

Using evolutionary noise to improve prediction of rapidly evolving targeting peptides

Mikael Bodén

mikael@itee.uq.edu.au

School of Information Technology and Electrical Engineering
University of Queensland, QLD 4072, Australia.

Abstract- Targeting peptides are responsible for directing proteins to the appropriate subcellular location. As a group of biological sequences, targeting peptides are evolving at a relatively high rate and exhibit diversity. We investigate if evolutionary noise – simulated mutation at the molecular level – improves target classification for a neural network predictor. Comparison with the well-known TargetP prediction service illustrates some advantages of the approach. Specifically, classification of signal peptides, which exhibit an extremely high rate of evolution, is improved.

1 Introduction

The subcellular translocation of proteins usually relies on a short N-terminal targeting peptide (sequence of amino acid residues). The peptide directs the protein to its appropriate subcellular location. After delivery the targeting peptide is cleaved from the mature protein and is presumed to be of no further use and digested. Some properties that correlate with the subcellular sorting are known, but statistical indicators are very weak. Numerous forms of sequence can serve the same role. As an example, for signal peptides (a particular class of targeting peptide) it has been argued that the only requirement is that the peptide core is consisting almost exclusively of hydrophobic amino acids (see Williams et al., 2000, for discussion).

A series of neural network based predictors have shown potential in handling the task of classifying sorting signals. SignalP, ChloroP and TargetP all predict localization and cleavage sites of various proteins using simple feed forward neural networks (see Emanuelsson and von Heijne, 2001, for a review, but also Chou, 2001, and Cai et al., 2002, 2003 for variations). We focus on the most general predictor: TargetP. TargetP distinguishes between proteins destined for mitochondrion, for chloroplast, for the secretory pathway, and others.

By experimenting with various configurations, Emanuelsson et al. (2000) found that target specific feed forward neural networks which slide over a window of residues can work as a target sequence recognizer, i.e. detect which residues that belong to the target sequence (to be cleaved) and those that belong to the mature protein. The detection outputs for the first 100 residues (from each of the target specific neural networks) are fed into another neural network which makes a final decision on which subcellular compartment the protein is destined for.

In general, a neural network is trained from example data to search for a solution using a learning algorithm. The so-

lution – a network with a particular set of weight values – is later evaluated by presenting novel data. In contrast to conventional statistical tools, the network architecture imposes a bias (or constraint) on the search for the solution. Restricting a network's capacity prevents overfitting to training data since the network exhibits insufficient capacity to represent models that are too complex. This idea is consistent with a Vapnik Chervonenkis (VC) dimension analysis of a neural network's capacity versus its generalization. The more adaptable weights, the larger the VC dimension of the solution space, and the less likely the training set is large enough to select a correct solution (Baum and Haussler, 1989). A second aspect concerns the data set directly – which also specifies a constraint to meet for the learning algorithm. In high-dimensional data (as in the case of the combinatorial space of amino acid sequences) the data is usually sparse. Even if we collect many thousand proteins we will still only cover a fraction of the possible input space. Potential remedies include various types of regularization techniques, e.g. weight decay and additive noise (Bishop, 1995), bagging and mixture/ensemble models (Breiman, 1996; Bauer and Kohavi, 1999). Technically, adding noise to input data is equivalent to a smoothness constraint on the learning error function. Mixture models promote generalization by virtue of multi-versioning.

The diversity of sequences nevertheless presents a major hurdle for machine learning techniques: Finding representations that include sequences that share functionality but not necessarily share componential structure, and (concurrently) exclude sequences that perform different functions becomes difficult.

Proteins perform various functions and it is important to acknowledge that over evolution it is not their structures (or components) that are conserved. Rather, it is their function that is selected for (Lesk, 2001). Here, we are concerned with a particular kind of function: that of sorting and transporting the mature protein to subcellular compartments. There is considerable variation between proteins in the rate of evolution. There is even variation within peptide chains. On average, signal peptides evolve five times faster than the flanking mature peptide (Williams et al., 2000).

