Nucleic Acids Research, 2004, Vol. 32, Web Server issue ***W365–W371***
DOI: 10.1093/nar/gkh485

Proteome Analyst: custom predictions with explanations in a web-based tool for high-throughput proteome annotations

Duane Szafron, Paul Lu*, Russell Greiner, David S. Wishart, Brett Poulin, Roman Eisner, Zhiyong Lu, John Anvik, Cam Macdonell, Alona Fyshe and David Meeuwis

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

Received February 15, 2004; Revised April 19, 2004; Accepted May 5, 2004

ABSTRACT

Proteome Analyst (PA) (http://www.cs.ualberta.ca/~bioinfo/PA/) is a publicly available, high-throughput, web-based system for predicting various properties of each protein in an entire proteome. Using machine-learned classifiers, PA can predict, for example, the GeneQuiz general function and Gene Ontology (GO) molecular function of a protein. In addition, PA is currently the most accurate and most comprehensive system for predicting subcellular localization, the location within a cell where a protein performs its main function. Two other capabilities of PA are notable. First, PA can create a custom classifier to predict a new property, without requiring any programming, based on labeled training data (i.e. a set of examples, each with the correct classification label) provided by a user. PA has been used to create custom classifiers for potassium-ion channel proteins and other general function ontologies. Second, PA provides a sophisticated explanation feature that shows why one prediction is chosen over another. The PA system produces a Naïve Bayes classifier, which is amenable to a graphical and interactive approach to explanations for its predictions; transparent predictions increase the user's confidence in, and understanding of, PA.

INTRODUCTION

There are now more than 1200 complete or partially sequenced genomes deposited in public databases (http://www.ebi.ac.uk/genomes/) and this number is growing rapidly. Given the size and complexity of these data sets, most researchers are compelled to use automated annotation systems to identify or classify individual genes/proteins in their genomic data. A number of systems have been developed over the past few years that permit automated genome-wide or proteome-wide annotation. These include GeneQuiz (1), GeneAtlas (2), Ensembl (3), PEDANT (4), Genotator (5), MAGPIE (6) and GAIA (7).

The Proteome Analyst (PA) system (8–10) (http://www.cs.ualberta.ca/~bioinfo/PA/) focuses on the task of predicting (classifying) various aspects of a protein. Our results show that classification can be used for many annotations, including general function, subcellular localization, specific function and many specialized predictors, such as the potassium-ion channel predictor described later in this paper.

Although there are a variety of tools for protein annotation, PA has unique capabilities. In addition to being the most accurate and most comprehensive (i.e. broadest range of organisms and organelles, highest number of proteins annotated) predictor of subcellular localization (9),

(i) PA provides a single, integrated, high-throughput and web-based interface to a number of different tools for proteome annotation;
(ii) PA allows the user to create custom predictors in a simple train-by-example way, for any user-specified ontology of labels; and
(iii) PA provides clear and transparent explanations for each of its predictions.

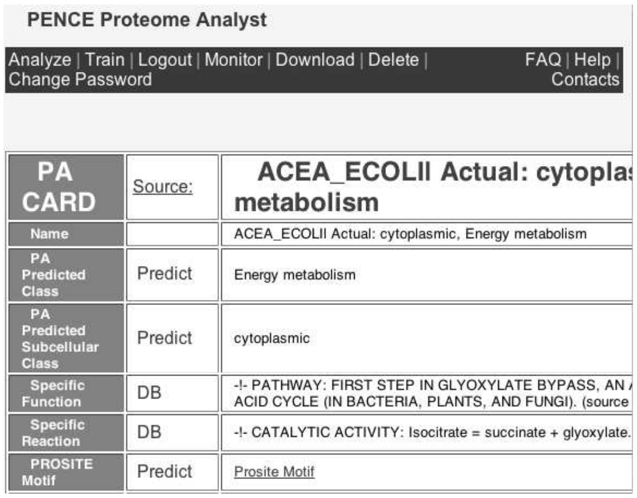
In the context of PA, transparency is the ability to provide formally sound and intuitively simple explanations for predictions. PA bases its predictions on well-understood concepts from probability theory. Its explanations use stacked-bar graphs (Figure 8) and hyperlinks to clearly display the evidence for each prediction.

