

# Predicting Peroxisomal Proteins

John Hawkins & Mikael Bodén

School of Information Technology and Electrical Engineering

The University of Queensland

QLD 4072, Australia

Email: jhawkins@itee.uq.edu.au, mikael@itee.uq.edu.au

**Abstract**—PTS1 proteins are peroxisomal matrix proteins that have a well conserved targeting motif at the C-terminal end. However, this motif is present in many non peroxisomal proteins as well, thus predicting peroxisomal proteins involves differentiating fake PTS1 signals from actual ones. In this paper we report on the development of an SVM classifier with a separately trained logistic output function. The model uses an input window containing 12 consecutive residues at the C-terminus and the amino acid composition of the full sequence. The final model gives a Matthews Correlation Coefficient of 0.77, representing an increase of 54% compared with the well-known PeroxiP predictor. We test the model by applying it to several proteomes of eukaryotes for which there is no evidence of a peroxisome, producing a false positive rate of 0.088%.

## I. INTRODUCTION

The localization of proteins to the peroxisomal matrix is known to be generally dependent upon the presence of one of two dominant motifs. The vast majority of peroxisomal matrix proteins rely on a motif on the C-terminus called the PTS1 signal, often described as SKL with variation.<sup>1</sup> The PTS1 has also been described using a number of different PROSITE patterns, each allowing more or less flexibility in the allowed substitutions. Even though these motifs are highly conserved in peroxisomal proteins, they are present in a many non-peroxisomal proteins. It is known that there are other elements of the sequence that are involved. Investigation of the PTS1 transport mechanism and statistical analysis has indicated that the last twelve C-terminal residues of the protein sequence are the most significant localization determinant of PTS1 proteins [1], [2].

The most accurate peroxisomal protein prediction service in operation is PeroxiP by Emanuelsson, Elofsson, Heijne and Cristóbal [3]. PeroxiP has three stages in its processing of candidate sequences. An initial pre-processing step makes use of existing localization predictors to eliminate sequences that are predicted to be bound for other organelles or predicted to have a membrane spanning region.<sup>2</sup> Secondly, a motif identification module examines the C terminus of the sequence and filters out those proteins that do not fit the pattern of approved PTS1 motifs. In the final step a machine learning model examines the 9-mer of amino acids preceding the PTS1

motif and provides a prediction of whether the sequence is peroxisomal or not. In the original PeroxiP model the machine learning module consists of a neural network and a support vector machine (SVM) that are trained independently and operate as a miniature ensemble. The final output is the union of their individual outputs, i.e. the sequence is predicted as peroxisomal if either model indicates so [3].

In a previous set of simulations we demonstrated that the prediction accuracy could be improved by including the PTS1 motif in the window of residues supplied to the machine learning module. The model was making use of inter-dependencies between the PTS1 motif and the 9-mer [7]. In the current study we perform a detailed exploration of different SVM architectures for the prediction task, employing jack knife testing on the same data set. The final model utilizes the final 12 C-terminal residues and the amino acid composition of the entire protein. The best performing classifier utilized a fifth order polynomial kernel with a separately trained logistic output function, and providing an increase in prediction accuracy compared with PeroxiP of 54% (56% using an output threshold of 0.44). The PTS1Prowler prediction service is now integrated into the Protein Prowler predictor suite available at <http://pprowler.imb.uq.edu.au>

## II. DATASETS

We use two datasets for evaluating our predictor. Firstly, a replicated version of 2003 PeroxiP dataset, created by performing redundancy reduction on the dataset published on the internet by Emanuelsson *et al.*. Secondly a new 2005 dataset created using the procedures for extraction, curation and redundancy reduction outlined by Emanuelsson *et al.* on SWISS-PROT release 45. The datasets and the details of the development of these datasets are available via the predictor web site.

The 2005 dataset is used for the majority of simulations to establish the kernel and parameter settings, the 2003 dataset is used in a final training run to compare the machine learning architecture chosen against the the original PeroxiP.