Operators in evolutionary computation are useful for exploring complex search landscapes. More specifically, point mutations can serve to explore alternative solutions close to known solutions. Recombination operations (e.g. crossover) traverses the search landscape in a more disruptive manner.

The basic idea investigated in this paper is motivated by two observations:

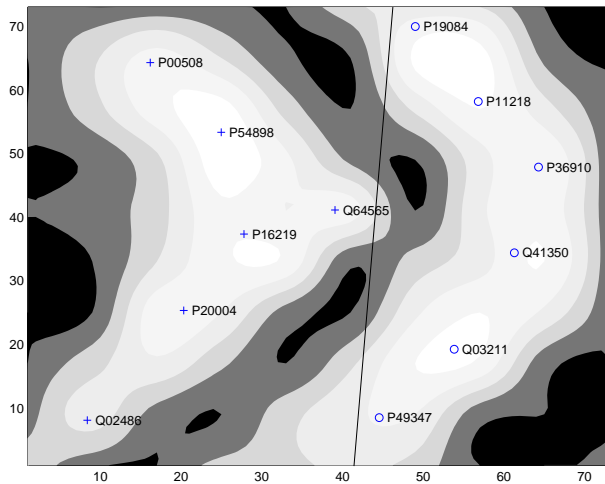


Figure 1: A hypothetical sequence space in which the learning algorithm tries to fit a decision boundary between mitochondrial and signal peptides. 12 sequences are plotted (mitochondrial targeting peptides plotted as '+', signal peptides plotted as 'o') and a simple line can separate the two classes. The background shade indicates if contained sequences are functional (white) or not (black).

- Due to a high rate of evolution, targeting peptides exhibit great diversity. Presumably some stabilizing selection occurs, but most variation seems to be due to more or less incidental changes over evolutionary time. Even random sequences can act as signal peptides.
- In the presence of noise, machine learning techniques (including neural networks) tend to avoid overfitting training data and focus on main characteristics. Consequently, generalization to novel data is improved.

We hypothesize that the addition of noise to target sequence training data provides a useful bias which improves generalization to novel target sequences. We use point mutation to jump from a sample in the data set to a point nearby in sequence space. In doing so, we rely on an appropriate sequence space to search. In a good sequence space related proteins are related spatially (see Gustafsson et al., 2001, for a discussion).

In Figure 1, 12 proteins have been plotted in a hypothetical sequence space. Related proteins are confined to restricted regions. With a small and sparse data set as in Figure 1, a simple decision line can partition the samples of the two classes (mitochondrial targeting and signal peptides). However, by performing mutations we create new training samples and thus prohibits the too simple generalization represented by the line (see Figure 2). The mutated sample is not necessarily within the boundaries of fully functional peptides but nevertheless serves to guide learning in an otherwise sparse space.

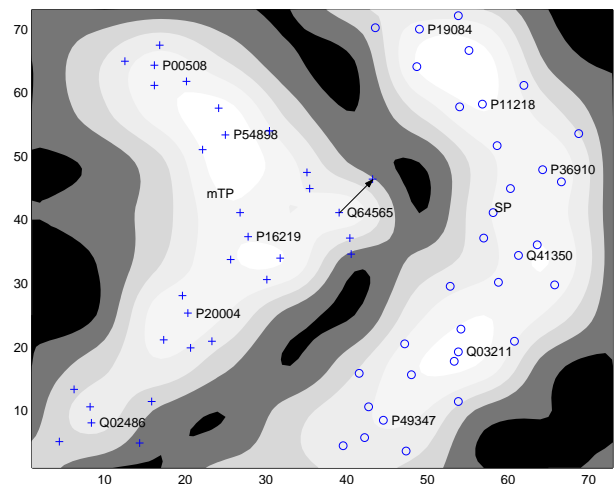


Figure 2: With several mutated versions of each of the 12 sample sequences the sequence space is more densely populated and a simple line does no longer separate sequences of the two classes. The mutational change of Q64565 is illustrated by an arrow.