USING PROTEOME ANALYST

Proteome Analyst is web based. The user may choose to either analyze (annotate) a proteome using built-in (previously

*To whom correspondence should be addressed. Tel: +1 780 492 7760; Fax: +1 780 492 1071; Email paullu@cs.ualberta.ca
Correspondence may also be addressed to Duane Szafron. Email: duane@cs.uslberta.ca

The authors wish it to be known that, in their opinion, the first two authors should be regarded as joint First Authors


PENCE Proteome Analyst
Analyze | Train | Logout | Monitor | Download | Delete | Change Password
FAQ | Help | Contacts

PA CARD	Source:	ACEA_ECOLII Actual: cytoplas metabolism
Name		ACEA_ECOLII Actual: cytoplasmic, Energy metabolism
PA Predicted Class	Predict	Energy metabolism
PA Predicted Subcellular Class	Predict	cytoplasmic
Specific Function	DB	-!- PATHWAY: FIRST STEP IN GLYOXYLATE BYPASS, AN ACID CYCLE (IN BACTERIA, PLANTS, AND FUNGI). (source
Specific Reaction	DB	-!- CATALYTIC ACTIVITY: Isocitrate = succinate + glyoxylate.
PROSITE Motif	Predict	Prosite Motif

Figure 1. The top part of a sample PACard.

trained) classifiers or train a new custom classifier, which can afterwards be used to analyze specific properties of proteins. We explain both options.

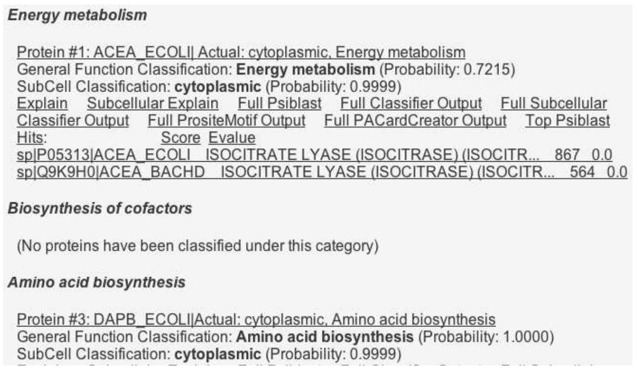
Analysis of a proteome

To analyze a proteome, the user first uploads a FASTA-format file containing the sequences to be analyzed. Two important tools are a classifier-based predictor and the PACardCreator. Currently, a user may select from several built-in general function classifiers that use the GeneQuiz (GQ) ontology and were trained on sequences from individual organisms: *Escherichia coli*, Yeast, *Drosophila* or a multiorganism-trained classifier trained with sequences from all three organisms. A Gene Ontology (GO)-based classifier for molecular function is also available in the latest version of PA. Alternatively, a user may select any custom classification-based predictor that has been trained as described below.

The PACardCreator generates a PACard for each sequence—a summary of all the predicted properties of each protein specified in the input. The top of a typical PACard is shown in Figure 1. A PACard is based on the *E.coli* cards from the CyberCell Database (CCDB) (http://redpoll.pharmacy.ualberta.ca/CCDB/).

Currently, PA can fill in over 30 different fields: Name, GeneQuiz general function, subcellular location, GeneOntology molecular function, Specific Function, Pfam Domain/Function, EC Number, Specific Reaction, General Reaction, PROSITE, BLAST, Important Sites, Inhibitor, Interacting Partners, Sequence, Secondary Structure, Metabolic Importance, Copy Number, RNA Copy No., Similarity, Number of Amino Acids, Molecular Weight, Transmembrane, Cys/Met Content, Structure Class, Quaternary Structure, Cofactors, Metals Ions, Kcat Value (1/min), Specific Activity (μmol) and Km Value (mM).

Figure 2 shows an example analysis, sorted by general function (GeneQuiz ontological class). The probability of the predicted general function class is shown for each sequence. The output also shows the predicted subcellular localization (and the probability of this prediction), the top homologs found during the BLAST search, a link to the full



Energy metabolism

Protein #1: ACEA_ECOLI| Actual: cytoplasmic, Energy metabolism
General Function Classification: **Energy metabolism** (Probability: 0.7215)
SubCell Classification: **cytoplasmic** (Probability: 0.9999)
Explain Subcellular Explain Full Psiblast Full Classifier Output Full Subcellular Classifier Output Full PrositeMotif Output Full PACardCreator Output Top Psiblast Hits:

```
Hits:                                                    Score  Evalue
sp|P05313|ACEA_ECOLI   ISOCITRATE LYASE (ISOCITRASE) (ISOCITR...   867   0.0
sp|Q9K9H0|ACEA_BACHD   ISOCITRATE LYASE (ISOCITRASE) (ISOCITR...   564   0.0
```

Biosynthesis of cofactors

(No proteins have been classified under this category)

Amino acid biosynthesis

Protein #3: DAPB_ECOLI|Actual: cytoplasmic, Amino acid biosynthesis
General Function Classification: **Amino acid biosynthesis** (Probability: 1.0000)
SubCell Classification: **cytoplasmic** (Probability: 0.9999)

Figure 2. Proteins by ontological class (partial screenshot).