## III. SIMULATIONS

In our initial study of this problem we found that under five fold cross validation the best performance was achieved with a support vector machine with a second-order polynomial kernel [7]. However, due to the large number of algorithms tested, these benchmarks were performed using only standard

<sup>1</sup>A much smaller number of peroxisomal proteins rely on an N-terminal bipartite signal, called the PTS2 motif, which we do not attempt to predict in this study.

<sup>2</sup>The localization of peroxisomal membrane proteins is governed by a separate set of signals [4], often consisting partly of a membrane spanning region e.g. [5], [6].

values for all SVM parameters (for example excluding low-order terms for the polynomial kernels). In order to develop the best possible predictor for the task, in this study we perform a more thorough search of the parameter space. We also chose to perform our evaluations using the more accurate jack knife approach, in which we train as many models as there are samples in the dataset. Each model is trained on all but one sample, and then each is tested on only that sample excluded from its training set.

The support vector machine uses a kernel function which projects input samples to a feature space. The power of the SVM stems partly from its natural ability to generalize by maximizing a margin of separation between classes in the feature space. The choice of kernel function is a major issue and needs to be carefully investigated. Kernel functions often utilize parameters that also need to be considered. Finally, there are parameters governing the selection of so-called support vectors that affect the size of the separating margin.

It has been shown that essential sequence dependencies occur within the last twelve residues of PTS1 targeted proteins, in particular between the PTS1 motif and the preceding 9 residues. [1] A fact which previous studies into classification has corroborated [7]. However, when the translocation machinery of the cell interacts with an inchoate protein it has access to general information about the overall structure of the protein from the physical interactions that occur between areas flanking the active sites. The holistic information about a protein used by the cellular processes can be very difficult to capture in a classifier, indeed attempts to use complicated encoding structures often succeed only in confusing the task by flooding the machine with superfluous information.

The PeroxiP predictor was reported to have employed an amino acid composition window, as well as the 9-mer preceding the PTS1 motif, as input to the machine learning module[3]. The amino acid composition encoding is vector is formed by summing the number of occurrences of each amino acid in the entire protein, and then producing a probability vector with each element giving the likelihood that a random position in the sequence would contain the corresponding amino acid. The amino acid composition has been shown to be at least correlated crudely with localization [8], and had been used as an encoding for machine learning prediction of localization to various subcellular locations (including the peroxisome), with moderate success [9]. However, previous studies have made no attempt to distinguish how much information (if any) a classifier based on primary sequence can glean from this additional encoding.

In the first set of simulations conducted in this study we report extensively on the performance of models trained with and without the amino acid composition window in order to shed some light on the proportion of the targeting information that comes from the global properties of the protein, as opposed to merely the targeting peptide. The amino acid composition was made available as an option to all classifiers, presented via a window separated from the primary sequence information. Following previous results we use the orthonormal encoding

TABLE I

AMINO ACID COMPOSITION STUDY - AVERAGE MCC		
<sup>3</sup> SVM Kernel	Without AAC	Including AAC
Gaussian ( $\gamma = 0.01$ )	0.0983	0.0808
Gaussian ( $\gamma = 0.1$ )	0.4805	0.5688
Gaussian ( $\gamma = 0.2$ )	0.4279	0.4746
Gaussian ( $\gamma = 0.3$ )	0.2751	0.2751
Linear	0.3720	0.4245
Poly 2	0.6738	0.6927
Poly 2 +lo	0.6658	0.6925
Poly 3	0.6322	0.7044
Poly 3 +lo	0.6473	0.6955
Poly 4	0.5259	0.6666
Poly 4 +lo	0.5412	0.6834
Poly 5	0.4212	0.5626
Poly 5 +lo	0.4775	0.5739
Increase due to AAC		0.0659 (0.0536)

for the primary sequence window [3], [7]. The optimization of the support vector machines is implemented using the Weka library of machine learning tools [10].

