Artificial Neural Networks for Protein Structure Prediction

W. Willie Wells

May 2015

1 Introduction

1.1 Motivation

Pathogens are infectious agents that cause disease and illness to a host organism. Various substrates and pathways are means for pathogens to infect their hosts. Although the human immune system and helpful internal bacteria fight of some harmful pathogens, pathogens still threaten human life. This necessitates the engineering of fungicides, vaccines, and antibiotics to neutralize this threat to humanities existence [7]. Accurately designing drugs to neutralize pathogens or to enhance normal functions involves knowledge of the target protein's function. Designed drugs normally activate or inhibit the function of a target protein. This then results in a therapeutic benefit to the organism. The target protein is usually a prominent molecule in the metabolic or signaling pathway that leads to a disease condition within the host organism. A protein's function is determined by its secondary structure as well as how the protein interacts with DNA, RNA, and enzymes [3, 6].

1.2 Difficulties

The specific shape of a protein's structure allows the protein to perform its specific function. The overall three-dimensional structure of a protein can be parsed into a sequence of secondary structure elements. The observed three-dimensional structure of a protein is obtained by X-ray diffraction (X-ray crystallography) and Nuclear Magnetic Resonance (NMR) Spectroscopy. Although these methods provide ground truth they have not yet provided an exhaustive database. This is due to the fact that these methods are slow. The equipment involved is expensive. Obtaining and exhaustive database of each isolated protein using these methods is a difficult process. Thus, machine learning methods

were applied to predict protein secondary structure [3].

1.3 Solution

The use of machine learning techniques saves individuals and their respective corporations time and money in engineering new drugs to neutralize life threatening pathogens. Hidden Markov Models, Support Vector Machines, Bayesian Belief Networks, and Artificial Neural Networks are machine learning methods utilized in the prediction of protein secondary structure. Artificial Neural Networks provide $\geq 70\%$ accuracy in predicting protein secondary structure .

2 Problem definition

How does one use machine learning techniques to predict protein secondary structure?

2.1 Protein Secondary Structure

Secondary structure elements are defined by their folding which is regulated by hydrogen bonding between amino acids. Although the Dictionary of Protein Secondary Structure (DSSP) defines eight types of secondary structure, there are only three major types: α-Helix (H), β-Sheet (E for extended strand), and Coil or Loop (C or L). Various notations for the third type are used besides Coil (C) and Loop (L). Among these is other (-) and γ. The prediction accuracy category notation associated with the three type division is Q3. Protein secondary structure is sensitive to changes in a single amino acid and is reliant on local and long range interactions. [1, 3].

2.1.1 α-helices

Every fourth residue in the backbone has a hydrogen bonded link in α-helices. Hydrophilic portions of the α-helices face outwards while hydrophobic portions face inwards with relation to the overarching structure of the protein. This gives α-helices a lower ratio of hydrophobic amino acids than β-sheets [8].

2.1.2 β-sheets

In β-sheets, hydrogen and oxygen elements of two sequence chains form a bond in either a parallel or an anti-parallel configuration. Anti-parallel arrangements align inwards facing R groups in β-sheets which is an arrangement that is stabler than the parallel arrangement with single inwards facing R groups alternating with hydrogen bonds. There can be more than two strands in a β-sheet and each strand can have anti-parallel or parallel relationship with its neighbors. An inner strand can have an anti-parallel relation with one neighboring strand and a parallel relation with the other neighboring strand [8].

2.1.3 Coils

Coils are structures that are not α-helices or β-sheets, Coils often connect α-helices or β-sheets to other α-helices or β-sheets. Insertions and deletions are most likely to occur in Coils that are on the surface of the protein, these Coils are hydrophilic [8].

2.2 General Machine Learning Techniques

Machine learning techniques are used for predicting regression values (exact values), interpolation of a relation between features, density estimation (probability) of the existence of a set of features, or for predicting class inclusion.

2.2.1 Input

A feature is a distinct aspect of an object under study that can be encoded into a numerical form. Possibly relevant features for use in the desired discrimination (regression, interpolation, density estimation, classification) are extracted. Selecting invariant features that are not perturbed by small transformations is ideal. The ideal features should be invariant to translation, rotation, and scale. Feature encoding involves taking an aspect of an object such as "Apache" and mapping it to a numeral, e.g. every instance of "Apache" in the data set maps to 42. If the number of features is excessive the dimension of the feature space can be reduced by combining features. Two common feature dimensionality reduction techniques are Principle Component Analysis (PCA) and Multiple Discriminant Analysis (MDA). A projection that best represents the data is obtained using PCA which uses eigenvalues of the scatter matrix to arrive at this effect while a projection that best separates the data is obtained using MDA which uses the differences between class means to arrive at its results. After the features are extracted, encoded, and reduced (if desired/required) then the data can be discriminated [4].

