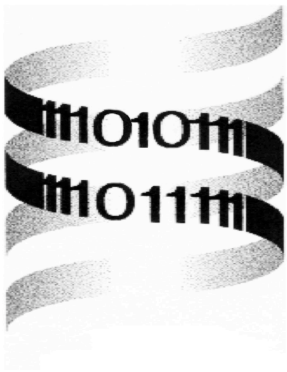
BIOINFORMATICS



Predicting Subcellular Localization of Proteins using Machine-Learned Classifiers

Z. Lu, D. Szafron, R. Greiner, P. Lu, D.S. Wishart, B. Poulin, J. Anvik, C. Macdonell, and R. Eisner

Department of Computing Science, University of Alberta, Edmonton, AB, Canada, T6G 2E8

Received line

ABSTRACT

Motivation: Identifying the destination or localization of proteins is key to understanding their function and facilitating their purification. A number of existing computational prediction methods are based on sequence analysis. However, these methods are limited in scope, accuracy and most particularly breadth of coverage. Rather than using sequence information alone, we have explored the use of database text annotations from homologs and machine learning to substantially improve the prediction of subcellular location.

Results: We have constructed five machine-learning classifiers for predicting subcellular localization of proteins from animals, plants, fungi, Gram-negative bacteria and Gram-positive bacteria, which are 81% accurate for fungi and 92% to 94% accurate for the other four categories. These are the most accurate subcellular predictors across the widest set of organisms ever published. Our predictors are part of the Proteome Analyst (PA) web-service.

Availability:
http://www.cs.ualberta.ca/~bioinfo/PA/Sub
http://www.cs.ualberta.ca/~bioinfo/PA

Supplementary Information:
http://www.cs.ualberta.ca/~bioinfo/PA/Subcellular

Contact: bioinfo@cs.ualberta.ca

INTRODUCTION

High-throughput sequencing technology has made it possible for many laboratories to sequence the genomes of new organisms. There are more than 1200 genome sequences deposited in public databases (EBI, 2003). Given the size and complexity of these data sets, most researchers are compelled to use automated annotation systems to identify or classify individual genes and proteins. As part of this annotation process, a number of systems have been developed that support automated prediction of subcellular localization, based on amino acid sequence information. There are three basic approaches. One approach is based on amino acid composition, using artificial neural nets (ANN) such as NNPSL (Reinhardt and Hubbard, 1998), or support vector machines (SVM) like SubLoc (Hua and Sun, 2001). A second approach uses the existence of peptide signals, which are short sub-sequences of approximately 3 to 70 amino acids to predict specific cell locations, such as TargetP (Emanuelsson, *et al.*, 2000). A third approach, such as the one used in LOCkey (Nair and Rost, 2002), is to do a similarity search on the sequence, extract text from homologs and use a classifier on the text features. Some tools, like PSORT (Nakai and Kanehisa, 1992; Horton and Nakai, 1997), combine a variety of individual predictors. Many tools, like SubLoc, PSORT, and TMHMM (Krogh *et al.*, 2001), are available for public use on the web. Unfortunately, most tools accept only a single sequence at a time, with TMHMM being a notable exception. Emanuelsson (2002) provides a good survey of these tools.

Better Accuracy and Coverage are Needed

There are two limitations to current techniques. First is the limited accuracy of the predictors, especially for some organelles. The second is limited coverage. The term coverage can be used in three ways: location coverage, sequence coverage and taxonomic coverage. All three kinds of coverage are limited in current tools.

First, *location coverage* defines the sub-regions (nuclear, cytoplasmic, extracellular, etc.) in the cell that are supported by a predictor. Most existing tools limit the location coverage to just membranes or just a few organelles.

Second, given a training/test set, *sequence coverage* is defined as the ratio of sequences for which a prediction is made to the total number of sequences of interest. For

example, the LOCkey dataset consists of 3146 labeled sequences from Swiss-Prot and the predictor obtained an accuracy of .87 on a subset of 1161 sequences (coverage = .37). Sequence coverage can be measured on one organism (*1-organism sequence coverage*) or multi-organisms. The 1-organism measure is important for high-throughput prediction for newly sequenced organisms.

A third coverage measure is *taxonomic coverage* – the range of organisms for the predictor such as: animal, green plant, Gram-negative bacteria, etc. Most existing predictors have only been evaluated on a limited number of sequences from a specific taxonomic category of organism (for example, just Gram-negative bacteria or just green plants).

Table 1 lists some predictors and gives a measure of accuracy and the kind of technique employed. It also provides an informal indication of combined sequence coverage and taxonomic coverage. Unfortunately, no standardized sequence coverage ratios have been published for these predictors.

Table 1. Acc(uracies) and informal sequence/taxonomic coverage of current subcellular localization predictors. Gram-negative bacteria and Gram-positive bacteria are denoted GN and GP respectively.

Name	Acc	Coverage	Technique
PSORT-B	.75	1443 GN bacterial	combination
LOCkey	.87	1161 assorted	homology
SubLoc	.91	291 prokaryotic	AA composition
	.79	2427 eukaryotic	
TargetP	.85	940 plant	signal prediction
	.90	2738 non-plant	
Proteome	.93	16284 animal	homology and
Analyst	.93	3420 plant	machine learning
	.81	2104 fungal	
	.92	3218 GN bacterial	
	.94	1571 GP bacterial	

Using Classifiers for Prediction

This paper describes a novel classification technique for predicting subcellular localization (Lu, 2003). This technique is used in our publicly available web-based Proteome Analyst (PA). Two tools are available for subcellular localization – a simple tool (PA-SUB) that only predicts subcellular localization (http://www.cs.ualberta.ca/~bioinfo/PA/Sub) and a more comprehensive tool that predicts subcellular localization along with other annotations, including general function (http://www.cs.ualberta.ca/~bioinfo/PA). The second tool also allows a user to build a custom classifier from custom training data.

A controlled vocabulary or ontology is required for subcellular localization. In fact, since cell structure varies across organisms, several ontologies are required and PA supports five: animal, plant, fungi, Gram-negative bacteria (GN) and Gram-positive bacteria (GP), which are based on the PSORT ontologies. Among them, PSORT (bacteria/plants), PSORT II (animals/yeast) (Nakai, 2000) and PSORT-B (GN bacteria) provide a set of predictors over the same classes of organisms as PA. However, PSORT and PSORT II are older systems with poor accuracy, whereas PSORT-B is a newer system with much better accuracy (Gardy *et al.*, 2003).

