Amino Acid Encoding Schemes for Machine Learning Methods

Masood Zamani
School of Computer Science
University of Guelph
Guelph, Canada
mzamani@uoguelph.ca

Stefan C. Kremer
School of Computer Science
University of Guelph
Guelph, Canada
skremer@uoguelph.ca

***Abstract*—In this paper, we investigate the efficiency of a number of commonly used amino acid encodings by using artificial neural networks and substitution scoring matrices. An important step in many machine learning techniques applied in computational biology is encoding the symbolic data of protein sequences reasonably efficient in numeric vector representations. This encoding can be achieved by either considering the amino acid physicochemical properties or a generic numerical encoding. In order to be effective in the context of a machine learning system, an encoding must preserve information relative to the problem at hand, while diminishing superfluous data. To this end, it is important to measure how much an encoding scheme can conserve the underlying similarities and differences that exist among the amino acids. One way to evaluate the effectiveness of an amino acid encoding scheme is to compare it to the roles that amino acids are actually found to play in biological systems. A numerical representation of the similarities and differences between amino acids can be found in substitution matrices commonly used for sequence alignment, since these substitution matrices are based on measures of the interchangeability of amino acids in biological specimens. In this study, a new encoding scheme is also proposed based on the genetic codon coding occurs during protein synthesis. The experimental results indicate better performances compared to the other commonly used encodings.**

***Keywords*-machine learning; artificial neural networks; amino acids; substitution matrix;**

I. Introduction

Recent developments in high throughput sequencing have made available vast databases of DNA and protein sequences. The phenomenal increase in the rate of acquisition of data, has thus-far not been matched by complementary analytic techniques. Machine Learning (ML) techniques offer a promising opportunity to correct this imbalance, by extracting useful rules and patterns from data in an automated fashion, and allowing researchers to apply these rules and patterns to novel data in order to classify, predict and generate a variety of features of interest.

One of the first things that a ML user must address when tackling a new problem is that of encoding the data in a format suitable for processing with a ML tool. The encoding of input data can have a critical effect on the applicability and resulting quality of ML approaches. An effective encoding must, first and foremost, preserve the information required by the ML technique in order to solve the problem at hand. Specifically, any inputs that should generate different outputs in the problem space must be distinguishable. Next, it is desirable for an encoding to attenuate any noise in the input data, effectively shielding the subsequent processing from extraneous data. This can be achieved by selecting an encoding in which groups of input patterns that are intended to generate similar output patterns are represented in a similar, consolidated way. It is important to note that both *distinguishability* and *consolidation* are problem specific concepts. For example, the important data required to solve one problem may be noise in the context of another. This proves to be a challenge in any work attempting to identify a "best" encoding scheme [1]–[3].

The task of encoding input patterns for a ML system can be particularly challenging in problem domains such as bioinformatics where data exemplars often poses a structural or sequential component and can vary in size from one test-case to another. In this paper, we focus on a part of the encoding of protein sequences.

Protein sequences are composed of chains of amino acids and vary in length. The varying length nature of these chains is often addressed by using either a sliding-window approach [4], [5] or a recursive network [6]. Within the sliding window or recursive network, one still needs to encode the individual amino acids themselves. In this paper, various amino-acid encodings are evaluated based on their capability to simulate commonly used substitution matrices via best-trained neural networks. The substitution matrices such as Point Accepted Mutation (PAM) [7] and Blocks Substitution Matrix (BLOSUM) [8] are the standard benchmark tools to measure the sequence similarities of protein sequences. Our objective in this paper is to shed some light on the effectiveness of various amino-acid encoding schemes. It is our hope that a reader, informed by our results, will be able to judiciously and confidently select the best encoding method.

The paper is organized as follows. In Section II, we describe our method for evaluating encodings, and the premises of our work. In Section III, we detail the experimental method used, and in Section IV, the results obtained. Next, in Section V we analyze our results and provide some

978-1-4577-1613-3/11/$26.00 ©2011 IEEE

comments on different encodings. Finally, in Section VI, we present concluding remarks, limitations and future work.

