A graphical model for predicting protein molecular function

Barbara E Engelhardt BEE@CS.BERKELEY.EDU
Computer Science Division, University of California, Berkeley, CA 94720 USA

Michael I Jordan JORDAN@CS.BERKELEY.EDU
Computer Science Division and Department of Statistics, University of California, Berkeley, CA 94720 USA

Steven E Brenner BRENNER@COMPBIO.BERKELEY.EDU
Department of Plant and Microbial Biology, University of California, Berkeley, CA 94720 USA

Abstract

We present a simple statistical model of molecular function evolution to predict protein function. The model description encodes general knowledge of how molecular function evolves within a phylogenetic tree based on the proteins' sequence. Inputs are a phylogeny for a set of evolutionarily related protein sequences and any available function characterizations for those proteins. Posterior probabilities for each protein are used to predict the molecular function of that protein. We present results from applying our model to three protein families, and compare our prediction results on the extant proteins to other available protein function prediction methods. For the deaminase family, our method achieves 93.9% where related methods BLAST achieves 72.7%, GOtcha achieves 87.9%, and Orthostrapper achieves 72.7% in prediction accuracy.

1. Introduction

The number of sequenced nucleotide sequences encoding proteins is growing at an extraordinarily fast rate due to technologies developed in the last decade that enable rapid sequence acquisition. Such acquisition is a prelude to the understanding of the molecular function and tertiary structure of these protein sequences, and thence to an understanding of the role these proteins play in a particular organism. The experimental technologies that enable us to understand molecular function and structure have, however, not progressed nearly as fast as those for sequencing. One important role of computational biology is to make accurate predictions for these additional properties of a protein based on sequence alone.

A standard approach to the prediction of molecular function is to compare a query sequence to sequences with known function. The underlying assumption is that similarity in sequence implies similarity in molecular function. Whether this assumption is warranted or not, a significant practical problem is that there are very few annotated sequences, and thus any given query protein often does not have a significant similarity with any sequences in the database.

Moreover, pure sequence comparison methods fail to take advantage of one of the most important sources of constraint in biology, the structure of evolutionary relationships among biological molecules. In particular, *homologous* proteins—proteins that are evolutionarily related to each other through a single common ancestor—may have little sequence similarity. And yet homologous proteins often have a similar function, because function tends to be evolutionarily conserved.

Phylogenomics is an approach to the study of molecular function that exploits the observation that protein function and protein sequence tend to evolve in parallel (Eisen, 1998). This observation suggests that a phylogeny built using protein sequence can accurately capture the evolution of molecular function within the homologous proteins, despite the lack of a direct connection between sequence and function. Indeed, there is often enough information contained in a set of aligned homologous sequences to accurately reconstruct a *phylogeny* (a bifurcating tree describing the evolutionary relationships among homologous protein sequences) using one of the many available methods

Appearing in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006.

(e.g., (Felsenstein, 1989; Swofford, 2001)).

Phylogenomics also emphasizes the tendency of proteins to mutate function more rapidly after undergoing a duplication event in a single species than after a speciation event (Ohno, 1972). A gene duplication event creates two copies of a gene in a single genome; if both genes produce protein, then the levels of that protein are often higher than necessary to perform its function. This redundancy takes selective pressure off the individual genes, and one of the copies will often mutate away from the original function; an example relevant to the experiments in this paper is the emergence of adenine deaminase in the deaminase family (Ribard et al., 2003). In speciation events, on the other hand, only one copy of the gene is available in the two genomes. Since no redundancy exists, selective pressure makes a function mutation less likely in this case.

While phylogenomics was developed by Eisen as a manual procedure, more recent work has attempted to automate the concepts of phylogenomics, and thereby to provide tools for phylogenetic-based inference of molecular function at a genomic scale. Resampled Inference of Orthologs (RIO) (Zmasek & Eddy, 2002) and Orthostrapper (Storm & Sonnhammer, 2002) are examples of this effort. Both of these methods bootstrap sequence alignments to produce ensembles of phylogenies, and extract from the ensemble a set of pairwise relationships between a query protein and other proteins. They then use heuristic methods to determine which annotation to transfer, using some combination of average distance in the tree and frequency of not having a duplication in the tree path.

In recent work we have presented an alternative approach to automated phylogenomics that we refer to as SIFTER (Statistical Inference of Function Through Evolutionary Relationships) (Engelhardt et al., 2005). In the current paper we present a novel model for which we retain the name SIFTER, but which provides a more fully statistical methodology for phylogenomics. In particular, the earlier approach was based on a noisy-OR model with user-specified parameters; here we develop an alternative model in which the parameters are learned from data via an EM procedure.