All potential kernels are evaluated using the Matthews' correlation coefficient [11] (also known as the Pearson Coefficient). The MCC is a performance statistic that takes into account the numbers of true positives ( $tp$ ), true negatives ( $tn$ ), false positives ( $fp$ ) and false negatives ( $fn$ ). It is calculated as follows:

$$\frac{tp \cdot tn - fp \cdot fn}{\sqrt{(tp + fn)(tp + fp)(tn + fp)(tn + fn)}} \quad (1)$$

By default a support vector machine will produce a binary classification on a two class problem. For many reasons it is desirable that a classifier service produces a posterior probability score for a given input. This can be achieved with SVMs by fitting a logistic model to the output of the trained classifier [12]. For this model the output probability of the target class is given by:

$$P(y = 1|f(x)) = \frac{1}{1 + e^{Af(x)+B}} \quad (2)$$

Where  $f(x)$  is the output of the SVM for sample  $x$ , and the parameters  $A$  and  $B$  must be estimated. Following Platt's recommendation and examples, we used an internal cross validation within the training set folded into three to estimate the parameters for the logistic function [12]. This means that two thirds of the training set are used for training the SVM and the final third is used for estimating the parameters of the logistic function via the maximum likelihood method.

In the second set of simulations we fit logistic output models on top of SVMs with the same set of kernels used in the first run. The optimization procedures for adapting the SVMs and the logistic output models are drawn from the Weka implementation. These studies are conducted as jack knife simulations, however, due to the use of an internal cross validation for training the SVM and logistic model separately, we conducted five runs using different seeds for the internal spitting of training data. Results are reported using the MCC as before, but include standard deviations across the five runs.

TABLE II  
LOGISTIC OUTPUT STUDY - AVERAGE MCC (STD)

<sup>a</sup> SVM Kernel	Without AAC	Including AAC
Gaussian ( $\gamma = 0.01$ )	0.4565 (0.0412)	0.4990 (0.0178)
Gaussian ( $\gamma = 0.1$ )	0.5978 (0.0188)	0.6516 (0.0101)
Gaussian ( $\gamma = 0.2$ )	0.6116 (0.0258)	0.6566 (0.0267)
Gaussian ( $\gamma = 0.3$ )	0.5799 (0.0561)	0.5629 (0.0608)
Linear	0.3842 (0.0145)	0.4301 (0.0173)
Poly 2	0.6874 (0.0149)	0.7064 (0.0128)
Poly 2 +lo	0.6792 (0.0160)	0.7009 (0.0123)
Poly 3	0.7124 (0.0261)	0.7278 (0.0159)
Poly 3 +lo	0.7115 (0.0187)	0.7344 (0.0082)
Poly 4	0.7109 (0.0081)	0.7389 (0.0183)
Poly 4 +lo	0.7125 (0.0107)	0.7294 (0.0191)
Poly 5	0.7024 (0.0113)	0.7410 (0.0128)
Poly 5 +lo	0.7142 (0.0096)	0.7397 (0.0189)
Increase due to AAC		0.0276 (0.0184)

<sup>a</sup>The average Matthews Correlation Coefficient (standard deviation) for each of the models trained to produce a probability distribution by fitting a logistic model to the SVM output. Simulations are performed for all kernels with and without the amino acid composition window (+lo indicates the inclusion of lower order terms in polynomial kernels). Results produced from a jack knife test trained on the 2005 dataset, the logistic model parameters are estimated using an internal 3-fold cross validation. Final row contains the average increase in MCC due to the inclusion of the amino acid composition window.

#### IV. RESULTS

The results for the first round of simulations are shown in Table I. It can be seen that the amino acid composition window improves the performance for all bar one kernel (Gaussian  $\gamma = 0.01$ ). The best performance is obtained with a third order polynomial kernel, however the difference between the second and third order kernels is marginal.

The results for the second round of simulations are shown in Table II. Two points are made very clear by these results. Firstly, the use of a logistic output improves the accuracy of the models. The capacity for logistic outputs to improve performance was noted by Platt, but is deemed to be difficult to predict in advance [12]. Secondly, the amino acid composition window almost invariably improves the performance of a classifier trained with a logistic output. The best performing model, judged by highest average, is the fifth order polynomial kernel without lower order terms. However, there is a reasonable overlap of the distribution of MCCs for the best performing models. In general, a fourth or fifth order polynomial kernel with or without lower order terms is performing close to the best. We can take these results to be relatively informative due to the extensive nature of the jack knife test procedure.

