Pathway Analyst—Automated Metabolic Pathway Prediction

Luca Pireddu, Brett Poulin, Duane Szafron, Paul Lu, and David S. Wishart
Department of Computing Science
University of Alberta
Edmonton, AB, T6G 2E8 CANADA
{luca, poulin, duane, paullu}@cs.ualberta.ca, david.wishart@ualberta.ca

Abstract—**Metabolic pathways are crucial to our understanding of biology. The speed at which new organisms are being sequenced is outstripping our ability to experimentally determine their metabolic pathway information. In recent years several initiatives have been successful in automating the annotations of individual proteins in these organisms, either experimentally or by prediction. However, to leverage the success of metabolic pathways we need to automate their identification in our rapidly growing list of sequenced organisms. We present a prototype system for predicting the catalysts of important reactions and for organizing the predicted catalysts and reactions into previously defined metabolic pathways. We compare a variety of predictors that incorporate sequence similarity (BLAST), hidden Markov models (HMM) and Support Vector Machines (SVM). We found that there is an advantage to using different predictors for different reactions. We validate our prototype on 10 metabolic pathways across 13 organisms for which we obtained a cross-validation precision of 71.5% and recall of 91.5% in predicting the catalyst proteins of all reactions.**

I. Introduction

Understanding the complex metabolic processes of organisms has been a long-standing challenge for biologists. These processes consist of a map of chemical reactions, each catalyzed by special-purpose proteins. The complexity of the metabolic system motivated the creation of an abstraction to provide a simpler view of this complex network: the metabolic pathway. Metabolic pathways are a way to segment the map of the metabolic process into logical sections of related reactions. Each pathway contains two kinds of information: the *pathway structure* and the *pathway components*.

The *pathway structure* is the graph or map that defines the relationships between the reactions in the pathway, along with the compounds (small molecules) that are their reactants and products. For example, Fig. 1 shows the relationship between seven reactions in the Gluconeogenesis pathway. The reactant of reaction 1 is Lipoamide, the catalyst (enzyme) is a Dihydrolipoyl dehydrogenase protein (labeled by the enzyme classification number of this family of enzymes, EC 1.8.1.4) and the product is Dihydrolipoamide. Some reactions have more than one reactant or product. For example, reaction 3 has two reactants (2-Hydroxy-ethyl-Thpp and Lipoamide) and two products (ThPP and 6-S-Acetyl-dihydrolipoamide). Some reactions have more than one family of catalysts. In such cases, catalysts from any of these families can catalyze the reaction. For instance, reaction 4 has two potential catalyst families, Pyruvate dehydrogenase (EC 1.2.4.1) and Pyruvate decarboxylase (EC 4.1.1.1).

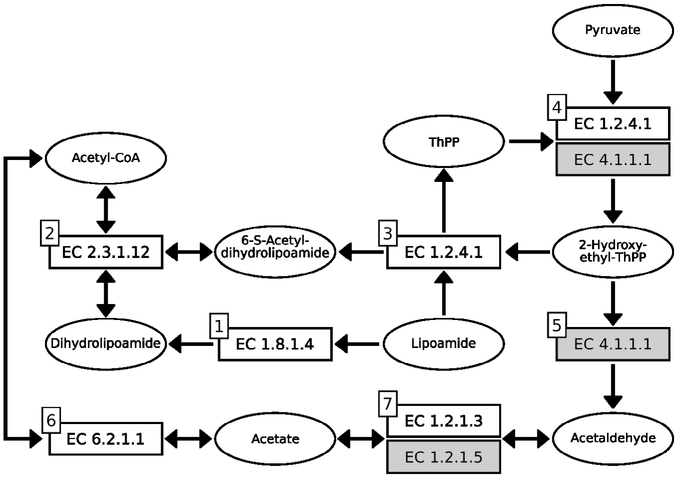

Fig. 1. Seven reactions from the Gluconeogenesis pathway.

The *pathway components* are the specific proteins that catalyze each of the pathway's reactions in a specific organism. For example, reaction 1 is catalyzed by protein HSA:1737(DLD)[1] in the organism *H. sapiens* (NCBI-GI 4557525, Uniprot P09622) and by protein CEL:LLC1.3 in the organism *C. elegans*. Sometimes more than one protein from the same catalyst family can catalyze the same reaction in a single organism. For example, in the organism *A. thaliana*, reaction 1 is catalyzed by three proteins in family EC 1.8.1.4: ATH:At1g48030, ATH:At3g16950 and ATH:At3g17240.

Although two species may have similar metabolic pathways, evolution generates some organism-specific variations. We refer to the variants of a specific metabolic pathway across different organisms as *pathway instances*. Pathway instances can differ in their pathway components, as illustrated by the organism-specific proteins in a single enzyme family. However, pathway instances can also differ in their pathway structure. For example, in Fig. 1, two of the catalyst families denoted by gray boxes (EC 4.1.1.1) are present in the pathway instance for *A. thaliana*, but are not present in either *C. elegans* or *H. sapiens*. The other catalyst family in a gray box (EC 1.2.1.5) is present in *C. elegans* and *H. sapiens*, but is not present in *A. thaliana*. Since there is no

[1] We use the notation of the Kyoto Encyclopedia of Genes and Genomes (KEGG) PATHWAY database for protein names. Each name consists of a three character organism identifier followed by an abbreviated gene name from the original source, such as NCBI, Wormbase, TAIR, TIGR, MIPS, etc.

catalyst for reaction 5 in *C. elegans* or *H. sapiens*, reaction 5 is not known to take place in either of these organisms, so it is not part of the structure of their pathway instances.