The paper is organized as follows. In Section 2 we briefly outline the phylogenomic method and describe the specific probabilistic approach behind SIFTER. Section 3 presents a set of experiments comparing three state-of-the-art methods (BLAST, Orthostrapper and GOtcha) to SIFTER in the context of three protein families (secretins, deaminases and aminotransferases). We present our conclusions in Section 4.

2. Methodology

The basic phylogenomic flow diagram underlying SIFTER is shown in Figure 1. The first four steps are generic and we briefly describe their implementation in SIFTER in Section 2.1. Step 5 (Function overlay) and Step 6 (Infer function) are more specific to SIFTER and we describe them in detail in Section 2.2 and Section 2.3, respectively.

2.1. Query to phylogeny

A query protein sequence can be matched to a set of homologous sequences using, for example, the HMMER program (Durbin et al., 1999), which finds the best single domain alignment based on Pfam domain profiles. The Pfam database (Bateman et al., 2002) stores over 8000 manually-curated, aligned, homologous protein domains (functional segments of proteins) with their associated species phylogeny, and builds a sequence profile for each one. Once we find that the query protein has one or more Pfam domains, we obtain a set of homologous proteins and an alignment of those sequences using the domain profile.

Given a set of aligned proteins, we perform sequence-based phylogeny reconstruction and reconciliation using standard methods. *Reconciliation* labels internal nodes of a phylogeny with speciation or duplication events based on reconciling the structure of the protein sequence phylogeny with the structure of a species phylogeny, which are inconsistent based on the locations of duplication and deletion events in the history of the protein family (Goodman et al., 1979). Our implementation uses Paup* version 4.0*b*10 (Swofford, 2001), using parsimony with the BLOSUM50 matrix for phylogeny reconstruction, and Forester version 1.92 (Zmasek & Eddy, 2001) for reconciliation. The result is a rooted phylogenetic tree with branch lengths, where each internal node is labeled with either a duplication or speciation event.

2.2. Molecular function: terms and data

Protein function is defined here as the action of a protein *in vivo*. *Molecular function* includes the ability of a protein to bind to or transport a molecule, or catalyze a reaction. To provide a basic set of terms that capture these concepts, we make use of the Gene Ontology (GO) (Ashburner et al., 2002). GO not only provides a vocabulary of basic terms for SIFTER, it also organizes these terms into a directed acyclic graph (DAG), a feature that SIFTER exploits. A four node subgraph of the GO DAG for molecular function (containing 7395 nodes total) is shown in Figure 2.

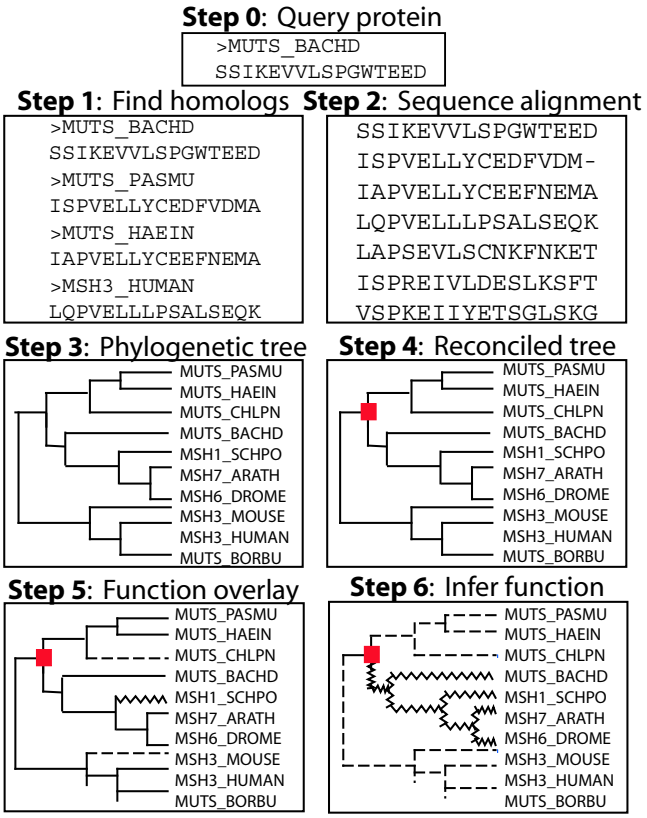

Figure 1. Flow diagram for phylogenomic methodology.