In order to refine the final prediction model we perform an exploration of the various free parameters controlling the training of SVMs: the complexity constant  $C$ , the round-off error  $\epsilon$  and the tolerance parameter  $T$  [13]. Due to computation time constraints it was assumed that optimal values for these parameters could be established independently of one another.

It was found that altering  $C$  had very little affect on the results. If the value of  $C$  dropped below 0.0007 the results

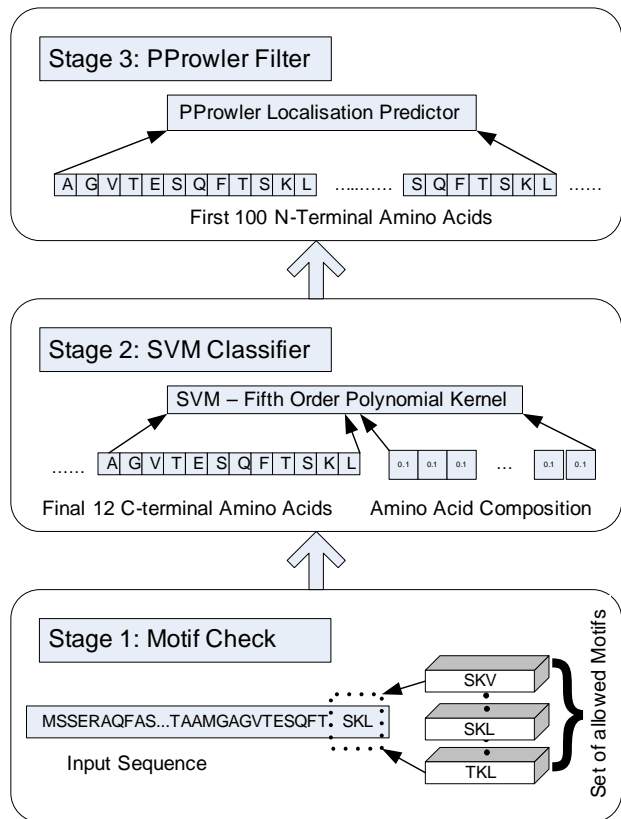


Fig. 1. Overview of the three stages of the final PTS1Prowler model. The first stage filters out sequences without a PTS1 motif. The second stage runs the SVM classifier over the final 12 residues with an orthonormal encoding and an amino acid composition window. If this stage produces a positive prediction, then the third stage is executed, by which the PProWler application examines the sequence for a potential Signal Peptide. If the PProWler application gives a score greater than 0.85 then the prediction is changed to non-peroxisomal, otherwise the output from stage two is given.

rapidly degraded, however for all values of  $C$  greater than this, the results were identical (values up to the maximum double value of  $1.8 \times 10^{308}$  were tried). Similarly alteration of the  $\epsilon$  parameter had no effect in most instances, and deleterious effects for extreme changes. Minor improvements were achieved by tuning the tolerance parameters so that the average MCC increased from 0.741 to 0.749, with a final value of  $T = 0.038$ . The insensitivity to the setting of regularization parameters indicate that the separation in the feature space defined by the polynomial kernel is simple.

#### V. FINAL MODEL

The overall structure of the model is similar to that of PeroxiP, as shown in Figure 1. An application is used to filter proteins 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 gives a classification as PTS1 targeted or not.

The filtering of secreted proteins is performed by the Protein Prowler subcellular localization model [14]. However, in our

model this step is performed last rather than first. Its use is deferred as it is more computationally expensive than either the motif filter or the SVM driven classification. The reordering has no effect on the classification, only the efficiency of the model.

Running PProWler over the training sets revealed only one protein P24552 (D-amino-acid oxidase) with a significant prediction of containing a signal peptide (0.832). A threshold value of 0.85 was chosen as the cutoff for the PProWler filter stage, yielding no predictions from the positive set and 49 predictions from the negative training set.