In general, a classifier takes a query instance, described by a set of feature-value pairs, and returns one of a fixed number of labels (Mitchell, 1997). In PA, each query instance is a primary sequence that is *BLAST*ed against the Swiss-Prot database to obtain a set of homologs. Each feature of the query instance is a Boolean value corresponding to the presence or absence of a token (word or phrase) from certain fields of the homologous sequences' Swiss-Prot database entries.

We use a machine learning (ML) algorithm to learn a mapping from the features of a query instance to the appropriate subcellular localization label for that instance. A common technique is to apply a ML algorithm to a set of labeled training items to produce a classifier. In our case, each training item consists of a primary protein sequence and the ontological label it has been assigned by an expert. Each training instance is first *BLAST*ed against Swiss-Prot to identify its features in the same manner as query instances. Features are not provided in the training set – they are computed automatically from Swiss-Prot data.

In this paper, we use three different sources for labeled training data: Swiss-Prot database entries that have unambiguous subcellular localization annotations (26,458 sequences), a subset of the Swiss-Prot database developed for LOCkey (3146 sequences) and the set of GN bacteria sequences (1443) used in PSORT-B. These three data sets are used to evaluate the PA classifiers. However, a PA user can also create a custom subcellular localization classifier using custom training data, by simply uploading a file of labeled training sequences (Szafron *et al.*, 2003b). No programming is required.

In the context of PA, *transparency* is the ability to provide formally-sound and intuitively-simple reasons for each prediction (Szafron *et al.*, 2003a). PA bases its predictions on well-understood concepts of conditional probabilities. Its explanations are presented as stacked bar-graphs that clearly display the evidence for each prediction.

Contributions

This paper describes a subcellular localization prediction technique that makes the following scientific contributions:

1) This machine learning technique makes the most accurate subcellular localization predictions over the broadest range of organisms (animals, plants, fungi,

GN bacteria and GP bacteria) of all subcellular localization prediction techniques published to date.
2) This technique is publicly available as a high-throughput web-based tool in Proteome Analyst.
3) Proteome Analyst provides the first explanation facility for subcellular localization predictions.
4) Proteome Analyst can be used to easily create new subcellular classifiers using custom training data, without any programming.

SYSTEMS AND METHODS

The Prediction Process

PA predicts the subcellular localization of a query protein sequence using its primary sequence and the organism taxonomic category: animal, plant, fungi, GN bacteria, GP bacteria. Here is the *five-step prediction process* used by PA.

P1. The primary sequence of the query protein is *BLAST*ed against the Swiss-Prot database and a set of homologous sequences is selected.
P2. Potential features are computed by extracting text from the Swiss-Prot records of the best homologs. A feature has the value *true* if a token representing that feature is extracted and *false* if no such token is extracted.
P3. The user-provided taxonomic organism category is used to select one of five pre-built Naïve Bayes classifiers (Duda and Hart, 1973**)**: animal, plant, fungi, GN bacteria, GP bacteria.
P4. The features are used by the appropriate classifier to compute the probability of each label in the ontology of that classifier. The label with the highest probability is considered the primary location for the protein.
P5. The user can view a graphical explanation of the prediction (Szafron *et al.*, 2003a).

We use the GP bacterial protein *Exodeoxyribonuclease* from *Streptococcus pneumoniae* (*EXOA_STRPN*) as an example. If this organism was newly sequenced, its proteins would not appear in Swiss-Prot. Therefore, we removed all *EXOA_STRPN* entries from our Swiss-Prot database for this demonstration. We experimented with many variations of steps P1 (homolog selection) and step P2 (feature extraction), as described in the Discussion Section. In this Section, we describe only the best configuration. We select up to three homologs with the lowest *BLAST* E-values that are less than 0.001.

Fig. 1 shows three homologs of our query protein sequence. For feature selection, we obtained the best results using phrases extracted from selected fields of the Swiss-Prot homologs. Specifically, we extracted each semi-colon delimited phrase from the Swiss-Prot KEYWORD field of each selected homolog, as well as all InterPro numbers from the DBSOURCE field. Finally, we checked for the inclusion of a pre-defined set of phrases in the SUBCELLULAR LOCALIZATION subfield of the COMMENT field. For ease of reference in this paper, we will denote these fields by: KWORD, IPR and SCELL respectively. This set of phrases forms the potential feature set. The Discussion Section describes alternative feature definition strategies that produced less accurate classifiers.

After computing the potential feature set, we remove all ubiquitous phrases like: "complete proteome", that are contained in a stop-word list (van Rijsbergen, 1979). For example, **Fig. 2** shows the potential feature set for the demonstration query sequence (*EXOA_STRPN*) that were extracted from the top three homologs. They appear under the heading "Unique tokens extracted from Protein #6".

Our classifiers remove other features as well. When PA builds a classifier, it actually learns the *best* set of features to use. This process of *feature selection* is a standard ML technique for improving accuracy (Kohavi and John, 1997). In fact, the five classifiers (animal, plant, fungi, GN bacteria and GP bacteria) use different machine-learned feature sets. **Fig. 2** shows the features that were actually used by the GP bacteria classifier to classify the demonstration sequence (*EXOA_STRPN*). They appear under the heading "Relevant Tokens for Protein #6". For example, the features *ipr003034* and *polymorphism* appear in the "Unique Tokens" list, but are not used by the classifier, so they are not in the "Relevant Tokens" list.

PA uses a Naïve Bayes classifier, which generates a probability for each label. Fig. 3 shows the probabilities of each of the GP bacteria labels for the demonstration sequence (*EXOA_STRPN*) as shown in PA.

Sequences producing significant alignments:		Score (bits)	E Value
sp|P37454|EXOA_BACSU	Exodeoxyribonuclease	196	4e-50
sp|P45951|ARP_ARATH	Apurinic endonuclease-redox..	183	3e-46
sp|P27695|APE1_HUMAN	DNA-(apurinic or apyrimidi..	167	2e-41

Fig. 1. The Swiss-Prot homologs of *EXOA_STRPN*.

Psi-blast Output

Unique Tokens Extracted for Protein #6:

ipr003034, ipr000097, lyase, nuclear, nuclease, cytoplasmic, nuclear protein, polymorphism, exonuclease, ipr005135, dna repair, ipr004808, hydrolase,

Relevant Tokens for Protein #6:

lyase, nuclease, cytoplasmic, nuclear protein, ipr005135, hydrolase,

Fig. 2. The features for *EXOA_STRPN*.