2 Background

Mitochondrial targeting peptides generally display high representation of Alanine (A), Arginine (R) and Serine (S). There are also some indications that the target sequence forms an α -helix to enable importation into the mitochondrion. However, there is little known of any specific signalling motifs – though in some cases Alanine appears 2 or 3 residues from the cleavage site. It has also been noted that mitochondrial target sequences evolve at a relatively slower pace and are generally longer (Williams et al., 2000).

Signal peptides belong to a very general sequence class that is responsible for transportation of proteins to the endoplasmic reticulum for subsequent transport to the secretory pathway. They typically display a hydrophobic core and have a small neutral amino acid at specific positions near the cleavage point (Nielsen et al., 1997; Chou, 2001). It has been argued that 20% of random sequences can act as signal sequences (Kaiser et al., 1987).

Peptides that direct proteins to chloroplast also lack known distinctive features. Weak characteristics include small numbers of acidic amino acids. In ChloroP (a specialized predictor; Emanuelsson et al., 1999) some confusion was noted between sorting signals for mitochondrion and chloroplast. Moreover, several proteins have been found to be recognized by both systems and subsequently become imported into both organelles (Peeters and Small, 2001).

We also use a set of nuclear and cytosolic proteins as negative samples. These lack a targeting peptide.

3 Method and simulations

To allow objective assessment of the proposed approach we use the standard data set which accompanies TargetP. Each simulation is evaluated by 5-fold crossvalidation: The data set is divided into five subsets (of approximately equal size).

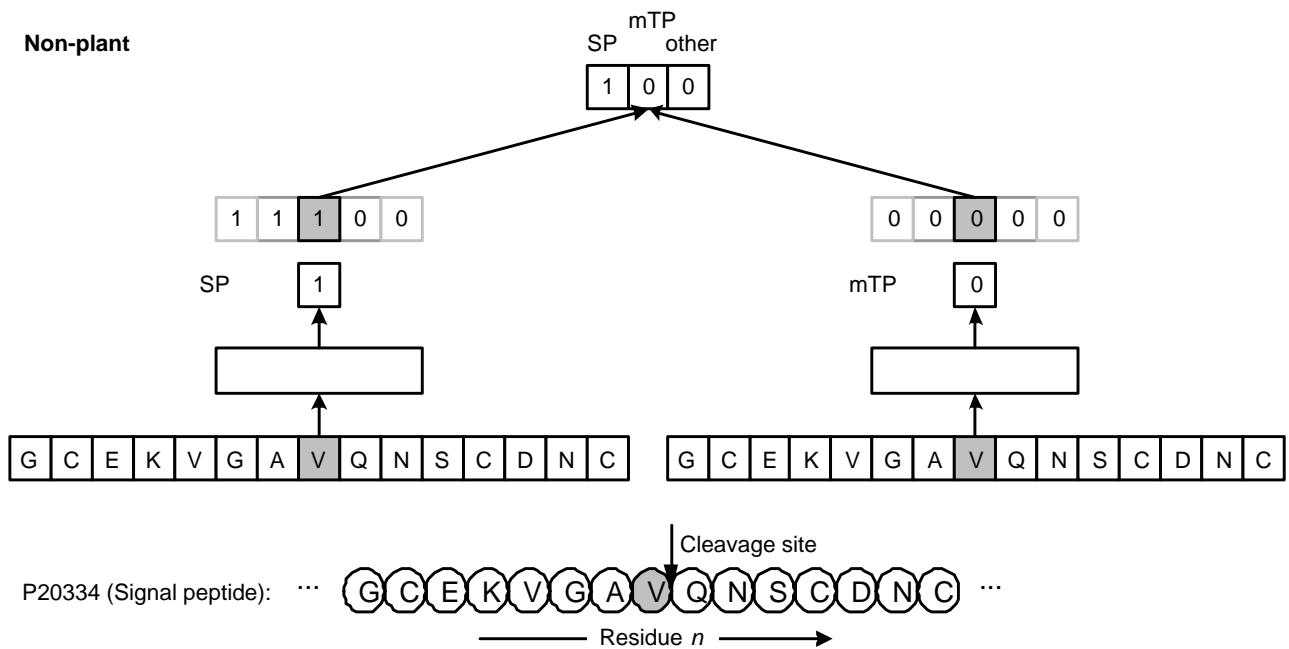


Figure 3: The TargetP neural network architecture. A sequence of amino acids is processed by presenting a window (of a predetermined size) of residues. There is one network for each class of target sequence. The target output of the first layer of networks is 1 if the middle residue (of the current sequence) is part of the specific target sequence, 0 otherwise. The 100 first actual outputs of each target specific network are fed to a final network which classifies the current sequence into one of the target sequence classes or other.