A goal of biology is to understand the pathway instances of all known organisms. Biologists who study pathways tend to approach their work in one of two ways. The first approach is to study a particular pathway over many organisms. The second is to study one organism by trying to analyze all of its pathways. The current rate of knowledge acquisition prompts a broader systems approach. Genomic and proteomic sequence data is being generated so fast that analytical experimental methods to determine pathway structure and components cannot keep pace. To cope with this deluge of sequence data, we propose an automated computational approach to the prediction of organism-specific metabolic pathways that can assist the study of many pathways in many organisms. We hypothesize that the biological and sequence similarities between organisms can be exploited to predict the structure and the components of metabolic pathway instances. This paper makes three main research contributions:

1. We describe Pathway Analyst, a prototype high-throughput system that predicts metabolic pathway reactions and catalysts
2. We demonstrate a simple but effective pathway prediction algorithm that incorporates machine learning techniques.
3. We provide empirical results that suggest the need for reaction-specific classifiers.

II. REPRESENTING METABOLIC PATHWAYS

We represent the reactions in each metabolic pathway as a directed graph. Each reaction node is annotated with the identifier of the reaction it represents (in this paper we use 1, 2, 3 …), and a set of proteins that catalyze it. The EC number of the enzyme family that catalyzes the reaction is also displayed in this paper for easy reference. The arcs of the graph follow the flow of chemical compounds through the metabolic process. Fig. 2 shows an example of this data model.

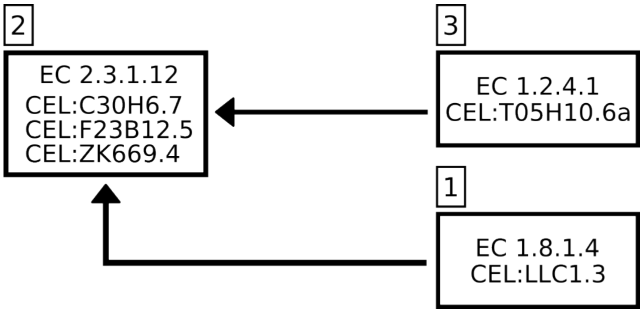

Fig. 2. A data model of three reactions from the Gloconeogenesis pathway in *C. elegans* expressed as a graph of reaction nodes.

This graph represents a section of the instance of the Gluconeogenesis pathway for *C. elegans* that spans reactions 1, 2 and 3 from Fig. 1. Each node contains the names of the proteins in *C. elegans* that catalyze the reaction. The arcs from reactions 1 and 3 to reaction 2 indicate that the chemical products of 1 and 3 are the reactants of 2.

III. THE PATHWAY PREDICTION ALGORITHM

Having a data model to represent pathways in a structured way lays the foundations for an algorithm that can use the pathway model to make predictions. To exploit the similarity between organisms, we would like to learn from well-studied versions of a pathway (such as in model organisms) to predict what the pathway may look like in an organism of interest, where the pathway is not as well characterized. We first introduce some terminology. The *target organism* is the organism whose pathway instance we are predicting. A *training pathway* is a pathway instance that is given as input to the prediction algorithm. A *training reaction* is a reaction in a training pathway. A *training protein* is a protein that labels a training reaction. We devise a prediction algorithm that takes as input a single training pathway[2] and the proteome of a target organism. The goal of the algorithm is to predict:

1) whether the pathway exists in the target organism and, if it exists, then
2) the structure of the pathway, and
3) for every reaction in the pathway, its set of potential catalyst proteins.

To achieve the goal, we analyze each of the training pathway's reactions. For each of these reactions, we must decide whether or not it exists in the target organism. We assume that the reaction needs to have at least one protein catalyst to occur. Therefore, we determine whether the reaction exists in the target organism by determining whether the target organism has one or more proteins capable of performing the same function as the training reaction's proteins. If such candidate proteins are found, the reaction is added to the predicted pathway. In addition, the predicted reaction is annotated with the candidate proteins from the target organism's proteome. On the other hand, if no such proteins are found then we predict that the training reaction does not exist in the target organism. Finally, if none of the training reactions are predicted to exist in the target organism then the algorithm decides that the entire pathway does not exist in the organism. The pathway prediction algorithm appears as Algorithm 1.

The prediction algorithm is quite intuitive. However, it abstracts a critical step – how to decide whether a protein from the target organism is capable of performing the same function as the training protein. This particular task is handled by a classifier and is hidden in the algorithm as the `find_able_proteins_in(proteome)` invocation.

