Review

Advances in the prediction of protein targeting signals

Gisbert Schneider and Uli Fechner

Johann Wolfgang Goethe-Universität, Institut für Organische Chemie und
Chemische Biologie, Frankfurt, Germany

Enlarged sets of reference data and special machine learning approaches have improved the accuracy of the prediction of protein subcellular localization. Recent approaches report over 95% correct predictions with low fractions of false-positives for secretory proteins. A clear trend is to develop specifically tailored organism- and organelle-specific prediction tools rather than using one general method. Focus of the review is on machine learning systems, highlighting four concepts: the artificial neural feed-forward network, the self-organizing map (SOM), the Hidden-Markov-Model (HMM), and the support vector machine (SVM).

Keywords: Bioinformatics / Machine learning / Review / Signal peptidase / Signal sequence

Received 14/11/03
Revised 9/2/04
Accepted 11/2/04

Contents

1 Introduction

Knowledge of the subcellular localization of a protein is an important piece of information for the target identification process in drug discovery. Secreted proteins and integral plasma membrane proteins are of special interest since they play key roles in important biological processes, *e.g.*, signal transduction and transmission, and cellular differentiation. Moreover, these protein families are comparably easily accessible by drug molecules, due to their localization in the extracellular space or on the cell surface. On the other hand, proteins that are located in special organelles of parasites, *e.g.*, represent candidate targets for the development of novel anti-infective agents [1, 2]. Knowing which proteins are cytosolic and which are targeted to an organelle will help assembling metabolic pathways that putatively occur in the organelle [1]. The prediction of the subcellular localization of proteins has a tradition in bioinformatics, and many such computational tools have been developed over the past two decades, facilitating the identification and even the design of targeting signal features [3–6]. The first prediction methods yielded 70–80% accuracy for secretory proteins [7, 8], current techniques reach up to 95% accuracy with a reduced risk of false-positive predictions [9, 10]. This review highlights some of the more recent additions to the method repertoire, in particular selected machine learning methods, and addresses some conceptual issues.

Protein targeting principles and processes have been described in much detail elsewhere, and the reader is referred to the respective literature [11–14]. Briefly, many thousands of proteins must be transported from their site of synthesis to various cellular compartments involving translocation across at least one membrane [15]. In most

Correspondence: Gisbert Schneider, Beilstein-Professor of Cheminformatics, Johann Wolfgang Goethe-Universität, Institut für Organische Chemie und Chemische Biologie, Marie-Curie-Str. 11, D-60439 Frankfurt, Germany
E-mail: gisbert.schneider@modlab.de
Fax: +49-69-79829826

Abbreviations: ANN, artificial neural network; **COX**, cyclooxygenase; **cTP**, chloroplast transit peptide; **HMM**, hidden Markov model; **mTP**, mitochondrial targeting peptide; **SOM**, self-organizing map; **SP**, signal peptidase; **SVM**, support vector machine; **TP**, transit peptide of the cases, intracellular protein transport and translocation is guided by “targeting signals”, *i.e.*, short stretches of amino acid residues containing information about the target compartment and interaction with appropriate membrane receptors [16]. Having arrived at the place of destination, the targeting signal is generally removed by proteolytic activity of signal peptidases (SPs) [17, 18]. Besides encoding for secretion, transport to a cellular compartment or import into organelles, targeting signals play additional roles. For example, they critically influence the interaction between the ribosome and the translocon [19] and have been shown to affect glycosylation [20] as well as transmembrane domain orientation and integration [21]. The mechanisms of these processes are still unclear for the most part, and additional functions of targeting signals are likely to be discovered in the future [22].

Primary focus has always been on the prediction of secreted proteins, but several tools also exist for prediction of nuclear-encoded organellar proteins. Recently, a large-scale effort has been made by researchers at Genentech to identify novel human secreted proteins relying on a combination of biological and bioinformatical approaches [23]. An outcome of this proteome analysis was the identification of approximately 200 novel secreted protein candidates with hitherto unknown function. Bioinformatical scrutinizing of these sequences is of paramount importance for guiding further experimental work. To be able to do this, the prediction methods need to be very reliable. It has been demonstrated that there exists no cure-all method for prediction of subcellular localization, rather methods should be adapted to the particular organism and subcellular compartment under investigation [1, 24]. Also, it is advised to use several prediction techniques simultaneously whenever possible, since the error rate, *i.e.*, the fraction of false-positive and false-negative predictions, of any individual method can be high, despite methodological advances that have been made during the past years. Results can often be complemented by similarity searching studies, *e.g.*, by heuristic sequence alignment using BLAST or FastA algorithms, with the aim to find annotated protein homologues in databases [25, 26].

2 Concepts in targeting signal prediction

Two basically different approaches for predicting targeting signals have been followed and implemented. First, the “sliding window” technique may be used to perform an analysis of local sequence patterns. This approach is motivated by the fact that continuous stretches of residues may encode for a targeting signal, *e.g.*, a signal peptide carrying the secretion signal, or a transit peptide (TP) for targeting to organelles, *e.g.*, mitochondria (mitochondrial targeting peptide, mTP) and chloroplasts (chloroplast transit peptide, cTP). The idea is to move a window of a defined number or residues along the amino acid sequence, starting at the *N*-terminal end, and calculate a score value for each residue position that was passed by the sliding window. Maximal scores indicate the potential presence of a local targeting signal. This technique is also the most widely used for predicting signal peptidase cleavage sites. Two predictions are made, one for the existence of a targeting signal, and the other for the SP cleavage site. If both types of sequence “filters” produce a consistent result, the existence of a cleavable targeting signal is assumed.

