Detecting and Sorting Targeting Peptides with Neural Networks and Support Vector Machines

John Hawkins

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

jhawkins@itee.uq.edu.au

Mikael Bodén

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

mikael@itee.uq.edu.au

This paper presents a composite multi-layer classifier system for predicting the subcellular localization of proteins based on their amino acid sequence. The work is an extension of our previous predictor PProwler v1.1 which is itself built upon the series of predictors SignalP and TargetP. In this study we outline experiments conducted to improve the classifier design. The major improvement came from using Support Vector machines as a 'smart gate' sorting the outputs of several different targeting peptide detection networks. Our final model (PProwler v1.2) gives MCC values of 0.873 for non-plant and 0.849 for plant proteins. The model improves upon the accuracy of our previous subcellular localization predictor (PProwler v1.1) by 2% for plant data (which represents 7.5% improvement upon TargetP).

Keywords: Protein Subcellular Localization; Machine Learning; Neural Network; Recurrent Neural Network; Support Vector Machine; Targeting Peptide; Amino Acid Sequence

1. Introduction

The localization of proteins to specific subcellular organelles frequently relies on the presence of targeting peptides. These peptides are sequences within the protein which are recognized and used by the localizing machinery. Understanding and recognizing targeting signals is of immense importance to biomedical science for two reasons. Firstly the misplacement of proteins is known to be a cause of numerous debilitating diseases (e.g. hypercholesterolemia and cystic fibrosis). Secondly, drug design requires accurate predictors to verify that products are properly localized within the cell or secreted. Although *in silico* prediction may never inspire the confidence given by *in vivo* testing, it can provide short cuts in developing drugs.

There have been a wide range of techniques applied to the problem of predicting subcellular localization of proteins (see the Emanuelsson[8] or Schneider & Fechner[16] review articles). Here, as in previous studies, we have adopted the general design solution adopted by one of the most successful of these classifiers, TargetP.[9] TargetP

is composed of two layers of machines; the first layer of machines are trained to recognize amino acid residues that belong to a specific targeting signal. The outputs of these machines are then fed into a sorter which produces an overall classification.

Both layers of TargetP are composed of feed forward neural networks (FFNN). We have argued previously that, for problems with certain features, recurrent neural networks (RNN) have a greater ability for recognizing patterns in biological sequences than feed forward architectures.[4] There is also a growing body of literature that demonstrates the practical utility of recurrent neural networks for bioinformatics problems.[2,14,19,15] Indeed, in previous studies we replicated the classifier produced by Emanuelsson *et al.* and showed by extensive simulation and careful analysis that recurrent networks were able to recognize residues as belonging to targeting peptides with an accuracy exceeding that of feed forward networks, in some cases by 25%.[3] However, although recurrent neural networks are able to significantly improve the accuracy at the residue detection level, the sorting layer networks were not able to fully exploit this improvement and provided a much smaller increase in the overall prediction accuracy.[5]

In this paper we extend our previous research in two ways. Firstly, we train a second class of recurrent neural network for the residue detection problem. Pollack's Sequential Cascaded Neural Network (SCNN)[13] is a second-order network architecture that, while having similar overall performance to a standard recurrent neural network,[7] has a slightly different bias that may provide further advantage to the classifier.

Secondly, we have sought to improve the overall prediction produced by the classifier by employing support vector machines at the sorting layer. We were previously able to improve the overall performance of PProwler by using a naive ensemble approach, in which several machines each combining a different residue network and sorting machine are trained on the same problem. The results of each machine in the ensemble are averaged to produce the final output (implemented in PProwler V1.1). In this study we seek to generate a more sophisticated ensemble model in which a single sorting machine operates to exploit the strengths of a range of residue detection networks and identify the conditions under which they can each be trusted.

We use SVMs for this task for a number of reasons. First, we are dealing with a fixed input window, so a static window classifier is more appropriate than a recurrent architecture. Secondly, SVMs have an affinity for performing efficient classifications in a high dimensional input space.[18,6]

We test a range of kernel functions in order to find a sorting support vector machine able to make use of the increased accuracy at the residue detection level. In particular, we construct composite machines that make use of two or three different machine architectures at the residue detection level, whose outputs are then sorted by a support vector machine. The composite machines are able to exploit the variability of different architectures and demonstrate superior classification accuracy.

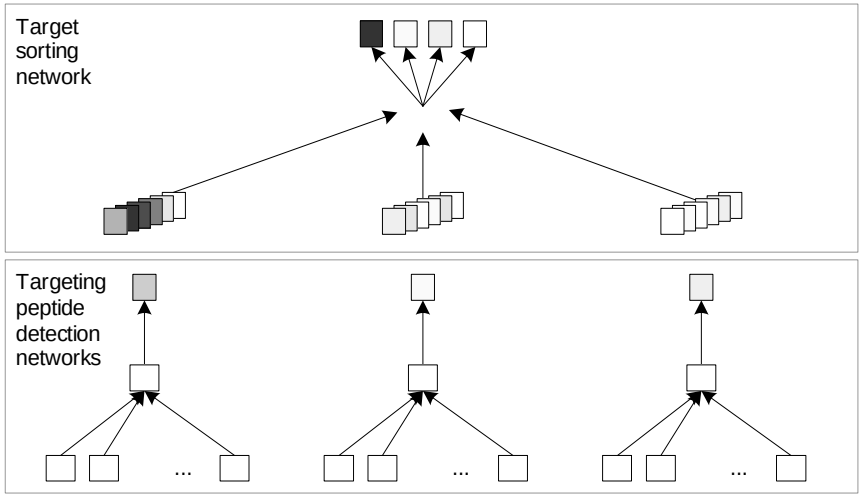

Fig. 1. The TargetP neural network architecture. An initial set of residue-level targeting peptide detector networks (one for each target) receive as input a window of residues and output the status of the middle residue. The outputs for each of the residue detection networks are presented to the target sorting network, which outputs the probabilities of the presence of the targeting peptide types (SP, mTP, cTP and the probability of not having a targeting peptide at all).

