A Probabilistic Classification System for Predicting the Cellular Localization Sites of Proteins

Paul Horton
Computer Science Division 387 Soda Hall
University of California, Berkeley, CA 94720
paulh@cs.berkeley.edu

Kenta Nakai
Institute of Molecular and Cellular Biology
Osaka University,
1-3 Yamada-oka, Suita 565, Japan
nakai@imcb.osaka-u.ac.jp

Abstract

We have defined a simple model of classification which combines human provided expert knowledge with probabilistic reasoning. We have developed software to implement this model and have applied it to the problem of classifying proteins into their various cellular localization sites based on their amino acid sequences. Since our system requires no hand tuning to learn training data, we can now evaluate the prediction accuracy of protein localization sites by a more objective cross-validation method than earlier studies using production rule type expert systems. 336 *E.coli* proteins were classified into 8 classes with an accuracy of 81% while 1484 yeast proteins were classified into 10 classes with an accuracy of 55%. Additionally we report empirical results using three different strategies for handling continuously valued variables in our probabilistic reasoning system.

Keywords: Protein Localization, Probabilistic Reasoning, Classification, Yeast, *E.coli*

Introduction

The field of computational biology has mainly been characterized by successes in the efficient organization of biological data into databases and the development of algorithms, e.g. string algorithms, to manipulate that data. This research has been of great utility but it stops short of modeling aspects of biology at the cellular or higher levels. Any system which does that must have a language for knowledge representation as well as a robustness to errors or incomplete knowledge. Expert systems using production rules, which have been applied to biology (Nakai & Kanehisa 1991; 1992), have a rich language for representing knowledge but are not well suited for reasoning under uncertainty. For this reason we believe that probabilistic reasoning systems are very promising as a tool in computational biology. Based on probability theory, probabilistic reasoning is inherently designed to handle uncertainty, and has been shown to have a wide range of applications (Pearl 1992). In this paper we describe a model for classification which can be viewed either as a probabilistic analog to decision trees or as a restricted form of Bayesian network.

The localization site of a protein within a cell is primarily determined by its amino acid sequence. Nakai and Kanehisa have exploited this fact to develop a rule based expert system for classifying proteins into their various cellular localization sites, using their amino acid sequences, in gram-negative bacteria (Nakai & Kanehisa 1991) and in eukaryotic cells (Nakai & Kanehisa 1992). These expert systems can predict the localization sites of protein sequences with good accuracy once appropriate certainty factors have been given to each rule. However there is no direct probabilistic interpretation of their certainty factors and for optimal prediction accuracy they must be hand-tuned for each dataset.

In order to remedy that shortcoming we have constructed a probabilistic reasoning system for the same classification problem. The reasoning system classifies objects on the basis of an input vector of real valued feature variables for each object. The relationship between the classes and the feature variables is provided by a human expert, who must organize the relationships into a tree structure.

Model

We define a simple model for probabilistic classification of objects. The model, consists of a rooted binary tree of "classification variables" and a "feature variable" associated with each non-leaf node of the binary tree, as in figure 1a. The leaves of the tree represent the possible classes into which an object can be classified. A non-leaf node n represents of all the classes which belong to leaves that are descendants of n (in this section we will refer to that set of classes as the class of node n). When performing inference, each node has a probability associated with it, the probability of n being true represents the probability that an object belongs to n's class. Thus since the children of a node represent a mutually exclusive partioning of the parent's class, it follows that the probability that a node is true must equal the sum of the probabilities that its children are true.