Four are used for training the system, the remaining subset is used for testing. The procedure is repeated with randomly initialized networks and by shuffling the data subsets so that each of the five subsets appears as a test set exactly once. Consequently, the five systems are tested on, for each individual system, unseen sequences. We are also ensured that each sequence in the original data set appears as a test sample exactly once. The final score is the result for all five test sets.

The default neural network architecture is the same as used by TargetP. A fixed architecture (known to produce reliable results) ensures that any relative performance differences can safely be attributed to be caused by the few parameters we choose to change.

3.1 Data sets

TargetP is able to classify sequences from eukaryotic organisms. There are two versions: one for plants, and one for non-plants. The plant version is trained to classify sequences into three specific target classes (mitochondrial, chloroplast, signal peptides) or “other”. The non-plant version is trained to classify sequences into two specific target classes (mitochondrial, signal peptides) or “other”.

The plant data set consists of 940 proteins (368 mitochondrial [mTP], 141 chloroplast [cTP], 269 signal peptides [SP] and 162 nucleus and cytosolic [other]). The non-plant set consists of 2738 proteins (371 mitochondrial, 715 signal peptides and 1652 nucleus and cytosolic).

All sequences are presented to the networks as one-hot

bit-strings, i.e. vectors with all elements set to 0 except one element set to 1. The set element is unique for the amino acid, resulting in a 20 bit vector for each residue in the sequence, mutually orthogonal to all others.

3.2 Neural networks

To classify a sequence into one of the target classes, a series of neural networks is used. First, for each target class there is a target sequence recognition network. The recognition network has an input window of a predetermined number of residues. The recognition network slides over the sequence, one residue at a time. At each instant the output target is 1 if the middle residue in the current window belongs to the specific target sequence, or 0 otherwise. The recognition networks for all target classes are trained concurrently, and each produces an output value. The first 100 recognition outputs for each recognition network are collated and then presented to another neural network which classifies the sequence into one of the target classes. The process is illustrated in Figure 3 (non-plant version).

The TargetP plant version is equipped with three target sequence recognition networks: one for mitochondrial, one for chloroplast and one for signal peptides. These networks are fitted with an input window of sizes 35, 55, and 31 amino acid residues respectively. Each recognition network was also fitted with a hidden layer consisting of four hidden nodes. All networks were reportedly optimal with

these configurations (Emanuelsson et al., 2000). The classification network had 300 input nodes (100 from each peptide recognition network), no hidden nodes and four outputs (one for each target class and one for “other”).

Similarly, the TargetP non-plant version had two recognition networks: one for mitochondrial and one for signal peptides, fitted with input windows of sizes 35 and 29 residues respectively, and four hidden nodes. The non-plant classification network had 200 input nodes, no hidden nodes, and three outputs (mTP, SP and other, see Figure 3).

We implemented the configuration above and replicated the reported simulations. We use the softmax output function and the negative log-likelihood error function. All networks are trained using backpropagation (Rumelhart et al., 1995). The learning rate (η) is 0.005 in the recognition networks and 0.05 in the classification network. The classification result for a class is defined as the fraction between the number of samples correctly classified as belonging to the class and the total number of samples in the class.

The results after 20000 presentations are presented in Table 1. The overall classification result is 0.815 for the plant version (0.858 in Emanuelsson et al., 2000) and 0.903 for the non-plant version (0.900 in Emanuelsson et al., 2000).¹ The discrepancy between our simulations and the original TargetP results for the plant version may be attributed to the fact that we do not exclude proteins deemed redundant for specific recognition networks (see Emanuelsson et al., 2000, p. 1013, for details). Also, we did not fine-tune learning rates and stopping criteria. Importantly, the results herein should be interpreted as measuring the relative difference between configurations differing only in the extent mutations occur in the training data set.