II. Methodology

In order to evaluate different amino-acid encoding methods, we needed to develop a benchmark test-set. Since there have been many efforts to use a protein's primary sequence to predict a wide variety of properties of that protein (interactions, disulphide bonds, shape, secondary structure, etc.), there are many different possible test-sets to choose from. Our goal was to try to select the most generic possible test set in the hope that our results would be applicable to the largest variety of ML problems.

Our test-set selection is based on two assumptions. First, that the goal of any ML approach is ultimately to predict the function of a protein. Most ML methods that are applied to protein sequences are used to work on a part of this problem, whether it be by elucidating some aspect of the shape of a protein, or its chemical properties. Our second premise is that this biological function tends to be preserved across related proteins in related organisms, and thus that the conservation of specific amino acids in specific locales in a protein is a valid objective to predict in a comparison of encoding schemes. Thus, we decided to base our test-set on the substitutability of specific amino acids in biological organisms. Specifically, we decided to predict similarity matrices which measure just this type of substituability.

A. Similarity Matrices

Similarity Matrices were developed to help in the task of aligning protein sequences. Sequence alignments can reveal the underlying relations existing among protein sequences such as their evolutionary distances, functions and structures. In general, sequence alignments fall in two main categories: (1) global alignment by which a pair or multiple sequences are aligned by considering the entire length of the sequences. The aligned sequences could be originated from the same ancestor or different evolutionary paths and families, and (2) in local alignments the aim is to find out the segments of protein sequences which are related in terms of similarity or evolutionary stand points. The main component in both types of sequence alignment is the evaluation of sequence similarities. This is accomplished by a dynamic programming algorithm which attempts to find the most likely alignment between the amino acids in two (or more) sequences. The probability measure that the algorithm uses to measure and guide its success, is based on the likelihoods of substituting, deleting and inserting symbols in one sequence until it matches the other. These likelihoods have been the subject of much research.

The mutation data matrices or Point Accepted Mutation (PAM) matrices proposed in [7] are considered one of the pioneering and standard works in measuring sequence similarities based on likelihoods for sequence alignment methods. PAM matrices are derived based on an empirical dataset of 1572 mutations in 71 groups of closely related proteins. PAM substitution matrices are generated by measuring substitution frequencies in sequences that have been aligned by human experts with similarities of 85% or more. PAM matrices are indexed with a number (e.g. PAM10) which indicates that the substitution probabilities were derived from the sequences that have 10 mutations per one hundred amino acids. Thus, large indices are based on more divergent datasets and should be applied to such, as well.

An alternative to the PAM approach, the Blocks Substitution Matrix (BLOSUM) proposed in [8], was a solution for detecting sequences with higher evolutionary distances (or lower homology) than previously detected by Dayhoff substitution matrices. To obtain the BLOSUM matrix, a frequency table is derived from a database of ungapped blocks by counting the relative frequencies of amino acids and their substitution probabilities. Each block represents the conserved region of a protein family. Then a logarithm of odds matrix for 210 possible substitutions based on the 20 amino acids is calculated. Each obtained log-odds score in BLOSUM is the ratio of two amino acid's frequency of appearance in natural sequences to the two amino acids if they appeared randomly based on their independent frequencies. BLOSUM matrices are also given numeric indices. For example BLOSUM50 is derived from sequences that are 50% homologous and higher numbers indicate higher sequence similarity.

An important difference between PAM and BLOSUM matrices is that in PAM, mutation frequencies are estimated from sequences that are closely related proteins but in BLOSUM the frequencies are computed from the blocks that are highly conserved regions, regardless of the evolutionary distances of the sequences of origin.

B. Substitution Matrices as a Benchmark

For the reasons outlined above, we selected the task of predicting substitution matrix likelihoods as our benchmark problem. Our premise is that if an encoding scheme allows a ML technique to effectively predict the likelihoods of amino acid substitutions that occur in nature, that same encoding scheme can be expected to perform well on a number of different ML tasks involving the prediction of protein function, structure, composition, etc..