For example in figure 1, $Pr[C_1] = Pr[C_{10}] + Pr[C_{11}]$. Each non-leaf node n has a feature variable F_n and a conditional probability table (or function) associated with it. The influence of the features of an object on whether it should be classified as being a left descendant of n versus being classified as a right descendant of n is assumed to be completely summarized by F_n. In our example, this would imply that $Pr[C_{11}|C_1, F_1] = Pr[C_{11}|C_1, F_1, F_{root}]$. This conditional independence allows us to calculate the probability of each node given a set a values for the feature variables, and the appropriate conditional probability tables, with one traversal of the tree. The traversal starts with the root which always has a probability of 1. Although we did not originally conceptualize this model as a family of Bayesian networks, it can be expressed as such. For readers who prefer to think in terms of Bayesian networks, the translation of our example to a Bayesian network is shown in figure 1b.

Conditional Probabilities

If the feature variables are discrete variables then the conditional probabilities, e.g. $Pr[C_{11} = 1|C_1 = 1, F_1 = 1]$ may be estimated by counting the frequency of the examples in the training data for which $F_1 = 1$ & $C_{11} = 1$ and dividing by the frequency of examples for which $F_1 = 1$ & $C_1 = 1$. However, in this application we were generally faced with continuous variables.

We tried three approaches for dealing with continuous variables. The first step of all three methods was to normalize the variable values to a range of $[0, 1]$. In the first two methods we then discretized the values by dividing $[0, 1]$ into intervals and treated each interval as a single value. The intervals were chosen such that a roughly equal number data points (i.e. values of the feature variable in question for the sequences in our training data), fell into each interval. Unfortunately, we were not able to derive a well principled criterion for how many intervals the range $[0, 1]$ should be divided into. Instead we somewhat arbitrarily tried making a number of intervals equal to either the log to the base 2 of the number of relevant examples, or the square root of the number of those examples. Here relevant means that the examples belong to the class of the node whose feature value we are discretizing.

The third method we employed does not discretize the values but learns a conditional probability function in the form of the sigmoid function $G(F_i) = \frac{1}{1+e^{(aF_i+b)}}$. More specifically suppose that we want to learn the conditional probability function $Pr[C_1|F_1]$ from figure 1. Let F_{1i} denote the value of F_1 for the ith example of the training data. We used gradient descent to choose values for a_1 and b_1 which minimize

$$\sum_i (G(F_{1i}) - C_{11})^2$$

where C_{11} equals one when true, i.e. for examples of class C_{11}, and zero otherwise. The summation is over all the examples of class C_1. We subscripted a and b here to indicate that a separate pair of a and b parameters are learned for each mapping of a feature variable to its associated conditional probability function. This sigmoid function does not have a local minimum and therefore gradient descent is sufficient for learning optimal values for a and b. The reader may observe that this procedure is equivalent to using a feed-forward neural network with just one input node and one output node to learn the mapping from feature variable values to conditional probabilities.

The Classification Tree for E.coli Sequences

Proteins from *E.coli* were classified into 8 classes: inner membrane lipoproteins (imL), outer membrane lipoproteins (omL), inner membrane proteins with a cleavable signal sequence (imS), other outer membrane proteins (om), periplasmic proteins (pp), inner membrane proteins with an uncleavable signal sequence (imU), inner membrane proteins without a signal sequence (im), and cytoplasmic proteins (cp). 7 features were calculated from the amino acid sequences for use in classification: A modification of McGeoch's (MeGeoch 1985) signal sequence detection parameter (mcg), the presence or absence of the consensus sequence (von Heijne 1989) for Signal Peptidase II (lip), the output of a weight matrix method for detecting cleavable signal sequences (von Heijne 1986) (gvh), the output of the ALOM program (Klein, Kanehisa, & DeLisi 1985) for identifying membrane spanning regions on the whole sequence (alm1), and on the sequence excluding the region predicted to be a cleavable signal sequence by von Heijne's method (von Heijne 1986) (alm2), the presence of charge on the N-terminus of predicted mature lipoproteins (chg), and the result of discriminant analysis on the amino acid content of outer membrane and periplasmic proteins (aac). The classification tree used is shown in figure 2. These classifications and features are discussed in detail by Nakai and Kanehisa (Nakai & Kanehisa 1991).

