

# Exploiting Sequence Dependencies in the Prediction of Peroxisomal Proteins

Mark Wakabayashi<sup>1,2</sup>, John Hawkins<sup>2</sup>, Stefan Maetschke<sup>2</sup>, and Mikael Bodén<sup>2†</sup>

<sup>1</sup> ARC Centre for Complex Systems

<sup>2</sup> School of Information Technology and Electrical Engineering,  
The University of Queensland, Australia

†mikael@itee.uq.edu.au

**Abstract.** Prediction of peroxisomal matrix proteins generally depends on the presence of one of two distinct motifs at the end of the amino acid sequence. PTS1 peroxisomal proteins have a well conserved tripeptide at the C-terminal end. However, the preceding residues in the sequence arguably play a crucial role in targeting the protein to the peroxisome. Previous work in applying machine learning to the prediction of peroxisomal matrix proteins has failed to capitalize on the full extent of these dependencies. We benchmark a range of machine learning algorithms, and show that a classifier – based on the Support Vector Machine – produces more accurate results when dependencies between the conserved motif and the preceding section are exploited. We publish an updated and rigorously curated data set that results in increased prediction accuracy of most tested models.

## 1 Introduction

A cell requires that each of its many proteins are localized to the appropriate compartment or membrane. Of the smaller cellular compartments, the peroxisome is an organelle lined by a single membrane, lodging essential enzymes for a variety of specialized functions (e.g. lipid metabolism). All peroxisomal proteins are nuclear encoded, synthesized on free ribosomes in the cytosol, folded, and inserted into the organelle via at least two pathways. The import process is not fully understood but involves receptor proteins in the cytosol which recognize a signal of the newly synthesized protein and protein docking complexes at the surface of the peroxisome. Several diseases are caused by deficient peroxisomal import (e.g. Zellweger’s disease), making the peroxisomal import machinery a prime research target.

The vast majority of proteins localised to the peroxisomal matrix rely on a short motif on the C-terminal end of the sequence called the PTS1 signal. PTS1 is often described as the tripeptide SKL with some substitution flexibility. Even though this motif is highly conserved in most peroxisomal proteins, it is also present in many non-peroxisomal proteins. Statistical analysis of the PTS1 transport mechanism has indicated that the last twelve C-terminal residues of

the protein sequence are the most significant determinants of PTS1 peroxisomal proteins [6, 5].

The PeroxiP predictor [3] uses a three stage process to predict peroxisomal localisation via the PTS1 pathway. Stage one makes use of existing predictors to eliminate sequences that are targeted to other organelles or are predicted to have a membrane spanning region.<sup>3</sup> Secondly, a motif identification module examines the C-terminus of the sequence and rejects all sequences without an approved PTS1 motif. In the final stage a machine learning module predicts whether the PTS1 bearing protein is peroxisomal. For this last step, PeroxiP employs a neural network and a Support Vector Machine that are trained independently and operate in union. The sequence is predicted as peroxisomal if either model indicates so [3]. Importantly, the models in PeroxiP are distinctive in evaluating only the 9-mer of amino acids that precede the PTS1 motif, implying that the specific PTS1 motif had no further information to offer. However, independent statistical analysis demonstrated significant correlation between positions -1 and -2 of the PTS1 with positions between -3 and -6 of the 9-mer [6]. As previous work in machine learning has failed to capitalize on the full range of the PTS1 signal, we develop a new, extensive but non-redundant data set and explore a variety of machine learning techniques. We show that dependencies between the 9-mer and the elements of the PTS1 motif can be exploited.