Another virtue of GO is that it is accompanied by a database (the GOA database (Camon et al., 2004)) of function annotations. These annotations are labeled with *evidence codes*, including *IDA* (Inferred from Direct Assay), *TAS* (Traceable Author Statement), *IMP* (Inferred from Mutant Phenotype), *NAS* (Non-traceable Author Statement) and *IEA* (Inferred from Electronic Annotation). The first three codes are those for which an experimental assay was performed; they tend to be correct. The latter two codes are those for which no experimental assay was performed; they are often incorrect. In SIFTER, these annotations are treated as likelihoods, and we associate expert-elicited probabilities with each of the codes. Specifically, we use likelihoods of 0.9 for *IDA* and *TAS* annotations, 0.8 for *IMP*, 0.3 for *NAS* and 0.2 for *IEA*.

Given a query protein, we gather a list of candidate molecular functions for the corresponding family of proteins by taking the union of all experimental GO annotations associated with the proteins in this family (e.g., the subgraph in Figure 2). We prune this list so that only the function terms at the leaves of the DAG are left in the list and call this set of terms *candidate functions* (e.g., the double ovals in Figure 2). This choice of terms distinguishes SIFTER from many other protein function prediction methods. We view the most specific terms as being the hardest to predict and of the greatest utility for biologists. In Figure 2, for example, biologists may be able to readily infer that a query protein is a G-protein-coupled receptor (GPCR) based on an annotation from a distant homolog. However, often attention will center on the more difficult problem of differentiating glucagon receptor activity from parathyroid hormone receptor activity in a single protein. Many fewer annotations are available for transfer at this level of specificity; moreover, mutations occur more rapidly at this level. Furthermore, the first function is involved in insulin regulation where as the second regulates bone growth. Although they are nearby ontologically, it is important for biological research to assess which of the two different roles a query protein performs *in vivo*.

Annotations at nonterminals in the DAG provide evidence for the leaves below those nonterminals. This is achieved by treating evidence at an ancestor node as evidence for all possible combinations of its descendants, according to the distribution $Q(S) = 1/\eta^{|S|}$, where S is an arbitrary non-empty subset of the descendant nodes, $|S|$ is the cardinality of that subset, and the value η is fixed by the requirement that $\sum_S Q(S) = 1$. Annotations are combined at a given node by taking one minus the product of their errors (where error is one minus their probabilities).

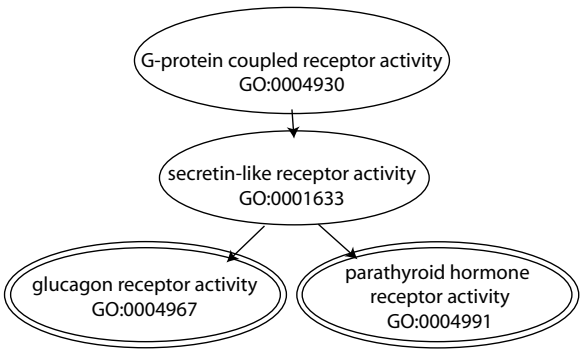

Figure 2. A segment of the GO DAG representing terms used in the secretin protein family. The double circles represent the candidate functions used in SIFTER. The arrows here represent "is a" relationships between the terms, with the more general terms as the ancestors.

2.3. Model specification

Classical phylogenetics uses probabilistic methods to model the evolution of states along the branches of a tree and to infer ancestral states in that tree. SIFTER borrows this machinery in the service of a procedure for inferring molecular function. Specifically, having assembled the candidate functions for a query protein, we treat that list as a Boolean state vector for phylogenetic analysis. Note in particular that the representation of function as a Boolean vector implies that multiple functions can be asserted as present in a single protein; there is no assumption of mutual exclusivity.