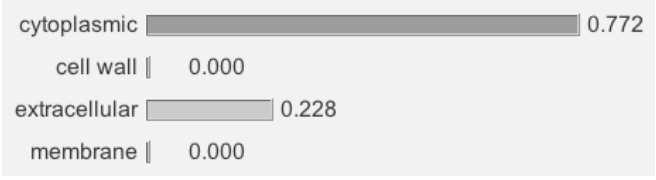


Fig. 3. Predicted subcellular locations for *EXOA_STRPN*.

Table 2. Confusion matrix for the PA Gram-positive classifier, trained on Swiss-Prot data. The ontological labels are: cyt(oplasmic), (cell) wal(l), mem(brane), and ext(racellular). N.P. represents no prediction, ASum and PSum are the sums of the actual and predicted labels, respectively. cov is sequence coverage. The superscripts: TP – true positive, TN – true negative, FP – false positive and FN – false negative are relative to the mem(brane) label and are used in the text for illustration, along with the bolded entries. $\overline{R}$, $\overline{P}$ and $\overline{S}$ denote overall sensitivity (recall), precision and specificity respectively.

	predicted label →							
actual↓	cyt	wal	**mem**	ext	N.P.	ASum	cov	sensitivity
cyt	881^{TN}	0^{TN}	13^{FP}	26^{TN}	10^{TN}	930	.989	.947
wal	1^{TN}	16^{TN}	0^{FP}	1^{TN}	1^{TN}	19	.947	.842
mem	8^{FN}	1^{FN}	$\mathbf{291}^{TP}$	17^{FN}	23^{FN}	**340**	**.932**	**.856**
ext	4^{TN}	2^{TN}	8^{FP}	217^{TN}	21^{TN}	252	.917	.861
PSum	894	19	**312**	261	55	1541	.964	$\overline{R}$=**.912**
precision	.985	.842	**.933**	.831				$\overline{P}$=**.945**
specificity	.979	.998	.983	.966				$\overline{S}$=**.978**

Building a Classifier

A classifier must be trained (built) before it can be used. PA uses labeled training data to build a simple Naïve Bayes classifier using these basic steps:

B1. Each labeled training instance consists of a primary sequence and a label from the ontology of the classifier being built.

B2. The primary sequence of each training instance is run through steps P1 and P2 described in the previous Section to produce a set of potential features.

B3. A set of *sufficient statistics*, c^{+}_{ij} and c^{-}_{ij}, are computed for the set of training instances, where c^{+}_{ij} is the number of training sequences that were labeled by label j with $F_i = true$, and c^{-}_{ij}, is the number of training sequences that were labeled by label j with $F_i = false$.

B4. A Naïve Bayes classifier is built using these sufficient statistics.

In fact, as mentioned earlier, we modify this basic process by using feature selection (Kohavi and John, 1997) to improve the accuracy. After building and computing the accuracy using all of the potential features, we remove the 5% of the features that have the lowest information content. The *information content* (*information gain*) of a feature is a measure of the amount that a feature contributes to classifications in general (Mitchell, 1997). For example, if a feature appears in every training instance, it is useless in discriminating between labels and its information content is zero. On the other hand, a feature that appears in all training instances that have a single label and no training instances with any other labels is very good for discriminating the one label. Therefore, it has high information content. After removing this 5% of low information content features, we build a second classifier, and measure its accuracy. We then remove another 5% of low information content features and continue in this way until we have computed the accuracy of 20 different classifiers with 0%, 5%, 10%, … 95% of the original features removed. We identify the threshold that produced the classifier with the highest accuracy. The most accurate classifiers for subcellular localization typically had 75%-80% of the least discriminating features removed.

Classifier Evaluation

To compare classifiers, it is important to define the evaluation criteria precisely. Most techniques start with a confusion matrix or contingency table (van Rijsbergen 1979). Table 2 shows the confusion matrix for the PA GP bacteria classifier trained on Swiss-Prot data.

We will use Table 2 to illustrate our evaluation techniques. Each entry in Table 2 represents the number of sequences in the test set whose actual label is the row label and whose predicted label is the column label. For example, the number of sequences with actual label mem(brane) that were incorrectly predicted as ext(racellular) is 17. The $ASum$ column indicates the number of test sequences whose actual label is specified by the row label. For example, 340 sequences were actually labeled mem(brane). The *PSum* row indicates the number of test sequences whose predicted label is specified by the column label. For example, 312 sequences had predicted label, membrane.

Various statistics can be computed from a confusion matrix to evaluate a classifier. In this paper we will use four standard statistics: specificity, precision, sensitivity and recall (the last two are identical). Given a confusion matrix M and a set of labels $\{L_i\}$, the standard definitions (van Rijsbergen, 1979) (Altman and Bland, 1994) of these statistics are as follows.

The *precision* for each label L_i is P_i defined by:

$$P_i = \frac{TP}{TP + FP} = M_{ii} \Big/ \sum_{k=1}^{n} M_{ki} = M_{ii} / PSum_i$$

Here, *true positives* (TP) is the number of labels correctly predicted as L_i which were actually labeled L_i. The *false positives* (FP) is the number of labels incorrectly predicted as L_i that were actually not labeled as L_i. For example, consider the label mem(brane) in Table 2. The TP and FP counts are denoted by superscripts, where there is a single count for TP, but the three FP entries must be summed. From Table 2, we have $TP = 291$ and $FP = 13+0+8 = 21$. Therefore, the precision for membrane is: $P_{(mem)} = 291/(291+21) = 291/312 = .933$.

The *specificity* for each label L_i is S_i defined by:

$$S_i = \frac{TN}{TN + FP} = \frac{sum\_ASum_i - PSum_i + M_{ii}}{sum\_ASum_i}$$

Here, *true negatives* (TN) is the number of labels correctly predicted as not L_i, that were actually not labeled L_i and *sum*

is the total number of sequences (1541 in Table 2). For example, in Table 2, the TN and FP counts for the label mem(brane) are denoted by superscripts, where the superscripted numbers must be summed. We have $TN = 881+0+26+10+1+16+1+1+4+2+217+21 = 1180$ and $FP = 13+0+8 = 21$. The specificity of label mem(brane) is: $S_{(mem)} = 1180/(1180+21) = 1180/1201 = .983$.

