Prediction of Enzyme Classification from Protein Sequence without the use of Sequence Similarity

Marie desJardins
Peter D. Karp
Markus Krummenacker
Thomas J. Lee
Christos A. Ouzounis+

SRI International, 333 Ravenswood Avenue, Menlo Park CA 94025, USA, pkarp@ai.sri.com
+ Current Address: The European Bioinformatics Institute, EMBL Outstation, Wellcome Trust Genome Campus, Cambridge UK CB10 1SD

Abstract[1]

We describe a novel approach for predicting the function of a protein from its amino-acid sequence. Given features that can be computed from the amino-acid sequence in a straightforward fashion (such as pI, molecular weight, and amino-acid composition), the technique allows us to answer questions such as: Is the protein an enzyme? If so, in which Enzyme Commission (EC) class does it belong? Our approach uses machine learning (ML) techniques to induce classifiers that predict the EC class of an enzyme from features extracted from its primary sequence. We report on a variety of experiments in which we explored the use of three different ML techniques in conjunction with training datasets derived from PDB and from Swiss–Prot. We also explored the use of several different feature sets. Our method is able to predict the first EC number of an enzyme with 74% accuracy (thereby assigning the enzyme to one of six broad categories of enzyme function), and to predict the second EC number of an enzyme with 68% accuracy (thereby assigning the enzyme to one of 57 subcategories of enzyme function). This technique could be a valuable complement to sequence-similarity searches and to pathway-analysis methods.

Introduction

The most successful technique for identifying the possible function of anonymous gene products such as those generated by genome projects is performing sequence-similarity searches against the sequence databases (DBs). Putative functions are assigned on the basis of the closest similarity of the query sequence to proteins of known function. These techniques have achieved a high level of performance: more than 60% of *H. influenzae* (Casari *et al.* 1995) and around 40% of *M. jannashii* (NC *et al.* 1996) open reading frames (ORFs) have been assigned a specific biochemical function, at varying degrees of confidence. However, many unidentified genes remain in those genomes, and the only way that functional predictions can increase is by repeating the searches against larger (and hopefully richer) versions of the sequence DBs. For unique proteins, or large families of hypothetical ORFs, function remains unknown, with the current similarity-based methodology.

We have developed a novel approach for accurately predicting the function of a protein from its predicted amino-acid sequence, based on the Enzyme Commission (EC) classification hierarchy (Webb 1992). Given features that can be computed from the amino-acid sequence in a straightforward fashion, the technique allows us to answer questions such as: Is the protein an enzyme? If so, in which EC class does it belong?

Our approach uses machine learning (ML) techniques to induce classifiers that predict the EC class of an enzyme from features extracted from its primary sequence. We report on a variety of experiments in which we explored the use of three different ML techniques in conjunction with training datasets derived from PDB and from Swiss–Prot. We also explored the use of several different feature sets.

Problem Definition

The aim of this work is to produce classifier programs that predict protein function based on features that can be derived from amino-acid sequence, or a 3-D structure. The classifiers will predict whether the protein is an enzyme, as opposed to performing some other cellular role. If the protein is an enzyme, we would prefer to know its exact activity: however, we have assumed that learning to predict exact activities is too difficult a problem, partly because sufficient training data is not available. We therefore focus on the problem of predicting the general class of activity of an enzyme, which can also be valuable information.

Our work makes use of the EC hierarchy. This classification system organizes many known enzymatic activities into a four-tiered hierarchy that consists of 6 top-level classes, 57 second-level classes, and 197 third-level classes; the fourth level comprises approximately 3,000 instances of enzyme function. The organizing principle of the classification system is to group together enzymatic activities that accomplish chemically similar transformations. The central assumption underlying our work is that proteins that catalyze reactions that are similar within the EC classification scheme will also have similar physical properties.

We constructed classifiers that solve three different problems:

- **Level-0 problem:** Is the protein an enzyme?
- **Level-1 problem:** If the protein is an enzyme, in which of the 6 first-level EC classes does its reaction belong?
- **Level-2 problem:** If the protein is an enzyme, in which of the 57 second-level EC classes does its reaction belong?