## 2 Sequence data

Data for training and testing classifiers is collected in accordance with the methodology employed in [3] but from release 45 of SWISS-PROT. In brief, all entries with a `SUBCELLULAR LOCATION` annotation in the comments field that included any of `PEROXISOM`, `GLYOXYSOM`, or `GLYCOSOM` using a case-insensitive search were initially considered as positives. Each protein was also required to identify a `MICROBODY TARGETING SIGNAL` in the feature table, indicating a PTS1 target signal, resulting in an initial set of 202 proteins. The initial set was then filtered manually but conservatively for proteins not likely to be targeted by a PTS1 and for membrane proteins. An initial set of 573 negatives was similarly created, requiring a C-terminal tripeptide identical to one of the initially identified positives, and a subcellular location not specified as peroxisomal, glyoxysomal or glycosomal. By consulting the literature for suspicious cases, a few records appeared to be erroneously annotated in SWISSPROT and shuffled between the positive and negative subsets. The resulting data set had 206 positives and 564 negatives. Given this overrepresentation of non-peroxisomal proteins and to increase the quality of the data set, the negative subset was cleaned of all proteins whose subcellular location was qualified as `POTENTIAL`, `PROBABLE`, or `BY SIMILARITY` leaving 348 proteins, a sufficient size given the smaller size of the positive subset.

---

<sup>3</sup> The localisation of peroxisomal membrane proteins is governed by a separate set of signals [4], often consisting partly of a membrane spanning region.

Both steps of redundancy reduction employed in [3] were performed. Highly similar proteins were first removed such that each pair of proteins differed in at least two positions in the nine residues preceding the C-terminal tripeptide. The final stage of redundancy reduction was performed using BLASTClust (clustering on basis of pairwise sequence alignments). In order to reproduce a data set of the same size as in [3] we found that a similarity threshold of 1.675 was required. The final sequence data set, henceforth called the 2005-set, consisted of 124 peroxisomal proteins and 214 non-peroxisomal proteins. To ensure fair comparisons we also created what we believe to be a close replica of the data set originally used in [3] consisting of 90 positives and 160 negatives, henceforth referred to as the 2003-set (based on SWISS-PROT release 39.27).

### 3 Simulations

We conducted two sets of exploratory simulations during the development of the PTS1 peroxisomal predictor. In the first set of simulations we benchmarked a range of machine learning models on both the replicated 2003-data set and our new 2005-data set.

In the second set of simulations we took the better of the machine learning models and explored the effect of using different input window sizes. This was done in the interests of producing an optimal window size for the final classifier.

#### 3.1 Machine learning algorithms

For the benchmarking study we used the WEKA library [7] and deployed the following machine learning algorithms: a naive Bayes classifier, multilayer and single layer perceptrons, Support Vector Machines, a k-nearest neighbor classifier and the C4.5 decision tree algorithm. The algorithms were evaluated using the Matthews correlation coefficient (MCC)<sup>4</sup>. We tested a range of encodings of which an orthonormal encoding gave consistently better performance for all of the machine architectures, thus all results reported herein were generated with this encoding.

The outcome of benchmarking machine learning algorithms is summarised in Table 1. In agreement with [3] the polynomial support vector machine performed best. The Naive Bayes classifier achieved the lowest MCC, which we conclude is due to the non-negligible dependencies between sequence positions. Also the C4.5 decision tree algorithm seemed unable to identify these dependencies and was only slightly better than the Naive Bayes classifier. The k-nearest neighbor classifier performed surprisingly well (second best) with a k-value equals one, indicating that the class boundaries are not strongly overlapping, thus the noise level of the data is quite low. We tested a number of neural network architectures,

<sup>4</sup> The MCC is defined as  $(tp \cdot tn - fp \cdot fn) / \sqrt{(tp + fn)(tp + fp)(tn + fp)(tn + fn)}$ , where  $tp$  is the number of true positives,  $tn$  is the number of true negatives,  $fp$  is the number of false positives and  $fn$  is the number of false negatives. Higher MCC is better (max is 1).

Data set	Machine Learning Model					
	NB	SLP	SVM(P2)	SVM(G)	KNN(1)	C4.5
2003	0.30 (0.12)	0.40 (0.13)	0.46 (0.12)	0.38 (0.12)	0.43 (0.12)	0.23 (0.12)
2005	0.27 (0.10)	0.40 (0.11)	0.59 (0.09)	0.47 (0.09)	0.53 (0.11)	0.28 (0.12)

**Table 1.** The average MCC (std) over all models (5-fold cross validation over 10 runs). NB – Naive Bayes, SLP – Single Layer Perceptron, SVM(P2) – Support Vector Machine with Polynomial Kernel Order 2, SVM(G) – Support Vector Machine with Gaussian Kernel (var=0.1), KNN(1) – K-nearest neighbours (k=1), C4.5 – Decision tree algorithm (rev. 8).

single and multilayer perceptrons, varying the number of hidden neurons. Of these, the Single Layer Perceptron (i.e. no hidden neurons at all) performed best, although the 2 and 3 node MLPs were very close behind (results not shown). The highest MCC was achieved with a Support Vector Machine with a polynomial kernel of order two. The linear and the Gaussian kernels were inferior.