2.2.2 Algorithm Implementation

The discrimination step is when the machine learning algorithm is employed. Supervised training involves using input data, a feature space, which has labels associated with each object/instance. Unsupervised training involves using a feature space without labels assigned to train the corresponding model. Clustering is another name used for unsupervised learning methods. In reinforcement learning, the agent is given an indication if its tentative decision is correct or incorrect. A training and a testing phase are associated with supervised machine learning techniques. During the training phase is when the algorithm learns the

appropriate values of parameters in the model. Discriminator learning implies reduction of error on the next iteration during the training phase. Gradient descent algorithms are used to accomplish minimization of error and to facilitate updating an algorithm's parameters. A set of feature vectors is selected as the training set. Decisions are based on maximizing a discriminant function (f_i) corresponding to a class for supervised learning methods

$$f_i(X) > f_j(X), \forall j \neq i, X =< x_1, \cdots, x_n >, x_k = feature, k = [1, n]. \quad (1)$$

and minimizing an internal class distance metric (f_i) for clustering

$$f_i(X) < f_j(X), \forall j \neq i, X =< x_1, \cdots, x_n >, x_k = feature, k = [1, n]. \quad (2)$$

The object that X represents is labeled as a member of the class (c) corresponding to f_i. An inactive learning, testing phase consists of inputting a set of feature vectors that were not in the training set into the model, bypassing the parameter update stage, and arriving at the output [4].

2.2.3 Output

Post-processing of the output of a machine learning method is usually required. The output of a machine learning method is numerical while the desired output may not be, e.g. the actual output is a probability while the desired output is "is X probable: yes or no". Thus, a probability greater than target threshold maps to "yes"and otherwise probability maps to "no". If the classes or values desired are already known, an error rate corresponding to how the machine learning algorithm performed can be computed. Conversely, the amount of correctly discriminated objects out of the total objects, the accuracy, can be determined [4].

2.2.4 Problems with Machine Learning

Accurately determining invariant features, small differences between classes, inherent noise in input, and feature vectors with missing values are problems that can arise with designing the feature space used as an input to machine learning techniques. Designing a machine learning algorithm that has finite computation time, uses finite storage, is easy to implement, and avoids over-fitting the training set data are issues that arise during the design process. Over-fitting involves obtaining high accuracy for the training set but low accuracy for all other sets. Determining the amount of training set vectors to use, determining the rate at which parameters are updated during the learning process and determining window size for non-parametric methods are design considerations that can affect accuracy after an algorithm is chosen [4].

3 Technical Approach

Support Vector Machines and Artificial Neural Networks are the two most successful machine learning techniques for protein secondary structure prediction but other machine learning methods have been applied such as Hidden Markov Models and Bayesian Belief Networks. The prediction of all β-sheets is an area in which artificial neural networks are lacking. Despite this deficiency Artificial Neural Networks provide a protein secondary structure prediction accuracy above 70% [8]. Perceptron, Feed Forward(Multi-Layer Perceptron), Radial Basis Function, Bidirectional Recursive, Kohonen Self-Organizing Maps (SOMs), Competitive Layers, Learning Vector Quantification, and Hopfield Network are several types of artificial neural networks.

3.1 Structure of Artificial Neural Networks

Artificial neural networks are based on the network of neurons in the brain. Billions of neurons populate the biological neural network that exists within the human brain. A neural network is inherently a parallel signal processing unit due to its structure. The structure of a neural network consists of layers. Each layer has a distinct, finite number of neurons. Any layer that is not the input or output layer is a hidden layer The neurons in the same layer process data simultaneously. Each neuron has a weight (w_{ij}) associated with each of its inputs (x_j). These weights are often initialized as random small numbers. All neurons from the previous layer have a weighted input ($w_{ij}x_j$) to each neuron in the current layer. A minimal neural network consists of two neurons in the input layer and a single neuron in the output layer. The output (y_i) of a neuron (i) is the response of an activation function (f) to the sum of the weighted inputs to the neuron:

$$y_i = f(\sum_j w_{ij}x_j). \tag{3}$$

Initial inputs to the network are feature vectors. Final outputs are determined by the activation functions in the network, which are specified by the designer. Activation functions take their name from the action potential required for a neuron to fire (activate). Non-linear activation functions allow neural networks to approximate non-linear functions of the inputs to the neural network [2, 4].