Given a Boolean vector of functions at a given node in the phylogeny, we modeled the conditional probability of a single function at a child node as follows. Let X_i denote the Boolean random vector of candidate functions associated with protein i. Let M denote the number of components of this vector; this is the number of candidate functions. The conditional probability of protein i having a function m present (represented by Boolean random variable $X_i^m = 1$) conditioned on its parent protein π_i and the branch length b_i is modeled as follows:

$$\begin{aligned}
\theta_{i,m} &= p(X_i^m = 1 \,|\, X_{\pi_i} = x_{\pi_i}, b_i, \sigma, \phi) && (1)\\
&= \mathbf{1}\left[X_{\pi_i}^m = 0\right](1 - e^{-b_i\sigma}) \prod_{n \neq m}^{M} e^{-\phi_{n,m} x_{\pi_i}^n} && (2)\\
&+ \mathbf{1}\left[X_{\pi_i}^m = 1\right](1 - e^{-\phi_{m,m}}(1 - e^{-b_i\sigma})), && (3)
\end{aligned}$$

where the model parameters are a symmetric, nonnegative matrix of parameters $\phi = \{\phi_{1,1}, \phi_{1,2}, \ldots, \phi_{M,M}\}$ and a nonnegative rate parameter σ. Overall, the model has $\frac{M(M-1)}{2} + 2$ parameters.

To interpret this probability model, consider the case in which the immediate parent of X_i^m ($X_{\pi_i}^m$) has value 0. From Equation 2, we see that the probability of X_i^m being 1 is the probability of mutating at least once within time interval b_i, $(1 - e^{-b_i\sigma})$, scaled by the product of the probabilities that all functions n that are 1 in the ancestor protein mutate to function m. When $X_{\pi_i}^m$ has value 1, Equation 3 shows that the probability of X_i^m being 1 is the probability of not mutating within time interval b_i ($e^{-b_i\sigma}$) scaled by the probability of retaining function m ($e^{-\phi_{m,m}}$).

In this model, the parameter σ captures the rate of mutation. There are actually two different rate parameters in the model, which we henceforth denote as $\sigma_{speciation}$ and $\sigma_{duplication}$. That is, the mutation rate is indexed by the type of the internal node in the phylogeny—speciation versus duplication.

The overall probability model for protein i is obtained by taking a product over the M possible functions:

$$p(x_i|\theta_{i,m}) = \prod_{m=1}^{M} \theta_{i,m}^{x_i^m}(1 - \theta_{i,m})^{1 - x_i^m},$$

which reflects a conditional independence assumption for the functions under consideration.

We estimate the parameters ϕ and σ in this model using a generalized expectation maximization (GEM) algorithm. Noting that a phylogeny is a tree in which the nonterminal nodes are unobserved, the E step of the algorithm is simply a standard graphical model inference procedure in which messages are propagated upward and downward in the tree (Felsenstein, 1981). As for the M step, there is no closed-form solution for the maximizing values of the parameters $\Theta = \{\phi, \sigma\}$; thus, we implement a generalized M step via gradient ascent. The gradient for this model is presented in Appendix A. In practice, we take a single gradient step for each iteration of GEM. We stop EM iterations when the sum of the absolute value of the change in parameters is less than some cutoff c. For our experiments, we set the step size ρ to 0.01, and the cutoff c to 0.005. We initialized all of the parameters in the ϕ matrix to 2.0, and initialized $\sigma_{speciation} = 1.5$, $\sigma_{duplication} = 2.0$. We found that the convergence of the algorithm was robust across a range of initializations.

3. Results

We tested the performance of SIFTER on three different protein family sequences and alignments from the Pfam database downloaded October 7, 2005 (Bateman et al., 2002). The annotations are from the GOA database downloaded October 7, 2005 (Camon et al., 2004).

In two of the families below (deaminases and aminotransferases), we use an additional set of experimental annotations, besides those found in the GOA database, derived from manual literature searches, and include them as *TAS* annotations.

We conducted cross validation experiments, which involved removing each protein's annotations from the training set and running EM, then checking whether the maximum posterior probability for that protein using the estimated parameters agrees with the original held-out annotation. If the held-out annotation was not a member of the candidate functions (e.g., the annotations for GPCR in the secretin family, which is an ancestor term of both candidate terms), we did not use that protein for comparison. For each family we ran cross validation with experimental annotations, and also with both experimental and electronic anno-

tations. We used the experimental cross validation result as the gold standard and in our comparisons.

We chose the three protein families because they are fairly well studied due to their biological importance and roles in human disease. Despite this, the problem of predicting protein function for the unannotated members of the families involves sparse data. For the most part, the experiments that we ran on each of the three protein families are identical. But each family posed unique challenges to function prediction, and we discuss each family below focusing on those challenges.

3.1. Methods for comparison

Before we present results on the three families, we describe the three methods we use for comparison and how they were run. BLAST and GOtcha are both methods that transfer function annotations based on sequence comparisons; Orthostrapper comes from a family of methods that rely on phylogenomic assumptions to transfer annotations based on a pairwise similarity heuristic.