For each prediction problem we ran several machine-learning algorithms to examine which performed best. We also employed several different training datasets for each prediction problem to determine what features are most informative, and to explore the sensitivity of the method to different distributions of the training data.

The only similar work we are aware of is Wu's work on learning descriptions of PIR protein superfamilies using neural networks (CH 1996).

Methods

Our methodology for applying ML to the enzyme classification problem was as follows:

1. Characterize the classification problem, and identify the characteristics of this problem that would influence the choice of an appropriate ML method.
2. Select one or more ML methods to apply to the classification problem.
3. Create a small dataset from available data sources.
4. Run the selected ML methods on the small dataset.
5. Evaluate the results, and make changes to the experimental setup by (a) Reformulating the classification problem (e.g., adding new prediction classes), (b) Eliminating noisy or problem data points from the dataset, (c) Eliminating redundant or useless features, or adding new features to the data, (d) Adding or deleting ML methods from the "toolkit" of methods to be applied
6. When the above process is complete, create a larger alternative dataset, run the selected ML methods, and evaluate the results.
7. Evaluate the results on all datasets, with all ML methods, with respect to the baseline test of a sequence-similarity search, currently the most widely used method of approaching this problem (P, C, & C 1994).

We started with a small dataset to familiarize ourselves with the domain, identify the features to be learned, and provide a testing ground for exploring the space of experimental setups, before scaling up to larger datasets. These larger datasets served to check the generality and scalability of the experimental approach in real-world situations. The sequence-similarity baseline provides a means of assessing the overall performance of the approach: Do ML methods make better predictions than sequence similarity? Are there some classes or particular cases for which ML methods perform better or worse?

Problem characteristics The features in this domain are mostly numerical attributes, so algorithms that are primarily designed to operate on symbolic attributes are inappropriate. The prediction problem is a multiclass learning problem (e.g., there are 6 top-level EC classes and 57 second-level EC classes to predict), for which not all learning algorithms are suited. The features are not independent (e.g., the sum of the normalized proportions of the amino acids will always be one), so algorithms that rely heavily on independent features may not work well. Most important, there may be noisy data (incorrect or missing feature values or class values), and we do not expect to be able to learn a classifier that predicts the EC class with perfect accuracy, so the algorithm must be able to handle noise. Such examples are sequence entries that are fragments but do have an assigned EC number, or real enzymes with no EC numbers assigned to them.

ML methods Based on the above problem characteristics, we selected three learning algorithms: discretized naive Bayes (DNB), C4.5, and Instance-Based Learning (IBL).

Discretized naive Bayes (J, R, & M 1995) is a simple algorithm that stores all the training instances in bins according to their (discretized) feature values. The algorithm assumes that the features are independent given the value of the class. To make a prediction, it does a table lookup for each feature value to determine the associated probability of each class, given the feature's value, and combines them using Bayes' rule to make an overall prediction.

C4.5 (JR 1993) induces classifiers in the form of decision trees, by recursively splitting the set of training examples by feature values. An information-theoretic measure is applied at each node in the tree to determine which feature best divides the subset of examples covered by that node. Following the tree construction process, a pruning step is applied to remove branches that have low estimated predictive performance.

The term *instance-based learning* (IBL) covers a class of algorithms that store the training instances, and make predictions on a new instance I by retrieving the nearest instance N (according to some similarity metric over the feature space) and then returning the class of N as the class of I (or by making a weighted prediction from a set of nearest instances) (Aha DW 1991).

Feature engineering Feature engineering, or the problem of identifying an appropriate set of features and feature values to characterize a learning problem, is a critical problem in real-world applications of ML algorithms. This process frequently represents a substantial part of the time spent on developing the application, and this project was no exception. Section describes the features we identified, and their biochemical significance. Section discusses the process by which we identified and removed redundant features. Section gives the results for the alternative datasets and feature sets that were explored.