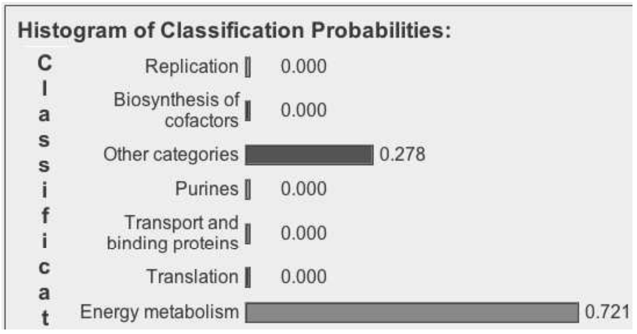


Figure 3. Full classifier output for ACEA_ECOLI (partial screenshot).

BLAST output in standard format, links to the full general function classifier output (Figure 3), the subcellular classifier output, the PACard (Figure 1) and explanations for each classifier prediction. The 'explain' facility is discussed later in the paper and is one of the most novel characteristics of PA. We believe explanations are essential for widespread acceptance of computational prediction techniques in bioinformatics.

Figure 3 shows that the predicted ontological class (Energy metabolism) of the ACEA_ECOLI protein (Protein #1 from Figure 2) has a probability of 72.1%. It also shows that the next most probable class is Other categories, with a probability of 27.8%.

Prediction techniques in PA

PA makes extensive use of machine-learning (ML) classifiers to predict annotations. PA can help the user build novel classifiers, for new annotations, by applying a standard ML algorithm to a set of labeled training items—a list of known proteins with their respective class labels (i.e. annotations). The classifier is later used to provide labels (predictions or annotations) to previously unlabeled proteins. In PA, each training item consists of a primary protein sequence and the ontological class it has been assigned by an expert.

In general, an ML classifier algorithm requires features to be associated with each training item. Note that PA is given only the primary sequence of the protein; the features are automatically computed by the system. Once built, a classifier takes a protein sequence with unknown class and uses the values of these features (i.e. the presence or absence of the associated word or phrase) to predict its class.

Specifically, PA uses a preprocessing step that maps each sequence to a set of features, as shown in Figure 4. First, the sequence is compared to the SWISS-PROT database using BLAST. Second, the SWISS-PROT entries of (up to) three top homologs (whose *E*-values are <0.001) are parsed to extract a feature set from the SWISS-PROT KEYWORDS field, any Interpro numbers (11) contained in the DBSOURCE field, and the SUBCELLULAR LOCATION field. The union of the features for the selected homologs forms the feature set. If no homologs match the *E*-value cutoff or if all features are removed by feature selection (described later) then the sequence has no features, so no prediction is made.

The feature set is then used as input for both the training and classification phases of PA (discussed below). In essence, PA learns a mapping from feature sets to classes (also known as 'annotations'). The same extraction algorithm is used to determine the prediction or annotation for each protein sequence, whether the classifier is a built-in one or a custom user-trained one.

Training a custom classifier

Since PA provides several built-in classifiers, many users will not need to build their own custom classifier. However, for user-specified ontologies or other specialized purposes, the ability to build and explain a custom classifier, without requiring any programming, is a key advantage of PA.

As shown in Figure 5, classification-based prediction is a two-step process: training/learning and prediction. In the training/learning step, a classifier is built using an ML classification algorithm by analyzing a set of training sequences, each tagged by a known class label. In the prediction step, the generated classifier is used to predict the class label of an unknown query sequence.

Of course, when building any classifier, it is necessary for the training data to satisfy two criteria. First, they must be broad enough to contain representative examples of each labeled class. Second, the training data must be relatively free from errors. Training data with narrow coverage or labeling errors cannot produce an accurate classifier using any ML technology. However, PA's explanation system can actually be used to find errors in the training data, if necessary (10). If the training data contain too many errors, PA will indicate the poor quality of the trained classifier by reporting low accuracy in the automatic validation that is done after training a classifier.

The production version of PA includes a general function (GeneQuiz ontology) classifier and a series of subcellular localization classifiers (based on organism type). However, a user can also train a Naïve Bayes (NB) custom classifier. The first step in training a custom classifier is to provide a name for the classifier and a corresponding training file in FASTA format. Each sequence in the file must have a FASTA tag that starts with a known class label. For example, Figure 6 shows part of a training file for a custom K-ion channel classifier, where the two training sequences have known class labels KV1 and KV2, respectively.

After uploading the training file, the user has a choice of two configuration parameters: feature wrapping (12) and the value of *k* for the *k*-fold cross-validation. The wrapping (feature selection) process is a standard ML technique (13). It removes the less discriminating features from the trained classifier and has the overall effect of improving accuracy by reducing overfitting. The default configuration uses wrapping.