The Classification Tree for Yeast Sequences

Proteins from yeast were classified into 10 classes: cytoplasmic, including cytoskeletal (CYT);nuclear (NUC);vacuolar (VAC); mitochondrial (MIT); peroxisomal (POX); extracellular, including those localized to the cell wall (EXC); proteins localized to the lumen of the endoplasmic reticulum (ERL); membrane proteins with a cleaved signal (ME1); membrane proteins with an uncleaved signal (ME2); and membrane proteins with no N-terminal signal (ME3), where ME1, ME2, and ME3 proteins may be localized to the plasma membrane, the endoplasmic reticulum membrane, or

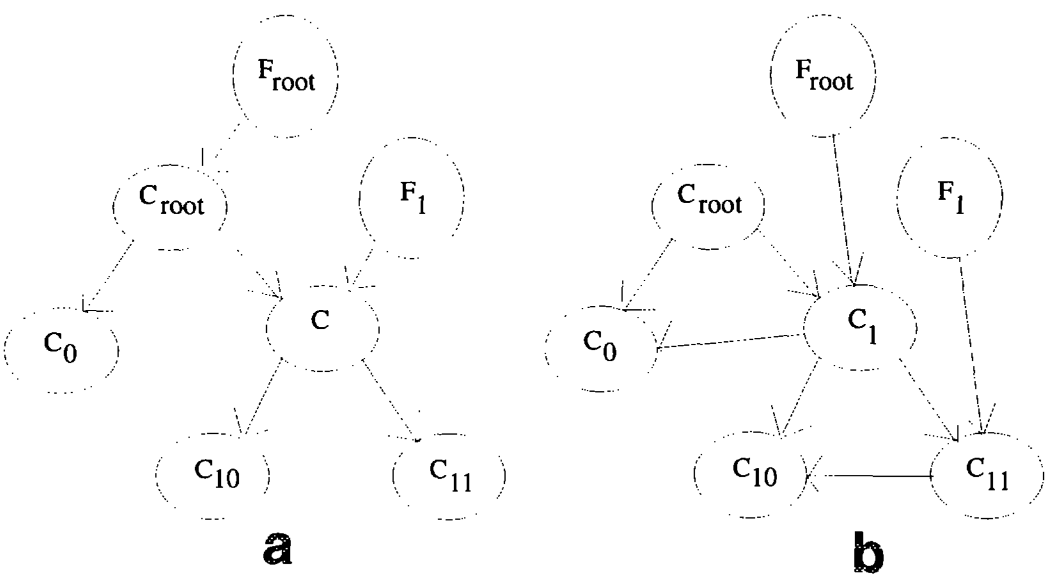

Figure 1: (a.) An example of a classification tree with its feature variables. The dotted edges represent feature variables associated with classification variables. (b.) One possible equivalent Bayesian network; where F_1 and F_{root} are clamped variables.

the membrane of a golgi body. Ideally those three membranes would be classified separately, but in this work we chose to group them together to obtain a more tractable classification task. The classification tree used is shown in figure 3. 8 features were calculated from the amino acid sequences for use in classification: the presence or absence of an HDEL pattern (substring) as a signal for retention in the endoplasmic reticulum lumen (Pelham 1990) (erl); the result of discriminant analysis on the amino acid content of vacuolar and extracellular proteins (vac); the result of discriminant analysis on the amino acid composition of the 20-residue N-terminal region of mitochondrial and non-mitochondrial proteins (mit); the presence or absence of nuclear localization consensus patterns combined with a term reflecting the frequency of basic residues (nuc); and some combination of the presence of a short sequence motif and the result of discriminant analysis of the amino acid composition of the protein sequence (pox). Several versions of the pox variable were tried (see discussion). The feature variables mcg, gvh, and alm described in the E.coli section were also used for classifying yeast proteins. The classes and feature variables are discussed in detail by Nakai and Kanehisa (Nakai & Kanehisa 1992).