The classifier is a critical component of the pathway prediction algorithm. It is a computational device that predicts which proteins from the target organism are capable of catalyzing a training reaction. The classifier is a "black box" whose inputs consist of the target proteome and a training reaction. The classifier filters the target organism's proteome, returning only those proteins that are functionally compatible with the training reaction's catalysts. Therefore, the classifier in the pathway prediction problem must make a very specific function prediction based on a small number of positive training samples. In our experimental data set the number of

[2] In the next section we generalize this approach to utilize multiple training pathways.

catalysts in a single reaction of a single pathway instance varies from 1 to 17, with a mean of 1.8. In the next section we illustrate how the number of positive training samples can be increased to the range 1 to 50 with a mean of 11.5. However, even with this increase, the problem is different from many other protein function prediction problems, such as high level Gene Ontology [1] classification, because a much more specialized function must be predicted for the target protein and the number of positive training samples is still very low.

In Section VI we describe the different classifiers we have used in our pathway prediction architecture. In Section VII, we compare the prediction accuracies of these classifiers.

```
Algorithm Pathway prediction algorithm.
Require: training_pathway
Require: proteome
Ensure: prediction
  prediction ← Pathway.new
  for all reaction in training_pathway do
    pred_proteins ← reaction.find_able_proteins_in(proteome)
    if not pred_proteins.empty? then
      new_reaction ← prediction.add_reaction(reaction)
      new_reaction.add_catalysts(pred_proteins)
    end if
  end for
  if prediction.empty? then
    prediction ← nil
  end if
  return prediction
```

Algorithm 1. The pathway prediction algorithm.

IV. THE MODEL PATHWAY

There are two major problems in using a pathway prediction algorithm that tries to predict the structure and components of a pathway based on a single pathway instance. First, we have shown that the structure of a pathway varies between organisms. Using only a single training pathway increases the chance that the training pathway has a different structure than the target pathway. In this case, predicting the true structure of the target pathway becomes impossible since no predictions would be attempted on any reaction not found in the training instance. For example, if the training pathway is the *C. elegans* pathway for Gluconeogenesis and the target pathway is the same pathway in *A. thaliana* then there is no chance of finding the reaction denoted 5 in Fig. 1, since this reaction does not appear in the training pathway. Second, as indicated in the last section, if a classifier uses only a few positive training instances to predict the pathway components, then the classifier will have poor accuracy.

Our approach is to make our structure and component predictions using all available pathway instances. This enables the algorithm to use all of the diverse instances of the training pathway to match the structure of the target pathway and to predict components more accurately. To add this capability to the algorithm we introduce the notion of a *model pathway*. A *model pathway* is an abstract version of a pathway that combines multiple pathway instances. To create a model pathway we effectively take the "union" of a number of training pathways. At the structural level, we define the union of two pathways A and B as $U = A \cup B$, where U is a new pathway whose structure includes all the reactions occurring in either A or B. For each reaction in pathway U, if that reaction existed in both A and B, then the reaction's protein catalyst set in U is the union of the catalyst sets from the same reaction in A and B. Fig. illustrates the union of part of the *C. elegans* instance of the Gluconeogenesis pathway (left subfigure) and part of the *S. pombe* instance of the same pathway (middle subfigure). *C elegans* has reaction 6 and 7, from Fig. 1, but not reaction 5. *S. pombe* has reaction 5 and 7, but not reaction 6. The union pathway has all three reactions. The catalyst set for reaction 7 contains all of the proteins that catalyzed this reaction in either organism, regardless of the EC family where the protein originated. For brevity, reactions 5 and 6 are shown with only one of their protein catalysts.

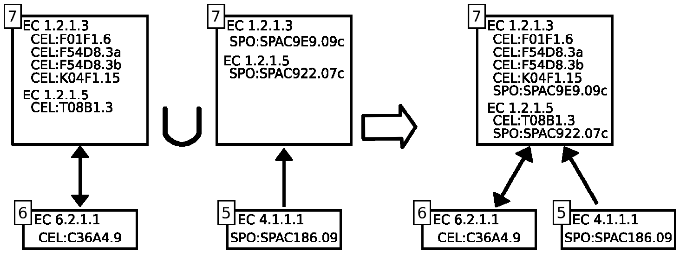

Fig. 3. The union of two partial pathway instances in a model pathway.

Algorithm 1 is not modified, except that the model pathway is used as the training pathway input instead of using a single instance pathway. By using model pathways, the pathway prediction algorithm can even predict instances of a pathway with variations in structure that were never observed in the training pathway set – and perhaps never found in any physical laboratory. Such *emergent structures* can be computationally predicted before being observed.

V. EXPERIMENTAL METHODOLOGY

To evaluate the effectiveness of our automated pathway prediction technique, we performed a cross-validation of pathway predictions to simulate the situation where the pathways of one organism are completely unknown. We began with a data set consisting of n instances of the same pathway, where each instance was from a different organism. We used $n - 1$ organisms to build a model pathway and then used this model pathway to build a classifier. We used this classifier to predict the remaining n^{th} pathway instance and compared the predictions to the known structure and components of this n^{th} pathway instance. We repeated this cycle n times, each time predicting the pathway instance of a different one of the organisms. Since our data set had 125 different pathway instances, we repeated this cross-validation process 125 times, once for each instance. Finally, we aggregated all of the results to compute overall statistics.