In *k*-fold cross-validation (12), the labeled training instances are 'randomly' divided into *k* groups ($G_1, \ldots, G_k$), while keeping the number of training instances with each label approximately the same in each training group. Then, *k* different classifiers are constructed ($C_1, \ldots, C_k$), 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 *k* classifiers, C_i, using the sequences in group G_i (which were not used in its training) as test data. The confusion matrix records the number and type of classification mistakes made by the newly trained classifier (false positives, false negatives, and so on). The final confusion matrix is then computed by summing the entries in all of the confusion matrices. The PA default value of *k* is 5 (common in ML).

Once the classifier has been trained, the user may view a classifier information page (Figure 7) that contains three lists that summarize the training. The first list shows the training sequences that PA excluded (none in this example) because the BLAST search did not produce any usable features. The second list contains training sequences that are most probably labeled incorrectly, sorted from the highest to the lowest probability. The third list contains the rest of the training sequences, sorted from the highest to the lowest probability of being labeled correctly. The user can discover why PA inferred that a training sequence was labeled correctly or incorrectly by selecting an Explain hyperlink or by looking at the raw BLAST results.

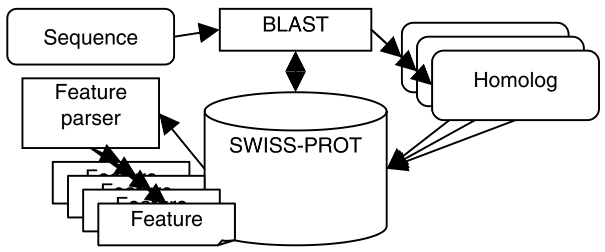


Figure 4. The feature extraction algorithm for a protein sequence in PA.

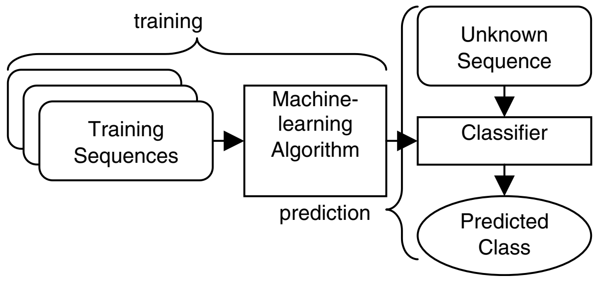
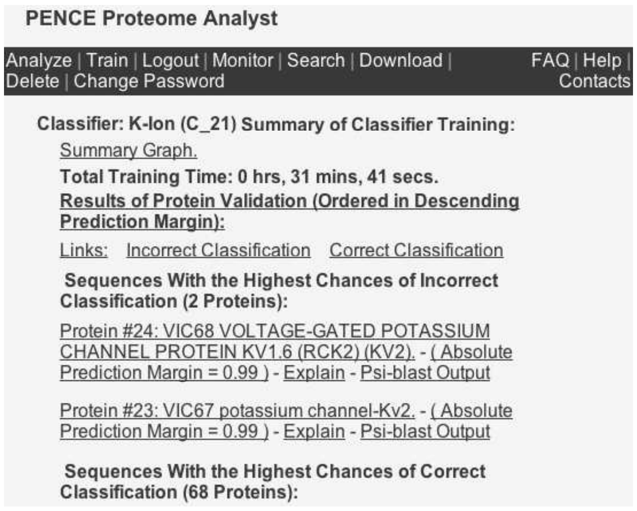


Figure 5. The training and prediction phases of classification.

```
>KV1<VIC0 potassium voltage-gated …
mtvatgdpadeaaalpghpqdtydpeadhecce …
>KV2<VIC171 potassium channel protein …
mvgqlqggqaagqqqqqqqatqqqqhskqqlqg …
```

Figure 6. The FASTA-based format of a classifier training file.



PENCE Proteome Analyst

Analyze | Train | Logout | Monitor | Search | Download | FAQ | Help | Delete | Change Password Contacts

Classifier: K-Ion (C_21) Summary of Classifier Training:

Summary Graph.

Total Training Time: 0 hrs, 31 mins, 41 secs.

Results of Protein Validation (Ordered in Descending Prediction Margin):

Links: Incorrect Classification Correct Classification

Sequences With the Highest Chances of Incorrect Classification (2 Proteins):

Protein #24: VIC68 VOLTAGE-GATED POTASSIUM CHANNEL PROTEIN KV1.6 (RCK2) (KV2). - (Absolute Prediction Margin = 0.99) - Explain - Psi-blast Output

Protein #23: VIC67 potassium channel-Kv2. - (Absolute Prediction Margin = 0.99) - Explain - Psi-blast Output

Sequences With the Highest Chances of Correct Classification (68 Proteins):

Figure 7. Information for a trained classifier (partial screenshot).

