Supplemental material for: Genome-scale phylogenetic function annotation of large and diverse protein families

Barbara E Engelhardt, Michael I Jordan, John R Srouji, Steven E Brenner

Contents

1 Supplemental Introduction

In the manuscript "Genome-scale phylogenetic function annotation of large and diverse protein families", we present SIFTER 2.0, a new method for predicting protein molecular function based on a phylogeny. SIFTER 2.0 includes a new statistical model, chosen for its robustness to noise, that is more general than the previous version of SIFTER and creates a platform where additional biological information can be easily incorporated to aid prediction. SIFTER 2.0 also includes approximate computation of posteriors, which enables a phylogenetic-based protein function prediction method to be applied to large and functionally diverse protein families for the first time. In the main manuscript, we show how the new model for SIFTER, using exact computation, produces results comparable to the previous version of SIFTER (where the complete details of the two supporting experiments are in this supplement). We also show that the approximation produces equivalent results at all levels of computation truncation. We then apply the new version of SIFTER using approximation to families that were previously beyond the scope of a phylogenetic-based method for protein function prediction: the Nudix family and a large number of proteins from *S. pombe*. We conclude that this version of SIFTER is capable of genome-scale annotations.

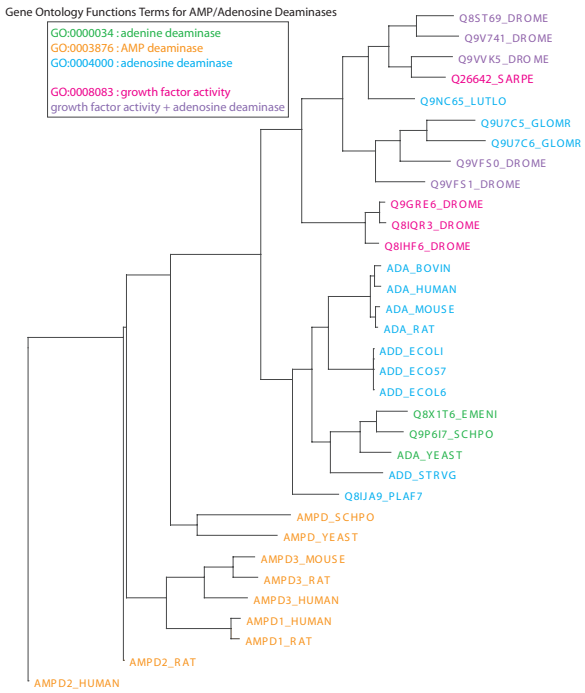
In this Supplement to the main manuscript, we present a number of additional results and discussions, including a complete set of results including parameter estimation results for the deaminase family, additional truncation results for the deaminase and sulfotransferase families, a more detailed look at the large-scale comparison between SIFTER 2.0 and SIFTER 1.1, and a discussion and short analysis of changes in prediction based on the Pfam/GOA database versions. We extend the results for the Nudix family by discussing the diversity of function in the family, detailing how we generalized the functional annotations, and quantifying the benefit of more observations in function prediction. We conclude with a section that goes into depth on the model we used for transforming GO annotations to probabilities

(which has been presented before), we provide a deeper intuition behind the transition rate matrix and the parameters, and we describe the method for estimating those parameters from available data.

2 Results: supplemental information

2.1 AMP/Adenosine deaminase family: complete results

We applied SIFTER 2.0 to the Pfam adenosine/AMP deaminase family (PF00962), which contains 251 proteins in Pfam 18.0. We use an older release of Pfam because of the corresponding gold-standard data set that has been built using these data, in conjunction with a manual literature search and a protein characterization experiment (Engelhardt et al., 2005), and also because Pfam release 24.0 has 1607 sequences, making prediction difficult for some related methods. These proteins remove an amine group from the purine base of three possible substrates: adenine, adenosine, and AMP. There are four candidate functions, three of which are deaminase activity with different substrates. Additionally, a subset of proteins, known as adenosine deaminase-related growth factors (Maier et al., 2005), shows growth factor activity. A phylogeny reconstructed for the 33 proteins with experimental annotations from the GOA database, the literature search, and the characterization experiment (Supplemental Figure 1) shares the branching structure with the phylogeny in a previous study regarding the relative positions of the adenosine, adenine, and AMP deaminases, and adenosine deaminase-related growth factors (Maier et al., 2005). It is hypothesized that adenosine deaminase activity confers growth factor activity through the destruction of adenosine, which induces apoptosis in some types of cells (Maier et al., 2001), so annotations for proteins with only growth factor activity annotations may be incomplete. Besides being an important family in the study of human immunodeficiency disease (Hirschhorn and Ellenbogen, 1986), this family is interesting in the context of evolution because the active site residues are shared across the different substrates (i.e., in all cases the substrate binds to an amine) (Ribard et al., 2003); substrate specificity in this protein is modified by molecular changes in areas not associated with amine binding. Thus a closer look at the active site will not result in better discrimination of the protein substrate but only a general evolutionary divergence.