For the sake of a thorough comparison with the PeroxiP predictor we include two extra performance statistics for evaluating the final model. The *sensitivity* is a measure of the classifiers predilection for always identifying positive examples, calculated thus:

$$\frac{tp}{tp + fn} \quad (3)$$

and the *specificity* of the classifier is a measure of the classifier's tendency to avoid making false positive predictions, calculated as follows:

$$\frac{tp}{tp + fp} \quad (4)$$

These values, like the MCC, are averaged over five runs of the simulation and reported with standard deviations.

## VI. RESULTS

The final performance statistics for PTS1Prowler (our full Peroxisomal/PTS1 Localization Predictor now part of our Protein Prowler suite) are shown in Table III. The final estimate of the Mathews Correlation Coefficient for the model is 0.77, a 54% improvement over PeroxiP.

Because no peroxisomal proteins are predicted as secreted by the PProWler filter, the sensitivity of the model does not decline by including the filter. The PProWler filter increased specificity by an average of 0.020 (an increase of 2%) and the MCC by an average of 0.017 (an increase of 2%). Decreasing the threshold could further increase the gain in accuracy, but may result in reduced sensitivity when exposed to a larger dataset.

Emanuelsson *et al.* published values of 0.50, 0.78, and 0.64 for MCC, sensitivity and specificity respectively, values were obtained from a single run of five-fold cross-validation. When trained on the same dataset our model yielded an average MCC of 0.76 which is a 52% improvement on PeroxiP and only 1% less than the value achieved by training on the 2005 dataset. It is interesting to note that the sensitivity of our final model is identical to that of the original PeroxiP, all of the improvement in performance has come from an increase of 45% in the specificity of the model.

A ROC curve analysis was performed using an average of the outputs of the five classifiers produced from the five runs of the simulation (See Figure 2). The ROC curve revealed that the optimal threshold for accepting a positive peroxisomal localization prediction occurs for thresholds between 0.439

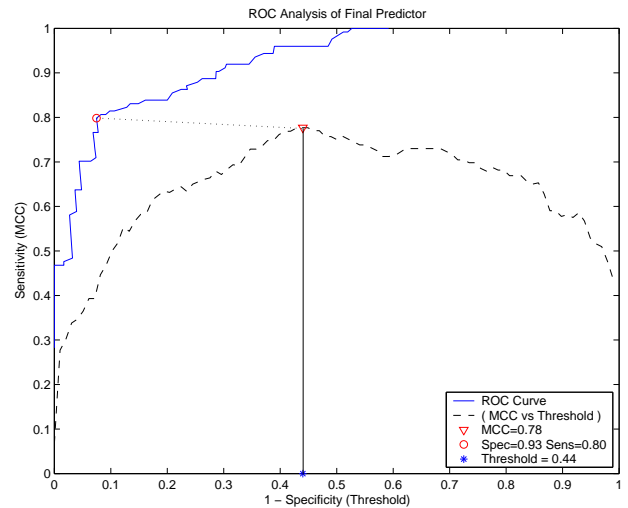


Fig. 2. ROC curve for the final SVM classifier (Polynomial kernel degree five, without lower order terms), trained with a logistic output function and using the PProWler filter. Dashed graph gives average MCC across the five runs plotted against the threshold values. Solid line depicts ROC analysis Sensitivity plotted against 1-Specificity. The points marked on both graphs correspond to a threshold value of 0.44, for which the optimum MCC of 0.78 occurred.

and 0.456 yielding an MCC of 0.78. In the name of making a conservative estimate we do not use this as the final estimate of MCC, but we recommend the threshold value for users of PTS1Prowler.

## VII. ORGANISMS WITHOUT PEROXISOMES

Although peroxisomes are present in most eukaryotic cells, there are numerous species of parasitic eukaryotes which are believed not to possess a peroxisome. For protozoa of the genus *Giardia*, *Trichomonas* (Agent of Vaginitis), or amoebas of the genus *Entamoeba* (agents for dysentery and ulceration of the colon and liver) there is no evidence for the presence of a peroxisome [15]. Thus, no reason to expect their proteomes to contain peroxisomally targeted proteins.