CUSTOM CLASSIFIER EXAMPLE

Voltage-gated potassium channels (VKCs) are intrinsic membrane proteins that respond to changes in the transmembrane electric field by changing shape and selectively allowing potassium ions to pass through the lipid bi-layer (14). We obtained 78 protein sequences that were divided into 4 classes (KV1, 23 sequences; KV2, 19 sequences; KV3, 17 sequences; and KV4, 19 sequences) from W. Gallin's laboratory. Many of the VKC sequences have close homologs that lie in classes other than their own class.

In a process that mirrors how users of PA might create a custom classifier, we iteratively trained a classifier, used PA's Explain (and other capabilities) to find the reasons for any inaccurate predictions on the training set itself [i.e. re-substitution or training set errors (12)], fixed the source of the inaccuracies and then trained a new classifier. After only two rounds, we were able to create a final custom classifier that is 100% accurate, over 5-fold cross-validation, with respect to the training data.

Initially, PA produced an NB classifier that made only three errors during 5-fold cross-validation. However, one error was a labeling error in the training set. After consulting with an expert and fixing the labeling error, we re-trained another classifier. The output in Figure 7 shows the two remaining errors. We eliminated these two errors by modifying the feature extraction algorithm's parameters. Originally, we performed three PSI-BLAST iterations before picking the top three homologs to use for feature extraction. We found that when there are many homologs in different ontological classes, better accuracy can be obtained by using only a single PSI-BLAST iteration. Since one iteration of PSI-BLAST is equivalent to BlastP, PA uses BlastP.

From this case study, we now understand that PSI-BLAST with multiple iterations is likely to perform poorly because multiple iterations tend to promote sequences from the most prevalent organisms in the SWISS-PROT database, at the expense of sequences from minority organisms, even though the minority sequences may have better similarity. The lesson is that when you train a classifier for predicting properties that differentiate based on small differences, a single iteration is better. After making this change, the accuracy increased to 100% on the K-ion training set. PA's explanation mechanism was key in improving the custom classifier.

Table 1. Accuracies and informal sequence/taxonomic coverage of current subcellular localization predictors

Name	Accuracies	Coverage	Technique
PSORT-B	0.75	1443 GN bacterial	Combination
LOCkey	0.87	1161 assorted	Homology
SubLoc	0.91	291 prokaryotic	AA composition
	0.79	2427 eukaryotic	
TargetP	0.85	940 plant	Signal prediction
	0.90	2738 non-plant	
Proteome analyst	0.93	16 284 animal	Homology and machine learning
	0.93	3420 plant	
	0.81	2104 fungal	
	0.92	3218 GN bacterial	
	0.94	1571 GP bacterial	

Gram-negative bacteria and Gram-positive bacteria are denoted GN and GP respectively. This table is reproduced from (9).

ACCURACY AND COVERAGE OF PA

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.

We constructed five custom 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 92–94% accurate for the other four categories (Table 1). These are the most accurate subcellular predictors across the widest set of organisms published to date (9). In a series of experiments, we showed that PA makes highly accurate subcellular localization predictions, for many different organisms (e.g. the five custom classifiers listed above), for a variety of different data sets (e.g. SWISS-PROT, LOCkey, PSORT-B) and using a variety of ML techniques. We tested Naïve Bayes, artificial neural networks (ANNs), support vector machines (SVMs), and nearest-neighbor classifiers.

PA uses Naïve Bayes classifiers, since the accuracy is always within 3–5% of the best technique for all of the classifiers we have trained (the best technique varies for different training sets, so no particular ML technique is always best) (9). Since there is very little quantitative difference in accuracy between NB and the best technique for any training set, we select NB for qualitative reasons. Specifically, as described in the next section, it is possible to transparently explain NB predictions to non-computational scientists. Since the production version of PA uses only NB, it is the only ML technique discussed in this paper.

The subcellular predictors began their lives as custom predictors in PA. However, after their success, we constructed a simple standalone web tool for predicting just subcellular localization (PA-SUB), available at (http://www.cs.ualberta.ca/∼bioinfo/PA/Sub). In addition, these predictors have also become built-in classifiers in the production version of PA (http://www.cs.ualberta.ca/~bioinfo/PA).

TRANSPARENCY AND EXPLAINABILITY

While it is necessary for a protein prediction tool to be accurate, it is also important that it can clearly explain its predictions to the user. This is important for two main reasons. First, it helps biologists to develop confidence in the tool. Second, it can help locate and correct errors that occur in the training set, the underlying database (which might give rise to incorrect predictions) or the system parameters, as with the K-ion custom classifier in PA.