Our algorithm is evaluated by comparing the structure (reactions) and components (catalysts) of the predicted pathway instance to the known pathway information. We computed statistics for the components predicted to catalyze each reaction in a pathway, and called these statistics the *catalyst scores*. We also separately computed statistics for the existence of each reaction in a pathway. We call these statistics *reaction scores*. We computed three standard statistics for the catalyst and reaction scores: precision (P), recall (R) – also called sensitivity – and f-measure (F). In the calculation of these statistics, a true positive (TP) is a correct (according to the known information) positive prediction, a false positive (FP) is an incorrect positive prediction (e.g., a protein is predicted to catalyze a reaction, but it is not known to do so), and a false negative (FN) is an incorrect negative prediction (e.g., a protein is not predicted to catalyze a reaction, when it is known to do so). Precision, recall, and f-measure are defined by:

$$P = \frac{TP}{TP + FP} \quad R = \frac{TP}{TP + FN} \quad F = \frac{2 \times PR}{P + R}$$

In particular, f-measure is important, since it combines recall and precision. It is easy to inflate either precision or recall separately at the expense of the other. For example, a classifier that always predicts "yes" has perfect recall since FN = 0. A classifier that always predicts "no" has perfect precision since FP = 0. Note also that we do not compute the specificity, which gives credit for true negatives, since in the context of this problem the negative set is extremely large compared to the positive set so the specificity would be high even for a poor classifier that always predicts negative.

A. False positives versus discoveries

In imperfect information situations there is an alternative to the standard definition of true positive (TP). In the case of pathway data (and other biological data), we cannot assume that our empirical testing data is complete. Most publications usually report only positive results, so the absence of a protein from a reaction's list of catalysts could indicate that an experiment has not been performed to determine whether or not that protein catalyzes that reaction. Given that the data is incomplete, it may be desirable not to penalize the algorithm for predicting the existence of a catalyst that is not known to catalyze a particular reaction. Such a prediction may not really be a false positive – it may be a *discovery*. On the other hand, every false positive could be treated as a discovery, causing the predictor that always says "yes" to result in a perfect score. This is not desirable either.

We propose a compromise. To predict pathways, an algorithm makes catalyst predictions and reaction predictions. The probability of discovering another protein that catalyses a reaction that is known to occur in an organism is higher than the probability of discovering that a reaction occurs in an organism, when it was previously not known to occur. In other words, small discoveries are more probable than large discoveries. When evaluating reaction scores, false positives should be considered normally, since discoveries are improbable. For example, there is no reaction 5 for *C. elegans* in the Gluconeogenesis pathway. If a predictor predicted such a reaction then it would be considered a false positive reaction. The predictor knows about reaction 5, since the model pathway contains this reaction (from *S. pombe*). Therefore, it would be possible for the classifier to predict a catalyst for it. In fact, to predict such a reaction for *C. elegans*, the classifier would only need to predict that a single protein in *C. elegans* has the same function as a protein in *S. pombe* that catalyzes reaction 5. However, this would be the first protein ever discovered in *C. elegans* that catalyses this reaction, which is improbable. On the other hand, if a reaction is known to exist in an organism and a false positive occurs in predicting a catalyst for it, we could have discovered another protein that catalyses the reaction (many reactions have multiple catalysts). This is a small discovery so it is more probable.

For a molecular biologist, a false positive on a catalyst in an organism that is known to have a reaction could be considered a "lead" for an experiment that makes a "small" discovery of a new catalyst for a known reaction in the metabolic pathway of this organism. A false positive on a catalyst in an organism that is not known to have this reaction, is a "lead" for a riskier experiment that could lead to a "large" discovery of a new reaction in the metabolic pathway for this organism.

In this paper, all of the statistics we quote use the traditional (more conservative) definition of false positive, even though some false positives are probably discoveries. Therefore the actual accuracies of our classifiers are probably higher than we report.

B. Experimental data set

To perform an experiment that fairly and reliably tests the utility of the pathway prediction algorithm, we placed some requirements on the data set.

1) There must be at least two instances of every pathway (necessary for cross-validation), though more are preferred to favor fair training data sets.
2) The structure and components should be experimentally verified or at least manually curated - not automatically generated by computational methods.
3) All data must be available in machine-readable format.

The structure and components of every pathway in the data set must be specified.

TABLE I
THE 13 SPECIES SELECTED FOR THE EXPERIMENTAL DATA SET.

Species	Strain
Agrobacterium tumefaciens	*C58 (Cereon)*
Arabidopsis thaliana	
Bacillus subtilis	
Caenorhabditis elegans	
Chlamydia trachomatis	
Drosophila melanogaster	
Escherichia coli	*K-12 MG 1655*
Helicobacter pylori	*J99*
Homo sapiens	
Mycobacterium tuberculosis	*CDC1551*
Mycoplasma pneumoniae	
Saccharomyces cerevisiae	
Schizosaccharomyces pombe	

Many potential data sources were evaluated—for examples see [3]-[7]—but only one satisfied our requirements, the Kyoto Encyclopedia of Genes and Genomes (KEGG) PATHWAY database [3]. A subset of KEGG numbering 10 metabolic pathways spanning 13 organisms—a total of 125 pathway instances—was extracted from the database to be used as our experimental data set. The complete listing of pathways and organisms are in Tables I and II respectively.

Two organisms do not have instances of all 10 pathways. Therefore, only 8- or 9-fold cross-validation was done for those pathways. The missing instances are listed in Table III.