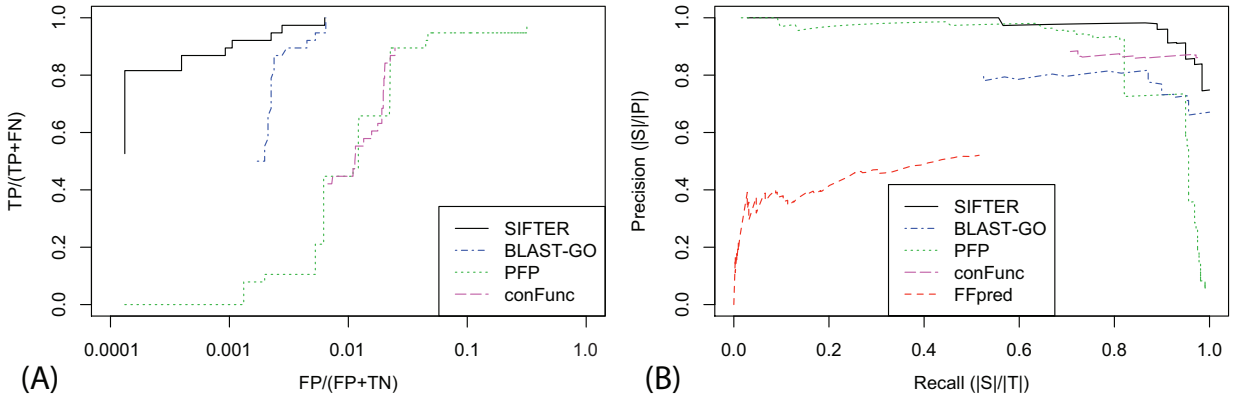


Supplemental Figure 1. Phylogeny of experimentally characterized AMP/adenosine deaminase proteins. The phylogeny of the experimentally characterized set of proteins from the AMP/adenosine deaminase family. The branching structure is the same as that of the full tree used in the SIFTER experiments at the top levels of the phylogeny. The colors indicate the experimentally characterized protein functions, as specified in the key.

Evaluating SIFTER using leave-one-out cross-validation (see Methods) on this family yields 93.9% accuracy (31 out of 33 proteins). Of the two proteins with incorrect predictions, one protein (Q9NC65_LUTLO) with adenosine deaminase activity located near the growth factor activity clade is incorrectly predicted to have growth factor activity (Charlab et al., 2000), and one protein (ADD_STRVG) with adenosine deaminase activity is incorrectly predicted to have activity on adenine. In comparison, BLAST achieves 66.7% accuracy (22 of 33), PFP achieves 78.8% accuracy (26 of 33), conFunc achieves 81.8% accuracy (27 of 33), FFPred achieves 3.0% accuracy (1 of 33), and Orthostrapper achieves 78.8% accuracy (26 of 33).

The ROC-like analysis looks at the relative rate of increase of true positives versus false positives as the cutoff threshold gets more permissive (see Methods in the manuscript for details). In the ROC-like

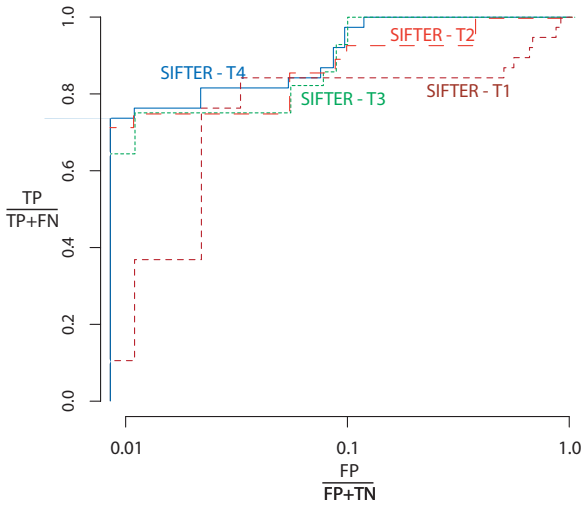
analysis, SIFTER outperforms all of the methods on this family at all error rates (Supplemental Figure 2A). Within the area of high specificity, which is the most relevant area for quantifying performance on biological sequence analysis, SIFTER's performance advantage is striking. The precision-recall analysis (Supplemental Figure 2B) shows that SIFTER outperforms all of the methods at high levels of precision and recall, with conFunc matching SIFTER's performance for recall close to one. FFPred has a strange curve because so few of the proteins had functional predictions.