The proteomes of these organisms offer an opportunity to test the integrity of a peroxisomal targeting classifier. By feeding the set of known proteins from each genus through the classifiers we report how many proteins are fallaciously labeled peroxisomal, thus giving an independent estimate of the specificity of the models. The datasets for this study were extracted using the online Swiss-Prot(47.5)/TrEMBL(30.5) Advanced Search, retrieving all proteins within each taxonomy, excluding protein fragments. Entries from Swiss-Prot and TrEMBL are kept separate for the sake of distinguishing known gene products from putative ones. We conduct the a benchmark study of PTS1Prowler vs PeroxiP to provide a comparison of specificity between the two classifiers. For the PeroxiP predictions we accepted only Method 1 predictions, in which the most restrictive PTS1 criteria is used, corresponding to the 32 PTS1 motifs found in the 2003 training set. For PTS1Prowler we ignore the results of the ROC study and use a cutoff of 0.5 for the sake of fairness. The two versions of

TABLE III  
FINAL MODEL PERFORMANCE COMPARISON

<sup>a</sup> Prediction Service	Matthews Correlation Coefficient	Sensitivity	Specificity
PeroxiP (2003 Dataset)	0.50	0.78	0.64
PTS1Prowler (2003 Dataset)	0.760 (0.016)	0.771 (0.022)	0.930 (0.003)
PTS1Prowler (2005 Dataset)	0.766 (0.018)	0.777 (0.023)	0.931 (0.003)

<sup>a</sup>Performance measures of the final model, (including Prowler filter) as depicted in Figure 1. The SVM Classifier has a fifth order polynomial kernel without lower order terms, with tolerance parameter  $T = 0.038$ ,  $C = 1.0$  and  $\epsilon = 1.0 \times 10^{-12}$  and a logistic output function. The input for the SVM is an orthonormal encoding of the final 12 C-terminal residues and an amino acid composition window for the entire protein. Results are generated from a jack knife simulation run five times using different seeds for the internal cross validation used to separate the training of the SVM and the fitting of the logistic function parameters.

PTS1Prowler, on for each datasets, are employed in order to increase the quality of the comparison between the underlying models.

The results displayed in Table IV reinforce the findings of the preceding simulations. PeroxiP has a false positive rate of approx 1.88% whereas PTS1Prowler (trained on either dataset) has a false positive rate of 0.088%, over an order of magnitude smaller. PTS1Prowler appears to operate with a far superior specificity than PeroxiP.

### VIII. CONCLUSION

We have taken the overall three stage classifier design of Emanuelsson *et al.* and focused on improving the machine learning stage of the process. The final model is a finely tuned support vector machine with a polynomial kernel of order five. Unlike Emanuelsson we include the PTS1 tripeptide in the input window and additionally we train the model to produce a probability vector by fitting a logistic function to the output of the SVM. The combination of all of these modifications resulted in a predictor with an average MCC of 0.77, an improvement of 54% on the results reported for the original PeroxiP predictor.

By training PTS1Prowler on both the original 2003 PeroxiP dataset and the 2005 dataset, it is possible to obtain an indication of how much improvement is gained by the model alone. The final performance results in Table III indicate that the vast majority of the improvement in performance is due to the superiority of the new model. Only an extra 1% increase in MCC can be attributed to the use of the new dataset. An independent analysis using all available protein sequences for several genus of eukaryotes without peroxisomes reveals that the false positive rate for PTS1Prowler is over an order of magnitude smaller than PeroxiP (Table IV).

In our exploration of the effects of the amino acid composition window we have observed that it almost invariably improves the performance of an SVM trained on PTS1 recognition. The improvement for SVMs augmented with a logistic output function provided is quite reliable, only a single kernel was affected deleteriously. Inclusion of the amino acid composition window increases the performance of the classifiers by an average value of 0.03 points of MCC.

ROC curve analysis reveals that the optimum cutoff value for accepting a protein as peroxisomal occurs at 0.44. Using

this value, and by averaging the outputs of the five models trained with different seeds, for the splitting of data between training the SVM and the logistic output, the estimated MCC increased a further 1% to a value of 0.78. This results in a final increase in the estimated MCC of the classifier of 56% when compared to the original PeroxiP.

