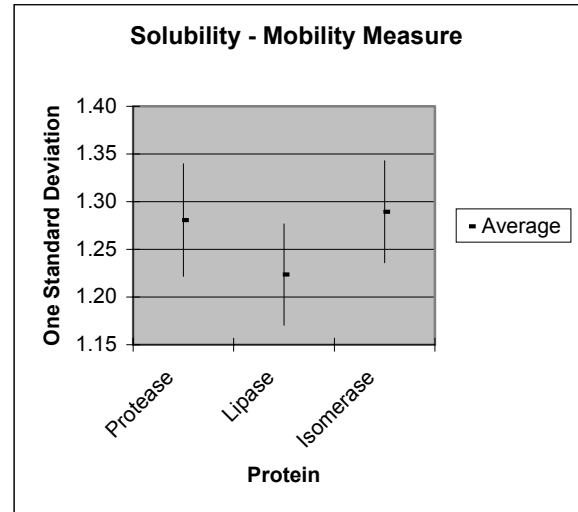


Figure 5: Plot showing one standard deviation of mobility measures for the solubility encoding of three protein classes studied.

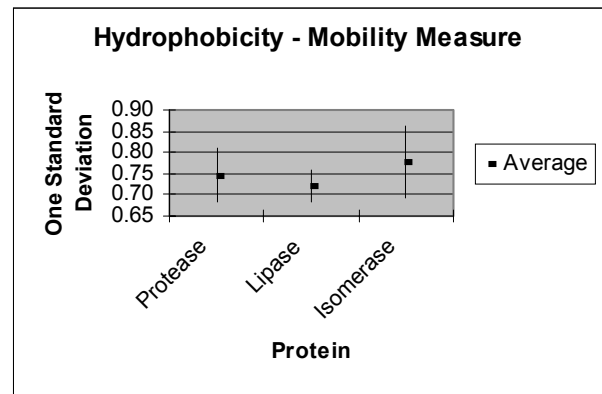


Figure 4.6: Plot showing one standard deviation of mobility measures for the hydrophobicity encoding of three protein classes studied.

The mobility measures are similar to the complexity measures in that the Lipase enzymes are the most easily differentiated. Also, the solubility encoding shows greater differentiation for Lipase than the hydrophobicity encoding.

The neural classifiers used for length-independent classification, the input vectors form a set of 46 measures developed from their corresponding sequence. Each measure is calculated for a given protein sequence (with arbitrary length), and fed to a neural network. These measures are calculated for sequences formed by the hydrophobicity encoding and

solubility encoding of the protein sequence. These measures are calculated only on the actual signal portion of the sequence, i.e. there is no zero padding.

Many network architectures were explored but only the best performing with the least number of neurons are discussed. The network architecture was trained three times using the same training/test data sets to determine the variability due to randomized starting weights for the network. The percentage correctly classified are shown in Tables 2 – 5.

Isomerase – 5,2 50e	Run 1	Run 2	Run 3
Isomerase Train	85.9	84.0	96.1
Isomerase Test	81.2	73.9	81.2
Other Train	81.1	87.0	93.7
Other Test	78.2	86.1	81.2

Table 2. Isomerase network: percentage correctly classified.

The Isomerase classification network has 46 input nodes, 5 hidden nodes, and 2 output nodes and was trained for 50 epochs during each run. The resulting networks are not overtrained (low difference between testing and training results) and generalize well (good results for testing data).

Lipase – 5,2 20e	Run 1	Run 2	Run 3
Lipase Train	93.7	97.9	88.4
Lipase Test	87.5	96.9	87.5
Other Train	99.1	99.4	97.2
Other Test	97.2	93.4	98.1

Table 3. Lipase network: percentage correctly classified.

The Lipase network is even better performing than the Isomerase network. It also has 46 input nodes, 5 hidden nodes, and 2 output nodes. It was trained for only 20 epochs (this network overfitted the data when trained for 50 epochs). The resulting network generalizes well and is not overtrained.

An optimal Protease classification network was not found. The networks were easily overtrained even when relatively large numbers of neurons were used (up to a 10,10,2 network was tried and also a 20,2 network). Typically, the training data was twice as accurate as the testing data for the Protease sequences. In Table 4, run 1, the protease training data is 48.6% correct and the testing data is 24.3% accurate (2:1). The network in Table 4.4 was trained for only 20 epochs (attempting to not overfit the data) but is comparable in accuracy to the network in Table 5 which was trained for 300 epochs. The training data for run 1 of Table 5 is 97.3% correct and the testing data is only 56.8% correct (still roughly 2:1).

Protease – 5,2 20e	Run 1	Run 2	Run 3
Protease Train	48.6	76.6	58.6
Protease Test	24.3	51.4	40.5
Other Train	93.4	95.2	97.7
Other Test	92.5	88.0	89.5

Table 4: Protease network trained for 20 epochs.

Protease – 5,2 300e	Run 1	Run 2	Run 3
Protease Train	97.3	66.7	93.7
Protease Test	56.8	40.5	48.6
Other Train	100	97.5	100
Other Test	87.2	88.7	87.2

Table 5: Protease network trained for 300 epochs.

The variability among the runs is much greater for the Protease networks than for the Isomerase or Lipase networks. These networks simply cannot generalize very well; but, it should be noted that the number of training samples for Protease is much smaller than for the other enzymes.

A final network was created and tested using the best of the networks above. This network first picked out the Lipases, any non-classified data were passed to the Isomerase classifier, and finally to the Protease classifier. All data

was fed to this network and the results are shown in Table 6.