TABLE II
THE 10 PATHWAYS IN THE EXPERIMENTAL DATA SET.

Category	Pathway
Amino acid metabolism	Alanine and aspartate metabolism
	Cysteine metabolism
	Glutamate metabolism
	Methionine metabolism
	Urea cycle and metabolism of amino groups
Carbohydrate metabolism	Aminosugar metabolism
	Citrate cycle (TCA cycle)
	Galactose metabolism
	Glycolysis / Gluconeogenesis
	Propanoate metabolism

TABLE III
SPECIES MISSING TEST PATHWAYS.

Species	Pathways
C. trachomatis	Galactose m., Urea cycle
M. pneumoniae	Aminosugars m., TCA cycle, Urea cycle

VI. CLASSIFIERS

Several different classifiers were implemented, tested, and evaluated in our prototype pathway prediction system. The classifiers were based on three different technologies: BLAST, hidden Markov models (HMM) and Support Vector Machines (SVM). Several of the classifiers used combinations of these techniques. Other classification technology could be used, but these technologies were sufficient to establish the utility of our approach to high-throughput pathway prediction.

A. BLAST-based classification

One approach to the classification problem is to compare the primary sequence of the training proteins—which are known to catalyze a specific reaction node—to the target organism's proteins, and select the most similar ones. BLAST [8] is a tool that performs this type of comparison. Two classifiers based solely on BLAST were implemented.

1) *BLAST nearest-neighbour classifier:* The BLAST nearest-neighbour (NN) classifier selects the protein from the target organism's proteome that is most similar to any of the training reaction's proteins (as determined by BLAST). In other words, for a given reaction and target proteome, the BLAST NN classifier compares all the training proteins to all the sequences of the target proteome, and then *returns the single* protein with the smallest e-value. This simplistic classifier provides a baseline for comparison with the other classifiers. In particular, the classifier's limitation of only selecting a single protein from the target proteome makes it impossible for the classifier to attain good recall scores, since most reactions have more than one catalyst. In addition, the fact that the BLAST NN classifier always makes a prediction – regardless of the dissimilarity of the best-matching protein – undoubtedly harms its precision.

2) *BLAST threshold classifier*: Like the BLAST NN classifier, the BLAST threshold classifier uses BLAST as a metric to compare each training protein to each of element of the target proteome. However, it eliminates some of the BLAST NN classifier's obvious limitations by predicting all those proteins from the proteome whose comparison with any of the training proteins resulted in an e-value no greater (i.e. no worse) than a significant threshold.

B. Profile-HMM-based classification

In cases where the functionality of a protein depends mainly on a small conserved portion of its amino acid sequence (called a motif), profile HMMs may be more sensitive than alignment methods such as BLAST for identifying candidate catalysts. Profile HMMs can weigh similarity to these conserved regions more heavily than similarity to the rest of the sequence. They are constructed to recognize recurring protein segments, or motifs, that are common to most of the model catalysts.

To use profile HMMs for pathway prediction, we built a profile HMM model for each reaction in the training pathway. This process involved computing a multiple alignment of all the training reaction's catalysts with ClustalW [9], and then using HMMER [10] to build a profile hidden Markov model. The model was calibrated [10] to determine score significance.

1) *Profile HMM Threshold Classifier*. Our Profile HMM classifier uses HMMER to iterate over the organism's proteome and calculate for each protein the likelihood of being emitted by the training reaction's hidden Markov model. Like BLAST, HMMER returns an e-value for each protein (although the e-values for the two tools do not have exactly the same meaning), so our predictor uses a threshold to filter out proteins that do not match well enough.

2) Mixing BLAST and HMMs: The evaluation of both the BLAST threshold and the HMM classifiers shows a significant trade-off between precision and recall. In an attempt to take advantage of the strengths of each, we implemented a classifier that combines the BLAST and HMM threshold classifiers (our BLAST-HMM classifier). The BLAST-HMM classifier's prediction consists of the intersection of the predictions of the two component classifiers. This classification rule implies a tougher standard to be met by proteins before being classified as catalysts, since both BLAST and HMM classifiers need to agree. We used lower e-value thresholds to allow more true positive catalysts to be predicted by each classifier (increasing recall), while the requirement for agreement filtered the extra false positives generated by the individual classifiers because of their lower thresholds (increasing precision).

C. SVM-Pfam-based classification

A Support Vector Machine is a statistical classification technique to compute a separator for a two-class data set. Indeed, our classifier's task is to separate the proteins that catalyze a reaction from the ones that do not. Unlike BLAST and HMMER, the SVM itself is not a sequence analysis technique. Therefore, it cannot work with raw amino acid sequences. Instead, it is necessary to produce a representative feature vector for each protein and use it for training and prediction. Given that motifs and domains often determine enzymatic activity, the Pfam families [11] identify motifs and domains that may be used as features for our problem. The feature vectors used by our SVM classifier consisted of 7,673 boolean values, each stating the presence of absence in the protein of the corresponding Pfam motif.

Using the hmmpfam [10] tool we computed the motifs for each of the 120,054 proteins in our 13 test organisms. The training set for the classifier consisted of feature vectors for all the proteins of all the training organisms. The proteins that catalyze the reaction of interest were positive examples for the training process while the rest were negative examples. The training data set was used to compute a Support Vector Machine using LibSVM software [12]. The SVM was then used to predict which of the target organism's proteins were catalysts for the reaction, given their Pfam motifs.