The second concept is grounded on global sequence features, in particular, the amino acid composition of the proteins under investigation [27, 28]. This technique is motivated by the observation that the amino acid composition of a protein seems to be correlated with its subcellular localization [29]. Following this principle, Reinhardt and Hubbard [27] yielded 81% correct predictions for three possible subcellular locations (cytoplasmic, periplasmic, extracellular) in prokayotes. Schneider [28] reported 93% correct assignment of cytoplasmic and noncytoplasmic proteins and an estimated fraction of potentially noncytoplasmic proteins between 15% and 30% from an investigation of 15 bacterial genomes. For this approach the full sequence or only segments, *e.g.*, the *N*-terminal part, may be used. In contrast to the sliding window these tools are also applicable to faulty or incomplete sequences. They might represent methods of choice for rapid first-pass analysis of EST-, CONTIG- or cDNA-derived sequences. Their most obvious disadvantage is the inability to deliver information about the location of a potential targeting signal within a sequence. Furthermore, it is essential that these prediction methods are appropriately calibrated for their respective target species, as the amino acid usage differs among the organisms, one reason for which is the different G + C content of their genomes (Table 1) [30]. First such species-specific systems have been developed [1, 2, 31], but much work has still to be done to investigate the influence of species-specific sequence features on prediction accuracy and the limits of individual prediction methods.

Methods for targeting signal prediction can be further characterized by the representation of sequence information they rely on: amino acid composition, physicochemical and structural properties (hydrophobicity, charge, secondary structure, *etc.*), or canonic residue symbols. The most advanced methods use several sequence representations and a combination of different algorithms

Table 1. Amino acid compositions in percent of three eukaryotes (*Homo sapiens*, *Arabidopsis thaliana*, *Saccharomyces cerevisiae*), one archaeon (*Methanococcus jannaschii*), and two bacteria (*Mycoplasma genitalium*, *Bacillus subtilis*)[a)]

Amino acid	*Homo sapiens*	*Arabidopsis thaliana*	*Saccharomyces cerevisiae*	*Methanococcus jannaschii*	*Mycoplasma genitalium*	*Bacillus subtilis*
A	7.04	6.25	5.47	5.45	5.57	7.7
C	2.28	1.85	1.3	1.28	0.83	0.8
D	4.7	5.44	5.8	5.52	4.92	5.19
E	7	6.76	6.48	8.66	5.64	7.25
F	3.65	4.32	4.5	4.25	6.14	4.49
G	6.63	6.34	4.96	6.33	4.63	6.92
H	2.62	2.29	2.17	1.43	1.58	2.28
I	4.32	5.36	6.58	10.51	8.25	7.35
K	5.64	6.42	7.3	10.4	9.49	7.03
L	9.97	9.52	9.58	9.45	10.69	9.65
M	2.14	2.44	2.08	2.3	1.54	2.77
N	3.55	4.42	6.13	5.3	7.52	3.94
P	6.4	4.78	4.36	3.36	3	3.69
Q	4.74	3.47	3.91	1.45	4.73	3.84
R	5.72	5.4	4.44	3.84	3.1	4.13
S	8.33	8.98	9.04	4.51	6.66	6.3
T	5.35	5.12	5.89	4.05	5.4	5.42
V	6	6.71	5.57	6.8	6.11	6.74
W	1.26	1.26	1.05	0.73	0.97	1.03
Y	2.64	2.88	3.38	4.38	3.25	3.48
Other or unknown	2.65	2.88	3.38	4.38	3.25	3.48

a) from URL: http://www.ebi.ac.uk/proteome/index.html [84]

for prediction. Table 2 lists prediction methods which are accessible to the public *via* WWW, some of which are combinations of approaches. Numerous additional techniques have been published, most of them without providing a possibility for public access (for an overview of additional targeting signal prediction tools, see, *e.g.*, [3, 32]).

The first prediction systems for targeting sequences were linear discriminant functions using weight matrices that were grounded on observed residue patterns in sets of known targeting signals [7, 33]. The most prominent example probably is the method developed by von Heijne and the "-3,-1 rule" describing preferred residues in positions 3 and 1 relative to a signal peptidase I cleavage site [7, 8, 34]. These methods were based on the analysis of limited sets of known "positive examples". Genome projects and the use of modern biological screening methods (*e.g.*, "signal-trap" and similar techniques [35, 36]) have resulted in significantly larger sets of reference data, and consequently prediction methods have become more robust as they benefit from a broader sample set for statistical analysis of observed residue preferences (Table 1). In the following we describe selected machine learning methods that extend and complement the original weight matrix method [24].

3 Machine learning methods for predicting targeting signals

Hidden Markov Models (HMMs), supervised multilayer feed-forward artificial neural networks (ANNs), self-organizing maps (SOMs), and support vector machines (SVMs) have been employed for devising rules that can be used for targeting signal prediction (for reviews of these methods, see [37–40]). With the exception of HMMs which can be developed using only a set of "positive examples" (*e.g.*, known signal peptides), these prediction systems usually represent nonlinear classifiers that separate "positive examples" from "negative examples". All learning machines are developed in two stages, training and testing. During the training phase classifiers are established by adapting internal model parameters, and the performance and generalization ability is as-

Table 2. Selected prediction tools and databases for signal sequence analysis on the WWW