2. The Overall Architecture

A schematic of the TargetP classifier is shown in Figure 1. The essence of the design is as follows; each of the distinct subcellular targets is given a residue detection network that is trained to classify an individual amino acid as belonging to its respective targeting peptide. These detection networks perform this task using a window of residues surrounding the residue to be classified. The detection networks slide over the sequence starting from the N-terminal end (the location of the targeting peptide) and produce an output for each of the first one hundred residues. A second machine layer takes the output of each of the detection networks for a sequence and is trained to produce an overall classification. The final machine ideally attunes itself to the sequence regions for which each of the detection networks produce the most reliable results.[9]

3. Method and simulations

We construct two separate models: 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".

All sequences are presented to the networks as one-hot bit-strings. The set ele-

ment is unique for the amino acid, resulting in a 20 bit vector for each residue in the sequence, mutually orthogonal to all others. A single bit is added to accommodate unknown residues.

To allow objective comparison of our new models for subcellular localization prediction we use the standard data set which was used to develop and evaluate TargetP. Each simulation is evaluated by 5-fold cross-validation: The data set is divided into five subsets (of approximately equal size). 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 (and each data sample appears as a test case exactly once). Consequently, the five systems are only tested on, for each individual system, unseen sequences. The score we report is the aggregate result for all five test sets (over the five systems). All five-fold cross-validated simulations are then repeated five times with new data set divisions to ensure that final scores are significant.

3.1. *Networks*

The TargetP plant version is equipped with three targeting peptide detection networks. Each of which is a standard feed forward neural network. Each network corresponds to a particular localization: one for mitochondrial, one for chloroplast and one for signal peptides. These networks are equipped with an input window of sizes 35, 55, and 31 amino acid residues respectively. Each detection network is also fitted with a hidden layer consisting of four hidden nodes. All networks are reportedly close to optimal with these configurations.[9] Similarly, the TargetP non-plant version has two detection networks: one for mitochondrial and one for signal peptides, fitted with input windows of sizes 35 and 29 residues respectively, and four hidden nodes. We re-produce the simulations reported for the above configuration.

We constructed a set of analogous recurrent networks for scanning the sequence and detecting targeting peptides. We employ the bi-directional architecture proposed by Baldi *et al.*[2] for which the recurrent flow happens in both directions. Two input windows begin from a specified distance away from the target residue and move inward toward it. By iteratively creating a state from the residues next to each position in the sequence, the middle residue is classified as being part of the specific targeting peptide or not (see Figure 2). Thus the central residue is classified on the basis of the sequential content on either side. We tried a few configurations and the results reported below are taken from recurrent networks with wheels of $k = 10$ residues both from the N-terminal and the C-terminal flank. States consist of $h = 4$ nodes of which all are fully recurrent (all nodes feed back to all others within the same state layer). As configurations have yet to be fully explored, we do not claim that the reported configuration is optimal. We use the same configuration ($k = 10$ and $h = 4$) for both plant and non-plant data, and for all subcellular targets.

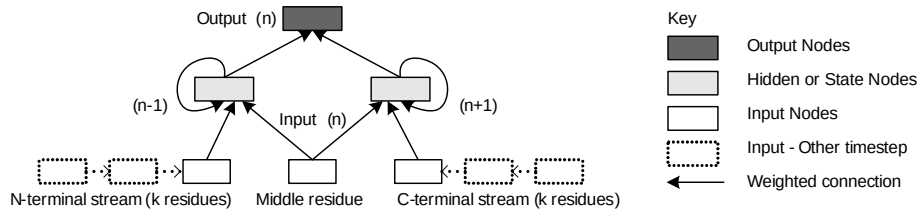

Fig. 2. The recurrent targeting peptide detection network operates by traversing the sequence from two directions, accumulating two separate states, until the middle residue is reached, and when the network produces the classification (part of targeting peptide '1' or not '0') at its output. As an example, the symbol G within the sequence ABCDEFGHIJKLM is classified by presenting a window of residues, say 2, from each direction: [AB:-:LM], then [CD:-:JK] and finally [EF:G:HI], where '-' represents a *nil* pattern (all zeros) and ':' indicates node bank boundaries between residues taken from the N-terminal flank, the residue at the point of prediction, and residues taken from the C-terminal flank, respectively.

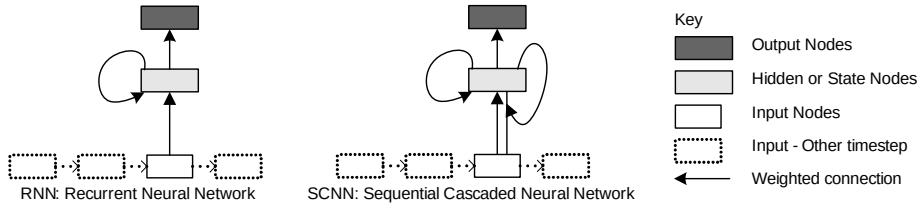

Fig. 3. Illustration of a uni-directional recurrent network and a uni-directional sequential cascaded neural network. The sequential cascaded architectures has a basic structure identical to the recurrent neural network, however, it possesses a second set of recurrent connections, that multiply current input activations with hidden node activations from the previous time step.