3.2 Artificial Neural Network Learning

The training error (E) is computed as a function of the output (Y) of the artificial neural network subtracted from the target value (T) such that:

$$E = \frac{1}{2}\|T - Y\|^2. \tag{4}$$

This training error (E) is then backward propagated using a gradient descent method:

$$\Delta w_{ij} = -\alpha \frac{\delta E}{\delta w_{ij}}, \tag{5}$$

where α is the learning rate. The learning rate (α) determines how fast a local minimum is approached. The negative guarantees a minimum is approached. The partial derivative is the gradient. After $\Delta \mathrm{w}_{ij}$ is calculated the weight for the next iteration is updated,

$$w_{ij}(\tau + 1) = w_{ij}(\tau) + \Delta w_{ij} \tag{6}$$

where τ represents the current iteration. The input sequence is ran through the artificial neural network again with the new weights and the process is repeated until the training error (E) reaches some minimum value (ϵ)

$$\|^{\bigtriangledown} E\| < \epsilon. \tag{7}$$

If there are any hidden layers in the artificial neural network, equation 5 requires the series of utilized activation functions to be differentiable[4].

3.3 Artificial Neural Network Testing

When all members of the selected training set have been ran through the model and the weights have been determined, the artificial neural network accuracy can be tested. Testing involves applying the generated artificial neural network to the testing set. If testing set labels/values are known, the accuracy of the artificial neural network can be determined.

3.4 Applying ANN to Protein SS Prediction

The first use of an artificial neural network did not use a multiple sequence alignment as an input but subsequent artificial neural networks have used multiple sequence alignments from such tools as BLAST as inputs to their artificial neural network models, which increases the overall accuracy of the protein secondary structure prediction.[1, 2]

3.4.1 Input

Thus, the input or feature space to an artificial neural network used for predicting protein secondary structure is either a windowed single protein chain or a windowed input profile built from a multiple sequence alignment. Amino acid encoding schemes include using a 20 bit representation with a different single bit as a 1 for all amino acids, using 5 bits for encoding with each code always containing either two or three 1's, orthogonal encoding, adaptive encoding, and in some manner encoding degeneracy information along with identity [5]. Choosing to use a binary code for input and output signals when compared to other representations decreases overall computation time. If hardware that is designed for neural networks is used, the computation time is further reduced.

3.4.2 Network Topology

A windowed, three layer neural network is the most widely used in protein secondary structure prediction for using a single artificial neural network or for the first artificial neural network in a cascaded system. The artificial neural network is predicting the protein secondary structure of the residue in the center of the window. Window size is normally between 11 and 41 residues to avoid overfitting and to model positional dependency of each amino acid. The artificial neural networks used usually are referred to by the amount of neurons in each

layer, e.g. 23 X 7 X 3 . Hidden layer size is typically less than input layer size and greater than or equal to output layer size. Amount of neurons in the outer layer depend on how many types of secondary structures are desired as output, typically three (H, E, C) [1, 2, 5].

3.4.3 Output and Testing

Codes such as H = [1 0 0], E = [0 1 0], and C = [0 0 1] or H = [0 0], E = [0 1], and C = [1 0] are used for the output layer. Leave-one-out cross-validation is used to train and test the artificial neural network. In some cases the "one" left out refers to one correlated group of sequences. All but one sequence or group of sequences is used to train the neural network and the resulting neural network is tested on the remaining sequence or group of sequences. This is repeated for every sequence or group of sequences. Cross validation using k-fold cross validation is another common training and testing method used. The data set is separated into k equal subgroups, k-1 subgroups are used for training and 1 is used for testing, this is repeated for all subgroups [2]. An accuracy $> 70\%$ is obtained using these methods.