Large datasets We used extended datasets for the final evaluation of the ML methods on the enzyme classification problem. We created several versions of these datasets — a full Swiss–Prot version, and several "balanced" datasets that contain a random sampling of the proteins in the Swiss–Prot DB, selected to have a class distribution (of enzymes vs. non-enzymes) similar to the PDB dataset in the first case, and to the distribution of enzymes versus non-enzymes in complete genomes in the second case.

Sequence similarity The predictions using ML have been compared with function assignments made through sequence similarity. We used BLAST (Altschul *et al.* 1990) with standard search parameters and a special filtering procedure (unpublished), against the equivalent datasets from the ML experiments. Query sequences (with or without an EC number) were predicted to have the EC number of the closest homologue (if applicable). Only significant homologies were considered, with a default cut-off P-value of 10^{-6} and careful manual evaluation of the DB search results. In this manner, we have obtained an accuracy estimate for the similarity-based methods. It is interesting to note that such an experiment is, to our knowledge, unique.

Features

For the core set of features used as inputs to the ML programs, we used properties that can be directly computed from primary sequence information, so they can be used for predicting the function of ORFs whose structure is also unknown. Those features are *length* of the amino acid sequence, the molecular weight *mw* of the sequence, and the amino acid composition, represented as 20 values {*pa pc pd pe pf pg ph pi pk pl pm pn pp pq pr ps pt pv pw py*} in the range from 0 to 1, each value standing for the respective residue frequency as a fraction of the total sequence length.

The feature *charge* was computed by summing the contributions of charged amino acids. The features *ip* (isoelectric point) and *extinction* coefficient were calculated by the program "Peptidesort" (Peptidesort is from the GCG package, version 8.0-OpenVMS, September 1994).

The secondary structural features *helix, strand,* and *turn*, which we used for one experiment, were extracted from information in the FT fields of Swiss–Prot records. For all such lines with a HELIX, STRAND, or TURN keyword, the numbers of amino acids between the indicated positions were summed up, to calculate the total percentages of amino acids that are part of these structures, respectively. We included this information, since it was available for the proteins in the PDB, to see how well it would improve the prediction quality of the learned classifiers if it *were* available for an unknown protein. Secondary structure can be estimated from the primary sequence (although not with perfect accuracy), and using this estimated secondary structure might be worthwhile in making predictions if secondary structure proved to be a strong enough predictor of enzyme class.

Datasets

We obtained EC classes from version 21.0 of the ENZYME DB (Bairoch 1996). We prepared datasets derived from the PDB and Swiss–Prot.

Dataset 1: This family of datasets originated from the PDB subset of Swiss–Prot,[2] containing 999 entries. Features for these protein sequences were calculated as

[2] See `ftp://expasy.hcuge.ch/databases/Swiss-Prot/special_selections/pdb.seq.180496`

in Section 3.1. EC numbers were extracted from the text string in the (DE) field of the Swiss–Prot records (more than one EC can occur in one entry). We created several variants of this dataset, containing different features.

Dataset 1a: Features:

{*length mw charge ip extinction pa pc pd pe pf pg ph pi pk pl pm pn pp pq pr ps pt pv pw py*}

Dataset 1b: We dropped the *ip* and *extinction/mw* features, because they were strongly correlated with *charge* and *length*, respectively. Features:

{*length charge pa pc pd pe pf pg ph pi pk pl pm pn pp pq pr ps pt pv pw py*}

Dataset 1c: We reduced the feature set further by combining the composition percentages of amino acids with similar biochemical properties (WR 1986). The following subsets were grouped together: *ag c de fwy h ilv kr m nq p st*, reducing the number of amino acid composition features from twenty to eleven. Features:

{*length mw charge pag pc pde pfwy ph pilv pkr pm pnq pp pst*}

Dataset 1d: Three new secondary structural features were added to the features in 1b. Features:

{*length mw charge pa pc pd pe pf pg ph pi pk pl pm pn pp pq pr ps pt pv pw py helix strand turn*}

Dataset 2: The raw data originated from the full release of Swiss–Prot version 33 (Bairoch & Boeckmann 1994), containing 52205 entries. Features were computed using the "aacomp" program (aacomp is part of the FASTA package). Secondary structural features were omitted, because only a small minority of entries carry this information. Feature set:

{*length charge pa pc pd pe pf pg ph pi pk pl pm pn pp pq pr ps pt pv pw py*}

Characterization of the Data

Table 1 provides a numerical overview of the entries in the datasets. There is a notable difference between Datasets 1 and 2 in the percentage of entries that have an EC-number, perhaps because enzymes are a common object of study by crystallographers. Dataset 2 is probably closer to the natural enzyme vs. non-enzyme distribution in the protein universe.