3.1.1. BLAST

The BLAST version 2.2.4 (Altschul et al., 1990) assessment was performed on the non-redundant (nr) set of proteins from SWISS-PROT downloaded from the NCBI website on April 27, 2005. We ran BLASTP with an E-value cutoff of 0.01.

For each query protein in the selected families we searched the BLAST output with the most significant E-value (probability of the alignment score based on an extreme value distribution for aligning protein sequences at random) removing any exact matches from the same species to ensure that the query protein did not receive its own database annotation. We used a keyword search with 265 GO terms to extract a set of annotations for each query protein ranked by E-values, facilitated by BioPerl (Stajich et al., 2002). The highest ranked candidate term for a particular family was considered the function prediction.

In practice, BLAST does not select from a set of candidate functions, but transfers a term from the entire set of annotation terms in its protein library. Often, either the most significant non-identity hit or the most significant non-identity annotated hit is used to transfer annotation onto the query protein. Here we transfer annotation from the most significant hit with a candidate term, which increases the accuracy of BLAST, and enables a comparative ROC-type analysis.

3.1.2. GOtcha

We ran the first publicly available version of the GOtcha software (Martin et al., 2004) on each of the three protein families. GOtcha predicts protein function using a statistical model applied to BLAST searches on a manually-constructed database containing complete GO annotations of seven genomes, including GO evidence codes. Because the annotation database is precompiled for fast querying we could not ensure that a query protein was not being annotated from its own annotation in the database. For one set of experiments (labeled GOtcha), we gathered results using annotations with both experimental and electronic evidence codes. For another set of experiments (labeled GOtcha-exp), we gathered results given only annotations with experimental evidence codes. The output is a ranked list of GO terms; we extracted the ranked list of candidate functions from this complete set, breaking ties in favor of the correct term.

3.1.3. Orthostrapper

We ran Orthostrapper (Storm & Sonnhammer, 2002), version from February 6, 2002, on each of the three families. We split the proteins in each family with experimental GO annotations into proteins from eukaryotes and non-eukaryotes respectively. We clustered the bootstrapped analysis according to the *cluster* program in Orthostrapper, using a bootstrap cutoff of 750 and then using a cutoff of 1, resulting in statistically significant clusters (Orthostrapper-750) and non-statistically significant clusters (Orthostrapper-1). In each cluster, we transferred all experimental annotations from member proteins onto the remaining proteins without experimental annotations. If a protein was present in multiple clusters, it would receive annotations transferred within all of those clusters. This method yields an unranked set of predictions for each protein; multiple annotations were resolved in favor of the correct one. We perform cross validation for each protein by removing its annotations and transferring the remaining annotations to make a prediction for the held out protein. The ROC analysis was performed by determining true positive and false positive annotations for all clusters generated by bootstrap cutoffs between 1000 and 0.

3.2. Secretin Proteins

We applied our model to a small subset of secretin proteins (within PF00002) with 14 proteins. This subset was selected by using a strict PSI-BLAST search (Altschul et al., 1997) with seed protein GLP2R_HUMAN. We removed all duplicate sequences

from this original set.

The activity of all secretin proteins is mediated by G-proteins. The glucagon receptor secretin proteins (GO:0004967) regulate blood glucose by controlling the rate of hepatic glucose production and insulin secretion. The parathyroid hormone receptor secretin proteins bind and regulate the parathyroid hormone (GO:0004991). This family plays a role in many human diseases such as osteoporosis. The GO terms associated with these proteins are shown in Figure 2, and their associated phylogeny and annotations from the GOA database are shown in Figure 3.

Cross validation using both electronic and experimental annotations yields 71.4% correct (5 of 7) (Figure 3), with the two proteins producing prediction errors when held out of parameter estimation being GLR_HUMAN and Q5IXF8_MOUSE. When we use only experimental proteins, cross validation yields 100% correct (4 out of 4), where after convergence of the parameters, 100% (15 of 15) of the three held-out electronic annotations are correct in each of the five iterations. This example shows how using a much larger proportion of one function may skew the estimated parameters towards the more highly represented function through the $\phi_{m,m}$ parameters.

Of the related methods, GOtcha-exp, Orthostrapper-750, and Orthostrapper-1 failed to annotate any of the three electronic annotations. GOtcha achieved 25% accuracy (1 of 4), whereas BLAST achieved 100% accuracy (4 of 4).