In this paper we also explore the use of sequential cascaded recurrent networks[13] for the task of peptide detection. This avenue of exploration is motivated by the fact that, as a second order network, the SCNN has a greater capacity to represent non-linear relationships between input and output. Secondly, our initial exploratory simulations indicate that it as least as promising as the simple recurrent neural network for recognizing the required features within biological sequences.[10]

The sequential cascaded recurrent neural networks have a design that is similar to that of the standard recurrent networks. The essential difference between a uni-directional SCNN and a simple recurrent network is depicted in Figure 3. The SCNN has an additional recurrent connection that merges with a second connection from the input layer. The weights in this merged connection are applied to all possible second order multiplicative combinations of the current activations on the input nodes with the activations on the hidden nodes from the previous time step. These second-order weights allows the machine to take input in a highly context sensitive fashion.[13] The sequential cascaded networks are constructed as bi-directional net-

works with input windows of $k = 10$ residues and $h = 4$ hidden nodes, just like the standard, first-order recurrent models.

In all cases, we use the softmax output function and the cross entropy error function. All networks are trained using backpropagation and for both the recurrent architectures the error is "unfolded" through the sequence both upstream and downstream as described by Baldi *et al.*[2] For practical reasons the error flow is truncated after five steps. For both feed forward and recurrent networks, the learning rate (η) is fixed to 0.01, and all weight values are randomly initialized with a Gaussian distribution around 0.0 (variance 0.1). By monitoring errors throughout learning, slow convergence and minor fluctuations were noted. However, the consistency of generalization results reported below denies the presence of major learning issues.

3.2. *Non-plant proteins*

The data is divided into plant and non-plant proteins. Non-plant proteins are used to train two separate targeting peptide detection networks: one for SP, and one for mTP. A third class (other) of proteins is used as additional negatives for both networks. Training is performed by presenting each network with a sequence randomly drawn from the training subsets (uniformly over the target classes). The sequence is processed by training the network to classify each residue as '1' or '0', in the same manner as TargetP.[9]

Table 1. The mean squared error over all non-plant sequences in the dataset for the each of the detector networks.

Target	FFNN	RNN	SCNN	NULL
SP	0.0183 (0.0006)	0.0146 (0.0008)	0.0146 (0.0006)	0.0593 (0.0)
mTP	0.0298 (0.0027)	0.0257 (0.0043)	0.0254 (0.0028)	0.0454 (0.0)

Note: The results are over five repeats of each five-fold cross-validated configuration. Standard deviations between repeats are given in parenthesis. The NULL column reflects the average error for a control machine that always emits a zero.

After 30,000 training sequences have been presented, the actual output for each position in each test sequence is recorded. Moreover, the squared difference between the target output ('1' or '0') and the actual output is used to assess the classification ability of the network. As the cleavage site determines the end of the string of 1's, the error indicates success of both the classification of the peptide and identification of the cleavage point. In Table 1 the mean errors are shown for all three types of networks on the two non-plant subcellular targets. Residues within signal peptides are generally easy to detect for both network types. However, both of the recurrent networks are 25% better than TargetP's SP detection network. When detecting mitochondrial targeting peptides the recurrent architectures also exhibit an advantage

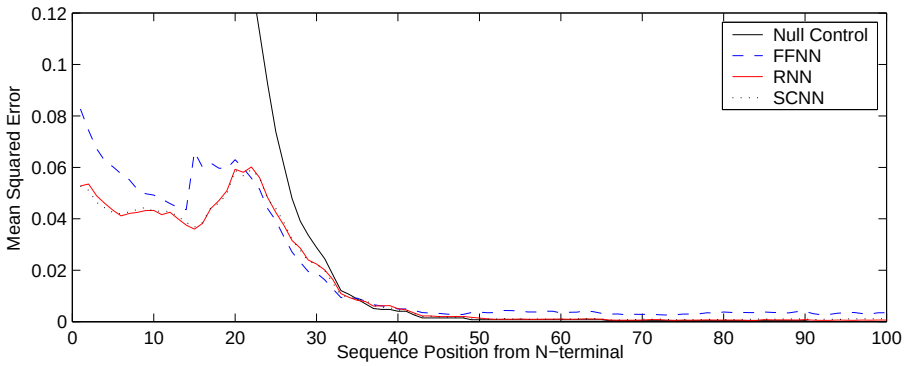

Fig. 4. Mean squared errors for the three types of SP targeting peptide detection networks averaged over all 2738 input sequences in the non-plant dataset.The results are displayed as a function of sequence position (1-100). All errors are means over five repeats of the five-fold cross-validated. The null control asymptotes to a value of approximately 0.27, but has been cut from the graph to show more detail of the detection network errors.

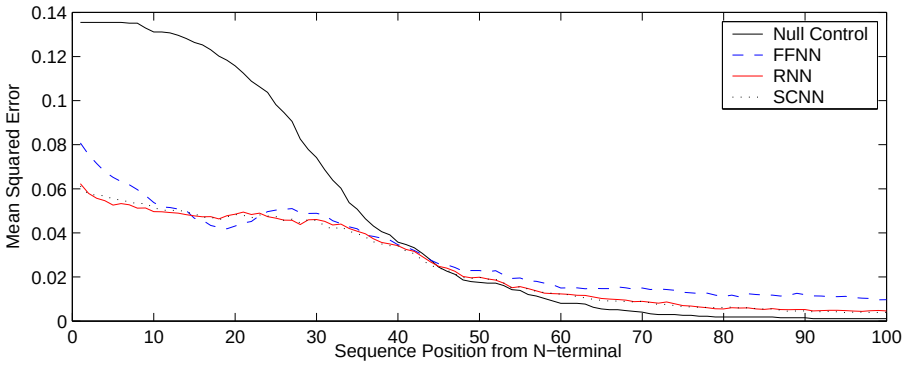

Fig. 5. Mean squared errors for the three types of mTP targeting peptide detection networks averaged over all 2738 input sequences in the non-plant dataset. The results are displayed as a function of sequence position (1-100). All errors are means over five repeats of the five-fold cross-validated. The null control asymptotes to a value of approximately 0.14.

over the feed forward networks (16%). The null column depicts the mean error for a control machine that always emits a zero, this being the most common desired output. All machines a performing significantly better than this control.