Name	URL	Description	Method
SignalP [65, 85]	http://www.cbs.dtu.dk/services/SignalP/	Predicts the presence and location of signal peptide cleavage sites in amino acid sequences from different organisms: Gram-positive prokaryotes, Gram-negative prokaryotes, and eukaryotes	Several ANNs, HMM
ChloroP [86]	http://www.cbs.dtu.dk/services/ChloroP/	Predicts the presence of chloroplast transit peptides (cTPs) in protein sequences and the location of potential cTP cleavage sites	ANN
TargetP [64]	http://www.cbs.dtu.dk/services/TargetP/	Predicts presence of any of the *N*-terminal presequences: cTP, mTP or secretory pathway signal peptides and their potential cleavage site	Several ANNs
LipoP [10]	http://www.cbs.dtu.dk/services/LipoP/	Predicts signal peptides and SP II cleavage site in lipoproteins from Gram-negative bacteria	HMM
NNPSL [27]	http://www.doe-mbi.ucla.edu/cgi/astrid/nnpsl_mult.cgi	Prediction method for the subcellular location of proteins	ANN
PSORT-B [45]	http://www.psort.org/psortb/index.html	Prediction of the subcellular localization of proteins (Gram-negative bacteria only); six analytical modules, each of which analyzes one biological feature known to influence or be characteristic of subcellular localization (binary or multicategory classifiers)	Six methods, based on SVM, BLAST-P, HMM
PredictNLS [87]	http://maple.bioc.columbia.edu/predictNLS/	Automated tool for the analysis and determination of nuclear localization signals (NLSs)	Local database searching
PlasMit [1]	http://gecco.org.chemie.uni-frankfurt.de/plasmit/index.html	Prediction of mTPs from *Plasmodium falciparum*	ANN
PATS [2]	http://gecco.org.chemie.uni-frankfurt.de/pats/pats-index.php	Prediction of apicoplast sequences from *Plasmodium falciparum*	ANN
Signal Peptide Prediction [66]	http://bioinformatics.leeds.ac.uk/prot-analysis/Signal.html	Prediction of signal peptides and location of their cleavage sites	Weight matrices
SIGFIND [88]	http://www.stepc.gr/~synaptic/sigfind.html	Predicts signal peptides at the start of protein sequences or searches open reading frames with a potential signal peptide coded in nucleotide sequences	Bidirectional recurrent neural networks with jury decision
SubLoc [60]	http://www.bioinfo.tsinghua.edu.cn/SubLoc/	Predicts three locations for prokaryotic sequences, and four localizations for eukaryotic sequences based on amino acid composition	SVM
PLOC [61]	http://www.genome.ad.jp/SIT/ploc.html	Predicts 12 subcellular localizations of proteins based on amino acid composition and gapped residue pairs	SVM classifiers with jury decision
iPSORT [54]	http://hypothesiscreator.net/iPSORT/	Rule-based system for predicting *N*-terminal sorting sequences based on physicochemical and biochemical properties; training data from TargetP [52]	Rule-based classifier
PATOSEQ [55]	http://www.expasy.org/tools/patoseq/	Prediction of lipoprotein signals for *Bacillus subtilis* sequence data	Sequence motif matching
LOChom	http://cubic.bioc.columbia.edu/db/LOChom/	Database of subcellular localization predictions based on sequence homology to experimentally annotated proteins	
LOC3d	http://cubic.bioc.columbia.edu/db/LOC3d/	Database of predicted subcellular localization of eukaryotic proteins with known 3-D structure	
LOCkey	http://cubic.bioc.columbia.edu/db/LOCkey/	Predicted subcellular localizations for entire proteomes	

sessed during the test phase by statistical methods (*e.g.*, jack-knifing or bootstrapping) using data sets which were not used for training [37]. Table 2 lists two data sets (LOChom, LOC3D) which were compiled from primary databases and employed for developing prediction systems (see the respective web links for reference).

3.1 Hidden Markov Models

HMMs are closely related to neural networks (*vide infra*), stochastic grammars, and Bayesian networks [38, 41]. The success of the HMM approach is critically influenced by an appropriate alignment of the training sequences. A standard HMM consists of a finite set of nodes representing "hidden states". These nodes are interconnected by links describing the probabilities of a transition between the individual states (Fig. 1a). Additionally, each hidden state has an associated set of probabilities of emitting a particular "visible state". A discrete alphabet *A* of symbols is assigned to the hidden and visible states. In the context of protein sequences, *A* is the standard 20-letter amino acid alphabet. The transition matrix *T* specifies the probabilities of going from the hidden state *x* to the hidden state *y*. The emission matrix *E* indicates the probabilities of emitting a certain symbol *S* in a certain hidden state. During HMM training the model parameters *T* and *E* are determined from an ensemble of training samples. No known learning method guarantees the obtainment of optimal system parameters, but there are algorithms which were shown to be well-suited for HMM optimization [42]. The architecture of HMMs explicitly considers deletion and insertion states (Fig. 1a). Thus, HMMs allow for the modeling of sequences of varying lengths. Because of this inherent property HMMs are fitting for the prediction of targeting signals. If HMMs are applied to pattern recognition, there exist several HMMs – one for each category. A test sequence is then classified according to the model with the highest probability [42].

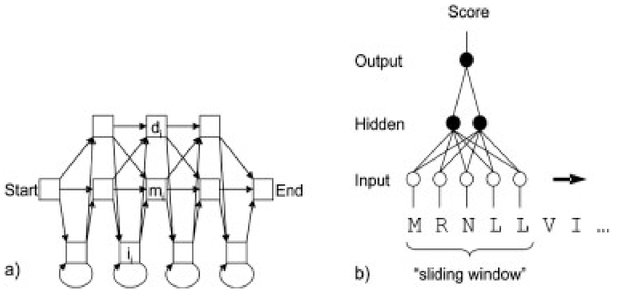

Figure 1. (a) Schematic of a standard HMM. In addition to the Start and End states, the HMM contains three classes of states, the Main states m_i, the Deletion states d_i, and the Insert states i_i. Arrows represent transition probabilities between the states. The Main and the Insert states always emit an amino acid residue, the Delete states are mute (gaps). The HMM represents a multiple sequence alignment, and new sequences can be compared to the HMM model. The probabilities of emitting a particular symbol from each Main and Insert state are omitted in this diagram. (b) Schematic of a three-layered ANN. Three layers of neurons (circles) and weights (lines) are connected to calculate a score value from an amino acid sequence. The input neurons are "fan-out" neurons distributing input values to the hidden layer neurons. The size of the "sliding window" is five residues in the example. Typically, the score value is computed for the central window position (here: asparagine).