A few entries have more than one EC number, for example, multifunctional or multidomain enzymes. We have excluded all these cases from the final dataset, on the assumption that they will introduce noise in the EC-classification experiments. Entries without any EC number are presumed not to be enzymes. However, we could envision data entry errors of omission that would violate this assumption. We performed a search for the string "ase" in the DE field of Swiss–Prot records that lack EC numbers to find potential mis-annotated enzymes. This search did pull out quite a few hits, which could act as false positives in the non-EC class. Dataset 2 contained too many such cases for a hand-analysis, but the Dataset 1 cases were examined. About half of the cases were enzyme-inhibitors, some were enzyme-precursors, and a few entries did seem to be enzymes.

Results

We present three groups of results. The sequence of experiments reflects our exploration of various subsets of features for the prediction problems under study. The first group involves the PDB datasets; the second group involves the Swiss–Prot dataset. We explored how well the learning algorithms scaled with training set size and composition. The third group compares the results of the learning experiments with sequence similarity — a mature technique for function prediction.

Learning experiments were conducted by preprocessing the dataset of interest to produce three different input files, one each for the level 0, level 1, and level 2 prediction problems. Preprocessing consisted of excluding non-enzymes from the level 1 and level 2 files, and formatting the class appropriately for the problem, that is, a binary value for level 0, one of six values for level 1, and one of 57 values for level 2 (actually 51, since the datasets only represented 51 of the 57 possible two-place EC numbers). We omitted level 2 experiments with the PDB datasets because 500 training instances are too few to learn 51 classes.

Each experiment consisted of performing a tenfold cross-validation using a random 90:10% partition of the data into a training set and test set, respectively. A suite of three experiments was run for each input file, one for each of the three learning algorithms DNB, C4.5, and IB. Results are reported as percentage of test set instances correctly classified by each algorithm, averaged over the ten cross-validation runs.

Experiments were conducted using *MLC++* version 1.2 (R & D 1995), a package of ML utilities. All experiments were run on a Sun Microsystems Ultra-1 workstation with 128 MB of memory, under Solaris 2.5.1.

Results for the PDB Datasets

Experiments involving Dataset 1 are shown in Table 2. Dataset 1a provides a baseline. The results for Dataset 1b show that the features extinction, isoelectric point, and molecular weight are redundant since each is strongly correlated with either charge or length (a principal component analysis of the feature sets also confirmed this fact – not shown). Those features were

	Dataset 1	Dataset 2
entries with exactly one EC number:	416 (41.6%)	14709 (28.2%)
entries without EC number:	565 (56.6%)	36997 (70.9%)
entries with multiple EC numbers (not used):	18 (1.8%)	499 (1.0%)
total number of entries in the raw dataset:	999	52205
entries with no EC number but “ase” in name:	35 (3.5%)	2156 (4.1%)

Table 1: Summary of data characteristics

Dataset and Features Used	Problem	Instances	IB	DNB	C4.5
(1a) Initial features	level 0	980	79.29	76.63	78.37
(1a) Initial features	level 1	416	60.10	48.54	50.53
(1b) Nonredundant features	level 0	980	77.96	76.33	78.98
(1b) Nonredundant features	level 1	416	62.74	48.54	48.64
(1c) Amino acid grouping	level 0	980	74.59	74.49	74.08
(1c) Amino acid grouping	level 1	416	57.46	45.90	46.63
(1d) Structural, unknowns	level 0	980	77.48	77.98	76.97
(1d) Structural, unknowns	level 1	416	55.06	47.64	49.59
(1d) Structural, no unknowns	level 0	630	81.59	80.16	76.19
(1d) Structural, no unknowns	level 1	266	63.50	47.35	48.52

Table 2: Classification accuracies on the PDB dataset for various representations

excluded from future experiments.