Explaining a prediction/classification

PA provides an explanation mechanism to help users understand why a classifier makes a particular classification (10). It allows a user to examine the query protein itself, as well as the proteins on which the classifier was trained. The user can then examine which particular features added the most evidence to a classification. We will use the protein ACEA_ECOLI as an example. If the user clicks the Explain hyperlink of the ACEA_ECOLI protein in Figure 2, then an Explain page (Figure 8) is displayed.

Each stacked bar in the graph represents a class in the ontology, and each of its five colored sub-bars corresponds to the presence of one of five selected features in the training sequences. In fact, a sub-bar may represent the absence of a feature. However, for simplicity, in this subsection, we will assume that sub-bars mark the presence of a feature (and this is the case in Figure 8, where the features are tricarboxylic acid cycle, glyoxylate bypass, ipr000918, lyase, and phophorylation).

Each composed bar on a single line represents the logarithm (base 2) of the combined probability that the protein is in the class represented by the line. For example the lengths of the Energy metabolism and Other categories bars are $\sim$43 and 41.6 units respectively. The difference is $\sim$1.4 units, which means that the ratio of the probabilities is $\sim 2^{1.4} \approx 2.6$. From Figure 3, the ratio is actually $72.1/27.8 \approx 2.6$. The logarithm is used so that the contributions to the probabilities represented by each feature can be added. Additive quantities can be visualized using stacked bar graphs. No simple visual mechanism is available for multiplicative values.

The (red) tricarboxylic acid cycle subbars occur in the class lines of Other categories, Purines, Energy metabolism and Amino acid biosynthesis and in no other class lines. This indicates that this feature occurred only in the training data of these four classes. The relative lengths of the sub-bars indicate the (logs of the) relative number of times the feature occurred in the different training sets. Similarly, the violet sub-bar represents the occurrence of the feature glyoxylate bypass, which appeared only in training data for the classes Other categories, Purines, Energy metabolism and Regulatory functions.

The (orange) reduced residual bar represents the combined contributions of all features that are not explicitly shown in the graph. The length of each (orange) reduced residual bar has been reduced by subtracting the length of the shortest one from all the (orange) reduced residual bars. Since bar lengths represent logarithms of probabilities, any fixed amount can always be subtracted from all bars in the graph without affecting the difference between the lengths of any two bars. For example, if the original lengths of two bars were 63 and 61.6 respectively, this would mean that the ratio of probabilities for the two predicted classes was $2^{(63-61.6)} = 2^{1.4} = 2.6$. If we subtract 20 from both bars the new lengths are $63 - 20 = 43$ and $61.6 - 20 = 41.6$. The ratio of probabilities of the two predicted labels becomes $2^{(43-41.6)} = 2^{1.4} \approx 2.6$ (unchanged). Since Transport and binding proteins had the shortest (orange) bar to begin with, its (orange) bar is eliminated (has zero length) after subtracting its length from all orange bars including itself. This subtraction is equivalent to applying a zoom to the graph that focuses on the five most important features.

The (gray) reduced prior bars account for the different sizes of the training sets. A similar subtraction of the shortest (gray) reduced prior bar has been performed (i.e. the bar for the class Other categories). The explanation facility also allows the user to change which features are explicitly displayed in the graph.