The number of model parameters of a HMM quickly increases with regard to the size of the alphabet employed. In the case of protein models this number can be unfavorably large. Another drawback is their inability to express dependencies between non-neighbored hidden states. This can prevent the identification of complex signal sequence features that might be formed by interaction of non-neighbored residues. One solution might be the use of correlation-based descriptors for sequence analysis and other modeling algorithms than HMM [43, 44]. Despite these handicaps HMM models have been very successfully applied to the prediction of protein targeting signals. Nielsen *et al.* [24] developed an HMM version of their previously established prediction system SignalP for secretory signal peptides. The three distinct regions of secretory signal peptides – the n-region, the h-region and the c-region – are explicitly incorporated by individual parts of the model. The PSORT-B prediction system for subcellular protein localization of Gram-negative bacteria integrates six different analyses yielding an overall precision of 97% and recall of 75% in fivefold cross-validation tests with a dataset containing 1443 proteins of experimentally known localization [45]. Two of these methods are based upon HMMs, namely the prediction of α-helical transmembrane regions and signal peptides.

Recently, an HMM was developed by Krogh and co-workers [10] for prediction of lipoprotein signal peptides ("LipoP") which are cleaved by SP II, a specific signal peptidase for bacterial lipoprotein precursors. The prediction accuracy was $>$ 98% with only 0.3% false positive assignments of other targeting signal-containing sequences as assessed by a leave-one-out statistics with 63 lipoproteins. The authors compared the HMM to an ANN model and obtained comparable results which is in accordance with observations by Apweiler and co-workers [46]. Another HMM model was recently presented by Zhang and Wood [9] who yielded approximately 95% sensitivity and specificity for eukaryotic signal peptide prediction using a collection of 892 human and 644 mouse signal-containing proteins.

3.2 Multilayer feed-forward networks

ANNs belong to the class of supervised neural networks, *i.e.*, the data used for training the network has to include information about the property (or category) the trained neural network should predict. The architecture of ANNs comprises two types of building blocks (Fig. 1b): formal neurons and connections between the neurons. The neurons are arranged in layers, whereas at least three layers of neurons are needed to form a multilayer feed-forward network. The first layer is called input layer, the last one output layer, and all layers in between are "hidden" layers. The number of neurons in the input layer equals the number of dimensions of the input data. Although different possibilities exist, in most cases the number of neurons in the output layer equals the number of classes in the input data minus one. The number of neurons in the hidden layers can be adjusted depending on the classification task at hand. A formal neuron transforms a numerical input to an output value. Every neuron of one layer is connected to every neuron of the following layer, and there are no connections between neurons of the same layer. The connections between neurons are numerical weight values that are optimized during network training according to an error function. Such a function describes deviation of predicted target values from observed values. Several algorithms are available for optimizing the free variables of an ANN during training, *e.g.*, gradient descent techniques, simulated annealing or evolutionary algorithms [37, 39].

There are several advantages of ANNs. If properly trained, they are able to cope with noisy data, and they have the ability to generalize [39]. It must be stressed that ANNs are especially qualified for modeling of nonlinear input/output relationships. There are, however, limitations of ANNs. A trained neural network is often considered as a "black box", *i.e.*, an easy understanding of the decisive features can hardly be obtained [47]. One possible complementary approach are "rule-based" systems that result in sets of human interpretable rules, typically of the "if . . . then . . ." type. First analyses of signal sequences were performed using inductive logic programming (ILP) systems [48, 49]. A more recent example of elaborated rule sets was obtained by Bannai *et al.* [50] using the same data set which was also employed for ANN training. It is important to stress that these rules yield only slightly lower prediction accuracy than the neural network system. A different concept resulting in interpretable rules was followed by Gonnet and Lisacek [51] who relied on probabilistic sequence motif generation. These motifs describe preferred patterns of amino acid residues and properties that can be used for database searching and sequence classification. Such complementary approaches can certainly help decipher targeting signals.

Another problem is overfitting of ANNs [52]. Generally, a neural network with perfect prediction can be found for every classification problem. But such a network is not able to generalize any more: it predicts the training data perfectly but it fails to classify data not used during training. In general, multilayer feed-forward neural networks have found a widespread use for classification tasks. This observation also holds for the classification of targeting signals as most of the prediction tools listed in Table 2 employ the ANN machine learning method.

3.3 Self-organizing map

SOMs or Kohonen-networks are unsupervised neural networks [39]. They map a high-dimensional input space to a lower-dimensional target space in a nonlinear fashion. As a result, the proximity of points in the target space (the map) reflects proximity of points in the source space, resulting in a "topological map" (Fig. 2). The basic building blocks of SOMs are similar to those of multilayer feed-forward networks, *i.e.*, they are composed of formal neurons and connections between these neurons (weights). SOMs consist of two layers: the input and the output layer. The number of input neurons and the number of weights connected to an output neuron is identical to the dimensionality of the input data. As each output neuron represents a data cluster the number of output neurons equals the

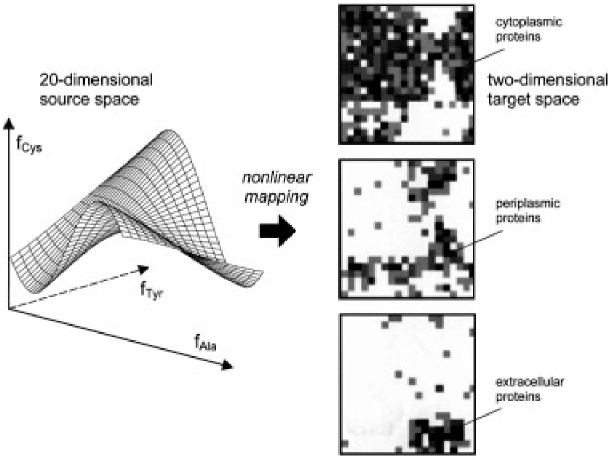

Figure 2. The SOM algorithm can be used to produce a low-dimensional map of the distribution of proteins in a high-dimensional space, *e.g.*, the space spanned by the relative amino acid composition. In the example, cytoplasmic, periplasmic, and extracellular bacterial proteins were encoded by their amino acid composition and projected onto a planar map. Shading indicates the density of these subsets on the map (dark: many; white: none). Each little square of the map represents a formal neuron. Obviously, the amino acid composition is a suited sequence descriptor for separation of the three protein classes. Adapted from [28].

number of clusters. Every data point is assigned to exactly one output neuron. This allocation is determined by the best match of a data point, *i.e.*, a protein sequence, with the weight vector of an output neuron. During the training phase the weight vectors of the SOM are determined [53].