In order to evaluate the encodings in this manner, we needed to select an ML method to tackle this problem. Since, the likelihoods in the matrices are numerical values (as opposed to categorical), we required a ML regression method. To keep things simple we used a "vanilla" backprop network and measured the effectiveness of the encoding scheme based on the speed of learning and the final error.

III. Experiment

In this section, we explain the amino acid encoding schemes and the experimental setups for evaluating the encoding schemes. The evaluation of each encoding scheme is performed by using a multilayer feed-forward neural networks and substitution scoring matrices.

A. Learning Substitution Probabilities

Artificial Neural Networks(ANNs) [9] have proven themselves as powerful machine learning tools in a number of research areas including various forms of function approximation. In this study, we used a multilayer feed-forward neural networks that has one hidden layer with 30 neurons. The output layer has one output and the number of input layer's neurons is varied and depends on the selected encoding scheme. For each encoding scheme, the supervised training of the neural network is performed by gradient descent back-propagation learning [10]. In each training epoch, 210 pairs of encoded amino acids, after normalizing to $\{-1, +1\}$, are fed to the neural network and the output errors computed by comparing to the corresponding values of the amino acid pairs from substitution matrices. In the experiment, each encoding scheme is applied separately on all possible (unordered, non-reflexive) 210 pairs of amino acids for generating training data to train five neural networks whose outputs are the approximation of five substitution matrices. The substitution matrices used in this study are BLOSUM50, BLOSUM62, BLOSUM80, PAM120 and PAM250. For each encoding scheme, the trainings of neural networks are repeated, and the average root mean square error ($RMSE$) is computed in 10 runs due to in each run (a complete training phase), a trained neural network may be achieved by a different number of training epochs and a varied final $rmse$ error. For each encoding scheme, the training is performed until a stop criterion is reached. Since our goal is to evaluate the effectiveness of a variety of encoding schemes with different expected performances, it would be counter productive to define a simple error threshold as our criterion (as is conventional with many ANN applications). Instead, we attempt to identify the convergence of the network, defined as a very small reduction in error (or an increase in error) over 5 training epochs. Specifically, the stop criterion is defined as follows:

$$E(t-5) - E(t) \leq \Delta E. \quad (1)$$

In (1), E, t denoted for the average $rmse$ and epoch number respectively, and ΔE was set empirically to $10e-7$.

B. Encoding Schemes

In this paper, we evaluated fifteen amino acid encoding schemes as described in Table I. Each amino acid encoding scheme generates twenty n-dimensional vectors in Euclidean space that correspond to the twenty amino acids [1]. Some of the encoding schemes are the commonly used techniques in the literature such as orthogonal [11], BLOMAP [2], polar distribution [12] and physicochemical properties [13], according to the classification of amino acid conservation proposed in [14]. Except our new proposed encoding scheme #12, the other encoding schemes which may not been commonly used and evaluated in this paper are the grouping of various properties of amino acids similar to the encoding schemes introduced in [15], [16]. However, it is worthwhile to use a few combinations of the amino acid features to examine the efficiency of the encoding schemes in machine learning applications. These properties range from amino acid preferences to form different types of secondary structures, molecular properties such as formula and molecular weight. A comprehensive list of amino acid features can be found in the AAindex database [17] which contains around five hundred features.

Note that, in this study we tried to identify effective encoding schemes by evaluating them using scoring matrices, as opposed to constructing an encoding scheme specifically designed to compute substitution scoring matrices such as [3]. In this regard, we hope to get a general assessment of the performance of these encoding schemes across a broad class of problems, rather than a specific method to perform optimally on the scoring matrix problem. In addition, we proposed a new encoding scheme according to genetic codon that are the building blocks of amino acids. Each codon is a combination of three naturally selected nucleotides from the set $\{A, T, G, C\}$ that create one of the twenty possible amino acids, and in addition an amino acid can be derived from a number of codons. For instances, the amino acid Valine can be derived by one of the codons from the set $\{GTT, GTC, GTA, GTG\}$. A graph representation of the codons leading to form the amino acid Valine is shown in Figure 1. This graph is created by using 4 nodes representing the nucleic acids $\{A, T, G, C\}$, and then adding directed vertices for each successive pair of nucleic acids in one of the four Valine encodings $G \rightarrow T, T \rightarrow T; G \rightarrow T, T \rightarrow C; G \rightarrow T, T \rightarrow A; G \rightarrow T, T \rightarrow G$. Next, the graph is represented in a 4×4 connectivity matrix, and the matrix is converted to a string of sixteen 0's and 1's. These steps are performed for the other amino acids with different sets of codons.