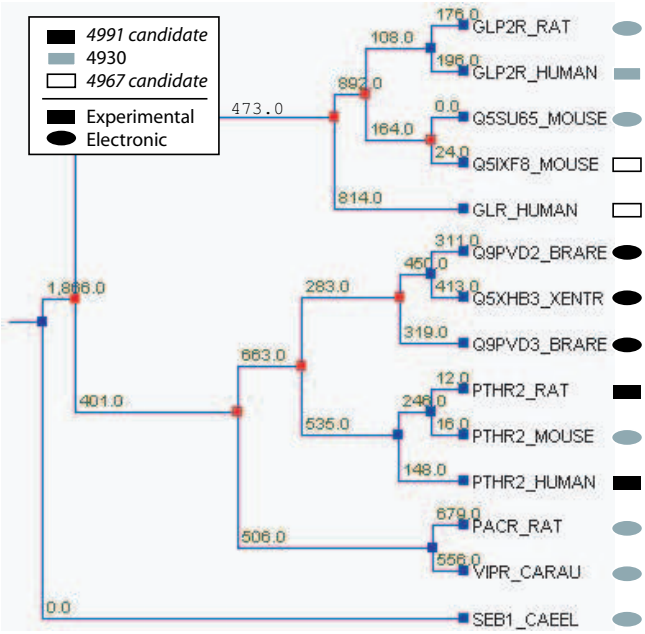

Figure 3. Secretin family dataset. The annotations from the GOA database are represented to the right of the phylogeny. The white and black annotations are in the set of candidate functions and make up the seven annotations used for cross validation; the gray annotations are ancestors of both candidate functions, and are too general to use for prediction. Branch lengths are on each edge. The red squares in the internal nodes represent duplication events; the blue ones speciation events.

3.3. Adenosine/AMP Deaminase Proteins

We applied SIFTER to the Pfam adenosine/AMP deaminase family (PF00962), containing 251 proteins. This family is responsible for removing an amine group from the purine base of three possible substrates. Determining which substrate (adenosine, adenine, or AMP) the protein acts on is critical to understanding its role in the cell, as each of the three substrates are a part of different biological processes. The GOA database contained experimental annotations for 13 proteins, and we found experimental annotations for 20 additional proteins through a manual literature search.

This family has the added difficulty of a subset of proteins with multiple functions. The second function is growth factor activity, conferred through an additional domain not found in proteins without this activity. Results here are based on the number of proteins with the correct annotation; if a protein had two different types of experimental annotations we considered a prediction correct if it was correct for one of the two.

Cross validation on experimental annotations yields 93.9% accuracy (31 out of 33). Cross validation on experimental and electronic annotations yields 96.3% accuracy (156 of 162). The comparison on the experimental annotations show that BLAST and GOtcha-exp achieve 66.7% accuracy (22 of 33), GOtcha achieves 87.9% accuracy (29 of 33), and Orthostrapper-1 achieves 78.8% accuracy (26 of 33). For GOtcha-exp, we broke ties in favor of the correct function 14 times over the 33 proteins.

The ROC analysis in Figure 4 uses a cutoff to determine the correct functions. It is a better method for comparison in this multifunction family because it accounts for accuracy in predicting both functions when multiple functions exist, by including them in the true positive and false negative count. In the figure, SIFTER outperforms all of the methods on this family.

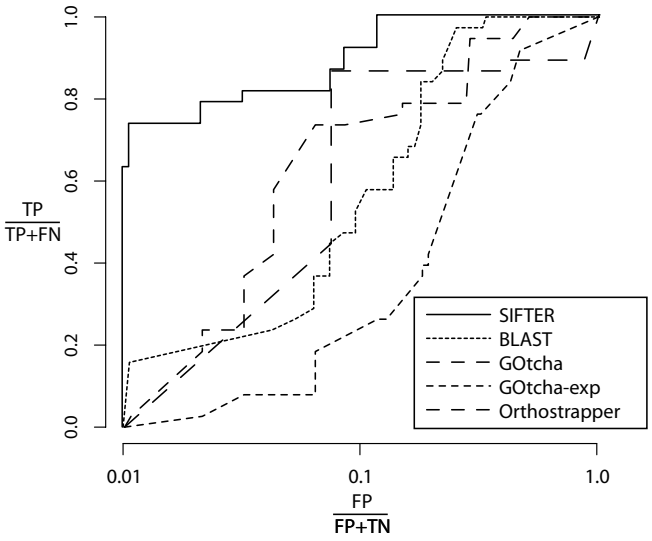

Figure 4. ROC figure comparing the five different methods on the deaminase family experimental annotations. There were 33 proteins annotated, with a total of 38 annotations (five proteins had multiple function annotations). In the axes, TP denotes true positives, FP denotes false positives, TN denotes true negatives, and FN denotes false negatives. Note that the X-axis is on log scale.