SOMs have first been used for comparative proteome analysis [54] and secondary structure prediction [55]. To the best of our knowledge there exists no public web-based prediction tool for targeting signals that is based upon a SOM. Nevertheless, SOMs were successfully applied to the clustering and visualization of proteins that are targeted to the mitochondrion and the extracellular space [28, 56].

3.4 Support vector machine

The SVM approach for solving classification tasks was introduced by Vapnik [57] about two decades ago. The classical SVM is a data-driven method for binary classification. SVM classifiers are generated by a two-step procedure: first, the sample data vectors are mapped ("projected") to a high-dimensional space. The dimension of this space is significantly larger than dimension of the original data space. Then, the algorithm finds a hyperplane in this high-dimensional space with the largest margin separating classes of data. It was shown that classification accuracy usually depends only weakly on the specific projection, provided that the target space is sufficiently high-dimensional [58]. Sometimes it is not possible to find a separating hyperplane even in a very high-dimensional space. In this case, a tradeoff is introduced between the size of the separating margin and penalties for every data vector which lies within the margin [58]. Points classified by SVM can be divided into two groups, support vectors and nonsupport vectors. Nonsupport vectors are classified correctly by the hyperplane and are located outside the separating margin. Parameters of the hyperplane do not depend on them, and even if their position is changed the separating hyperplane and margin will remain unchanged, provided that these points will stay outside the margin. Other points are support vectors, and they are the points which determine the exact position of the hyperplane. Informally speaking, support vectors contain the important information for the classification task.

One big advantage of support vector machines is the sparseness of the solution, *i.e.*, the separating hyperplane solely depends on the support vectors and not on the complete data set. Thus, SVMs tend to be less prone to overfitting than other classification methods. Whereas an ANN finds one of all possible separating hyperplanes, SVMs find the separating hyperplane with the largest margin. It is expected that the larger the margin is, the better the generalization of the classifier. SVMs are also very robust with regard to noisy features and are known to be able to cope with a large number of features [59]. But even though there is a smaller chance of overfitting than with ANNs, this problem is also present in SVM training. SVMs are not as widely used for classification tasks as ANNs yet [40]. A possible reason might be that they are not part of standard packages used for machine learning. The PSORT-B prediction tool employs an SVM to discriminate between cytoplasmic and noncytoplasmic sequences [45]. SUBLOC implements an SVM system that predicts three subcellular localizations based on amino acid composition, yielding over 91% positive correct predictions [60]. A more recent example is the SVM prediction tool developed by Cai *et al.* [61] who reached a comparable accuracy for a two-state classifier for secretory/nonsecretory proteins. These authors suggest the use of multiple prediction systems in parallel to reduce false-positives. The most comprehensive SVM application was described by Park and Kanehisa [62] who considered 12 subcellular localizations in eukaryotes for a prediction system (PLOC) consisting of 60 different SVMs and a jury decision. Amino acid frequencies and residue-pair frequencies were used for sequence encoding. The overall accuracy (total accuracy) of this system is approximately 80% as judged from a fivefold cross-validation test. A main outcome from this study is that a smart combination of different classifiers can result in more robust predictions than the individual systems. The study of Park and Kanehisa [62] complements earlier work by Chou and Elrod [63] who also considered 12 different subcellular compartments but used covariant discriminant analysis for classification but obtained slightly lower prediction accuracy.

4 Using prediction systems: an example application

Figure 3 demonstrates a typical result of a targeting signal prediction. Three prominent methods were used to analyze the precursor sequences of human cyclooxygenases 1 and 2 (COX-1, COX-2): TargetP [64], SignalP [65], and SignalPeptidePrediction [66]. An alignment of the two sequences reveals a high overall sequence similarity of 75% with the exception of the *N*-terminal portion. This observation substantiates earlier findings that related proteins do not necessarily have similar *N*-terminal signal peptides [67]. The three different prediction tools produce consensus predictions for COX-1 and COX-2 plus some additional potential SP I cleavage sites. Certainly one would most likely trust the consensus result: the exis-

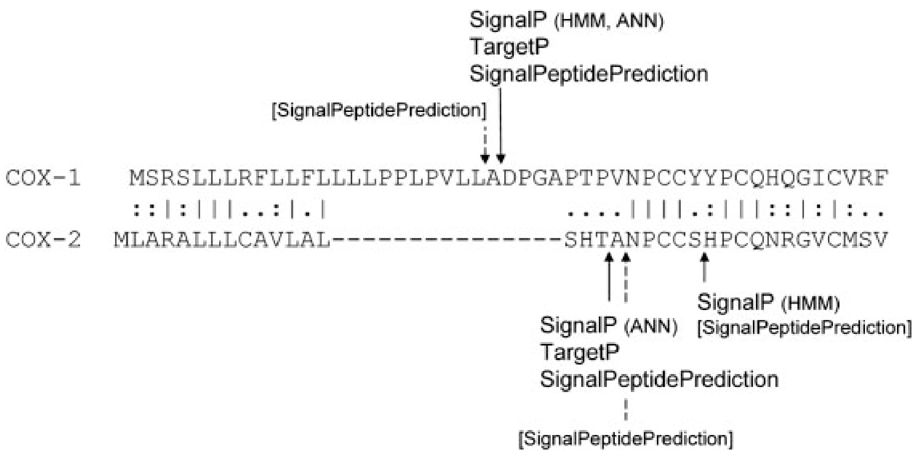
SignalP (HMM, ANN)
TargetP
SignalPeptidePrediction
[SignalPeptidePrediction]
COX-1 MSRSLLLRFLLFLLLLPPLPVLLADPGAPTPVNPCCYYPCQHQGICVRF
COX-2 MLARALLLCAVLAL---------------SHTANPCCSHPCQNRGVCMSV
SignalP (ANN)
TargetP
SignalPeptidePrediction
[SignalPeptidePrediction]
SignalP (HMM)
[SignalPeptidePrediction]