IV. Results

The fifteen amino acid encoding schemes were used separately to encode 210 amino acid pairs for training the multilayer feed-forward neural network as explained in Section III. Each encoding scheme is evaluated with five neural networks for the five substitution matrices. The experimental results indicate that the four encoding schemes #11, #12, #14 and #15 resulted in better training accuracies compared to the other encoding schemes' as described in Table I. The best four encoding schemes were selected based on the average $rmse$ when the training stop criterion is met.

Table I
THE SPECIFICATIONS OF THE FIFTEEN AMINO ACID ENCODING SCHEMES

Encoding scheme#	Description
1	The presences of nine physicochemical properties [14] of AAs are mapped in a string of 0s and 1s, and converted to an integer.
2	The presences of nine physicochemical properties [14], and six more physical properties [13] called extra tiny, pentagonal, hexagonal, forked and crossed are mapped in a string of 0s and 1s, and converted to an integer.
3	Based on AA preferences to form the secondary structures (β sheet, α helix).
4	AAs are represented based on the proposed codon encoding, and each of the four adjacent 0s and 1s is converted to an integer.
5	Based on BLOMAP [2] which converts BLOSUM62 to a 20×5 matrix.
6	AAs are indexed from 1 to 20, and the indexes are represented in 5 bits.
7	AAs' molecular properties including frequencies, volume, partial specific volume, hydration and R-group.
8	Information related to AAs' properties such as exposure to solvent, and the number of C, H, N, O, S atoms.
9	The 7 attributes of Chou-Fasmann's propensity values [18].
10	The presences of nine physicochemical properties [14] of AAs are mapped in a string of 0s and 1s.
11	The presences of nine physicochemical properties [14], and six more physical properties [13] called extra tiny, pentagonal, hexagonal, forked and crossed are mapped in a string of 0s and 1s.
12	The proposed codon coding scheme represented in a string of sixteen 0s and 1s.
13	The combination of all AAs' properties from the 3 encoding schemes 7,8,9.
14	Orthogonal or sparse encoding [11] string.
15	The combination of orthogonal [11] and AA polar distribution [12].

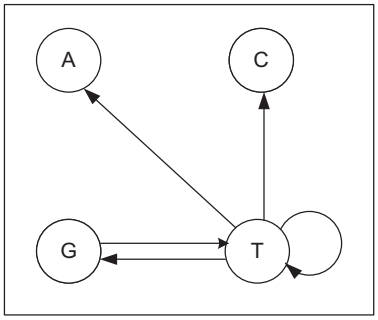

Figure 1. The graph representation of the codons forming the amino acid Valine based on 4 nucleotides $\{A, T, G, C\}$.

The comparison of the 4 encoding schemes in terms of the average $rmse$ and the number of required epochs for training are shown in Figures 2 and 3 respectively.

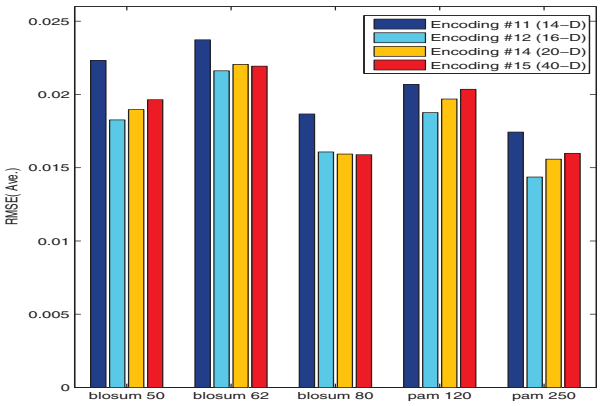

Figure 2. The comparison of average root mean square errors for encoding schemes #11, #12, #14 and #15 in 10 runs.