### AVAILABILITY

The PTS1Prowler peroxisomal prediction service has been integrated with the Prowler protein localization predictor available at: <http://pprowler.imb.uq.edu.au>

### ACKNOWLEDGMENTS

The authors wish to acknowledge the technical assistance of Mark Wakabayashi and Stefan Maetschke. This work was in part supported by the Australian Research Council Centre for Complex Systems.

### REFERENCES

- [1] 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*, vol. 328, no. 3, pp. 567–579, 2003.
- [2] 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*, vol. 273, no. 50, pp. 33 635–33 643, 1998.
- [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*, vol. 330, no. 2, pp. 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*, vol. 153, no. 6, pp. 1141–1150, 2001.
- [5] M. Biermanns, J. von Laar, U. Brosius, and J. Gaertner, "The peroxisomal membrane targeting elements of human peroxin 2 (pex2)," *European Journal of Cell Biology*, vol. 82, no. 4, pp. 155–162, 2003.
- [6] U. Brosius, T. Dehmel, and J. Gaertner, "Two different targeting signals direct human peroxisomal membrane protein 22 to peroxisomes," *Journal of Cell Biology*, vol. 277, no. 1, pp. 774–784, 2002.
- [7] M. Wakabayashi, J. Hawkins, S. Maetschke, and M. Bodén, "Exploiting sequence dependencies in the prediction of peroxisomal proteins," in *Intelligent Data Engineering and Automated Learning*, vol. LNCS 3578. Springer, 2005, pp. 454–461.
- [8] K. Nishikawa, Y. Kubota, and T. Ooi, "Classification of proteins into groups based on amino acid composition and other characters. i. angular distribution," *Journal Of Biochemistry*, vol. 94, no. 3, pp. 981–995, 1983.
- [9] A. Reinhardt and T. Hubbard, "Using neural networks for prediction of the subcellular location of proteins," *Nucl. Acids Res.*, vol. 26, no. 9, pp. 2230–2236, 1998.

TABLE IV  
NON-PEROXISOMAL PROTEOME COMPARISONS

<sup>a</sup> Genus	Source	Number of Proteins	PeroxiP (Method 1)	PTS1Prowler (2003 Dataset)	PTS1Prowler (2005 Dataset)
Trichomonas	SProt	15	0	0	0
	TrEMBL	104	3	0	0
Giardia	SProt	35	2	0	0
	TrEMBL	6690	127	3	4
Entamoeba	SProt	55	0	0	0
	TrEMBL	9081	168	11	10

<sup>a</sup>PeroxiP and PTS1Prowler applied to the all available proteins from the three genus of eukaryotes for which there is no evidence of peroxisomes. The number of available proteins is displayed in the first column, followed by the number of proteins predicted to be peroxisomal by PeroxiP and PTS1Prowler. The results acts as an independent confirmation of the specificity of the predictors.

- [10] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, San Francisco, 2000.
- [11] B. W. Matthews, "Comparison of predicted and observed secondary structure of t4 phage lysozyme," *Biochim Biophys Acta*, vol. 405, pp. 442–451, 1975.
- [12] J. Platt, "Probabilistic outputs for support vector machines and comparison to regularized likelihood methods," in *Advances in Large Margin Classifiers*, A. Smola, P. Bartlett, B. Schoelkopf, and D. Schuurmans, Eds. MIT Press, 2000, pp. 61–74.
- [13] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in kernel methods: support vector learning*. Cambridge, MA, USA: MIT Press, 1999, pp. 185–208.
- [14] M. Bodén and J. Hawkins, "Prediction of subcellular localisation using sequence-biased recurrent networks," *Bioinformatics*, vol. 21, pp. 2279–2286, 2005.
- [15] M. Parsons, T. Furuya, S. Pal, and P. Kessler, "Biogenesis and function of peroxisomes and glycosomes," *Molecular and Biochemical Parasitology*, vol. 115, no. 1, pp. 19–28, 2001.