Dataset	SVM(P) - Polynomial Order				
	1	2	3	4	5
2003	0.30 (0.11)	0.46 (0.12)	0.47 (0.11)	0.34 (0.12)	0.24 (0.12)
2005	0.34 (0.10)	0.59 (0.09)	0.59 (0.08)	0.47 (0.10)	0.37 (0.10)

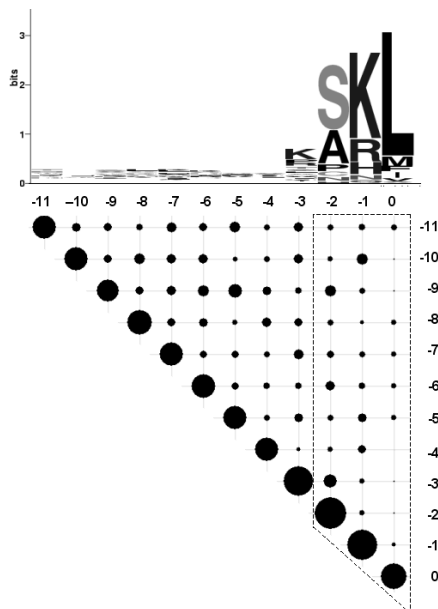
**Table 2.** The average MCC (std) for Support Vector Machines with a polynomial kernel of varying orders. 5-fold cross validation over ten runs.

In Table 2 we see that the optimum MCC occurs with a polynomial kernel of order 2. We noted that the specificity of the models increased with the degree of the polynomial kernel (advancing from 0.834 for order 2 to 1.000 for order 5 and beyond) as the sensitivity went down. Thus, in an ensemble of predictors a higher order may be preferred (cf. [3]).

### 3.2 Range of inputs

The developers of PeroxiP exclude the tripeptide from the input window because the sequences have already been filtered for PTS1 motifs. They further argue that including the tripeptide would cause the prediction to be dominated by the PTS1 motif neglecting the information contained in the adjacent 9-mer [3]. However, if there are dependencies in the structure of the motif and the preceding residues, then excluding the motif from the classifier – as is done in PeroxiP – may be hiding crucial information. Some co-dependencies between the physical characteristics of residues appearing inside and outside the PTS1 motif were recently put forward [6]. For instance, Neuberger *et al.* note that “Variations of the hydrophobicity level at position -1 can be compensated by positions -3 and

-2” (*ibid.* p. 571) and the -3 residue is outside the tripeptide motif. Amery *et al.*, on the motif ASL in protein Q9UHK6 state that it is “active [as a PTS1] only when preceded by a lysine residue (or likely a positively charged amino acid)” [1] (p. 1758).



**Fig. 1.** A logo of the C-terminal 12-mer aligned with inter-residue dependencies.

In order to obtain a deeper insight into the potential dependencies between the tripeptide and the 9-mer, we performed a probabilistic analysis of the maximal dependencies between locations within these last 12 residues.

The dependencies were extracted in the following manner: Given a set of aligned sequences  $S = \{s_k | k \in 1..n\}$ , where a sequence  $s_k$  is defined as a tuple  $(r_{k1}, \dots, r_{ki}, \dots, r_{km})$  of residues, the dependencies  $D_{ij}$  between two positions  $i$  and  $j$  over all sequences in  $S$  are calculated as the maximum difference between the joint probability  $P(r_{ki}, r_{kj})$  and the product of the independent probabilities  $P(r_{ki})$  and  $P(r_{kj})$ :

$$D_{ij} = \max_{k \in 1..n} (P(r_{ki}, r_{kj}) - P(r_{ki})P(r_{kj})) \quad (1)$$

If two positions are statistically independent (or perfectly conserved)  $D_{ij}$  equals zero. Figure 1 displays the logo and  $\log(D_{ij})$ <sup>5</sup> for the last 12 residues of the positive set.