The unbalanced nature of the training set—there are far more negative samples than positive samples—can be problematic in the application of SVM to this particular problem. In particular, when no perfect separator can be found between the set of catalysts and non-catalysts the SVM training algorithm may find it better – according to its optimization function – to take a small penalty for leaving the few known catalysts on the wrong side of the separator rather than leaving a larger number non-catalysts on the wrong side. To combat this symptom we raised the weight associated with the positive training samples. This raised the penalty for placing a positive training sample on the wrong side of the partition, making the training algorithm behave as if there were more positive samples in the training set.

VII. Experimental Results

The effectiveness of the pathway prediction algorithm using each of the classification techniques described in Section VI was evaluated via the experimental methodology described in Section V. The results of those experiments are presented in this section. In the interest of saving space, only catalyst prediction scores are presented in detail. This test is certainly the more stringent one, since good catalyst predictions will imply good reaction predictions, while the converse is not necessarily true. In fact, comparing the two measurements showed that for all of the tested classifiers the reaction prediction score was always higher than the corresponding catalyst prediction score. Most of the classifiers have parameters that affect their performance. Over the course of these experiments we varied some of them in an effort to obtain the best possible performance from each classifier. Results specific to each classifier type are presented in the following subsections, while all the classifiers' best component and structure scores are summarized together in Table V and Table VI for easy comparison.

A. BLAST NN and BLAST threshold

When testing the BLAST NN classifier, we used default values for all BLAST parameters. With the BLAST threshold classifier, only the e-value was changed. Figure 4 shows the precision, recall, and f-measure statistics for the pathway predictor using these two classifiers. The graph shows how the effectiveness of the classifier is significantly affected by its threshold. The threshold classifier is better than the baseline NN classifier over most of the range of reasonable e-values (10^{-14} to 10^{-128}) when comparing the results by f-measure. The threshold classifier's curves show that a reasonable threshold value needs to be chosen to guarantee good performance, so that enough similarity is required to try to match the functional parts of the protein, but enough variation is permitted to allow for the divergence between species.

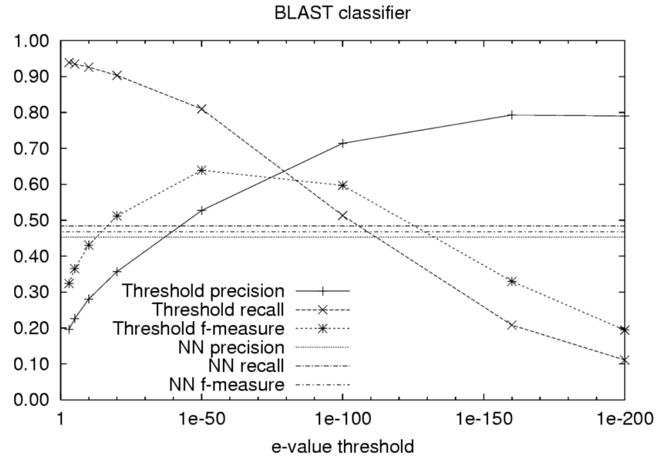

Fig. 4 Statistics for catalyst prediction using BLAST classifiers.

B. Profile HMM

A variety of threshold e-values were used for the profile HMM classifier. The results of the experiments are plotted in Fig. 5. This classifier is slightly more accurate than its BLAST-based counterpart. We see that both classifiers exhibit similar behavior when varying their e-value thresholds, with optimal overall performance at specific e-value.

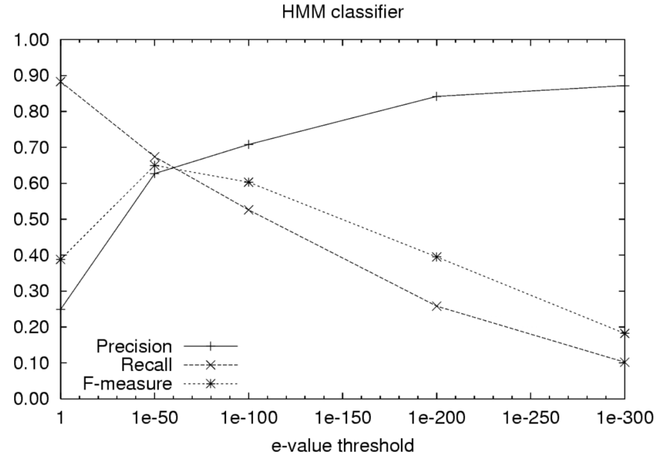

Fig. 5. Statistics for catalyst prediction using HMM classifiers.

C. BLAST-HMM

By intersecting the predictions of the BLAST Threshold and profile HMM classifiers (with lowered respective thresholds), the BLAST-HMM classifier reached an f-measure 2% above the HMM classifier and 3% above the BLAST Threshold classifier. This classifier represents an improvement in precision over both the individual classifiers (13% over BLAST, 3% over HMM), with a small increase in recall when compared to the HMM classifier. The relatively constant recall indicates that there probably is a large overlap between the true positive predictions of the two individual techniques.