The *sensitivity* or *recall* for each label L_i is R_i defined by:

$$R_i = \frac{TP}{TP + FN} = M_{ii} \Big/ \sum_{j=1}^{n+1} M_{ij} = M_{ii} / ASum_i$$

Here, *false negatives* (FN) is the number of labels incorrectly predicted as not L_i that were actually labeled L_i. For example, consider the label mem(brane) in Table 2. The TP and FN counts are denoted by superscripts, where the FN superscripted numbers must be summed. From Table 2, we have $TP = 291$ and $FN = 8+1+17+23 = 49$. Note that the FN number includes the no prediction (N.P.) column as well. Therefore, the sensitivity (recall) of label mem(brane) is: $R_{(mem)} = 291/(291+49) = 291/340 = .856$.

The precision and specificity statistics favor conservative predictors that make no prediction when there is doubt about the correctness of a prediction, while the sensitivity (recall) statistic favors liberal predictors that make a prediction if there is a chance of success. For example, if two predictions are changed from "no prediction" to a prediction, where one is correct and the other is incorrect, then TP increases by 1, FP increases by 1, TN decreases by 1 and FN decreases by 1. Therefore, the precision and specificity numbers both decrease, but the sensitivity (recall) increases:

$$\hat{P}_i = \frac{TP+1}{TP+1+FP+1} < \frac{TP}{TP+FP}$$

$$\hat{S}_i = \frac{TN-1}{TN-1+FP+1} < \frac{TN}{TN+FP}$$

$$\hat{R}_i = \frac{TP+1}{TP+1+FN-1} > \frac{TP}{TP+FN}$$

Information retrieval papers report precision and recall, while bioinformatics, medical and machine learning papers tend to report specificity and sensitivity. We include all of them. However, specificity is not as informative as precision for multi-labeled (non-binary) classifiers. We also include sequence coverage, which is the ratio of sequences for which a prediction was made to the total number of sequences in a specific class. For example, in Table 2, the coverage of mem(brane) is (340 – 23) / 340 = .932.

An overall version of each statistic is computed as a weighted average. For the overall sensitivity (recall) the weights are the number of sequences with each actual label ($ASum_i$), and we also refer to it as the *accuracy*, *A:*

$$A = \overline{R} = \frac{\sum_{i=1}^{n} ASum_i R_i}{sum} = \frac{\sum_{i=1}^{n} ASum_i \left(\frac{M_{ii}}{ASum_i} \right)}{sum} = \frac{\sum_{i=1}^{n} M_{ii}}{sum}$$

For example, in Table 2, the overall sensitivity (accuracy) is $A = \overline{R} = (881+16+291+217)/1541 = .912$.

The overall precision and overall specificity are weighted averages over the predicted labels (columns):

$$\overline{P} = \frac{\sum_{i=1}^{n} PSum_i P_i}{sum \sum PSum_{n+1}} = \frac{\sum_{i=1}^{n} M_{ii}}{sum \sum PSum_{n+1}}$$

$$\overline{S} = \frac{\sum_{i=1}^{n} PSum_i S_i}{sum \sum PSum_{n+1}}$$

For example, the overall precision and overall specificity of the classifier in Table 2 are $\overline{P} = (881+16+291+217)/(1541-55) = .945$ and $\overline{S} = .978$ respectively. The overall coverage is a weighted average of the label coverage, so $\overline{C} = .964$.

There are many different ways to organize test sets and we compute two different kinds of confusion matrices. First, we use a standard machine learning technique called *5-fold cross validation* (Mitchell, 1997). Each set of labeled training instances is "randomly" divided into five groups (G_1 ... G_5), while keeping the number of training instances with each label approximately the same in each training group. Then, five different classifiers are constructed (C_1 ... C_5), where C_i uses all of the training instances from all of the groups except G_i. Next, a confusion matrix is computed for each of the five classifiers, C_i using the sequences in group G_i (that were not used in its training) as test data. The final confusion matrix is then computed by summing the entries in all of the confusion matrices. In our application, there is one important modification that is necessary to ensure "fairness" of the evaluation. Our features are obtained by extracting them from Swiss-Prot homologs. Before searching for homologs, we remove the Swiss-Prot entries of each of the test sequences. This simulates the situation where the test sequences correspond to newly sequenced proteins that would not appear in the Swiss-Prot database. We used the 5-fold cross-validation accuracy to build the feature selection filter described in the previous section. A second technique for computing a confusion matrix is to build a single classifier from all training data except the sequences from one specific organism. This *1-organism classifier* is then applied to the specific organism and a confusion matrix is constructed. This simulates the situation in which a classifier is used to predict the subcellular locations of all sequences in a newly sequenced organism. In this case, for fairness, all Swiss-Prot entries for that specific organism are removed from the Swiss-Prot database.

After the evaluation is complete, we build a final classifier using all of the training instances. This final classifier typically has better accuracy than any of the five classifiers build during 5-fold cross-validation.

Table 3. Statistics for the PA animal classifier: count, spec(ificity), prec(ision) and sens(itivity), as well as the 1-organism statistics for *Bos taurus* (*Bovine*). The ontological labels are: nuc(lear), end(oplasmic reticulum), gol(gi), mit(ochondria), pe(ro)x(isomal), lys(osomal), cyt(oplasmic), mem(brane), and (ext)racellular.

location	5-fold cross-validate				1-organism: BOVIN			
	count	spec	prec	sens	count	spec	prec	sens
nuc	2846	.996	.979	.905	47	1.000	1.000	.894
mit	1194	.998	.973	.970	145	.993	.972	.952
cyt	1845	.981	.866	.919	84	.983	.878	.940
ext	3943	.991	.972	.927	197	.991	.974	.964
gol	167	.996	.723	.892	7	.996	.667	.857
pex	103	.999	.909	.971	4	.999	.800	1.000
end	457	.996	.868	.952	14	.996	.824	1.000
lys	170	.998	.861	.947	12	.997	.857	1.000
mem	4820	.981	.957	.938	218	.986	.966	.917
Overall	15549	.988	.946	.929	728	.990	.950	.941

Table 4. Statistics for the PA green plant classifier. See Table 3 for abbreviations. Additional labels are chl(oroplast) and vac(uole). The 1-organsim is *Zea mays*.