<sup>5</sup> A logarithmic scale was chosen to magnify smaller dependencies.

The marked area reveals dependencies between residues in the c-terminal tripeptide and the adjacent 9-mer. Any predictor for peroxisomes which analyses the tripeptide and the 9-mer independently neglect these dependencies and is therefore expected to perform worse than a classifier which takes all 12 residues into account.

In a final set of simulations we investigated the optimal window size for the classifier, both including and excluding the tripeptide. In all simulations excluding the tripeptide the classifier performed worse than those in which it was included, these results are summarised in Table 3.

Window size	Tripeptide	
	Included	Excluded
9	0.48 (0.03)	0.47 (0.04)
12	0.60 (0.03)	0.50 (0.04)

**Table 3.** MCC (std) of the SVM with two different window sizes both including and excluding the C-terminal tripeptide. 5-fold cross validation over 50 runs.

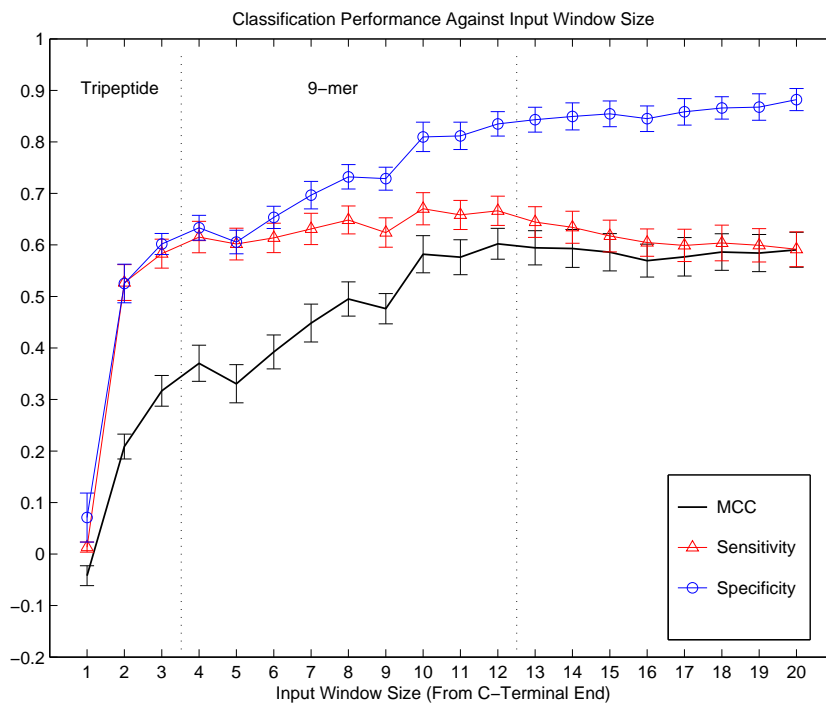
For the classifiers that included the tripeptide we ran a number of simulations for which the size of the window was varied between 1 and 20 residues. The accuracy of the models with different window sizes is shown in Fig. 2. In general, the classification accuracy increases up to a window of size 12, at which point the performance stabilises.

### 3.3 Final model

The overall structure of the model employs similar filtering steps as PeroxiP. Instead of TargetP, we use Protein Prowler – a subcellular localisation predictor with slightly better accuracy [2] – to disqualify sequences that are likely to be secreted. A motif filter rejects sequences with a C-terminal tripeptide not occurring amongst peroxisomal proteins in SWISS-PROT release 45, and an SVM analyses the sequence and classifies the protein as PTS1 targeted or not.

A scan by the Protein Prowler of the positive and negative sets (before redundancy reduction) showed only one protein, with a moderately high prediction as being secreted (0.82). The negative set, on the other hand, contained 100 proteins with scores above 0.82. 0.95 was chosen as a suitably high cutoff, including 74 proteins in the negative set.

Because no peroxisomal proteins are predicted to be secreted, the sensitivity did not decline by including this filter, however the filter increased specificity by an average of 0.034 (an increase of 3.9%) and the MCC by an average of 0.025 (an increase of 4%). Results were generated using 100 runs of five-fold cross-validation.