D. Motif SVM classifier

The Motif SVM classifier was tested with a linear kernel and varying weights for the positive class instances. The results in Fig. 6 show that a slight increase in the weight of the positive training samples is necessary to obtain good prediction scores. This classifier did not outperform any of the BLAST or HMM classifiers in the catalyst predictions, though it achieved high structure prediction scores.

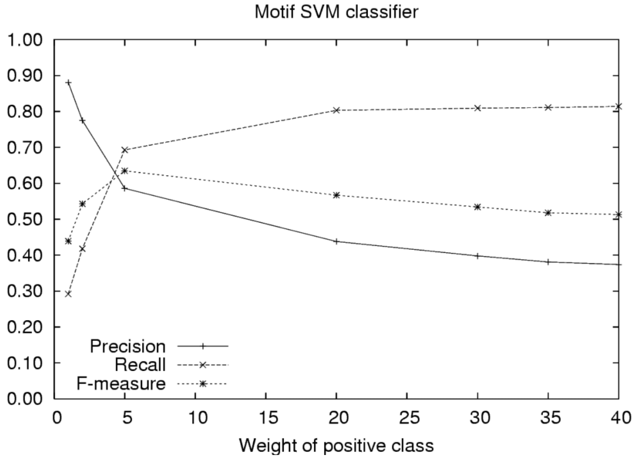

Fig. 6. Pfam motif linear kernel SVM classifier statistics.

VIII. REACTION-SPECIFIC CLASSIFIERS

The experimental results for both the BLAST threshold and HMM threshold classifiers show that different overall e-value thresholds vary the effectiveness of the predictor. However, when analyzing these overall scores, any sense of the success of the classifier at the individual reaction nodes is lost. A more detailed analysis shows that at different reaction nodes, a different e-value threshold maximizes the f-measure. Table IV shows the f-measure scores at different e-values for predicting reaction 1 (EC 1.8.1.4 - Dihydrolipoyl dehydrogenase) and reaction 2 (EC 2.3.1.12 - Dihydrolipoyllysine-residue acetyltransferase) for the *C. elegans* instance of the Gluconeogenesis pathway. This example shows that choosing a single e-value threshold for both nodes results in sub-optimal performance for the two classifiers.

TABLE IV
F-MEASURE METRIC AT DIFFERENT E-VALUE THRESHOLDS FOR THE BLAST THRESHOLD CLASSIFIER.

e-value	f-measure	f-measure
	1.8.1.4	2.3.1.12
0.001	0.250	0.750
1e-10	0.333	**0.857**
1e-20	0.400	**0.857**
1e-50	0.667	0.800
1e-100	**1.000**	0.500
1e-160	**1.000**	0.000

We analyzed the results of our experiments with the BLAST and HMM threshold classifiers and calculated the overall scores that could be achieved by using the best threshold at each reaction node. The results of this analysis are presented in Table V and Table VI, with the other predictors. They are named Opt BLAST and Opt HMM. The results show that the BLAST predictor gains 19% in precision and 12% in recall, while the HMM predictor gains 8% in precision and 17% in recall over their constant-threshold counterparts.

TABLE V
BEST CATALYST PREDICTION SCORES (SELECTED BY F-MEASURE) FOR EACH CLASSIFIER TYPE.

Classifier	F-measure	Precision	Recall
Opt BLAST	0.803	0.715	0.915
Opt HMM	0.767	0.706	0.839
BLAST-HMM	0.667	0.657	0.677
HMM	0.650	0.627	0.674
BLAST Thresh	0.639	0.527	0.810
Motif SVM	0.635	0.586	0.693
BLAST NN	0.468	0.453	0.484

TABLE VI
BEST PATHWAY STRUCTURE PREDICTION SCORES (SELECTED BY F-MEASURE) FOR EACH CLASSIFIER TYPE.

Classifier	F-measure	Precision	Recall
Opt BLAST	0.889	0.857	0.924
Opt HMM	0.864	0.847	0.881
Motif SVM	0.862	0.837	0.889
BLAST-HMM	0.860	0.850	0.871
BLAST Thresh	0.860	0.840	0.880
HMM	0.831	0.880	0.787
BLAST NN	0.665	0.506	0.970

Another interesting observation arises from the histogram of the best thresholds for the BLAST classifier (Fig. 7). The histogram shows two significant peaks, indicating that there are two categories of metabolic reactions. The first one (near 1e-100) is very sensitive to the variations in the catalyst's amino acid sequence. The second (near 1e-3) is rather forgiving of variations, perhaps only being functionally affected by a small section of the protein.

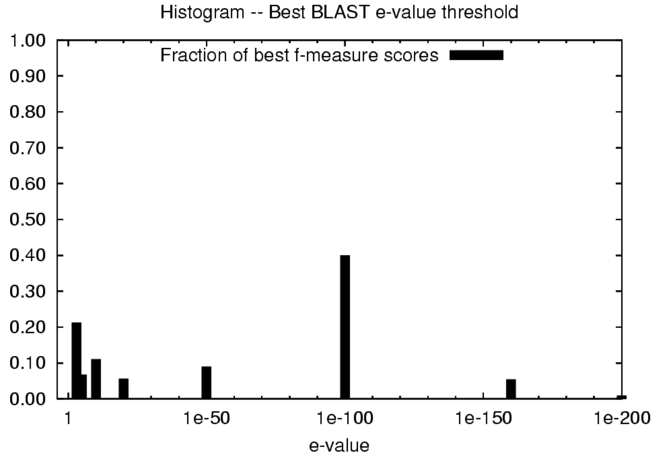

Fig. 7. Best e-values for BLAST threshold classifiers.