Results of *E.coli* Protein Classification Using Sigmoid Conditional Probability Functions

Examples	Class	High	Top2
77	im	77.9%	89.6%
143	cp	96.5%	100%
2	imL	50.0%	50.0%
5	omL	100%	100%
35	imU	71.4%	91.4%
2	imS	0.0%	0.0%
20	om	65.0%	85.0%
52	pp	78.9%	94.2%

Table 1: The accuracy of classification of *E.coli* proteins is displayed for each class when all of the data was used for training. For each class the number of examples in the training data, the percentage of sequences for which the correct class matched the class with the highest computed probability and the percentage which matched one of the classes with the 2 highest computed probabilities is shown.

Software

We have developed a C program to implement the computations necessary to perform the probabilistic inference described in the model section. The program also learns the conditional probability tables (function) using any of the three methods described in the section on conditional probabilities. Alternatively the program can read in conditional probability tables from a file and use them when doing the inference. The program inputs a file which describes the topology and feature variables of the classification tree in a simple language and another file which contains the values of the feature variables for the objects. The output of the program is the probability of each leaf class for each input object. This program is available upon request from paulh@cs.berkeley.edu.

Results of *E.coli* Protein Classification Using 3 Strategies

	sigmoid	log	square root
all data	84.2%	79.8%	82.7%
X-valid	81.1%, 7.7	79.1%, 10.1	80.6%, 7.1

Table 2: The accuracy of classifying *E.coli* proteins by three different strategies for defining and learning conditional probabilities. The cross-validation row gives the average accuracy and its standard deviation for each strategy.

Results of Yeast Protein Classification Using Sigmoid Conditional Probability Functions

Examples	Class	High	Top2
44	ME1	63.6%	81.8%
5	ERL	60.0%	80.0%
30	VAC	10.0%	13.3%
35	EXC	45.7%	60.0%
51	ME2	15.7%	52.9%
244	MIT	47.1%	56.6%
429	NUC	35.7%	90.2%
20	POX	0.0%	0.0%
163	ME3	85.3%	92.0%
463	CYT	74.3%	93.1%

Table 3: The accuracy of classification of yeast proteins is displayed for each class when all of the data was used for training. For each class the percentage of sequences for which the correct class matched the class with the highest computed probability and the percentage which matched one of the classes with the 2 highest computed probabilities is shown.

Results

E.coli

We gathered a dataset of 336 *E.coli* sequences from SWISS-PROT. The results with this dataset using sigmoid conditional probability functions are shown in table 1. Actually the lip and chg feature variables had only two values in the data set used. Our program automatically detects this and treats those variables as discrete variables.

We performed a cross-validation test by randomly partioning the data into 10 equally sized (±1) subsets and training on the remaining data (the reported accuracy is the average over the 10 subsets). The cross-validation set was used to evaluate the generalization ability of the three different strategies for computing conditional probabilities. The results of comparing these three strategies are shown in table 2.

Results of Yeast Protein Classification Using 3 Strategies

	sigmoid	log	square root
all data	54.5%	54.2%	56.5%
X-valid	54.9%, 4.9	53.9%, 4.1	54.3%, 4.4
non-red.	54.4%	55.6%	55.9%
X-valid	54.1%, 4.9	53.9%, 5.0	55.0%, 4.2

Table 4: The accuracy of classifying yeast proteins by three different strategies for defining and learning conditional probabilities. The third line reports the accuracy for training on all of the non-redundant dataset (described in the text), and the fourth line reports the cross-validation accuracy for that dataset. The cross-validation rows show the average accuracy and its standard deviation for each strategy.

Yeast

We gathered a dataset of 1484 yeast sequences from SWISS-PROT using the annotations from YPD. The results of training all of the 1484 yeast sequences in our database, using sigmoid conditional probability functions is shown in table 3. The only two-valued feature variable for the yeast data was the erl variable, which was recognized as such and treated as a discrete variable by the program.