The complete training results of all encoding schemes for BLOSUM62, BLOSUM80 and PAM250 are shown in Tables II, III and IV respectively. The training results of encoding scheme #5 (BLOMAP) is omitted from Tables III and IV since the encoded amino acids in 5-dimensional vectors for BLOSUM80 and PAM250 were not provided as we obtained the 5-dimensional vectors corresponding to each amino acid for BLOSUM62 matrix from [2].

V. DISCUSSION

The experimental results for each encoding scheme are evaluated according to the two criteria for NN's training performance. The criteria are the number of training epochs (training time) to reach convergence and the approximation accuracy (lower $rmse$ errors). For each encoding scheme, the two criteria were investigated with different substitution matrices used for the neural network input.

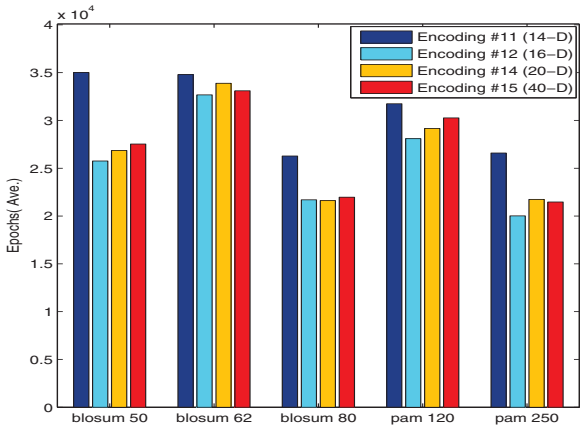

Figure 3. The comparison of average training epochs for encoding schemes #11, #12, #14 and #15 in 10 runs.

Table II
THE TRAINING RESULTS OF ALL ENCODING SCHEMES EVALUATED BY BLOSUM62

Encoding#	Dimension	Epoch	Ave. $rmse$
1	1	295	2.504639
2	1	415	2.502990
3	2	175	2.408614
4	4	8910	0.644371
5	5	120	2.286987
6	5	2680	0.580234
7	6	468705	0.350314
8	7	217025	0.900421
9	7	329990	0.253628
10	9	7050	0.693111
11	**14**	**34790**	**0.023730**
12	**16**	**32680**	**0.021617**
13	20	208785	0.282972
14	**20**	**33890**	**0.022051**
15	**40**	**33105**	**0.021930**

Table III
THE TRAINING RESULTS OF ALL ENCODING SCHEMES EVALUATED BY BLOSUM80

Encoding#	Dimension	Epoch	Ave. $rmse$
1	1	265	2.507192
2	1	410	2.503600
3	2	160	2.390761
4	4	7300	0.742286
6	5	1025	0.868625
7	6	492640	0.368557
8	7	341760	0.844570
9	7	371665	0.229421
10	9	5945	0.678370
11	**14**	**26280**	**0.018665**
12	**16**	**21700**	**0.016077**
13	20	349275	0.195062
14	**20**	**21615**	**0.015932**
15	**40**	**21975**	**0.015886**

Table IV
THE TRAINING RESULTS OF ALL ENCODING SCHEMES EVALUATED BY PAM250

Encoding#	Dimension	Epoch	Ave. $rmse$
1	1	520	1.904371
2	1	1275	1.896692
3	2	235	1.826631
4	4	18695	0.500904
6	5	2845	0.517455
7	6	417180	0.320489
8	7	612020	0.540101
9	7	267385	0.278978
10	9	5475	0.717099
11	**14**	**26590**	**0.017434**
12	**16**	**20020**	**0.014364**
13	20	338070	0.146217
14	**20**	**21745**	**0.015584**
15	**40**	**21470**	**0.015980**

At first glance, when the dimensionality of encoding schemes (the number of features) is increased, the training accuracies may improve as shown in Tables II, III and IV. However, the overall result indicates that the combination of the number of dimension and the type of selected amino acid properties greatly affect the approximation of the substitution matrices such as encoding schemes #7, #8. With the comparison of the amino acid encodings, encoding schemes #11, #12, #14 and #15 lead to a more precise approximation of the five substitution matrices based on the number of training epochs and lower $rmse$ errors. The number of epochs and $rmse$ of the 4 encoding schemes are compared in Figures 2, 3.