Figure 3. Predicted SP I cleavage sites in precursor sequences of human cyclooxygenases 1 (COX-1) and 2 (COX-2). Three different prediction tools were applied. Arrows indicate potential cleavage sites. Dotted arrows point to predicted alternative cleavage sites, which are also indicated by square brackets around the method's name. The alignment was generated by the EMBOSS method using default parameters (URL: http://www.ebi.ac.uk/emboss/align/index.html) and SWISS-PROT sequences PGH1_HUMAN (COX-1) and PGH2_HUMAN (COX-2).

tence of an *N*-terminal signal peptide at positions 1–24 in the COX-1 precursor, and at positions 1–17 of the COX-2 precursor sequence. The SWISS-PROT entry for human COX-1 (PGH1_HUMAN) confirms the existence of a signal peptide [68], but reports a clevage site between residues 23 and 24; the COX-2 entry (PGH2_HUMAN) lists a potential signal peptide at positions 1–17, which is identical to the consensus prediction. From the three methods used in the example only SignalPeptidePrediction identified a potential SP cleavage site after residue 23 which is in accordance with the SWISS-PROT annotation. Which one is correct? Going to the original literature, a putative 24-residue signal has been proposed for COX-1 [69, 70], and in a more recent work a 23-residue signal peptide has been reported [71]. To complicate things even more, a third variant COX-3 has recently been described containing an SP which is not cleaved off by signal peptidase [72].

The COX-2 example in Fig. 3 shows a potential downstream cleavage site which was predicted by two methods, and a second one immediately after the consensus site. Based on the predictions alone one might speculate that either SP I could cleave the precursor at the predicted sites, but downstream sites might be inaccessible for some reason. Or, there exists hitherto unidentified proteolytic activity which actually cleaves off one or more residues after primary cleavage by SP. Indeed, in mitochondria, an intermediate signal peptidase (IMP) has been found that cleaves off an additional stretch of eight or nine residues after primary cleavage by mitochondrial matrix processing peptidase (MPP) from approximately 30% of all imported protein precursors [73]. Many other cellular processes might be involved that result in variant or multiple *N*-termini of the mature protein. Proteome analyses and sequencing campaigns will be very important to help assess the *N*-termini of many mature proteins, and will be beneficial for database maintenance. Once such experimental facts will be available, potential discrepancies between predictions, literature reports, and database annotations might be resolved. The COX example also shows that methods for predicting targeting signals can only be as accurate as the data used for method development. A recent comparison of the performance of five different signal sequence prediction approaches by Apweiler and co-workers [46] led to the conclusion that SignalP (V2.0b2) seems to be most reliable and is thus employed for automatic SWISS-PROT annotation. The authors stress the fact that one should discriminate between classification and cleavage-site prediction when using current software tools, since different "best" methods exist for these two purposes.

5 Conclusions and outlook

The most crucial step during the development of a prediction system is the selection of appropriate training and test data, which should be representative of the underlying problem. A benefit from proteome analyses will certainly be the availability of augmented sets of reference data, not only of secreted proteins but also nuclear-encoded organellar proteins. These sets will allow for the refinement of prediction tools, and render the development of new systems feasible. The existence of "unconventional" targeting signals, like *C*-terminal or internal mTPs [74], has been known for some time, yet recognition of such signals in sequences is still not possible using automated methods. Very likely additional protein targeting principles and pathways exist which we are not aware of today, or have no training data set and thus escape our notion and consideration for prediction systems, *e.g.*, *C*- to *N*-terminal translocation [75], co-translational pathways for protein import into organelles [76], and structural aspects of targeting signals. Future research will also have to consider potential targeting signal features in the mature protein. Moreover, there are reports of splicing events resulting in alternative transcripts which lack the signal peptide [77–79]. A challenging question will also be addressable with more data being available from proteomics and genomics studies: How did *N*-terminal targeting signals evolve? What is the explanation for the existence of very different signal sequences in closely related proteins? If we are able to provide answers to these questions we will also be in a position to develop more reliable prediction systems.

Targeting signals might consist of multiple domains with either distinct or even overlapping functions [22]. Current analysis tools focus very much on *N*-terminal signals and amino acid composition, and machine learning methods have been used to make predictions increasingly robust. Recently, a model of the three-dimensional structure of the catalytic domain of bacterial SP I has been devised from X-ray spectroscopic data [80], which enables the application of structure-based methods for predicting potential substrates, *e.g.*, by ligand- and receptor-derived pharmacophores [81]. Such methods have not yet been applied to sequence analysis for targeting signal prediction. Still, the real challenge probably is to compile data sets that allow for the extraction and identification of the "atypical" targeting signals. Studies of proteomes will likely provide some of the required information. Also, codon usage can be employed for predicting subcellular localization, *e.g.*, to differentiate between cytoplasmic proteins and their organellar counterparts [82]. This possibility could be elaborated to make use of "genome signatures" for organism-specific signal sequence prediction systems [83].

Paul Wrede and Jürgen Paetz are thanked for helpful discussion and support during the preparation of the manuscript. This work was supported by the Beilstein-Institut zur Förderung der Chemischen Wissenschaften, Frankfurt.

6 References

[1] Bender, A., van Dooren, G. G., Ralph, S. A., McFadden, G. I., Schneider, G., *Mol. Biochem. Parasitol.* 2003, *132*, 59–66.