The average error for signal peptides in the detection networks as a function of position along the sequence is shown in Figure 4. The error calculation involves outputs for all sequences in the dataset, but only using those networks in the cross-validation sets that were not trained on the particular sequence. The cleavage site of signal peptides is usually located at position 15-30 of the nascent protein relative

to the N-terminal end (Mean=23, SD=6, in the data set). The classification error is generally higher around the cleavage site. However, the error is considerably higher for the feed forward network employed by TargetP for most residues preceding the cleavage site. After the cleavage site, both network types are generally performing equally well.

The errors for mitochondrial test sequences are similarly analyzed (see Figure 5). The performance of recurrent targeting peptide detection networks is considerably better before and around the cleavage sites of the nascent protein sequence. The cleavage sites of matrix mitochondrial processing peptidases occur further along the nascent protein (Mean=34, SD=16, in our data). Being very close to their mean, the error profiles of individual networks show little variation.

3.3. *Plant proteins*

Table 2. The mean squared error over all plant sequences in the dataset for the each of the detector networks.

Target	FFNN	RNN	SCNN	NULL
SP	0.0175 (0.0009)	0.0144 (0.0004)	0.0146 (0.0002)	0.0677 (0.0)
mTP	0.0607 (0.0018)	0.0558 (0.0033)	0.0557 (0.0018)	0.1305 (0.0)
cTP	0.0554 (0.0028)	0.0636 (0.0104)	0.0619 (0.0102)	0.0831 (0.0)

Note: The results are over five repeats of each five-fold cross-validated configuration. Standard deviations between repeats are given in parenthesis. The NULL column reflects the average error for a control machine that always emits a zero.

For plant proteins in our data set there are three targets and both the recurrent network architectures improve on the feed forward networks employed by TargetP for SP and mTP sequences. cTP sequences are handled better by the original feed forward detector networks. However, this advantage is only present in the latter end of the sequence (after position 55). See Table 2 for the average squared error for each of the detection networks. Figures 6, 7 and 8 show the position-specific error profiles for signal peptides, mitochondrial targeting peptides and chloroplast targeting peptides, respectively.

The factors determining localization are complex in nature, as exemplified by the existence of proteins with dual targets. A growing number of proteins have been observed to be translocated to both mitochondria and chloroplasts.[11] As an illustration of the comparative biases of the three different network architectures we have taken one example from the list in Peeters and Small's study.[11] The protein P27456, Glutathione reductase, was fed through each of the detection network architectures trained for both the mitochondrial and chloroplast targeting problems. The protein is more frequently found in the chloroplast, only 3% in mitochondria, which was well reflected by all three mTP detection networks. The feed forward architecture gave

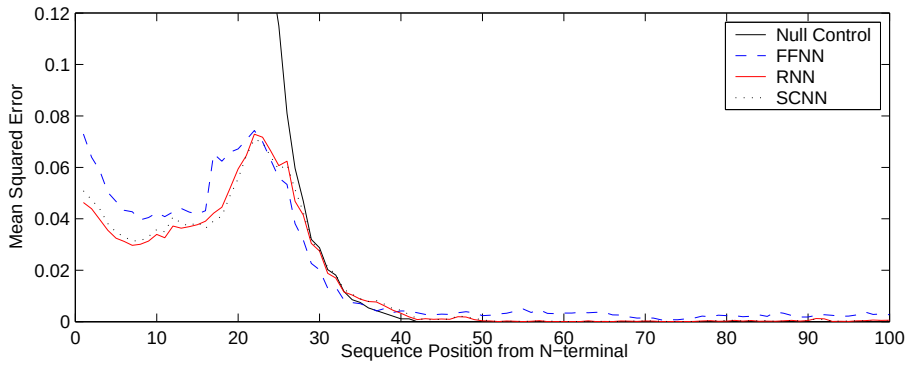

Fig. 6. Mean squared errors for the three types of SP targeting peptide detection networks averaged over all 940 input sequences in the plant dataset. The results are displayed as a function of sequence position (1-100). All errors are means over five repeats of the five-fold cross-validated. The null control asymptotes to a value of approximately 0.28, but has been cut from the graph to show more detail of the detection network errors.

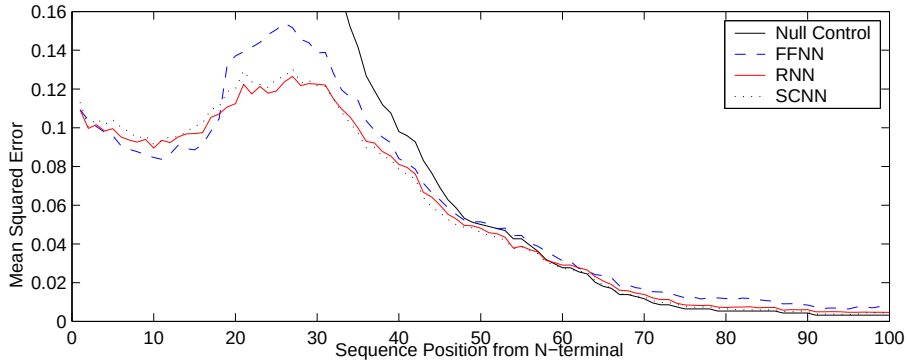

Fig. 7. Mean squared errors for the three types of mTP targeting peptide detection networks averaged over all 940 input sequences in the plant dataset. The results are displayed as a function of sequence position (1-100). All errors are means over five repeats of the five-fold cross-validated. The null control asymptotes to a value of approximately 0.4, but has been cut from the graph to show more detail of the detection network errors.