Experiment 1c asks whether accuracy can be improved by creating new features that group amino acids according to their biochemical properties. It is surprising that the results with this representation were universally worse. It is likely that useful information is lost with this reduction. We concluded that, since prediction was better with more features, we had not yet reached an upper bound on feature set size and could effectively add features without overwhelming any of the learning algorithms. We have not yet explored other groupings of amino acids.

Our next experiments added secondary structure information to the feature set by including the *helix*, *strand*, and *turn* features. Because the values for these features were not available for a high proportion (over 50%) of instances in PDB, we conducted two suites of experiments. The first suite used all instances in our PDB dataset, and annotated missing structure features as unknown values. The second suite excluded all instances for which structural data was missing. With unknowns excluded, accuracy did improve somewhat with the addition of structure features. However, the improvement is rather small. The value of structural composition is unclear, and further exploration has been left for future work.

Results from the Swiss–Prot Dataset

We conducted experiments using two subsets of Swiss–Prot, as well as with the full dataset and with some class-balanced datasets. Results are listed in Table 3. The first was a yeast subset, consisting of all instances for the species *Saccharomyces cerevisiae* (baker’s yeast) and *Schizosaccharomyces pombe* (fission yeast). The second was a prokaryotic subset, consisting of all instances for the species *E. coli*, *Salmonella typhimurium*, *Azotobacter vinelandii*, *Azotobacter chroococcum*, *Pseudomonas aeruginosa*, *Pseudomonas putida*, *Haemophilus influenzae*, and various *Bacillus* species.

As was observed with the PDB datasets, the IB algorithm performs the best overall. Although the other two algorithms are comparable for the simpler level 0 problems, they degrade substantially more than IB does as the number of classses increases. It also appears as if IB improves generally, though not universally, as the number of training instances increases. This is most apparent in the 67.6% accuracy it attains for the full 51-class problem. This is an encouraging trend, lending hope that classification accuracy will improve as more sequence data becomes available. IB consumes substantial machine resources during the training phase, however (57 hours of CPU for full Swiss–Prot), and it is not yet known if there a practical limit to its training set size.

We investigated the level 0 problem further by using different proportions of non-enzymes to enzymes. In full Swiss–Prot the proportion is 70:30%. We randomly excluded instances from it to generate a 57:43% ratio (which is the proportion for the PDB dataset) and an 80:20% ratio.

One simple way to evaluate a classifier is to compare it with a purely prior-based classifier that uses no features, simply classifying all instances as the class that is most represented in the training set (in this case, as a non-enzyme). All classifiers except DNB for the 80% non-enzyme dataset are above this baseline. IB is an impressive 25 percentage points better than this baseline for the 57% non-enzyme dataset, and 15% better for the 70% non-enzyme dataset.

Sequence Similarity

To compare our technique against the gold standard of sequence function prediction — sequence-similarity searching — we needed to evaluate the accuracy of sequence-similarity methods. We conducted a self-comparison of the full PDB dataset, with each query sequence treated as an unknown instance, and for those sequences with significant similarity to at least one entry in the DB, the EC number (or the absence of it) was used for the assignment of function to the query sequences. The decision algorithm in Figure 1 was used to automatically infer function from sequence analysis for a query sequence Q.

We will follow the convention enzymes/non-enzymes here to discuss the results. From the 999 entries (434/565), 731 (73%) did have a homologue (319/412), and 268 (27%) did not (115/153).

Table 4 summarizes the accuracy of sequence similarity at levels 0–4 (levels 3 and 4 refer to prediction of the first 3 elements, and all elements of the EC number, respectively). At level 4, for example, from the entries *with* a homologue, 625 were correctly predicted (219/406) while 106 were wrongly predicted (100/6). From the entries without a homolgue, the 115 enzymes are scored as incorrect, whereas the 153 non-enzymes are scored as correct. The combined accuracy of the rule is 78%. If we apply this rule to only those proteins that have a homologue (excluding non-homologues completely), the accuracy of the rule is 85%.