location	5-fold cross-validate				1-organism: MAIZE			
	count	spec	prec	sens	count	spec	prec	sens
nuc	168	.999	.988	.964	16	1.000	1.000	1.000
mit	307	.992	.926	.935	19	.986	.900	.947
cyt	447	.987	.923	.960	36	.992	.971	.917
ext	127	.996	.887	.866	6	.981	.667	1.000
gol	35	.998	.850	.971	2	1.000	1.000	1.000
chl	1899	.973	.980	.959	69	.979	.969	.913
pex	29	.999	.993	.966	1	.994	.500	1.000
end	64	.998	.903	.875	6	1.000	1.000	1.000
vac	82	.997	.870	.817	2	.994	.667	1.000
mem	135	.992	.805	.733	9	.987	.600	.333
Overall	3293	.982	.951	.939	728	.987	.926	.904

Table 5. Statistics for the PA fungi classifier. See Table 3 and Table 4 for abbreviations. The 1-organism is *Neurospora crassa*.

location	5-fold cross-validate				1-organism: NEUCR			
	count	spec	prec	sens	count	spec	prec	sens
nuc	621	.975	.933	.833	11	1.000	1.000	1.000
mit	406	.977	.888	.744	45	.976	.976	.889
cyt	395	.949	.786	.808	15	.958	.824	.933
ext	171	.993	.914	.871	2	1.000	1.000	1.000
gol	52	.991	.689	.808	0	1.000	-	-
pex	64	.993	.786	.859	0	1.000	-	-
end	64	.993	.750	.656	1	1.000	1.000	1.000
mem	302	.989	.932	.861	12	.987	.917	.917
vac	19	.996	.600	.632	1	1.000	1.000	1.000
Overall	2094	.975	.871	.811	87	.978	.940	.908

Table 6. Statistics for the PA Gram-negative bacteria classifier. See Table 2 for abbreviations. Additional ontological labels are: inn(ner membrane), per(iplasmic), (cell) wal(l) and out(er membrane). The 1-organism is *Haemophilus influenzae*.

location	5-fold cross-validate				1-organism: HAEIN			
	count	spec	prec	sens	count	spec	prec	sens
cyt	1861	.989	.992	.955	73	1.000	1.000	1.000
ext	253	.986	.838	.858	15	.990	.929	.867
per	385	.986	.898	.873	7	.991	.875	1.000
inn	432	.993	.958	.951	5	1.000	1.000	.800
wal	46	.999	.956	.935	0	.983	-	-
out	197	.996	.938	.919	15	1.000	1.000	.800
Overall	3174	.990	.959	.934	115	.990	.964	.922

Table 7. Statistics for the PA Gram-positive bacteria classifier. See Table 3 and Table 6 for abbreviations. The 1-organism is *Streptomyces coelicolor*.

location	5-fold cross-validate				1-organism: STRCO			
	count	spec	prec	sens	count	spec	prec	sens
cyt	930	.982	.988	.948	37	1.000	1.000	1.000
wal	19	.997	.750	.789	0	-	-	-
ext	252	.967	.841	.881	6	1.000	1.000	1.000
mem	340	.982	.929	.853	9	1.000	1.000	.889
Overall	1541	.980	.946	.914	52	1.000	1.000	.981

RESULTS

Proteome Analyst Accuracy

Table 3 to Table 7 show the statistics for the five classifiers we built using training instances from the Swiss-Prot database. The training sets are publicly available (PA-SUB, 2003), along with the confusion matrices that were used to compute these statistics. Each training set contains a set of sequences in FastA format that includes the correct label (from Swiss-Prot), the organism tag, the organism name, Swiss-Prot taxonomy information and the primary sequence.

These classifiers show excellent 5-fold cross validation and 1-organism statistics over all ontological classes. However, some small training and test sets produce poor results, such as the precision (.600) for the 19 training/test instances of vacuolar in the fungi classifier.

We performed additional experiments to compare our work with similar systems. To compare PA to LOCkey (Nair and Rost, 2002), we constructed two custom subcellular localization classifiers using their ontology and training data. The LOCkey paper contains a confusion matrix for a Swiss-Prot data-set with 1162 training instances. Table 8 shows the five-fold cross-validation prec(ision), rec(all) and acc(uracy) computed from their confusion matrix and from a PA classifier we built using their training data and ontology.

Table 8. A comparison of the statistics of a PA classifier built using the LOCkey 1161 sequence training data with the statistics produced by the LOC(Key) classifier on the their training data. See Table 3 and Table 4 for the abbreviations.

location	count	specificity		precision		sensitivity	
		LOC	PA	LOC	PA	LOC	PA
mit	190	.945	.993	.763	.964	.795	.979
ext	334	.947	.975	.879	.937	.953	.973
nuc	352	.926	.985	.850	.965	.971	.929
chl	94	.979	.997	.718	.966	.609	.894
cyt	136	.970	.973	.656	.804	.428	.846
end	14	.993	1.000	.200	1.000	.154	.500
lys	7	.999	.999	0.000	.833	.000	.714
gol	22	.998	.999	.895	.944	.810	.773
pex	8	1.000	1.000	0.000	1.000	0.000	.375
vac	4	1.000	1.000	0.000	1.000	0.000	.250
Overall	1161	.945	.983	.815	.936	.815	.912

Our accuracy results are consistently better than the LOCkey results, except for the sensitivity on the golgi class. Our accuracy (overall sensitivity) is almost 10% better at .912 versus .815. Even though our approaches are similar, there are two reasons for these accuracy differences. First, we are using a different classifier technology – Naïve Bayes versus an ad-hoc method. Second, we are using different Swiss-Prot database fields (including the IPR field). Their paper does not include a confusion matrix or accuracy statistics, for 100% coverage of a larger 3146 sequence set, other than to indicate that the accuracy is less than the .815 accuracy of their 34% coverage classifier. On this larger set (100% coverage), we achieved an accuracy (overall sensitivity) of .889 (PA-SUB, 2003).

We also built a custom classifier for GN bacteria using the reliable PSORT-B GN bacteria data (Gardy *et al.*, 2003) as a training set. Table 9 shows the five-fold cross-validation prec(ision), rec(all) and acc(uracy) presented in their paper and the same statistics computed from a PA classifier built using the PSORT-B training data and ontology. They do not report specificity, so it is not in Table 9. Note that our Swiss-Prot GN bacteria ontology has one extra label, (cell) wal(l), which they include in the ext(racellular) class. To compare our technique more directly with theirs, we did not include a (cell) wal(l) label in the classifier we built from their data. The PA approach is very different than the PSORT-B approach, since PA uses a simple Naïve Bayes classifier and features extracted from Swiss-Prot homologs, while PSORT-B uses a set of six sequence-based models. Nevertheless, PA produces results that are somewhat better for sensitivity and accuracy, and very close in precision. Furthermore, the PA technique produces excellent results for animals, plants, fungi and GP bacteria (with different classifiers of course).