4 Related Work and Experimental Results

4.1 Exploiting the past and future in protein secondary structure prediction

The authors of this paper present a bidirectional recurrent artificial neural network. This bidirectional recurrent neural network takes an entire sequence as input and runs a forward and a backward algorithm on the current amino acid

(t). Producing a probability for t being part of each distinct secondary structure

$$t \rightarrow (p_1, p_2, p_3), \sum_i p_i = 1, p_i \geq 0. \tag{8}$$

While the output prediction is of the form:

$$p_i = f(F_t, B_t, I_t). \tag{9}$$

Since fixed window sizes do not capture long range information, the authors devised the bidirectional recurrent artificial neural network to remedy this situation. Bidirectional recurrent artificial neural network architecture uses the fact that a protein sequence is non-causal (the next connected amino acid does not depend entirely on the previous amino acid, the "future" is known). The authors use adaptive dynamics, multiple sequence alignments, and mixture estimators in their algorithm. Effective window sizes of ± 15 was obtained giving a total effective window size of 31. Two data sets were used one with 826 sequences with 25% homology and one with 1180 sequences and 50% homology. This approach yields a 76% overall protein secondary structure prediction accuracy.

4.2 The Importance of larger data sets for protein secondary structure prediction with neural networks

Fully connected, feed forward artificial neural networks with either 2 or 3 layers are used by Chandonia and Karplus in their experiments. They used 318 high resolution sequences and 32-fold cross validation. A scaled conjugate gradient algorithm was used during the update process instead of a line search for the optimal learning rate (α), which is the most time consuming step in the average artificial neural network algorithm implementation. They smooth the outputs by averaging the protein secondary structure probabilities with those of their

immediate neighbors. Artificial neural network input layer size was varied while keeping hidden and output layer neuron numbers constant. Their results showed that 19 neurons in the input layer was optimal for a n X 2 X 2 artificial neural network. Hidden layer neuron count was varied while keeping neuron count the same in the other layers. A neuron count of 8 was optimal for a 19 X n X 2 neural network. They then varied the input neuron layer count for a n X 8 X 2 network and found that n = 15 was optimal. For these trials they were using single sequence data and not profile data and received optimal accuracies from 63.1 to 66.5%. They incorporated profile data and received a 72.9% for a 17 X 9 X 2 neural network.

4.3 Artificial Neural Network Aided Protein Structure Prediction

This paper provides a sufficient description of protein structure and motivation for predicting protein secondary structure more so than the preceding papers. The authors use 6 sequences from different secondary categories. Alphanumeric encoding is used for input and output. They use 4 hidden layers in a radial basis function artificial neural network and stopping criteria of 1650 iterations or $\epsilon = 10^{-4}$, which ever comes first. A forward approach primary structure to secondary structure and a backward approach tertiary structure to secondary structure is used to validate the secondary structure prediction. The forward approach out performed the backward approach for all 6 proteins for both α-Helices and β-Sheets.

4.4 Pattern Classification

A book discussing details of machine learning techniques.

4.5 Using a neural network to backtranslate amino acid sequences

A discussion of amino acid encoding methods is presented. The 5 bit code method performed the worst while simple 20 bit codes performed the best on their data set. All prediction accuracies were between 69 and 85%.

5 Conclusion

Future work includes an increased use of cascaded artificial neural networks, optimization of artificial neural network structure, finding the optimal set of activation functions, and increasing computational speed by designing complete artificial neural network systems from silica/germanium to the visual representation of each structural level. Artificial neural networks are powerful tools that will only be enhanced as time progresses and will not fade into oblivion. The accuracy of artificial neural network prediction of protein secondary structure will only increase with time. The choices of activation functions and gradient descent method are crucial design decisions in building an artificial neural network. Though, over-fitting is a constant bother it can be mediated partially by cross-validation methods and diversity in the training set(s). Accurate protein secondary structure prediction has the potential to save time and money as well as lives.

References

[1] P. Baldi, S. Brunak, P. Frasconi, G. Soda, G. Pollastri, "Exploiting the past and future in protein secondary structure prediction", in *Bioinformatics*, 1999. pp. 937-946.

[2] J. Chandonia, M. Karplus "The Importance of larger data sets for protein secondary structure prediction with neural networks", in *Protein Science*, 1996. pp. 768-774.

[3] A. Deka, K. K. Sarma, "Artificial Neural Network Aided Protein Structure Prediction", in *International Journal of Computer Applications*, 2012. pp. 33-37.

[4] R. O. Duda, P. E. Hart, D. G. Stork, *Pattern Classification*, 2nd Edition, John Wiley, 2001.

[5] W. Seffens, G. White, "Using a neural network to backtranslate amino acid sequences", in *Electronic Journal of Biotechnology*, 1998.

[6] "Drug Design". Wikipedia. Web. 2 May 2015

[7] "Pathogen". Wikipedia. Web. 2 May 2015

[8] "Protein Structure Prediction". Wikipedia. Web. 2 May 2015