We conducted a similar experiment using the full Swiss–Prot dataset and sevearl hundred sequences selected from it randomly. Similar results were obtained, alleviating some concern over the possible underrepresentation of homologues given the small size of the PDB data set.

We used the homologues that were identified using sequence similarity to conduct a final experiment. We withheld the 268 proteins with no homologues as a test set, and trained on the remaining proteins. The IB, DNB, and C4.5 algorithms were able to solve the level 0 problem on this test set at accuracies of 66%, 77%, and 68%, respectively. Since the accuracy with the full PDB was 76–79% for the three methods, the accuracy did degrade, but it is still above the 57% to be expected from random guessing. Therefore, in the absence of sequence similarity, our method has potential value.

Discussion

Several conclusions can be reached from these experiments. The relative performance of the ML algorithms is that IB is best, C4.5 is second best, and DNB is worst. C4.5 has other advantages over IB, such that decision trees can be stored more compactly, and are more amenable to inspection and understanding by scientists.

We were surprised that the feature set based on biochemical groupings of amino acids yielded worse accuracy that no groupings. Of course, many different groupings have been defined by other researchers, and could yield improvements. We were also surprised that the structural features yielded little improvement in accuracy. We plan to explore the utility of other features that can be computed from sequence, for example, coiled-coil and transmembrane regions may be associated with certain classes of enzymes.

The accuracy levels obtained here can be interpreted in several ways, as shown in Table 4. In general, these results support the rather surprising conclusion that protein function can be predicted fairly accurately from extremely gross features derived from the protein sequence, without the use of homology or pattern searches. We can compare the best accuracies from our method on the entire Swiss–Prot dataset (Column 2) with the accuracy that would be obtained by guessing purely by chance among the classes at each level (Column 3). Column 4 shows another guessing strategy based on always selecting the class with the highest prior probability. Column 5 shows the accuracy obtained by sequence analysis.

The comparison between Columns 2 and 4 is the least satisfying for level 0; it is difficult for our method to distinguish enzymes from non-enzymes. The performance might be improved if we trained the system on additional classes of proteins, such as transporters and DNA binding proteins. The results for levels 1 and 2 are much more encouraging: given knowledge that a protein is an enzyme, it is possible to predict

Subset	Problem	Instances	IB	DNB	C4.5
Yeast	level 0	4055	75.59	76.84	75.02
Yeast	level 1	936	53.00	41.56	39.32
Yeast	level 2	936	45.16	25.02	25.45
Prokaryotic	level 0	7598	75.40	72.03	72.49
Prokaryotic	level 1	2442	56.10	37.92	38.08
Prokaryotic	level 2	2442	48.15	15.82	23.40
Full	level 0	49550	85.20	70.91	78.74
Full	level 1	14709	74.18	46.87	53.14
Full	level 2	14709	67.61	37.38	43.03
80% non-enzyme	level 0	43614	87.01	73.74	81.91
70% non-enzyme	level 0	49550	85.20	70.91	78.74
57% non-enzyme	level 0	34626	82.58	69.48	74.77

Table 3: Classification accuracy for various subsets of Swiss–Prot

```
IF Q has no homologue,
  THEN infer Q is not an enzyme (the most common class)
  ELSE
    Let H be the homologue of Q with the smallest BLAST p-value
    IF H has no EC#, THEN infer Q is not an enzyme
        ELSE  infer Q is an enzyme with the same EC# as that of H
```

Figure 1: A decision algorithm was used to infer function from sequence analysis for a query sequence Q.

Level	ML Method	Guessing	Prior-based Guessing	Sequence Analysis
0	87%	50%	80%	87%
1	74%	17%	30%	87%
2	68%	2%	17%	86%
3				85%
4				78%

Table 4: Comparison of the performance of the best ML classifier with different guessing strategies and with the sequence analysis baseline.

its first two EC numbers with surprisingly good accuracy. One possible way of removing the restriction that the enzyme/non-enzyme distinction be known (which cannot be established accurately enough by the level 0 classifier) is to consult the certainty value produced by C4.5 in conjunction with each prediction. It is possible that if we applied the level 2 classifier to a mix of all proteins, and discarded all predictions with a low certainty value, we could obtain good accuracy on the remaining predictions.