Table 9. A comparison of the statistics of a PA classifier built using the PSORT-B training data with the statistics produced by the PSORT-B predictor, built from the same training data. See Table 5 for the ontological label abbreviations.

location	count	precision		sensitivity	
		PSORT-B	PA	PSORT-B	PA
cyt	252	.976	.947	.694	.853
inn	308	.967	.965	.787	.906
per	264	.919	.915	.576	.860
out	378	.988	.986	.903	.947
ext	241	.944	.876	.700	.880
Overall	1443	.965	.943	.748	.895

Note that 139/1443 training sequences in the PSORT-B training data have two labels. To accommodate double-labels in our Naïve Bayes classifier, we transformed each training instance that had two labels into two training instances, one with each label. Since we are comparing with the PSORT-B classifier (Gardy *et al.*, 2003), we followed their lead during predictor evaluation and counted a prediction as correct if it predicted either of the two labels. Note that subcellular location predictions were made on all of the 1443 sequences so the coverage is 100%.

As a final test, we applied our full Swiss-Prot trained GN bacteria classifier to the PSORT-B test set and obtained an accuracy of .869 (Lu, 2003) (PA-SUB, 2003).

Sequence Coverage

If PA is applied to an entire organism, there will be some sequences without homologs, so no features can be extracted and used by the classifier. In some cases, even though homologs are found, there will be no relevant tokens in the FUNCTION, IPR and SCELL fields used by PA to construct features. We call such sequences *excluded sequences* and PA makes no subcellular localization prediction for excluded sequences. Excluded sequences are the only ones that reduce the coverage of PA classifiers. To gain an appreciation for the PA subcellular localization sequence coverage on various organisms, we used the PA classifiers to classify all of the sequences in several organisms as shown in Table 10. A more complete table is online (PA-SUB, 2003).

Before running PA on an organism, we removed all of the sequences for that organism from Swiss-Prot, so that no exact sequence matches would be found. Of course, for these tests, we cannot report accuracy, since we do not know the "correct" subcellular localization for many of them. The organisms use the animal, plant, fungi, GN bacteria and GP bacteria classifiers respectively. Each was selected since its complete proteome is publicly available. We are currently developing pattern recognition and discovery software that can be used to extract local features from excluded sequences so that the coverage may approach 100%.

Table 10. Sequence coverage of the PA classifiers on some fully sequenced organisms. The count is the total number of genetic sequences for the organism. An exclude(d) sequence is one for which PA was unable to find at least one homolog whose E-value was less than .001 that contained at least one relevant feature. The cov(erage) is the ratio of all non-excluded sequences from an organism to the total number of sequences from that organism.

organism	class	count	exclude	cov
M. musculus	animal	27754	7099	.745
A. thaliana	plant	26032	10043	.600
S. pombe	fungi	5007	1023	.787
B. subtilis	GP bact.	4098	1346	.672
P. aeruginosa	GN bact.	5557	1355	.756

Table 11 The accuracy of PA Gram-negative classifiers that use different homolog selection techniques, different Swiss-Prot fields and different feature extraction techniques.

PSI-BLAST iterations	top homologs	KWORD	IPR	SCELL	acc
1	**3**	**phrase**	**yes**	**phrase**	**.934**
1	3	phrase	yes	no	.924
1	3	phrase	no	phrase	.934
1	3	no	no	phrase	.922
2	3	phrase	yes	phrase	.932
1	2	phrase	yes	phrase	.935
1	4	phrase	yes	phrase	.936
1	3	words	yes	words	.929

DISCUSSION

Extracting Ontological Labels for Training Sequences

We selected all mature sequences (≥ 40 amino acids) from the Swiss-Prot database and tried to extract their ontological labels. Although the Swiss-Prot database contains a subcellular localization field, this field does not contain a single ontological label for each sequence. Therefore, we had to construct a parser that extracted a simple ontological label, when possible. Here are the rules that our parser uses to label potential training sequences:

1) See if the field contains one of the ontological labels. If it does not, the sequence is rejected as a training sequence.
2) If it contains more than one ontological label, it is also rejected, unless one label is an organelle and the other is membrane.
3) If it contains an ontological label, but also contains the phrase "potential" or "by similarity" it is rejected if the number of training sequences with that label is high. However, if the number of training sequences with that label is small (< 1.5% of the total number of training instances), it is accepted.
4) If the ontological label is "cell wall" and the phrase contains the word "attached", it is rejected.

Steps 2), 3) and 4) require some explanation. For step 2) it is common to describe a protein as being in a membrane of a specific organelle. In this case, the correct label is the organelle. In Step 3), we want to reject any annotations that contain words like "potential " or "by similarity". However, for ontological labels with low numbers of training instances, we found that accepting "higher risk" annotations is necessary to obtain enough training data so that the classifiers have good accuracy. Note that we followed the PSORT-B lead of including any sequences that contain the phrase "cell wall" in the extracellular class for the plant and fungi ontologies, since the Swiss-Prot data is not very accurate in these cases. For Step 4), we found many Swiss-Prot SCELL annotations for proteins that are not in the cell wall, which contain the phrase "attached to the cell wall".

Selecting Homologs and Extracting Features

We experimented with many different implementations of the five-step prediction process described earlier in this paper. For step 1, we used PSI-BLAST instead of BLAST and varied the number of iterations. Second, we varied the number of homologs whose features were extracted. The highest accuracies were obtained by using one iteration of PSI-BLAST (so we reverted to BLAST). There is not much difference between using the top two, three or four homologs (whose E-values were smaller than 0.001), so we decided to pick three, while we investigate this further.