the only significantly non-zero outputs for mTP detection, but produced no clear indication of mitochondrial targeting. The cTP predictors were far more definitive, as can be seen in Figure 9. The protein was annotated with a potential cleavage site at position 60 which both recurrent architectures correctly locate. The feed forward architecture places it closer to position 50. It is worth noting that, despite their overall agreement, the recurrent architectures produce slightly different output profiles in this task, indicating sensitivity to different features in the sequence.

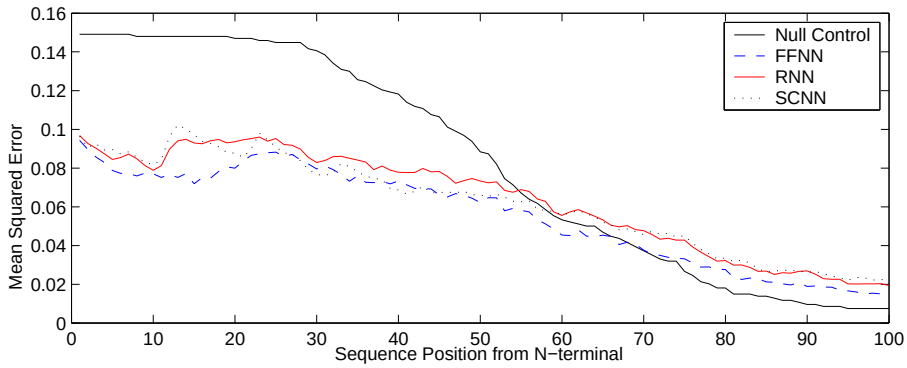

Fig. 8. Mean squared errors for the three types of cTP targeting peptide detection networks averaged over all 940 input sequences in the plant dataset. The results are displayed as a function of sequence position (1-100). All errors are means over five repeats of the five-fold cross-validated. As is shown the null control asymptotically approaches a value around 0.15 toward the N-terminal end.

4. Sorting Machines

In order to obtain a specific prediction of the localization of a protein the outputs of the target detection networks must be interpreted. In the original TargetP this was done using second 'sorting' neural network, trained on the outputs of the residue detection networks. In previous studies we replicated this design decision, as well as employing the k-nearest neighbors algorithm as an alternative sorting mechanism.[5] We found that neither sorting mechanism is able to make use of the increased accuracy of the recurrent neural networks at the residue detection level. We take this failure as an indication that the task requires a non-linear classifier capable of dealing with subtle distinctions in high dimensional input spaces.

In addressing the short comings of previous sorting techniques, we chose to experiment with a range of support vector machines due to their affinity for problems of a sparse and high-dimensional kind. We tested a Gaussian ($\gamma = 0.01$), linear and polynomial kernels of second- and third order. The SVM implementation employed is that provided in the WEKA library.[20] For our benchmark simulations we use the default value of C=1 (the complexity constant for soft-margin control).

In the initial simulations the support vector machines are trained to make an overall prediction of protein localization based on the outputs of a single targeting peptide detection network. For the purposes of a control we also train a set of SVM with the same kernels and parameters on the raw sequence encoded using an orthonormal encoding. The control provides a baseline of classification, and establishes the validity of employing targeting peptide detection networks. The difference between the control and the multi-layer classifiers gives an indication of the advantage being supplied by the targeting peptide detection networks.

We evaluate the comparative accuracy of each of the sorters by performing five

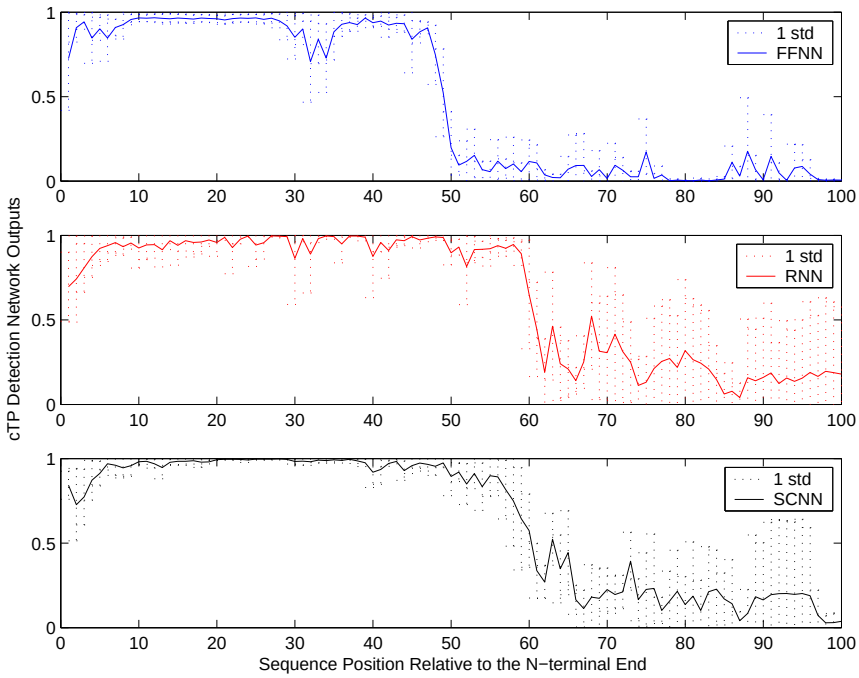

Fig. 9. The cTP detection network outputs for the dual-targeted protein P27456. Each graph shows the outputs with one standard deviation for a particular machine architecture. The sequence is predominantly a chloroplast protein (97%). It has a potential cleavage site at position 60.