The result indicate that encoding #12 needs fewer training epochs and leads to a more accurate approximation of the substitution matrices compared to the other three encodings. Also, encoding schemes #12, #14 and #15 perform closely when the substitution matrix is derived from sequences with less evolutionary distance such as BLOSUM80. Meanwhile, the performance of encoding #12, improves with substitution matrices derived from sequences with more evolutionary distance such as BLOSUM50 compared to the other three encodings. Therefore, encoding #12 is a better candidate for applications using sequences with low homology. The average epoch and $rmse$ of each encoding on 5 substitution matrices are shown in Figures 4 and 5 to illustrate the overall comparison. The results also indicate that increasing the encoding dimension may not even improve the approximation of the substitution matrices, and also it is not preferable with regards to magnifying "the curse of dimensionality" such as encoding #14 and #15. The encoding scheme that can best differentiate better 210 amino acid pairs, and meanwhile capture the similarities of the amino acid pairs is encoding #12. The superior performance of encoding #12 can be explained by knowing that each amino acid is encoded and formed based on a set of genetic codons. Therefore, the different arrangements and orders of the four nucleotides

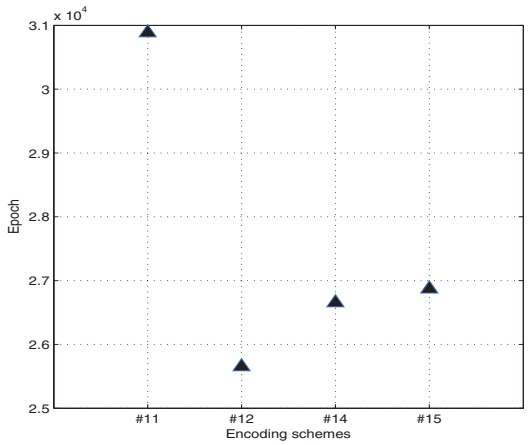

Figure 4. The average training epochs for encoding schemes #11, #12, #14 and #15 on the 5 substitution matrices.

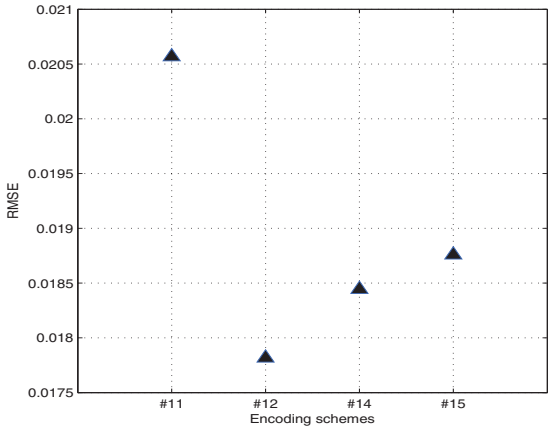

Figure 5. The average root mean square errors for encoding schemes #11, #12, #14 and #15 on the 5 substitution matrices.

are possibly important to conserve the information related to the underlying similarities and differences that exist between two amino acids.