3.4. Aminotransferase Proteins

We applied SIFTER to a subset of the Pfam aminotransferase family (PF00155) containing 90 proteins. This subset of the family was chosen using an iterative SATCHMO alignment (Edgar & Sjolander, 2003) to AATC_PIG of all Swiss-Prot proteins with "aminotransferase" in their annotations, and selecting the aminotransferase group Ia based on that alignment. This family is responsible for catalyzing the transfer of a nitrogenous group from a donor to an acceptor. There are two aminotransferases (ATases) in this family: aspartate aminotransferases (AATases; GO:0004069) and tyrosine aminotransferase (TATases; GO:0004838). In the GOA database, there are three experimental annotations supporting AATase proteins, but none supporting TATase proteins. We used five additional AATase and four TATase experimental annotations, based on manual literature curation, making 12 experimental annotations.

This family is difficult for function prediction using evolutionary assumptions because it contains a significant amount of homoplasy. *Homoplasy* occurs when a feature arises independently in different locations on the phylogeny; here it appears that the TATases have arisen multiple times in the phylogeny.

Because of the large amount of homoplasy, cross validation with only experimental annotations yields 75% correct (9 of 12); one of the errors was a TATase protein whereas two were AATase proteins. The cross validation with both experimental and electronic annotations yields 92.6% correct (50 of 54), getting the single TATase electronic annotation correct.

Neither BLAST nor either version of GOtcha predicted TATases for any of the experimental annotations. Hence, BLAST, GOtcha, and GOtcha-exp achieve 66.7% accuracy on the experimental annotations (8 of 12) by predicting only AATases. Orthostrapper-1 has a single cluster with all but one of the 12 experimental annotations; every correct annotation was a result of a tie broken in favor of the correct annotation. Orthostrapper-750 was not able to annotate any of the 12 proteins with experimental annotations.

4. Conclusions

We have described a methodology to predict protein molecular function given sequence and available molecular function annotations from homologous proteins. Evolutionary information and molecular function terms are incorporated in a graphical model of molecular function evolution. We use GEM to estimate the parameter values. We compare our method to state-of-the-art methods and outperform them in prediction on three diverse real world protein families.

We are involved in ongoing work to investigate the appropriateness of applying this method to predict other characteristics of a protein sequence that may evolve in parallel with protein sequence, such as transmembrane region boundaries or other tertiary protein structure characteristics.

5. Acknowledgments

The authors would like to thank Kathryn Muratore and Jack Kirsch (aminotransferase family), and Nandini Krishnamurthy and Kimmen Sjölander (secretin family) at UC Berkeley for their generous help. BEE was funded through the Google Anita Borg Scholarship. MIJ was funded through NIH grant R33 HG003070. SEB was funded through NIH K22 HG00056.

References

Altschul, S. F. et al. (1990). Basic local alignment search tool. *J Mol Biol*, *215*, 403–410.

Altschul, S. F. et al. (1997). Gapped BLAST and PSI-

BLAST: a new generation of protein database search programs. *Nucleic Acids Res*, *25*, 3389–3402.

Ashburner, M. et al. (2002). Gene ontology: Tool for the unification of biology. the gene ontology consortium. *Nat Genet*, *25*, 25–29.

Bateman, A. et al. (2002). The Pfam protein families database. *Nucleic Acids Res*, *30*, 276–280.

Camon, E. et al. (2004). The gene ontology annotation (GOA) database: sharing knowledge in uniprot with gene ontology. *Nucleic Acids Res*, *32*, 262–266.

Durbin, R. et al. (1999). *Biological sequence analysis: Probabilistic models of proteins and nucleic acids*. Cambridge University Press.

Edgar, R., & Sjolander, K. (2003). SATCHMO: sequence alignment and tree construction using hidden Markov models. *Bioinformatics*, *19*, 1404–1411.

Eisen, J. A. (1998). Phylogenomics: improving functional predictions for uncharacterized genes by evolutionary analysis. *Genome Res*, *8*, 163–167.