3.3 Sequence mutation

An amino acid is mutated by a probability p_{mut} . The mutation is performed in one of two ways:

- By stochastically determining the three nucleotides that coded for the amino acid, flipping one nucleotide (with even probability) and translating the new triplet back to an amino acid.
- By scanning the BLOSUM50 matrix, and selecting the amino acid substitution in accordance with the probabilities expressed therein.

Flipping first nucleotide is neutral in hydrophobicity. Some targeting peptides utilize hydrophobic amino acids and hence maintaining such may be evolutionary beneficial. The other two nucleotides can also be flipped. By experimentation we noted no significant difference if the mutation was confined to the first bit or not. It should be noted that the translation of a triplet uniquely determines the amino acid, but the triplet is redundant in the sense that more than one triplet results in a particular amino acid. Hence, before performing mutation, we convert an amino acid to one of the viable triplets on a random basis.

¹We managed to get an even lower error when using a larger learning rate, $\eta = 0.01$ resulted in 0.907.

Replication results						
Version	Class	Predicted class				Correct
		mTP	cTP	SP	other	
Plant	mTP	276	42	3	47	0.750
	cTP	26	102	1	12	0.724
	SP	8	1	243	17	0.904
	other	10	5	2	145	0.896
Non-plant	mTP	321	-	3	47	0.865
	SP	16	-	662	37	0.926
	other	134	-	28	1490	0.902

Emanuelsson et al., 2000						
Version	Class	Predicted class				Correct
		mTP	cTP	SP	other	
Plant	mTP	300	41	9	58	0.735
	cTP	14	120	2	5	0.851
	SP	7	7	245	15	0.894
	other	13	10	2	137	0.846
Non-plant	mTP	330	-	9	32	0.889
	SP	13	-	683	19	0.955
	other	152	-	49	1451	0.878

Table 1: Classification scores from replication of the original TargetP simulation and Emanuelsson et al’s published results. The correct classification score is the ratio between the number of correctly predicted sequences and the total number of sequences in the class.

The BLOSUM50 matrix (see Table 2) expresses the ratios, of the number of observed pairs of amino acids at any position, to the number of pairs expected from the overall amino acid frequencies. The data is based on an extensive protein data set. The matrix thus illustrates the frequencies of amino acid substitution in a general context.

4 Results

4.1 Configurations and outcomes

Several configurations with different mutations were simulated. Each simulation involved the presentation of about 20000 sequences, randomly drawn from the data set (uniform distribution over classes, rather than samples), each

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	+5																			
R	-2	+7																		
N	-1	-1	+7																	
D	-2	-2	+2	+8																
C	-1	-4	-2	-4	13															
Q	-1	+1	+0	+0	-3	+7														
E	-1	+0	+0	+2	-3	+2	+6													
G	+0	-3	+0	-1	-3	-2	-3	+8												
H	-2	+0	+1	-1	-3	+1	+0	-2	10											
I	-1	-4	-3	-4	-2	-3	-4	-4	-4	+5										
L	-2	-3	-4	-4	-2	-2	-3	-4	-3	+2	+5									
K	-1	+3	+0	-1	-3	+2	+1	-2	+0	-3	-3	+6								
M	-1	-2	-2	-4	-2	+0	-2	-3	-1	+2	+3	-2	+7							
F	-3	-3	-4	-5	-2	-4	-3	-4	-1	+0	+1	-4	+0	+8						
P	-1	-3	-2	-1	-4	-1	-1	-2	-2	-3	-4	-1	-3	-4	10					
S	+1	-1	+1	+0	-1	+0	-1	+0	-1	-3	-3	+0	-2	-3	-1	+5				
T	+0	-1	+0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	+2	+5			
W	-3	-3	-4	-5	-5	-1	-3	-3	-3	-2	-3	-1	+1	-4	-4	-3	15			
Y	-2	-1	-2	-3	-3	-1	-2	-3	+2	-1	-1	-2	+0	+4	-3	-2	-2	+2	+8	
V	+0	-3	-3	-4	-1	-3	-3	-4	-4	+4	+1	-3	+1	-1	-3	-2	+0	-3	-1	+5

Table 2: The BLOSUM50 substitution matrix used for selecting probable mutations for a specific residue. Each value/10 is \log_{10} of the relative expectation value of a mutation.