[2] Zuegge, J., Ralph, S., Schmuker, M., McFadden, G. I., Schneider, G., *Gene* 2001, *280*, 19–26.

[3] Emanuelsson, O., von Heijne, G., *Biochim. Biophys. Acta* 2001, *1541*, 114–119.

[4] Baldi, P., Brunak, S., Chauvin, Y., Anderssen, C. A., Nielsen, H., *Bioinformatics* 2000, *16*, 412–424.

[5] Nakai, K., *Adv. Prot. Chem.* 2000, *54*, 277–344.

[6] Wrede, P., Landt, O., Klages, S., Fatemi, A., Hahn, U., Schneider, G., *Biochemistry* 1998, *37*, 3588–3593.

[7] Von Heijne, G., *Eur. J. Biochem.* 1983, *133*, 17–21.

[8] Von Heijne, G., *Nucleic Acids Res*. 1986, *14*, 4683–4690.

[9] Zhang, Z., Wood, W. I., *Bioinformatics* 2003, *19*, 307–308.

[10] Juncker, A. S., Willenbrock, H., von Heijne, G., Brunak, S., *et al.*, *Prot. Sci*. 2003, *12*, 1652–1662.

[11] Pugsley, A. P., *Protein Targeting*, Academic Press, San Diego, CA 1989.

[12] Stroud, R. M., Walter, P., *Curr. Opin. Struct. Biol.* 1999, *9*, 754–759.

[13] McFadden, G. I., *J. Eukaryot. Microbiol*. 1999, *46*, 339–346.

[14] Baker, A., Kaplan, C. P., Pool, M. R., *Biol. Rev. Camb. Philos. Soc*. 1996, *71*, 637–702.

[15] Meacock, S. L., Greenfield, J. J., High, S., *Essays Biochem*. 2000, *36*, 1–13.

[16] Johnson, A. E., van Waes, M. A., *Annu. Rev. Cell. Dev. Biol*. 1999, *15*, 799–842.

[17] Von Heijne, G. (Ed.), *Signal Peptidases*, R. G. Landes, Austin, TX 1994.

[18] Paetzel, M., Karla, A., Strynadka, N. C., Dalbey, R. E., *Chem. Rev.* 2002, *102*, 4549–4580.

[19] Rutkowski, D. T., Lingappa, V. R., Hedge, R. S., *Proc. Natl. Acad. Sci. USA* 2001, *98*, 7823–7828.

[20] Ott, C. M., Lingappa, V. R., *J. Cell Sci*. 2002, *115*, 2003–2009.

[21] Kim, S. J., Rahbar, R., Hedge, R. S., *J. Chem. Biol*. 2001, *276*, 26132–26140.

[22] Bruce, B. D., *Biochim. Biophys. Acta* 2001, *1541*, 2–21.

[23] Clark, H. F., Gurney, A. L., Abaya, E., Baker, K., *et al.*, *Genome Res*. 2003, *13*, 2265–2270.

[24] Nielsen, H., Brunak, S., von Heijne, G., *Prot. Eng*. 1999, *12*, 3–9.

[24] Nielsen, H., Brunak, S., von Heijne, G., *Prot. Eng*. 1999, *12*, 3–9.

[25] Zhang, J., Madden, T. L., *Genome Res*. 1997, *7*, 649–656.

[26] Pearson, W. R., *Methods Enzymol*. 1990, *183*, 63–98.

[27] Reinhardt, A., Hubbard, T., *Nucleic Acids Res*. 1998, *26*, 2230–2236.

[28] Schneider, G., *Gene* 1999, *237*, 113–121.

[29] Nakashima, H., Nishikawa, K., *J. Mol. Biol*. 1994, *238*, 54–61.

[30] Lobry, J. R., *Gene* 1997, *205*, 309–316.

[31] Von Heijne, G., Abrahmsen, L., *FEBS Lett*. 1989, *244*, 439–446.

[32] Claros, M. G., Brunak, S., von Heijne, G., *Curr. Opin. Struct. Biol*. 1997, *7*, 394–398.

[33] Laforet, G. A., Kendall, D. A., *J. Biol. Chem*. 1991, *266*, 1326–1334.

[34] Von Heijne, G., *J. Mol. Biol*. 1984, *173*, 243–251.

[35] Klein, R. D., Gu, Q., Goddart, A. Rosenthal, A., *Proc. Natl. Acad. Sci*. USA 1996, *93*, 7108–7113.

[36] Baker, K., Guerney, A. L., US Patent 6, 060, 249 2000.

[37] Duda, R. O., Hart, P. E., Stork, D. G., *Pattern Classification*, John Wiley & Sons, New York 2001.

[38] Eddy, S. R., *Bioinformatics* 1998, *14*, 755–763.

[39] Schneider, G., Wrede, P., *Prog. Biophys. Mol. Biol*. 1998, *70*, 175–222.

[40] Byvatov, E., Schneider, G., *Appl. Bioinformatics* 2003, *2*, 67–77.

[41] Baldi, P., Brunak, S., *Bioinformatics*, second edition, MIT Press, Cambridge, MA 2001, pp. 128–138.

[42] Duda, R. O., Hart, P. E., Stork, D. G., *Pattern Recognition*, second edition, John Wiley & Sons, New York 2001, pp. 165–223.

[43] Chou, K. C., *Biochem. Biophys. Res. Commun*. 2000, *278*, 477–483.

[44] Schneider, G., Broger, C., in: Wagner, E., Normann, J., Greppin, H., Hackstein, J. H. P., *et al.* (Eds.), *From Symbiosis to Eukaryotism – Endocytobiology VII*, Geneva University Press, Geneva 1999, pp. 589–602.

[45] Gardy, J. L., Spencer, C., Wang, K., Ester, M., *et al.*, *Nucleic Acids Res*. 2003, *31*, 3613–3617.

[46] Menne, K., Hermjakob, H., Apweiler, R., *Bioinformatics* 2000, *16*, 741–742.