In addition to the 1484 sequence dataset we also created a "non-redundant" dataset of size 1368 by removing sequences until no pair of sequences in the dataset had over 50% residue identity. As in the *E.coli* results we performed a cross-validation test by randomly partitioning the datasets into 10 equally sized subsets. The classification accuracies for both datasets with the three different strategies for computing conditional probabilities described above is shown in table 4.

Discussion

The most common class of protein represents 41% and 32% of the *E.coli* and yeast data respectively. Thus the classification accuracies of 81.1% and 54.9% are dramatically superior than that obtained by simply choosing the most common class. A direct comparison of the classification accuracies of this system and the expert system of Nakai and Kanehisa (Nakai & Kanehisa 1991; 1992) is impossible because the difficulty of tuning the certainty factors of the rules makes cross-validation with their system infeasible. However the classification accuracy of our probabilistic system appears roughly comparable to theirs. The reason for the difficulty of this comparison however, underscores the utility of our method. Given a classification tree, our program computes everything it needs from the training data.

Although we failed to derive a well-principled method

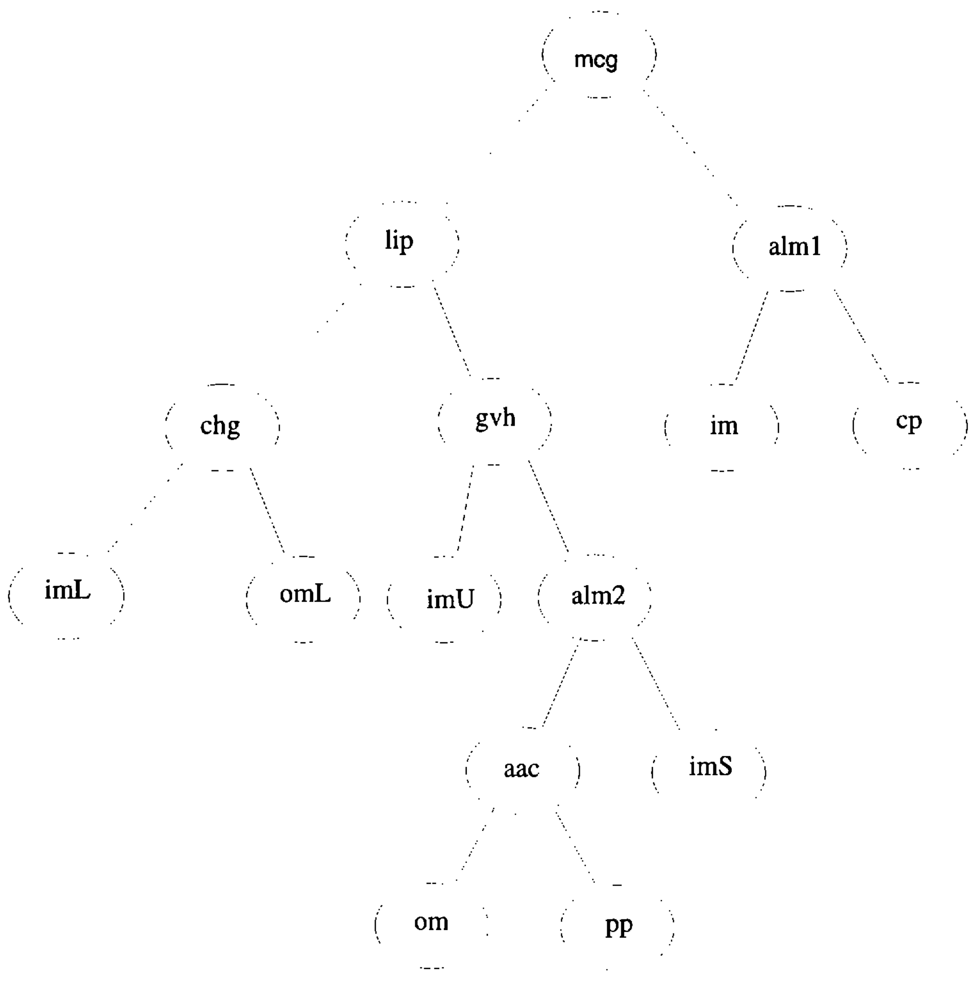

Figure 2: The classification tree used for *E.coli* protein localization. The leaf nodes are labeled with the class that they represent, while the other nodes are labeled with their feature variable. All the edges shown are directed downward and the edges connecting feature variable to their classification variables are not shown explicitly. The labels are abbreviations (see text).