	Isomerase Data	Lipase Data	Protease Data
As Isomerase	257	1	8
As Lipase	4	248	5
As None	8	4	9
As Protease	6	1	126

Table 6: Results for Classifier Built from Three Best Single Trained Classifiers

The feature vectors not classified by any of the three neural networks are counted as “None”. The networks used in Table 6 show an overall 93% correct classification.

4. DISCUSSION

From the results mentioned in the previous section the following general, the following observations can be made:

- a. The Isomerase and Lipase enzymes (of length 100-200) are easily classified using these 46 features whereas the Protease enzymes are not as easily classified.
- b. The Isomerase and Lipase classifiers generalize well and are not overtrained (they perform equally well with training and testing data).
- c. The new feature of complexity and mobility calculated for an amino acid binary encoded sequence resulted in much better accuracy over the previous studied classifiers [20].
- d. These computationally very inexpensive features are shown to be useful for characterizing biological function of protein sequences in these cases.

5. CONCLUSIONS

This study explored length-independent signal processing features for classification of protein sequences according to biological function. The features show promise for differentiating these classes. The resulting multilayer classifier is fairly accurate and reliable over the range of data available for this study. While this study used solubility and hydrophobicity values, there is the possibility of other chemically more meaningful encodings that might create better signals for processing.

In future work, we will continue exploring various encodings and signal processing features to extend these results to the larger sequences and other classifications.

REFERENCES

- [1] Burkhard Rost, “Evolution teaches neural networks to predict protein structure”, in *Scientific Applications of Neural Nets*, Ed. John W. Clark, Thomas Lindenau, and Manfred L. Ristig, pp. 207-223, Springer, Heidelberg, 1999.
- [2] Anfinsen, C.B., “I. Self Assembly of Macromolecular Structures: Spontaneous Formation of the Three-Dimensional Structure of Proteins”. 27th Symposium of the Society for Developmental Biology, *Developmental Biology Supplement 2*, 1968.
- [3] R. Bott, J. W. Shield, and A.J. Poulou, “Protein Engineering of Lipases” in *Lipases Their Structure, Biochemistry, and Applications*, Ed. Paul Woolly and Steffen B. Petersen, Cambridge University Press, New York, NY, 1994.
- [4] Robert K. Murray, Daryl K. Granner, Peter A. Mayes, and Victor W. Rodwell, “Harper’s Biochemistry”, 23rd Edition, Appleton & Lange, 25 Van Zant Street, East Norwalk, Connecticut, 1993.
- [5] C. Wu, “Artificial Neural Networks for Molecular Sequence Analysis”, *Computers Chem. Vol. 21, No. 4*, pp. 237-256, 1997.
- [6] C. Wu, S. Shivakumar, H. Lin, S. Veldurti, and Y. Bhatikar, “Neural Networks for

Molecular Sequence Classification”, Mathematics and Computers in Simulation, Vol. 40, p. 23, 1995.

[7] C. Wu, G. Whitson, J. McLarty, A. Ermongkonchai and T. Chang, “Protein Classification Artificial Neural System” in Protein Science, vol 1, p. 667, 1992.

[8] C. Wu, “Classification Neural Networks for Rapid Sequence Annotation and Automated Database Organization” in Computers and Chemistry, vol. 17, p. 219, 1993.

[9] Y. Xin, T. Carmeli, M. Liebman, and G. Wilcox, “Use of the Backpropagation Neural Network Algorithm for the Prediction of Protein Folding Patterns”, in Proc. Of the Second International Conf. On Bioinformatics, Supercomputing and Complex Genome Analysis, editors H. Lim, J. Fickett, C. Cantor and R. Robbins, pp. 359-375. World Scientific, River Edge, NJ.

[10] B. Rost and A. Sander, “Prediction of Protein Secondary Structure at Better than 70% Accuracy”, Journal of Molecular Biology, vol. 232, pp. 584-599, 1993.

[11] M. van Heel, “A new family of powerful multivariate statistical sequence analysis techniques” in Journal of Molecular Biology, vol. 220, p. 877, 1991.

[12] N. Harris, L. Hunter, and D. States, “Megaclassification: discovering motifs in massive data streams” in Proceedings of Tenth National Conference on Artificial Intelligence. AAAI Press. Menlo Park, CA, 1992.

[13] Higuchi, T., “Approach to an Irregular Time Series on the Basis of the Fractal Theory”, Physica D, Vol. 31, pp. 277-283, 1988.

[14] Swiss Protein Sequence Database <http://www.ebi.ac.uk/swissprot/>

[15] F.D. Foresee, and M.T. Hagan, “Gauss-Newton Approximation to Bayesian Regularization”, Proceedings of the 1997 International Joint Conference on Neural Networks, 1997.

[16] D.J.C. MacKay, “Bayesian Interpolation”, Neural Computation, Vol. 4, No. 3, pp. 415-447, 1992.

[17] Matlab 6 Release 12, The Mathworks, Inc. <http://www.mathworks.com>.

[18] Black, S.D., and Mould, .R., "Development of Hydrophobicity Parameters to Analyze Proteins Which Bear Post- or Cotranslational Modifications", Anal. Biochem. Vol. 193, pp. 72-82, 1991.

[19] The Merck Index, Merck & Co. Inc., Nahway, N.J., 11(1989); CRC Handbook of Chem.& Phys., Cleveland, Ohio, 58(1977).

[20] D.S. Warren & Kayvan Najarian, “Function Prediction of Fixed-Length Protein Using Neural Networks”, in Proceedings of the World Congress on Systemics, Cybernetics, & Informatics, Orlando, Florida, U.S.A., July 2002.