fold cross validation. For each simulation run five models are created and trained on four fifths of the data set. We ensure that for each run the data set is split as in the training of the residue detection networks. Thus, each machine is composed of a set of detection networks and a sorting machine that have all been trained on exactly the same subset of the full dataset. After training, each sorter is tested on the remaining fifth, each of the five models in the cross validation is tested on a different fifth of the dataset. We use the resulting performance results to produce a generalization metric called the Matthews Correlation Coefficient MCC.[1]

As can be seen in Tables 3 and 4, the control SVMs are not always able to perform the task adequately enough that the performance metrics could be calculated. In most of these cases the classifier simply classify everything as "other", resulting in a denominator of zero for some of these metrics. For all other classifiers the best MCC occurs with a Gaussian kernel, regardless of the type of detection network that is used. The best overall score for non-plant data is with feed forward detection networks, whereas for the plant data the best MCC is given by the simple recurrent neural networks. It is worth noting that when comparing to the original TargetP results the improved MCC comes at a cost of a slight reduction in sensitivity but

Table 3. Average (across all classes) of sensitivity, specificity and the MCC for all predictors on the non-plant data.

Sorting Results Non-plant Data					
Detection Network	Sorter Kernel	Sensitivity	Specificity	MCC	Accuracy
Null Control	SVM Gauss	0.333	NaN	NaN	60.3%
	SVM Linear	0.744	0.769	0.646	81.8%
	SVM 2order	0.530	0.883	0.452	75.1%
	SVM 3order	0.334	NaN	NaN	60.4%
FFNN	SVM Gauss	0.902	0.917	0.867	93.1%
	SVM Linear	0.898	0.912	0.860	92.7%
	SVM 2order	0.870	0.889	0.823	90.8%
	SVM 3order	0.857	0.880	0.807	90.0%
RNN	SVM Gauss	0.904	0.907	0.860	92.6%
	SVM Linear	0.899	0.906	0.856	92.4%
	SVM 2order	0.877	0.894	0.831	91.2%
	SVM 3order	0.863	0.887	0.815	90.4%
SCNN	SVM Gauss	0.905	0.909	0.863	92.8%
	SVM Linear	0.901	0.906	0.858	92.6%
	SVM 2order	0.872	0.890	0.826	91.0%
	SVM 3order	0.856	0.879	0.805	89.9%

Note: The performance statistics for a range of support vector machines trained to classify the subcellular localization of non plant proteins. The control results are generated by training the SVM to perform the classification based on an orthonormal encoding of the first hundred residue of the sequence. Each of the other rows of results reports on the performance of an SVM trained on the output of a particular residue detection architecture. Results are averages from five repeats of five-fold cross-validated test data. Accuracy depicts percentage of sequences correctly classified.

with improved specificity.

In a second set of sorting simulations we train SVMs to make the overall classification using the output vectors from some combination of detection networks. The goal of these simulations is to utilize the sensitivities of each of the network architectures to different sequence features. By using a combination of detection networks the sorting machine has access to several perspectives on the relevant features within the protein sequence. To highlight the usefulness of recurrent machines for target peptide detection we include a control simulation that involved a combination of feed forward detection networks. These are two sets of feed forward neural networks trained on the same cross validation sets using different initialization seeds for the network weights.

In Tables 5 and 6 it can be seen that sorting machines trained on the outputs of two different detection networks improve upon the the results of their respective individual sorters. Although the dual feed forward architecture offers and improvement over the single feed forward architectures shown in tables 3 and 4, it is not as significant as the improvement offered by the combination of the feed forward

Table 4. Sensitivity, Specificity and MCC for all predictors on the plant data.

Sorting Results Plant Data					
Detection Network	Sorter Kernel	Sensitivity	Specificity	MCC	Accuracy
Null Control	SVM Gauss	0.251	NaN	NaN	39.3%
	SVM Linear	0.699	0.740	0.628	74.6%
	SVM 2order	0.568	1.073	0.492	76.8%
	SVM 3order	0.397	NaN	NaN	57.7%
FFNN	SVM Gauss	0.866	0.870	0.827	88.4%
	SVM Linear	0.851	0.854	0.808	87.1%
	SVM 2order	0.838	0.839	0.789	85.8%
	SVM 3order	0.833	0.830	0.779	85.0%
RNN	SVM Gauss	0.872	0.870	0.831	88.6%
	SVM Linear	0.867	0.864	0.825	88.2%
	SVM 2order	0.855	0.850	0.807	86.9%
	SVM 3order	0.841	0.831	0.785	85.4%
SCNN	SVM Gauss	0.869	0.867	0.827	88.3%
	SVM Linear	0.857	0.859	0.815	87.7%
	SVM 2order	0.847	0.842	0.796	86.2%
	SVM 3order	0.835	0.825	0.777	84.7%

Note: See Table 3 for details.

and recurrent architectures. This indicates that the improvement is a mixture of an ensemble effect as well as an advantage of combining the biases of multiple architectures. However, the sorting machines trained on the output of all three do not deviate significantly from the performance of the best combined sorter using only two detection networks. The biases of the recurrent architectures are possibly too similar to provide a significant advantage. Again, when compared to TargetP, the improved MCC comes about by sacrificing some sensitivity for the sake of specificity.