For step 2, we varied the Swiss-Prot fields that we used to extract features. We used combinations of the KEYWORD field (KWORD), the InterPro numbers from the DBSOURCE field (IPR), and the SUBCELLULAR LOCALIZATION subfield of the COMMENT field (SCELL). We also varied the way we parsed the fields to extract features. For example, we tried stemming (Jurafsky and Martin, 2000) on the KWORD field so that the words: vacuole and vacuoles are the same. We also tried treating semi-colon delimited phrases like: "Purine biosynthesis" as a single feature versus two separate features in the KWORD field. The best results were obtained by using semi-colon delimited phrases without stemming. For the SCELL, we tried using all individual words as features and we tried using a fixed set of pre-defined phrases (PA-SUB, 2003). The pre-defined phrase approach worked the best. Table 11 shows accuracy results for some of our experiments.

Notice from Table 11 that using the SCELL, IPR and KWORD fields of the Swiss-Prot database gives the best prediction results, although the IPR field is the least important for predicting subcellular localization. Therefore, the better accuracy of PA compared to LOCkey cannot be

attributed only the inclusion of the IPR field. The NB classifier accounts for most of the improvement over LOCkey. However, there is hope that IPR numbers may be useful for some localizations. A domain projection technique based on SMART domains (Schultz *et al.*, 2000), which are included in Interpro, has been somewhat successful in identifying the labels: extracellular, cytoplasmic and nuclear (Mott *et al.*, 2002). In addition, we have found that using the IPR number is very significant for general function prediction and other specialized predictors we have constructed using PA, like K+-ion channel protein classification (Szafron *et al.*, 2003b).

Selecting a Classifier Technology

For step 3), we varied the kinds of classifiers. Table 12 shows a summary of results (PA-SUB, 2003) for Naïve Bayes (NB), Artificial Neural Nets (ANN), Support Vector Machines (SVM) and three nearest neighbor classifiers (1NN, 3NN and 5NN). For a *k*-nearest neighbor predictor, after *BLAST*ing for homologs, we ignored all Swiss-Prot fields except for the SCELL field. The *k* homologs with the smallest E-values (< 0.001), that had a non-empty SCELL field voted for a subcellular localization label, based on their own field label. In the case of a tie, the homolog with the smallest *BLAST* E-value won.

As shown in Table 12, the NB accuracy is better than any of the k-nearest neighbor classifiers, but is inferior to the ANN and SVM classifiers by 1% - 4 %. However, it is very difficult to explain the predictions of ANN and SVM classifiers, so we feel that this small decrease in accuracy is more than compensated by the ability to explain the predictions to users. Explanation is an important factor in getting users to trust predictors (Szafron *et al.*, 2003a).

The explain mechanism of PA allows users to review the evidence used by a classifier to make a prediction. For example, both the NB and ANN classifiers predict that the Gram-negative protein *OMP1_CHLMU* is an outer membrane protein, even though Swiss-Prot 41 SCELL entry is CELL WALL SURFACE. However, the PA explain mechanism for NB classifiers allow the user to view the evidence, while there is no way to do this in an ANN classifier. Fig. 4 shows part of an explain page for the *OMP1_CHLMU* classification. Each horizontal bar represents the evidence for a particular location on a logarithmic scale. Each sub-bar with different shading indicates the evidence due to the existence of a single feature (porin, outer membrane, ipr000604, integral membrane protein, and transmembrane). In PA, these sub-bars are different colors, but have been represented by different shadings in this paper. The long white bar represents the accumulated evidence of the other features that are not currently displayed (reduced residual).

Table 12. A comparison of the accuracy of Naïve Bayes (NB), artificial neural nets (ANN), Support Vector Machines (SVM), and three nearest neighbor classifiers (1NN, 3NN and 5NN) on the five Swiss-Prot datasets, the LOCkey dataset and the PSORT-B dataset.

Category	NB	ANN	SVM	1NN	2NN	3NN
animal	.929	.883	**.956**	.910	.919	.919
plant	.939	**.971**	.947	.900	.912	.911
fungi	.811	**.856**	.814	.726	.772	.752
GP bact	.914	**.949**	.898	.812	.845	.843
GN bact	.934	**.956**	.939	.868	.899	.892
LOCkey	.912	**.943**	.924	.720	.763	.768
PSORT-B	.895	**.927**	.888	.615	.652	.653

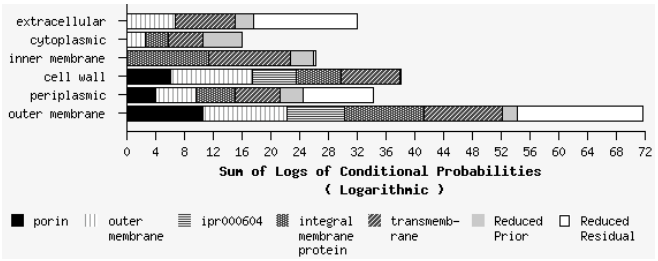


Fig. 4 Part of the PA explain page for protein *OMP1_CHLMU*.

PA contains a mechanism for changing the five features that are displayed and the remaining features that are combined into the white bar (reduced residual). Notice that evidence for label outer membrane over cell wall is overwhelming. Even though a PA-NB classifier and a PA-ANN classifier both predicted outer membrane, the advantage of using an NB classifier instead of an ANN classifier is the existence of this explanation facility. Note that in the revised Swiss-Prot version 42 database that will be released in September 2003, the SCELL entry of this protein is changed to outer membrane to match the PA prediction. Although the explanation mechanism of PA was not used to influence this annotation change, it could have been used in this way.

A complete description of the PA explanation facility is beyond the scope of this paper. However, Fig. 5 shows one more PA screen that can be used to view prediction evidence. This screen shows relative evidence from the most important features, in selecting between the predicted class (in this case, outer membrane) and any other class of interest (in this case, cell wall). The darker bars indicate evidence for outer membrane and the lighter bars indicate evidence for cell wall. The (P) notation indicates that a token for that feature was present in the query sequence (*OMP1_CHLMU*) and an (A) indicates that the token for that feature was absent. We believe that the convenience and power of the PA Naïve Bayes explanation facility is worth the loss of a few percentage points of precision accuracy that might be gained by using an ANN or SVM classifier.