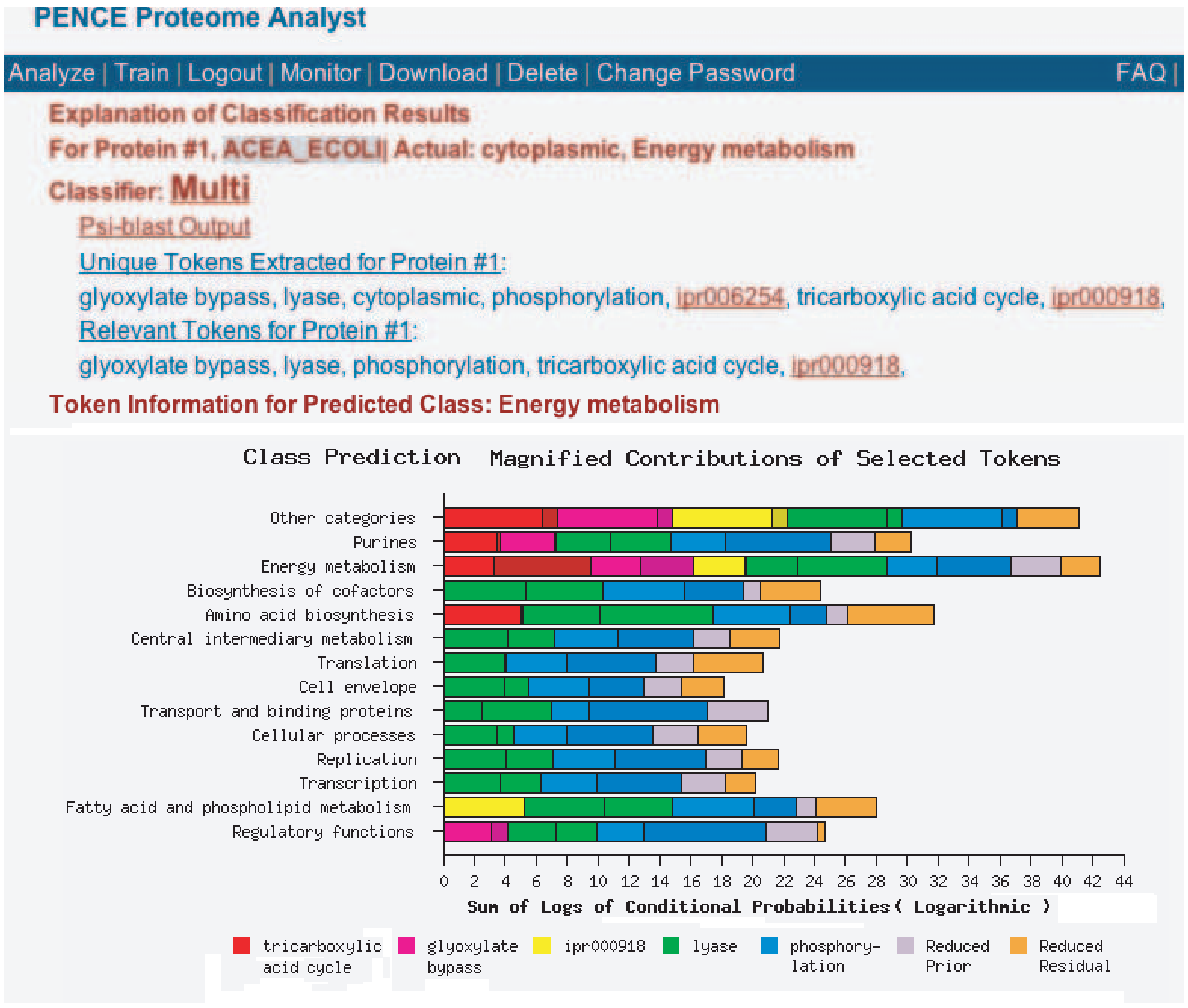
The importance of transparency

The PA Explain mechanism is the most important example of prediction transparency in PA. First, the Explain mechanism can be used to understand how a particular protein prediction was made. Second, it can be used to understand the internal structure of a predictor—how its training data affect its predictions. Prediction transparency is very important for two reasons. First, it is hard to accept predictions unless you understand how they were made. After using the Explain mechanism, you gain confidence that the predictor is working properly. Second, even the best predictors will make wrong predictions. They should not be trusted blindly.

There are three important situations in which classification-based predictors fail. First, classifiers are only as good as their training data, and the current databases that are used to obtain training data are not perfect. This is why PA clearly labels 'suspicious' training data as probably mislabeled, after it constructs any new classifier. This is another example of transparency in PA. Of course, the user decides on the training data, and whether to include the suspicious sequences in a classifier; PA's role is only to clearly identify the suspicious data. Given PA's feedback on the training data, a more conservative user can retrain a new classifier, without these suspicious sequences. We have found many suspicious sequences while training classifiers.

Second, predictors can fail if there is not enough training data to uniquely identify a single prediction class. In PA, this is characterized by a full-classifier graph (e.g. Figure 3) where there are multiple bars with significant probabilities.

Third, predictors can fail due to an inferior classifier algorithm which cannot adequately use the training data to differentiate between query sequences. A trend in ML in general, and recently in bioinformatics, has been always to select the algorithm with the best accuracy. If we had followed that advice we would be using an ANN or SVM classifier in PA (since they have accuracies that are a few percentage points higher) (9). But we are not. The ANN and SVM classifiers are not transparent,


PENCE Proteome Analyst

Explanation of Classification Results

For Protein #1, ACEA_ECOLI| Actual: cytoplasmic, Energy metabolism

Classifier: Multi

Psi-blast Output

Unique Tokens Extracted for Protein #1:

glyoxylate bypass, lyase, cytoplasmic, phosphorylation, ipr006254, tricarboxylic acid cycle, ipr000918,

Relevant Tokens for Protein #1:

glyoxylate bypass, lyase, phosphorylation, tricarboxylic acid cycle, ipr000918,

Token Information for Predicted Class: Energy metabolism

Figure 8. Part of the general function prediction (GeneQuiz ontology) Explain page for ACEA_ECOLI.

which, as argued above, is important. In the production version of PA, we have opted for a classifier with slightly lower accuracy so that we can provide transparent predictions. We believe that this is essential in this domain, where even the best predictors make errors due to bad training data or not enough training data, and these errors must be found.