We further refine these results by tuning the parameters γ and C. For practical reasons we assume independence between these parameters and tune γ first followed by C. This is only successful in improving the results of the plant protein classifier, for which we settle on a γ value of 0.001 and C value of 6.

In our previous experiments with sorting networks we found that performance could be improved using a naive ensemble consisting of four classifiers: two of which employ feed forward detection networks and two with recurrent networks. For each network class, one of the two classifiers used a feed forward sorting network, while the other used k-nearest neighbor.[5] For the non-plant data the combined SVM sorter is performing only marginally better than the naive ensemble method, but for the plant data we see an improvement from 0.834 to 0.849, a 2% increase.

For the final model we employ the technique of fitting a logistic model to the outputs of the SVM so it would produce a probability distribution. Following Platt's recommendation we use a three fold cross validation in this procedure to avoid problems with support vector bias.[12] The use of the logistic model has a minimal effect

Table 5. Average (across all classes) of sensitivity, specificity and MCC for the combined predictors on the non-plant data.

Combined Sorters Results Non-Plant Data				
Networks	Sorter	Sensitivity	Specificity	MCC
TargetP	-	0.910	0.853	0.823
PProwler V1.1	-	0.919	0.908	0.872
FFNN	SVM Gauss	0.902	0.921	0.869
FFNN	SVM Linear	0.895	0.908	0.855
	SVM 2order	0.869	0.887	0.820
	SVM 3order	0.854	0.875	0.801
FFNN	SVM Gauss	0.907	0.919	0.872
RNN	SVM Linear	0.895	0.907	0.854
	SVM 2order	0.878	0.896	0.833
	SVM 3order	0.866	0.883	0.814
RNN	SVM Gauss	0.907	0.914	0.868
SCNN	SVM Linear	0.897	0.907	0.856
	SVM 2order	0.872	0.890	0.824
	SVM 3order	0.862	0.881	0.810
FFNN	SVM Gauss	0.908	0.921	0.874
SCNN	SVM Linear	0.896	0.908	0.855
	SVM 2order	0.875	0.893	0.829
	SVM 3order	0.860	0.880	0.807
FFNN	SVM Gauss	0.909	0.921	0.875
SCNN	SVM Linear	0.894	0.906	0.853
RNN	SVM 2order	0.875	0.893	0.829
	SVM 3order	0.854	0.874	0.798

Note: The performance statistics for support vector machines trained as sorting classifiers using combinations of detection networks. The top two rows show the statistics for TargetP and PProwler V1.1 for comparison. All other major rows show results generated by training a support vector machine to make a classification based on the output of two or three different sets of residue detection networks. Results are averages from five repeats of five-fold cross-validated test data.

on the qualitative performance of the models. The performance statistics for the final tri-combined model compared to both the original TargetP and PProwler V1.1 are shown in Tables 7 and 8. We also include performance statistics broken down by each of the targets for subcellular localization. Although the overall improvements are small, it is worth noting that the standard deviations for PProwler V1.2 are significantly smaller than those of V1.1, thus we are confident of this as an estimate of the genuine accuracy of the model on the entire problem.

Table 6. Average (across all classes) of sensitivity, specificity and MCC for the combined predictors on the plant data.

Combined Sorters Results Plant Data				
Networks	Sorter	Sensitivity	Specificity	MCC
TargetP	-	0.858	0.830	0.790
PProwler V1.1	-	0.880	0.866	0.834
FFNN	SVM Gauss	0.868	0.874	0.832
FFNN	SVM Linear	0.847	0.851	0.804
	SVM 2order	0.845	0.846	0.798
	SVM 3order	0.832	0.830	0.778
FFNN	SVM Gauss	0.875	0.882	0.842
RNN	SVM Linear	0.854	0.859	0.813
	SVM 2order	0.850	0.851	0.804
	SVM 3order	0.840	0.838	0.788
RNN	SVM Gauss	0.876	0.879	0.839
SCNN	SVM Linear	0.864	0.865	0.823
	SVM 2order	0.852	0.848	0.804
	SVM 3order	0.836	0.827	0.780
FFNN	SVM Gauss	0.870	0.878	0.835
SCNN	SVM Linear	0.850	0.855	0.808
	SVM 2order	0.847	0.848	0.801
	SVM 3order	0.841	0.839	0.790
FFNN	SVM Gauss	0.873	0.884	0.841
SCNN	SVM Linear	0.852	0.858	0.811
RNN	SVM 2order	0.849	0.850	0.804
	SVM 3order	0.840	0.837	0.788

Note: See Table 5 for details.

Table 7. Final Plant Model

	Average Performance		
	Sensitivity	Specificity	MCC
TargetP	0.858	0.830	0.790
PProwler V1.1	0.880 (0.0199)	0.866 (0.0191)	0.834 (0.0192)
PProwler V1.2	0.882 (0.0084)	0.886 (0.0082)	0.849 (0.0099)
Target	PProwler V1.2 Performance by Subcellular Organelle		
SP	0.947 (0.0031)	0.964 (0.0054)	0.938 (0.0028)
mTP	0.911 (0.0015)	0.901 (0.0010)	0.845 (0.0012)
cTP	0.783 (0.0256)	0.859 (0.0197)	0.790 (0.0262)
Oth	0.886 (0.0103)	0.822 (0.0098)	0.822 (0.0118)

Note: Final performance statistics for PProwler V1.2 compared with V1.1 and TargetP. The classifier uses a Gaussian SVM sorter (γ =0.001, C=6) trained on the combined outputs of all three detection networks. Values are means calculated from five-fold cross-validations, run five times with a different seed for the data set split.