This evidence suggests that for accurate pathway prediction, the decision as to whether a particular protein from the target organism is suitable to catalyze a reaction node should be made by a reaction-specific classifier. The classifier should adopt prediction techniques and parameters that are specialized for recognizing proteins that meet its particular requirements. The parameter searching involved in constructing a large number of specialized classifiers may seem like a daunting task – choosing which of the presented classifiers to use, in addition to its particular parameters. However, we believe that this burden can be eased by utilizing machine learning techniques to select and tune the classifiers. We have started this process, but the results are beyond the scope of this paper.

IX. CONCLUSION

In this paper, we have presented a computational technique for predicting metabolic pathway reactions and catalysts that analyse the entire proteomes of organisms. We have shown that our algorithm and the classifiers that it uses can make accurate predictions, as measured by cross-validation. We have also shown that to achieve the best results, it is necessary to use reaction-specific classifiers, or classifiers that are tuned differently for each reaction. We are currently translating our prototype to a web-based tool that we plan to make available on-line.

ACKNOWLEDGMENT

This research was partially funded by research or equipment grants from the Canadian Protein Engineering Network of Centres of Excellence (PENCE), the Natural Sciences and Engineering Research Council of Canada (NSERC), the Informatics Circle of Research Excellence (iCORE), the Alberta Ingenuity Centre for Machine Learning (AICML), Sun Microsystems, Silicon Graphics Inc., and the Alberta Science and Research Authority (ASRA). We would like to thank the rest of the Proteome Analyst research team: Roman Eisner, Russ Greiner, Alona Fyshe, Kurt McMillan, Paul Nalos, Jordan Patterson and Brandon Pearcy for discussions and suggestions during this research.

REFERENCES

[1] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, et al., “Gene Ontology: tool for the unification of biology,” Nat. Genet., vol. 25, no. 1, pp. 25–29, 2000.

[2] C. J. Van Rijsbergen, Information Retrieval, 2nd edition. Dept. of Computer Science, University of Glasgow, 1979. [Online]. Available: citeseer.ist.psu.edu/vanrijsbergen79information.html.

[3] M. Kanehisa, S. Goto, S. Kawashima, Y. Okuno, and M. Hattori, “The KEGG resource for deciphering the genome,” Nucl. Acids Res., vol. 32, no. 90001, pp. D277—280, 2004.

[4] C. J. Krieger, P. Zhang, L. A. Mueller, A. Wang, S. Paley, M. Arnaud, et al., “MetaCyc: a multiorganism database of metabolic pathways and enzymes,” Nucl. Acids Res., vol. 32, no. 90001, pp. D438—442, 2004.

[5] G. Joshi-Tope, M. Gillespie, I. Vastrik, P. D'Eustachio, E. Schmidt, B. de Bono, et al., “Reactome: a knowledgebase of biological pathways,” Nucl. Acids Res., vol. 33, no. suppl. 1, pp. D428—432, 2005.

[6] C. Lemer, E. Antezana, F. Couche, F. Fays, X. Santolaria, R. Janky, Y. Deville, J. Richelle, and S. J. Wodak, “The aMAZE lightbench: a web interface to a relational database of cellular processes,” Nucl. Acids Res., vol. 32, no. 90001, pp. D443—448, 2004.

[7] S. Y. Rhee, W. Beavis, T. Z. Berardini, G. Chen, D. Dixon, A. Doyle, et al., “The Arabidopsis Information Resource (TAIR): a model organism database providing a centralized, curated gateway to Arabidopsis biology, research materials and community,” Nucl. Acids Res., vol. 31, no. 1, pp. 224—228, 2003. [Online]. Available: http://nar.oxfordjournals.org/cgi/content/abstract/31/1/224

[8] S. Altschul, T. Madden, A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. Lipman, “Gapped BLAST and PSI-BLAST: a new generation of protein database search programs,” Nucl. Acids Res., vol. 25, no. 17, pp. 3389—3402, 1997. [Online]. Available: http://nar.oxfordjournals.org/cgi/content/abstract/25/17/3389

[9] J. Thompson, D. Higgins, and T. Gibson., “CLUSTALW: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice,” Nucleic Acids Research, vol. 22, no. 22, pp. 4673—4680, November 1994.

[10] S. Eddy, “HMMER: Profile hidden markov models for biological sequence analysis,” http://hmmer.wustl.edu, 2001.

[11] A. Bateman, E. Birney, L. Cerruti, R. Durbin, L. Etwiller, S. R. Eddy, et al., “The Pfam protein families database,” Nucl. Acids Res., vol. 30, no. 1, pp. 276—280, 2002. [Online]. Available: http://nar.oxfordjournals.org/cgi/content/abstract/30/1/276

[12] C.-C. Chang and C.-J. Lin. (2001) LIBSVM: a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.