SUMMARY

We have presented Proteome Analyst (PA), a web-based tool for the high-throughput prediction of protein features. In addition to several built-in classifiers and tools for protein annotation, PA supports the construction of custom classification-based predictors. PA's custom classifiers can be for any user-specified ontology and, since classifiers are trained by example, no programming knowledge is required. Notably, using PA's custom classifier features, we have trained and then made available (as a built-in predictor) in PA, the (currently) most accurate and most comprehensive subcellular localization predictor. Furthermore, to increase the user's confidence in the system and to help improve the accuracy of the classifiers, every prediction made by PA can be explained in a transparent way. PA is publicly available at http://www.cs.ualberta.ca/~bioinfo/PA.

ACKNOWLEDGEMENTS

This research was partially funded by research or equipment grants from the Protein Engineering Network of Centres of Excellence, the Natural Sciences and Engineering Research Council of Canada, Alberta Ingenuity Centre for Machine Learning, Sun Microsystems, SGI, and the Alberta Science and Research Authority.

REFERENCES

1. Andrade,M.A., Brown,N.P., Leroy,C., Hoersch,S., de Daruvar,A., Reich,C., Franchini,A., Tamames,J., Valencia,A., Ouzounis,C. and

Sander,C. (1999) Automated genome sequence analysis and annotation. *Bioinformatics*, **15**, 391–412.

2. Kitson,D.H., Badretdinov,A., Zhu,Z.Y., Velikanov,M., Edwards,D.J., Olszewski,K., Szalma,S. and Yan,L. (2002) Functional annotation of proteomic sequences based on consensus of sequence and structural analysis. *Brief. Bioinformatics*, **3**, 32–44.
3. Hubbard,T., Barker,D., Birney,E., Cameron,G., Chen,Y., Clark,L., Cox,T., Cuff,J., Curwen,V., Down,T. *et al*. (2002) The Ensembl genome database project. *Nucleic Acids Res*., **30**, 38–41.
4. Frishman,D., Albermann,K., Hani,J., Heumann,K., Metanomski,A., Zollner,A. and Mewes,H.W. (2001) Functional and structural genomics using PEDANT. *Bioinformatics*, **17**, 44–57.
5. Harris,N.L. (1997) Genotator: a workbench for sequence annotation. *Genome Res*., **7**, 754–762.
6. Gaasterland,T. and Sensen,C.W. (1996) MAGPIE: automated genome interpretation. *Trends Genet*., **12**, 76–78.
7. Overton,G.C., Bailey,C., Crabtree,J., Gibson,M., Fischer,S. and Schug,J. (1998) The GAIA software framework for genome annotation. *Pac. Symp. Biocomput*., 291–302.
8. Szafron,D., Lu,P., Greiner,R., Wishart,D., Lu,Z., Poulin,B., Eisner,R., Anvik,J. and Macdonell,C. (2003) Proteome Analyst—transparent high-throughput protein annotation: function, localization and custom predictors. 12th International Conference on Machine Learning, Workshop on Machine Learning in Bioinformatics, Washington, DC, August 21, 2003, pp. 2–10. http://bioml-workshop.rutgers.edu/
9. Lu,Z., Szafron,D., Greiner,R., Lu,P., Wishart,D.S., Poulin,B., Anvik,J., Macdonell,C. and Eisner,R. (2004) Predicting subcellular localization of proteins using machine-learned classifiers. *Bioinformatics*, **20**, 547–556.
10. Szafron,D., Greiner,R., Lu,P., Wishart,D., Macdonell,C., Anvik,J., Poulin,B., Lu,Z. and Eisner,R. (2003) *Explaining Naive Bayes Classifications*, Technical Report 03-09. Department of Computer Science, University of Alberta.
11. Apweiler,R., Attwood,T.K., Bairoch,A., Bateman,A., Birney,E., Biswas,M., Bucher,P., Cerutti,L., Corpet,F., Croning,M.D.R. *et al*. (2001) The InterPro database, an integrated documentation resource for protein families, domains and functional sites. *Nucleic Acids Res*., **29**, 37–40.
12. Mitchell,T.M. (1997) *Machine Learning*. McGraw-Hill, NY.
13. Kohavi,R. and John,G.H. (1997) Wrappers for feature subset selection. *Artif. Intell*., **97**, 273–324.
14. Gallin,W.J. and Spencer,A.N. (2001) Evolution of potassium channels. In Archer,S.L. and Spencer,A.N. (eds), *Potassium Channels in Cardiovascular Biology*. Kluwer, Dordrecht, pp. 3–16.