Table 8. Final Non-Plant Model

	Average Performance		
	Sensitivity	Specificity	MCC
TargetP	0.910	0.853	0.823
PProwler V1.1	0.919 (0.0069)	0.908 (0.0092)	0.872 (0.0080)
PProwler V1.2	0.900 (0.0023)	0.929 (0.0033)	0.873 (0.0037)
Target	PProwler V1.2Performance by Subcellular Organelle		
SP	0.937 (0.0036)	0.948 (0.0033)	0.922 (0.0037)
mTP	0.799 (0.0068)	0.905 (0.0103)	0.829 (0.0079)
Oth	0.963 (0.0017)	0.934 (0.0018)	0.867 (0.0043)

Note: Final performance statistics for PProwler V1.2 compared with V1.1 and TargetP. The classifier uses a Gaussian SVM sorter (γ=0.05, C=1) trained on the combined outputs of all three detection networks. Values are means calculated from five-fold cross-validations, run five times with a different seed for the data set split.

5. Conclusion

We note that both recurrent network architectures are notably better than the feed forward networks used by TargetP at classifying residues as belonging to a targeting peptide. The advantage is particularly clear within the window believed to exhibit the strongest signals used by the translocation machinery.[8] The reason for the success lies partly in the fact that recurrent networks are naturally biased towards detecting sequential patterns.[4,17]

Previous iterations of the presented model used feed forward neural networks and k-nearest neighbors trained on the output of a single type of residue detection network.[9,5] Neither sorting mechanism is able to significantly exploit the improvements in targeting peptide detection accuracy offered by recurrent architectures. In our simulations we are able to improve on the performance of these individual sorters by using support vector machines with Gaussian kernels. The results are further improved by employing a composite machine that is trained on the output of several different residue detection networks. The most significant improvement, from machines using a single class of detection networks, comes by using the feed forward networks in combination with one of the recurrent architectures, boosting performance by approximately 1% for both non-plant and plant data. The use of the sequential cascaded network boosts performance when used in combination with the simple recurrent network, but contributes little to the combined sorter using all three detection networks. We suspect that the biases of the recurrent networks are too similar to provide added insight when used in combination with another architecture. It may still be possible to improve performance through the employment of a different class of detection networks with a unique bias.

Our previous release of PProwler outperforms TargetP, which has itself been demonstrated to be the most accurate of the available classifiers for subcellular

localization.[8] Our final model improves upon the accuracy of PProwler V1.1 by 2% for the plant data, and resulted in lower standard deviations across simulations for both predictors. PProwler V1.2 thus constitutes the most accurate predictor to date. The composite targeting peptide detection mechanism is integrated into the *Protein Prowler* prediction service available online at `http://pprowler.imb.uq.edu.au/`.

References

1. P. Baldi, S. Brunak, Y. Chauvin, C. A. F. Andersen, and H. Nielsen. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16(5):412–424, 2000.
2. P. Baldi, S. Brunak, P. Frasconi, G. Soda, and G. Pollastri. Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics*, 15:937–946, 1999.
3. M. Bodén and J. Hawkins. Detecting residues in targeting peptides. In *Proceedings of the Asia-Pacific Bioinformatics Conference*, 2005.
4. M. Bodén and J. Hawkins. Improved access to sequential motifs: A note on the architectural bias of recurrent networks. *IEEE Transactions on Neural Networks*, 16(2), 2005.
5. M. Bodén and J. Hawkins. Prediction of subcellular localisation using sequence-biased recurrent networks. *Bioinformatics*, 21:2279–2286, 2005.
6. M. P. S. Brown, W. N. Grundy, D. Lin, N. Cristianini, C. W. Sugnet, T. S. Furey, J. Ares, Manuel, and D. Haussler. Knowledge-based analysis of microarray gene expression data by using support vector machines. *PNAS*, 97(1):262–267, 2000.
7. J. L. Elman. Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning*, 7:195, 1991.
8. O. Emanuelsson. Predicting protein subcellular localisation from amino acid sequence information. *Briefings in Bioinformatics*, 3(4):361–376, 2002.
9. O. Emanuelsson, H. Nielsen, S. Brunak, and G. von Heijne. Predicting subcellular localization of proteins based on their n-terminal amino acid sequence,. *Journal of Molecular Biology*, 300(4):1005–1016, 2000.
10. J. Hawkins and M. Bodén. The applicability of recurrent neural networks for biological sequence analysis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2(2), 2005.
11. N. Peeters and I. Small. Dual targeting to mitochondria and chloroplasts. *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research*, 1541(1-2):54–63, 2001.
12. J. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In A. Smola, P. Bartlett, B. Schoelkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 2000.
13. J. B. Pollack. The induction of dynamical recognizers. *Machine Learning*, 7:227, 1991.
14. G. Pollastri, D. Przybylski, B. Rost, and P. Baldi. Improving the prediction of protein secondary strucure in three and eight classes using recurrent neural networks and profiles. *Proteins: Structure, Function, and Genetics*, 47:228–235, 2002.
15. M. Reczko and A. Hatzigeorgiou. Prediction of the subcellular localization of eukaryotic proteins using sequence signals and composition. *Proteomics*, 4(6):1591–1596, Jun 2004.
16. G. Schneider and U. Fechner. Advances in the prediction of protein targeting signals. *Proteomics*, 4(6):1571–1580, 2004.
17. P. Tino, M. Cernansky, and L. Benuskova. Markovian architectural bias of recurrent neural networks. *IEEE Transactions on Neural Networks*, 15(1):6–15, 2004.
18. V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

19. A. Vullo and P. Frasconi. Disulfide connectivity prediction using recursive neural networks and evolutionary information. *Bioinformatics*, 20(5):653–659, 2004.
20. I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, San Francisco, 2000.