VI. Conclusion

Amino acid encoding is a an important step to apply machine learning techniques in computational biology such as protein sequence alignment. Amino acid encoding aims to capture the underlying similarity and differences of each amino acid pair from symbolic data in a naturally meaningful way like that provided by substitution matrices. In this study, we evaluated the efficiency of a number of amino acid encoding schemes by the training of artificial neural networks to approximate the substitution matrices. In addition a new amino acid encoding was proposed based on genetic codon concept, and it was compared with a number of commonly used amino acid encoding schemes. The experimental results indicate the combination of the number (dimension) and types (properties) of the selected amino acid features are important for the efficiency of the encoding performance. The aim of this study was to demonstrate the efficiency of the commonly used encoding methods which can possibly improve the performance of machine learning techniques used in computational biology by choosing the correct encoding. The more meticulous evaluation of the encoding schemes, in addition to the approximation of various substitution matrices which is the important step to achieve this goal, can be fulfilled by employing the encoding schemes in a number of protein-related problems and comparing their performances. Moreover, the goal in this study was not to employ an optimization method or a feature selection technique to choose the best number of dimension and features of amino acids which would be an interesting future work to investigate the amino acid encoding in more details.

Acknowledgements

Dr. Stefan C. Kremer is funded by the NSERC of Canada.

References

[1] R. Swanson, "A vector representation for amino acid sequences," *Bulletin of mathematical biology*, vol. 46, no. 4, pp. 623–639, 1984.

[2] *Blomap: an encoding of amino acids which improves signal peptide cleavage site prediction*. Citeseer, 2005.

[3] K. Zimmermann and J. Gibrat, "Amino acid" little big bang": Representing amino acid substitution matrices as dot products of euclidian vectors," *BMC bioinformatics*, vol. 11, no. 1, p. 4, 2010.

[4] N. Qian and T. Sejnowski, "Predicting the secondary structure of globular proteins using neural network models," *Journal of Molecular Biology*, vol. 202, no. 4, pp. 865–884, 1988.

[5] B. Rost and C. Sander, "Prediction of protein secondary structure at better than 70% accuracy," *Journal of Molecular Biology*, vol. 232, pp. 584–584, 1993.

[6] G. Pollastri, D. Przybylski, B. Rost, and P. Baldi, "Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles," *Proteins Structure Function and Genetics*, vol. 47, no. 2, pp. 228–235, 2002.

[7] M. Dayhoff, R. Schwartz, and B. Orcutt, "A model of evolutionary change in proteins," *Atlas of Protein Sequence and Structure*, vol. 5, no. Suppl 3, pp. 345–352, 1978.

[8] S. Henikoff and J. Henikoff, "Amino acid substitution matrices from protein blocks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 89, no. 22, pp. 10 915–10 919, 1992.

[9] S. Haykin, *Neural networks and learning machines*. Prentice Hall Upper Saddle River, NJ, 2009, vol. 10.

[10] D. Rumelhart, G. Hinton, and R. Williams, "Learning internal representations by error propagation. parallel distributed processing, vol. 1," *Foundations. MIT Press, Cambridge*, 1986.

[11] P. Baldi and S. Brunak, "Bioinformatics: The machine learning approach." 1998.

[12] H. Hu, Y. Pan, R. Harrison, and P. Tai, "Improved protein secondary structure prediction using support vector machine with a new encoding scheme and an advanced tertiary classifier," *NanoBioscience, IEEE Transactions on*, vol. 3, no. 4, pp. 265–271, 2004.

[13] H. Lac and S. Kremer, "Inducing fold dynamics from known protein structures using machine learning," Ph.D. dissertation, CIS, University of Guelph, April 2009.

[14] W. Taylor, "The classification of amino acid conservation*," *Journal of theoretical Biology*, vol. 119, no. 2, pp. 205–218, 1986.

[15] C. Wu and J. McLarty, *Neural networks and genome informatics*. Elsevier Science, 2000, vol. 1.

[16] M. Zvelebil, G. Barton, W. Taylor, and M. Sternberg, "Prediction of protein secondary structure and active sites using the alignment of homologous sequences," *Journal of Molecular Biology*, vol. 195, no. 4, pp. 957–961, 1987.

[17] S. Kawashima and M. Kanehisa, "AAindex: amino acid index database," *Nucleic acids research*, vol. 28, no. 1, p. 374, 2000.

[18] P. Chou and G. Fasman, "Prediction of the secondary structure of proteins from their amino acid sequence." *Advances in enzymology and related areas of molecular biology*, vol. 47, p. 45, 1978.