[47] Sadowski, J., Kubinyi, H., *J. Med. Chem*. 1998, *18*, 3325–3329.

[48] King, R. D., Sternberg, M. J., *J. Mol. Biol*. 1990, *216*, 441–457.

[49] Schneider, G., Wrede, P., *Protein Seq. Data Anal*. 1993, *5*, 227–236.

[50] Bannai, H., Tamada, Y., Maruyama, O., Nakai, K., Miyano, S., *Bioinformatics* 2002, *18*, 298–305.

[51] Gonnet, P., Lisacek, F., *Bioinformatics* 2002, *18*, 1091–1101.

[52] Schneider, G., So, S. S., *Adaptive Systems in Drug Design*, Landes Bioscience, Georgetown, TX 2003, pp. 87–88.

[53] Kohonen, T., *Biol. Cybern*. 1982, *43*, 59–69.

[54] Ferran E. A., Ferrara, P., *Biol. Cybern*. 1991, *65*, 451–458.

[55] Hanke, J., Reich, J. G., *Comput. Appl. Biosci*. 1996, *12*, 447–454.

[56] Schneider, G., Sjöling, S., Wallin, E., Wrede, P., *et al.*, *Proteins* 1998, *30*, 49–60.

[57] Vapnik, V., *The Nature of Statistical Learning Theory*, Springer, Berlin 1995.

[58] Cortes, C., Vapnik, V., *Machine Learning* 1995, *20*, 273–297.

[59] Cristianini, N., Shawe-Taylor, J., *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press, Cambridge 2000.

[60] Hua, S., Sun, Z., *Bioinformatics* 2001, *17*, 721–728.

[61] Cai, Y.-D., Lin, S.-L., Chou, K.-C., *Peptides* 2003, *24*, 159–161.

[62] Park, K.-J., Kanehisa, M., *Bioinformatics* 2003, *19*, 1656–1663.

[63] Chou, K.-C., Elrod, D. W., *Protein Eng*. 1999, *12*, 107–118.

[64] Emanuelsson, O., Nielsen, H., Brunak, S., von Heijne, G., *J. Mol. Biol*. 2000, *300*, 1005–1016.

[65] Nielsen, H., Engelbrecht, J., Brunak, S., von Heijne, G., *Prot. Eng*. 1997, *10*, 1–6.

[66] Bradford, J. R., PhD Thesis, University of Leeds 2001.

[67] Watson, M. E., *Nucleic Acids Res*. 1984, *12*, 5145–5164.

[68] Bairoch, A., Apweiler, R., *Nucleic Acids Res*. 2000, *28*, 45–48.

[69] Yokoyama, C., Takai, T., Tanabe, T., *FEBS Lett*. 1988, *231*, 347–351.

[70] DeWitt, D. L., Smith, W. L., *Proc. Natl. Acad. Sci*. USA 1988, *85*, 1412–1416.

[71] Smith, T., Leipprand, J., DeWitt, D., *Arch. Biochem. Biophys*. 2000, *375*, 195–200.

[72] Chandrasekharan, N. V., Dai, H., Roos, K. L., Evanson, N. K., *et al.*, *Proc. Natl. Acad. Sci*. USA 2002, *99*, 13926–13931.

[73] Isaya, G., Kalousek, F., in: von Heijne, G. (Ed.), *Signal Peptidases*, R. G. Landes Company, Austin, TX 1994, pp. 87–103.

[74] Emanuelsson, O., von Heijne, G., Schneider, G., *Methods Cell Biol*. 2001, *65*, 175–187.

[75] Folsch, H., Guiard, B., Neupert, W., Stuart, R. A., *EMBO J*. 1998, *17*, 6508–6515.

[76] Crowley, K. S., Payne, R. M., *J. Biol. Chem*. 1998, *273*, 17278–17285.

[77] Duchange, N., Saleh, M. C., de Arriba Zerpa, G., Pidoux, J., *et al.*, *Neurochem. Res*. 2002, *27*, 1459–1463.

[78] Mu, W., Cheng, Q., Yang, J., Burt, D. R., *Brain Res. Bull*. 2002, *58*, 447–454.

[79] Luo, D., Mari, B., Stoll, I., Anglard, P., *J. Biol. Chem*. 2002, *277*, 25527–25536.

[80] Paetzel, M., Dalbey, R. E., Strynadka, N. C., *Nature* 1998, *396*, 186–190.

[81] Guner, O. F., Ed., *Pharmacophore Perception, Development, and Use in Drug Design*, International University Line, La Jolla, CA 2000.

[82] Chiapello, H., Ollivier, E., Landès-Devauchelle, C., Nitschké, P., Risler, J. L., *Nucleic Acids Res*. 1999, *27*, 2848–2851.

[83] Sandberg, R., Branden, C. I., Ernberg, I., Coster, J., *Gene* 2003, *311*, 35–42.

[84] Pruess, M., Fleischmann, W., Kanapin, A., Karavidopoulou, Y., *et al.*, *Nucleic Acids Res*. 2003, *31*, 414–417.

[85] Nielsen, H., Krogh, A., *Proceedings of the Sixth International Conference on Intelligent Systems for Molecular Biology (ISMB 6)*, AAAI Press, Menlo Park, CA 1998, pp. 122–130.

[86] Emanuelsson, O., Nielsen, H., von Heijne, G., *Prot. Sci*. 1999, *8*, 978–984.

[87] Cokol, M., Nair, R., Rost, B., *EMBO Rep*. 2000, *1*, 411–415.

[88] Reczko, M., Staub, E., Fiziev, P., Hatzigeorgiou, A., in: Guigo, R., Gusfield, D. (Eds.), *Algorithms in Bioinformatics, Proceedings of the 2nd Int. Workshop WABI* 2002, Rome, Italy, September 16–21, *Lecture Notes in Computer Science*, Springer, Berlin 2002, pp. 60–67.