**Fig. 2.** The prediction accuracy of SVM(P2) for a range of different input window sizes relative to the C-terminus.

Data set	Dataset performance comparison		
	MCC	Sensitivity	Specificity
2005	0.66 (0.03)	0.64 (0.03)	0.91 (0.02)
2003	0.51 (0.04)	0.52 (0.03)	0.82 (0.04)

**Table 4.** Performance measures and standard deviation of the final model (including filtering) 5-fold cross validation over 100 runs.

### 3.4 Results

The performance statistics for our full peroxisomal localisation predictor are shown in Table 4. When our model was trained on the replicated version of the PeroxiP data set it provided an overall comparable performance to PeroxiP. Emanuelsson *et al.* published values of 0.50, 0.78, and 0.64 for MCC, sensitivity and specificity respectively. Our model gives a comparable MCC, but has significantly worse sensitivity and much greater specificity than PeroxiP.

The performance increased significantly when our model was trained on the new data set, an increase of 29% in MCC. Interestingly, the reported sensitivity of PeroxiP still greatly exceeded that achieved by our final model. This suggests that a reasonable strategy for combined peroxisomal prediction would be to give greater weight to our positive predictions, and greater weight to PeroxiP's negative predictions.

## 4 Conclusion

In this paper we have outlined the process by which we developed a new PTS1 peroxisomal localisation classifier. The system uses a design similar to that of the PeroxiP classifier, first filtering for other signaling peptides and then for known PTS1 motifs, finally using a classifier to predict peroxisomal proteins. Unlike PeroxiP, our model consists of a single SVM that processes a window of 12 residues from the C-terminal. In addition we have achieved three outcomes with this research.

Firstly, the development of a updated and highly curated data set for peroxisomal localisation via the PTS1 motif. The quality of this data set improved the accuracy of our final prediction system, measured as a 29% increase in MCC.

Secondly we argued and demonstrated that a PTS1 classifier should include the terminal tripeptide within the input window, even when input is prefiltered for known PTS1 motifs. The argument relied on existing analysis of the dependencies between the tripeptide and the 9-mer, as well as simulation based results showing that its inclusion improved overall prediction accuracy.

Finally, through a series of benchmarking studies we have established that a Support Vector Machine with a polynomial kernel of order two produces the best performance of all individual classifiers tested on the new data set. When trained on our replicated version of the PeroxiP data set, our predictor provided comparable performance to the PeroxiP model but with a simpler structure.

## Acknowledgment

This work was supported by the Australian Research Council Centre for Complex Systems.

## References

1. L. Amery, M. Fransen, K. De Nys, G. P. Mannaerts, and P. P. Van Veldhoven. Mitochondrial and peroxisomal targeting of 2-methylacyl-coa racemase in humans. *J. Lipid Res.*, 41(11):1752–1759, 2000.
2. M. Bodén and J. Hawkins. Prediction of subcellular localisation using sequence-biased recurrent networks. *Bioinformatics*, 2005. Accepted pending minor revision.
3. O. Emanuelsson, A. Elofsson, G. von Heijne, and S. Cristobal. In silico prediction of the peroxisomal proteome in fungi, plants and animals. *Journal of Molecular Biology*, 330(2):443–456, 2003.
4. J. M. Jones, J. C. Morrell, and S. J. Gould. Multiple distinct targeting signals in integral peroxisomal membrane proteins. *Journal of Cell Biology*, 153(6):1141–1150, 2001.
5. G. Lametschwandtner, C. Brocard, M. Fransen, P. Van Veldhoven, J. Berger, and A. Hartig. The difference in recognition of terminal tripeptides as peroxisomal targeting signal 1 between yeast and human is due to different affinities of their receptor pex5p to the cognate signal and to residues adjacent to it. *Journal of Biological Chemistry*, 273(50):33635–33643, 1998.
6. G. Neuberger, S. Maurer-Stroh, B. Eisenhaber, A. Hartig, and F. Eisenhaber. Motif refinement of the peroxisomal targeting signal 1 and evaluation of taxon-specific differences. *Journal of Molecular Biology*, 328(3):567–579, 2003.
7. I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, San Francisco, 2000.