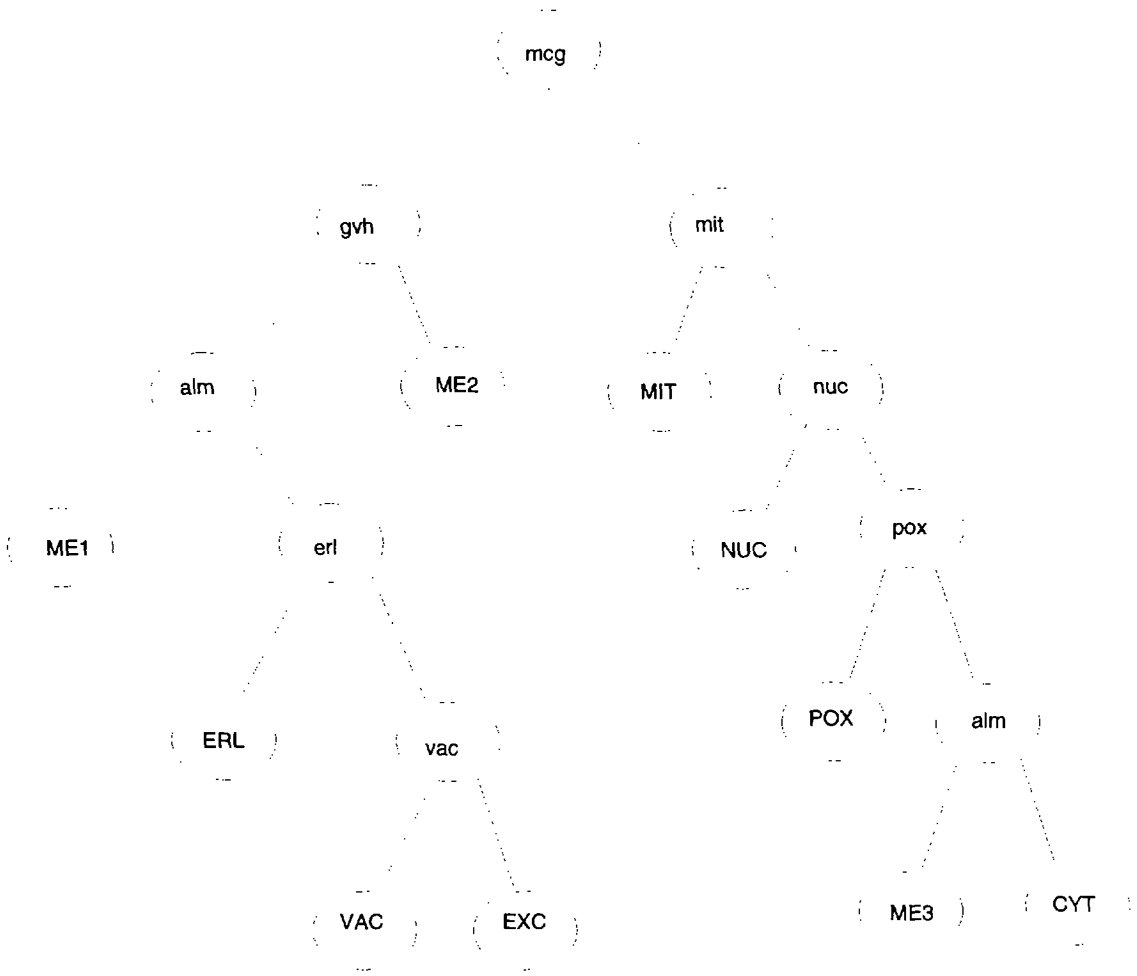

Figure 3: The classification tree used for yeast protein localization. The leaf nodes are labeled with the class that they represent, while the other nodes are labeled with their feature variable. All the edges shown are directed downward and the edges connecting feature variable to their classification variables are not shown explicitly. The labels are abbreviations (see text).

of treating continuously valued variables, we empirically evaluated three strategies. Of these three, the sigmoid conditional probability function appeared to be slightly better at generalizing than the square root number of intervals when discretizing, which in turn appeared to be slightly better than using the log number of intervals. This can be seen in the data in tables 2 and 4. In fact the sigmoid function actually showed a higher accuracy on the cross-validation test for the complete yeast dataset then when trained on all of the data. While this is surprising, it is not a contradiction because the sigmoid functions were fit to minimize the root mean squared error between the function and the data points rather then to directly minimize the number of misclassified sequences. We did not expect the sigmoid function to do as well as it did because it cannot model many distributions, for example bimodal ones. However when one considers that the sequence features used for our classification are fuzzy measures of essentially binary conditions (e.g. either a signal is cleaved or it is not), then one would expect the probability distribution to basically look like a fuzzy step function. The sigmoid function is well suited for use as a fuzzy step function and, apparently, therefore also well suited for this application.

Although we were generally happy with the classification results we were disappointed by the fact that none of the 20 POX (peroxisomal) proteins were predicted correctly. In addition to the pox feature variable used when generating table 3, we also tried two other variations, neither of which enabled the program to correctly predict any of the POX proteins. This failure can partially be explained by noticing the