No mutation			
$\eta =$	0.010	0.005	0.001
mTP	0.723	0.750	0.680
cTP	0.730	0.724	0.759
SP	0.907	0.904	0.934
other	0.877	0.896	0.889
all	0.803	0.815	0.800

Triplet mutation, $p_{mut} = 0.1$			
$\eta =$	0.010	0.005	0.001
mTP	0.747		
cTP	0.730		
SP	0.937		
other	0.852		
all	0.817		

First-bit mutation, $p_{mut} = 0.1$			
$\eta =$	0.010	0.005	0.001
mTP	0.758		
cTP	0.681		
SP	0.929		
other	0.889		
all	0.818		

BLOSUM50 mutation, $p_{mut} = 0.1$			
$\eta =$	0.010	0.005	0.001
mTP	0.804	0.813	0.720
cTP	0.617	0.695	0.773
SP	0.937	0.934	0.948
other	0.858	0.864	0.895
all	0.823	0.838	0.823

Table 3: The classification performance of various configurations for the plant data set. The simulations with $\eta = 0.010$ were stopped after 10000 presentations due to fluctuating, and seemingly non-converging outputs (and should consequently be interpreted with caution). All other simulations involved 20000 presentations.

residue mutated with a probability p_{mut} . Only the training sets were subject to mutation. Test sequences were presented in their original form.

The classification performance for each configuration is listed in Tables 3 and 4. To evaluate the effects of mutation over a broader range of searches, networks were trained with different learning rates (a greater rate means a more aggressive, accelerated search, whereas a smaller rate means a more cautious traversal of the search landscape).

We also tested architectures with more hidden units but no improvement was observed (in agreement with the original TargetP simulations; Emanuelsson et al., 2000).

4.2 Discussion

There is only subtle differences in overall classification performance if mutated samples are used for training (around the 2% mark for the plant data, no difference for the non-

No mutation			
$\eta =$	0.010	0.005	0.001
mTP	0.747	0.865	0.795
SP	0.934	0.926	0.945
other	0.933	0.902	0.910
all	0.908	0.903	0.904

BLOSUM50 mutation, $p_{mut} = 0.1$			
$\eta =$	0.010	0.005	0.001
mTP	0.825	0.873	0.836
SP	0.957	0.955	0.944
other	0.906	0.865	0.873
all	0.908	0.890	0.887

Table 4: The classification performance of various configurations for the non-plant data set. The simulations with $\eta = 0.010$ were stopped after 10000 presentations due to fluctuating, and seemingly non-converging outputs (and should consequently be interpreted with caution). All other simulations involved 20000 presentations.

plant data). However, the performance profile for both versions is changed.

For the plant-data, the signal peptides are classified correctly to a level of approximately 93-95% consistently with mutation, and 91-93% without mutation. This is in accordance with the high rate of evolution in signal peptides (Williams et al., 2000). Mitochondrial targeting peptides are also more frequently recognized in the presence of evolutionary noise. The other classes are less tolerant.

The observation that the quickly evolving peptides are better recognized in the presence of noise extended to the non-plant data. Both the signal and mitochondrial targeting peptides were predicted more successfully. Signal peptide prediction increases by approximately 2% and those destined for mitochondria were predicted better by at least 1% with noise. However, the sheer number of “other” proteins (for which prediction performance deteriorated) made the overall performance go down considerably when mutation was used.