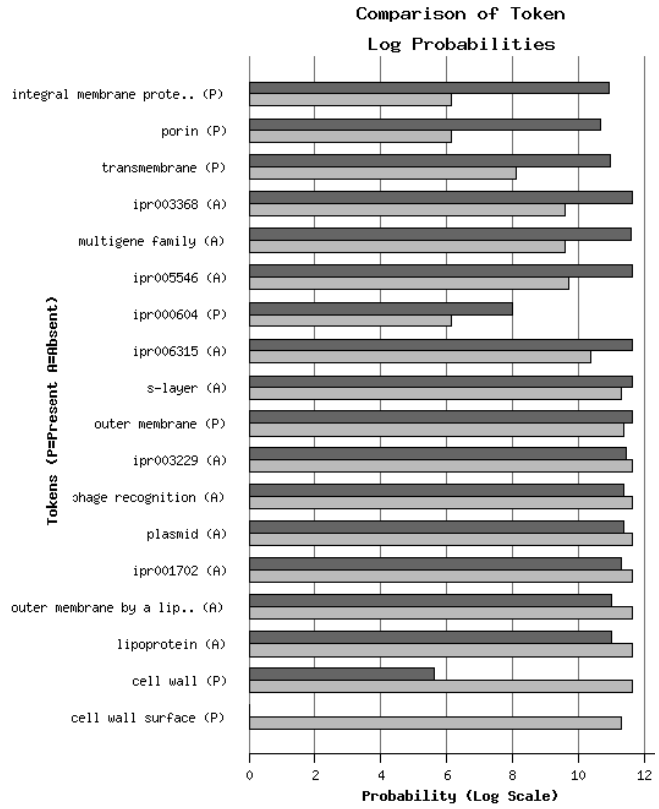


Fig. 5 Viewing feature contributions to a PA prediction.

It is also possible to use multiple classifier technologies and to report a consensus, although we are not currently using this approach. It is not clear how the explanation facility would fit with such an approach.

ACKNOWLEDGEMENTS

Thanks to the referees for several helpful suggestions. Thanks to Cynthia Luk, Samer Nassar and Kevin McKee for their contributions to the first prototype of PA. Thanks to Warren Gallin and Kathy Magor for their valuable feedback while using early versions of PA. Thanks to Rajesh Nair and Burkhard Rost, for providing us with their training data. Finally, a big thanks to Fiona Brinkman and Jennifer Gardy, for not only providing us with their Gram-negative bacteria training data, but also for many helpful pointers and ideas. This research was partially funded by research or equipment grants from the Protein Engineering Network of Centres of Excellence (PENCE), the National Science and Engineering Research Council (NSERC), Sun Microsystems and the Alberta Ingenuity Centre for Machine Learning (AICML).

REFERENCES

Altman,D.G. and Bland,J.M. (1994) Statistics notes: diagnostic tests 1: sensitivity and specificity. *BMJ*, **308**, 1552.

Duda,R.O. and Hart,P.E. (1973) *Pattern Classification and Scene Analysis*. John Wiley & Sons.

EBI (2003) European Bioinformatics Institute, http://www.ebi.ac.uk/genomes/

Emanuelsson,O. (2002) Predicting protein subcelluar localization from amino acid sequence information. *Briefings in Bioinformatics*, **3**, 361-376.

Emanuelsson,O., Nielson,H., Brunak,S. and von Heijne,G. (2000) Predicting subcellular localization of proteins based on their N-terminal amino acid sequence, *J. Mol. Biol*, **300**, 1005-1016.

Gardy,J.L., Spencer,C., Wang,K., Ester,M., Tusnády,G.E., Simon,I., Hua,S., deFays,K., Lambert,C., Nakai,K. and Brinkman, F.S.L. (2003) PSORT-B: Improving protein subcellular localization prediction for Gram-negative bacteria. *Nucleic Acids Res*. to appear.

Hua,S. and Sun,Z. (2001) Support vector machine approach for protein subcellular localization prediction. *Bioinformatics*, **17**, 721-728. http://www.bioinfo.tsinghua.edu.cn/SubLoc/

Horton,P. and Nakai,K. (1997) Better prediction of protein cellular localization sites with the *k* nearest neighbors classifier. Proc. of the Fifth ISMB, AAAI Press, 298-305. http://psort.nibb.ac.jp/

Jurafsky,D. and Martin,J.H. (2000) *Speech and Language Processing*. Prentice-Hall.

Kohavi,R. and John,G.H. (1997) Wrappers for feature subset selection.. *Artificial Intelligence*, **97**, 273-324.

Krogh,A., Larsson,B., von Hejne,G. and Sonnhammer, E. (2001) Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. *J. Mol. Biol*., **305**, 567-580. http://www.cbs.dtu.dk/services/TMHMM/

Lu,Z. (2003) Predicting protein subcellular localization from homologs using machine learning algorithms. PhD thesis, Department of Computing Science, University of Alberta.

Mitchell,T.M. (1997) *Machine Learning*. McGraw-Hill, N.Y.

Mott,R., Schultz,J., Bork,P. and Ponting,C.P. (2002) Predicting protein cellular localization using a domain projection method. *Genome Res*., **12**, 1168 - 1174.

Nair,R. and Rost,B. (2002) Inferring subcellular localization through automated lexical analysis. *Bioinformatics*, **18**, S78-S86.

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

NakaiK. (2000) PSORT II Users' Manual. http://psort.nibb.ac.jp/helpwww2.html.

PA-SUB (2003) http://www.cs.ualberta.ca/~bioinfo/PA/Subcellular

Reinhardt,A. and Hubbard,T. (1998) Using neural networks for prediction of the subcellular location of proteins. *Nucleic Acids Res*., **26**, 2230-2236.

van Rijsbergen,K. (1979) *Information Retrieval*. Butterworths, London (UK). http://www.dcs.gla.ac.uk/Keith/Preface.html

Sahami,M. (1999) Using machine learning to improve information access. PhD thesis, Computer Science Department, Stanford University.

Schultz,J., Cople,R.R., Doerks,T., Ponting,C.P. and Bork,P. (2000) SMART: A web-based tool for the study of genetically mobile domains. *Nucleic Acids Res*., **28**, 231–234.

Szafron,D., Greiner,R., Lu,P., Wishart,D., MacDonell,C., Anvik,J., Poulin,B., Lu,Z. and Eisner,R. (2003a) Explaining naïve Bayes classifications, TR03-09, Department of Computing Science, University of Alberta.

Szafron,D., Lu,P., Greiner,R., Wishart,D., Lu,Z., Poulin,B., Eisner,R, Anvik,J. and MacDonell,C. (2003b) Proteome Analyst – transparent high-throughput protein annotation: function, localization and custom predictors, International Conference on Machine Learning Workshop on Machine Learning in Bioinformatics (ICML-Bioinformatics), to appear.