severe under-representation of POX examples in the training data. In the yeast classification tree of figure 3 the POX, ME3, and CYT classes, with 20, 163, and 463 examples respectively, are in the same subtree. If the system chooses to ignore the POX feature variable and always predict that an object that belongs in that subtree is of the class ME3 or CYT rather then POX then the system will be correct $(163+463)/(20+163+463) = 96.9\%$ of the time. Thus, unless the POX feature variable distinguishes the data extremely well the system will do better just to ignore it and never classify any object as POX.

In conclusion we have made a system which implements a simple model for the probabilistic classification of objects. We have successfully applied this model to the problem of classifying protein sequences into their various localization sites.

Acknowledgements

KN was supported by a grant-in-aid for scientific research on the priority area "Genome Informatics" from the Ministry of Education, Science and Culture, Japan. PH was supported by DOE grant DE-FG0390ER60999 and would like to thank Geoff Zweig and Kevin Murphy for their helpful comments.

References

Klein, P.; Kanehisa, M.; and DeLisi, C. 1985. The detection and classification of membrane-spanning proteins. *Biochim. Biophys. Acta* 815:949–951.

MeGeoch, D. J. 1985. On the predictive recognition of signal peptide sequences. *Virus Research* 3:271–286.

Nakai, K., and Kanehisa, M. 1991. Expert system for predicting protein localization sites in gram-negative bacteria. *PROTEINS: Structure, Function, and Genetics* 11:95–110.

Nakai, K., and Kanehisa, M. 1992. A knowledge base for predicting protein localization sites in eukaryotic cells. *Genomics* 14:897–911.

Pearl, J. 1992. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann.

Pelham, H. 1990. The retention signal for soluble proteins of the endoplasmic reticulum. *Trends Biochem. Sci.* 15:482–486.

von Heijne, G. 1986. A new method for predicting signal sequence cleavage sites. *Nucleic Acids Research* 14:4683–4690.

von Heijne, G. 1989. The structure of signal peptides from bacterial lipoproteins. *Protein Engineering* 2:531–534.