We tried decreasing the mutation rate to 0.02 (when $\eta = 0.005$) to see the effects on the non-plant prediction. The overall prediction success rate went up (0.906) and the class-profile changed slightly. The signal peptides were still predicted well (0.947), but as the proportion of correctly labelled “other” went up, the prediction of mitochondrial targeting peptides went down (0.854).

On a cautionary note, the classification of the sequences (and in particular mitochondrial peptides) does not converge to a stable rate with a too high learning rate (≥ 0.01), and the outcomes from those simulations should only receive rough consideration.

We noted only marginal difference between different ways of performing point mutation. Mutation based on the BLOSUM50 matrix offered some advantage and was therefore used in the majority of our simulations. We do not rule out that uniform noise can provide a similar advantage. The

small performance differences between different mutation strategies only suggest that the evolutionary information is useful for guiding training set exploration.

We have yet to determine the effect of mutation for predicting the exact position of the cleavage site of sequences. We expect to investigate this in the near future.

5 Conclusions

We have not been able to show substantial evidence for the general advantage of mutation as a means to guide training of a protein feature predictor. For plant data, there is about 2% increase in prediction performance with mutated training data. For non-plant data we observed no or little difference in overall performance. However, in the case of signal peptides – a class of highly diverse and quickly evolving biological sequences – mutated samples were clearly advantageous to generalization. Similarly, mitochondrial targeting peptides were recognized to a greater extent with evolutionary noise. We foresee that our approach may be of value where sequences subscribe to such otherwise for a data mining tool difficult circumstances.

References

- Bauer, E. and Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36:105.
- Baum, E. B. and Haussler, D. (1989). What size net gives valid generalization? *Neural Computation*, 1(1):151–160.
- Bishop, C. M. (1995). Training with noise is equivalent to Tikhonov regularization. *Neural Computation*, 7(1):108–116.
- Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24:123–140.
- Cai, Y.-D., Lin, S.-l., and Chou, K.-C. (2003). Support vector machines for prediction of protein signal sequences and their cleavage sites. *Peptides*, 24(1):159–161.
- Cai, Y.-D., Liu, X.-J., and Chou, K.-C. (2002). Artificial neural network model for predicting protein subcellular location. *Computers & Chemistry*, 26(2):179–182.
- Chou, K.-C. (2001). Prediction of signal peptides using scaled window. *Peptides*, 22(12):1973–1979.
- Emanuelsson, O., Nielsen, H., Brunak, S., and von Heijne, G. (2000). Predicting subcellular localization of proteins based on their n-terminal amino acid sequence. *Journal of Molecular Biology*, 300(4):1005–1016.
- Emanuelsson, O., Nielsen, H., and von Heijne, G. (1999). ChloroP, a neural network-based method for predicting chloroplast transit peptides and their cleavage sites. *Protein Science*, 8:978–984.
- Emanuelsson, O. and von Heijne, G. (2001). Prediction of organellar targeting signals. *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research*, 1541(1-2):114–119.
- Gustafsson, C., Govindarajan, S., and Emig, R. (2001). Exploration of sequence space for protein engineering. *Journal of Molecular Recognition*, 14:308–314.
- Kaiser, C. A., Preuss, D., Grisafi, P., and Botstein, D. (1987). Many random sequences functionally replace the secretion signal sequence of yeast invertase. *Science*, 235:312–317.
- Lesk, A. M. (2001). *Introduction to Protein Architecture*. Oxford University Press, Oxford.
- Nielsen, H., Engelbrecht, J., Brunak, S., and von Heijne, G. (1997). Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. *Protein Engineering*, 10:1–6.
- Peeters, N. and Small, I. (2001). Dual targeting to mitochondria and chloroplasts. *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research*, 1541(1-2):54–63.
- Rumelhart, D. E., Durbin, R., Golden, R., and Chauvin, Y. (1995). Backpropagation: The basic theory. In Chauvin, Y. and Rumelhart, D. E., editors, *Backpropagation: Theory, architectures, and applications*, pages 1–34. Lawrence Erlbaum, Hillsdale, New Jersey.
- Williams, E. J. B., Pal, C., and Hurst, L. D. (2000). The molecular evolution of signal peptides. *Gene*, 253(2):313–322.