Although our method performs on par with sequence-similarity methods, these methods are intended to be complementary rather than competitive. In addition, our method produces less precise predictions than does sequence similarity: our method predicts only the first two elements of the EC number whereas sequence similarity predicts all elements.

Future Work

We expect to pursue four uses for our approach in the future. First is to apply it to ORFs that have not been identified by sequence-similarity methods (e.g., in bacterial genomes). Second is to apply the technique in conjunction with metabolic-pathway analysis of genomic sequence (Karp, Ouzounis, & Paley 1996). Pathway analysis often suggests missing enzymes that should be present in the genome; the technique presented here can be used to search for them since the pathway analysis gives their EC numbers. Third is to apply this approach as a check on the results obtained from other sequence-analysis techniques, if we can identify enzyme families for which our approach gives very accurate answers. The fourth use is to apply this same machine-learning approach to predicting the functions of other classes of gene products besides enzymes.

This project suggests several directions for ML research. The classifiers learned by the ML methods are often difficult for a user to understand. Tools for visualizing or summarizing these classifiers in an intuitive way would be immensely helpful in interpreting and evaluating the learning process. Other useful tools would enable the user to visualize error distribution (i.e., which instances and classes are misclassified, and in what way), and to compare classifiers in different representations (e.g., a decision tree versus an instance-based classifier).

We were very surprised to learn that little empirical work has been done to establish the accuracy of functional assignments obtained through sequence-similarity searches, considering how widespread is the use of this method. Future work should explore this question in more detail.

Acknowledgments

This work was supported by SRI International and by the Human Frontiers Science Program.

References

Aha DW, Kibler D, A. M. 1991. Instance-based learning algorithms. *Machine Learning* 6:37–66.

Altschul, S.; Gish, W.; Miller, W.; Myers, E.; and Lipman, D. 1990. Basic local alignment search tool. *J Mol Bio* 215:403–410.

Bairoch, A., and Boeckmann, B. 1994. The SWISS-PROT protein sequence data bank: current status. *Nucleic Acids Res* 22:3578–3580.

Bairoch, A. 1996. The ENZYME databank in 1995. *Nucl Acids Res* 24:221–222.

Casari, G.; Andrade, A.; Bork, P.; Boyle, J.; Daruvar, A.; Ouzounis, C.; Schneider, R.; Tamames, J.; Valencia, A.; and Sander, C. 1995. Challenging times for bioinformatics. *Nature* 376:647–648.

CH, W. 1996. Gene classification artificial neural system. *Methods in Enzymology* 266:71–88.

J, D.; R, K.; and M, S. 1995. Supervised and unsupervised discretization of continuous features. In *Proc of the Machine Learning Conference*.

JR, Q. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Karp, P.; Ouzounis, C.; and Paley, S. 1996. HinCyc: A knowledge base of the complete genome and metabolic pathways of *H. influenzae*. In States, D.; Agarwal, P.; Gaasterland, T.; Hunter, L.; and Smith, R., eds., *Proc of the Fourth International Conference on Intelligent Systems for Molecular Biology*, 116–124. Menlo Park, CA: AAAI Press.

NC, K.; GJ, O.; H-P, K.; O, W.; and CR, W. 1996. *Methanococcus jannaschii* genome: revisited. *Microbial and Comparative Genomics* 1:329–338.

P, B.; C, O.; and C, S. 1994. From genome sequences to protein function. *Curr Opin Struct Biol* 4:393–403.

R, K., and D, S. 1995. MLC++: Machine learning library in C++. See WWW URL `http://www.sgi.com/Technology/mlc`.

Webb, E. C. 1992. *Enzyme Nomenclature, 1992: Recommendations of the nomenclature committee of the International Union of Biochemistry and Molecular Biology on the nomenclature and classification of enzymes*. Academic Press.

WR, T. 1986. The classification of amino acid conservation. *J Theor Biol* 119:205–218.