Engelhardt, B. E., Jordan, M. I., Muratore, K., & Brenner, S. E. (2005). Protein molecular function prediction by Bayesian phylogenomics. *PLoS Comp Biol*, *1*, e45.

Felsenstein, J. (1981). Evolutionary trees from DNA sequences: a maximum likelihood approach. *JME*, *17*, 368–376.

Felsenstein, J. (1989). PHYLIP – phylogeny inference package (version 32). *Cladistics*, *5*, 164–166.

Goodman, M. et al. (1979). Fitting the gene lineage into its species lineage: a parsimony strategy illustrated by cladograms constructed from globin sequences. *Syst Zool*, *28*, 132–168.

Martin, D. M. A. et al. (2004). GOtcha: a new method for prediction of protein function assessed by the annotation of seven genomes. *BMC Bioinformatics*, *5*, 178–195.

Ohno, S. (1972). *Evolution by gene duplication*. Springer-Verlag.

Ribard, C. et al. (2003). Sub-families of alpha/beta barrel enzymes: a new adenine deaminase family. *J Mol Biol*, *334*, 1117–1131.

Stajich, J. E. et al. (2002). The BioPerl toolkit: Perl modules for the life sciences. *Genome Res*, *12*, 1611–1618.

Storm, C. E., & Sonnhammer, E. L. (2002). Automated ortholog inference from phylogenetic trees and calculation of ortholog reliability. *Bioinformatics*, *18*, 92–99.

Swofford, D. (2001). *Paup*: Phylogenetic analysis using parsimony*. Sinauer Associates.

Zmasek, C. M., & Eddy, S. R. (2001). A simple algorithm to infer gene duplication and speciation events on a gene tree. *Bioinformatics*, *17*, 821–828.

Zmasek, C. M., & Eddy, S. R. (2002). RIO: Analyzing proteomes by automated phylogenomics using resampled inference of orthologs. *BMC Bioinformatics*, *3*, 14.

Appendix A: Gradient Updates

The update equations for gradient ascent have the following form: for each $m = (1, ..., M)$, $n = (1, ..., M)$, and letting T be the number of nodes (besides the root) in the tree,

$$\begin{aligned}\phi_{n,m}^{(t+1)} &\leftarrow \phi_{n,m}^{(t)} \\ &+ \rho\left(\sum_{i=1}^{T}\left\langle \frac{x_i^m}{\theta_{i,m}}\left((1-x_{\pi_i}^m)(e^{-b_i\sigma}-1)x_{\pi_i}^n p_m\right)\right.\right. \\ &+ \left.\left.\frac{1-x_i^m}{1-\theta_{i,m}}(1-x_{\pi_i}^m)\left((1-e^{-b_i\sigma})x_{\pi_i}^n p_m\right)\right\rangle\right),\end{aligned}$$

$$\begin{aligned}\phi_{m,m}^{(t+1)} &\leftarrow \phi_{m,m}^{(t)} \\ &+ \rho\left(\sum_{i=1}^{T}\left\langle \frac{x_i^m}{\theta_{i,m}}\left(x_{\pi_i}^m(e^{-\phi_{m,m}}-e^{-\phi_{m,m}}e^{-b_i\sigma})\right)\right.\right. \\ &+ \left.\left.\frac{1-x_i^m}{1-\theta i,m}\left(x_{\pi_i}^m(e^{-\phi_{m,m}}e^{-b_i\sigma}-e^{-\phi_{m,m}})\right)\right\rangle\right),\end{aligned}$$

$$\begin{aligned}&\sigma^{(t+1)} \leftarrow \sigma^{(t)} \\ &+\rho\left(\sum_{i=1}^{T}\sum_{m=1}^{M}\left\langle \frac{x_i^m}{\theta_{i,m}}\left((1-x_{\pi_i}^m)(b_i e^{-b_i\sigma})p_m - x_{\pi_i}^m e^{-\phi_{m,m}}(b_i e^{-b_i\sigma})\right)\right)\right. \\ &+\left.\frac{1-x_i^m}{1-\theta_{i,m}}\left((1-x_{\pi_i}^m)\left((-b_i e^{-b_i\sigma})p_m\right)+x_{\pi_i}^m e^{-\phi_{m,m}}b_i e^{-b_i\sigma}\right)\right\rangle\right),\end{aligned}$$

for $\rho > 0$, where $\langle\cdot\rangle$ is the expectation under $\Theta^{(t)}$ computed in the expectation step and $p_m = \prod_{n\neq m} e^{-\phi_{n,m}x_{\pi_i}^n}$.