Supplemental Figure 2. Function annotation methods comparisons on AMP/adenosine deaminase family. Panel (A) shows a ROC-like analysis of results for SIFTER and other annotation methods on the AMP/adenosine deaminase protein family. We did not include FFPred because there were not sufficient numbers of true positive predictions to show up well on this plot. Note that the x-axis is on a log scale. Panel (B) shows a precision-recall analysis of results for SIFTER and other methods on the same family.

To assess the quality of the truncation approximation, we compared the results using approximation against the results using exact computation of posteriors. As with exact computation (level 4), truncation levels 3 and 2 achieved 93.9% accuracy (31 of 33), whereas truncation level 1 achieved 90.9% accuracy (30 of 33), missing one additional protein. The ROC-like analysis (Supplemental Figure 3) shows that the results remain accurate at all levels of truncation. The relatively small size of this family and low functional diversity enabled us to perform two additional experiments. First, we estimated the model parameters from the data itself. Second, we were able to run the previous version of SIFTER on this family, and found that it produced identical predictions and near-identical ROC-like curves to the predictions using exact computation from SIFTER 2.0. These results on this small family (in addition to the broad

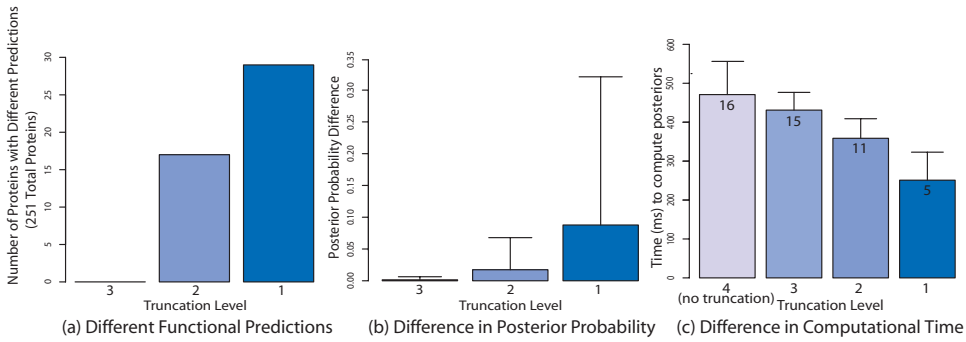
comparison below) serve to illustrate the equivalence of SIFTER 2.0 to SIFTER 1.1, and the high quality of the predictions produced by the approximation.



Supplemental Figure 3. Truncation approximation accuracy in the AMP/adenosine deaminase family. This figure shows the results of the ROC-like analysis on SIFTER leave-one-out cross-validation runs on the AMP/adenosine deaminase family of proteins. The curves are labeled SIFTER-TN where N is the level of truncation (4 is exact computation). Recall there are four candidate functions for the deaminase family. Levels 4, 3, and 2 all achieved the same accuracy (93.9%), and level 1 achieved 90.9% accuracy for leave-one-out cross-validation, where each run estimates the model parameter using GEM. Note that the x-axis is on a log scale.

2.1.1 Parameter Estimation

We ran GEM to estimate the parameters for the AMP/adenosine deaminase family, including all of the available experimental annotations. Leave-one-out cross-validation results (estimating the parameters after leaving each protein's annotations out) yields the same level of accuracy as the standard results, 93.9% (31 of 33). Examination of the parameter estimates for this family gives no obvious insight into how the functions evolved, and one should be wary of interpreting these estimated parameters in an evolutionary light. In particular, the parameter governing the spontaneous appearance of growth factor activity is estimated to be less than a quarter of the corresponding parameter for the other three functions (0.288 versus 1.233 for adenine, 1.204 for AMP, and 1.275 for adenosine). It appears that the growth fac-



Supplemental Figure 4. Truncation approximation performance in the AMP/adenosine deaminase family. There are four candidate functions for the AMP/adenosine deaminase family, which has 251 proteins in our data set. Panel (a) shows the number of inconsistencies in molecular function predictions for every extant protein in this family, truncating at each of the three possible levels for a maximum of 251 possible proteins with functions that were predicted differently than in the exact version. This does not evaluate whether the predictions on the entire family of proteins were correct or not, only that the approximate function prediction for each protein matched the exact prediction. Panel (b) shows the mean absolute difference between the approximate posterior probabilities and the exact posterior probabilities, including the standard deviation of that difference. This figure also is for proteins at the leaves of the phylogeny, and includes bars for each of the three possible levels of truncation as compared to exact computation. Panel (c) shows the average time to compute posterior probabilities for all levels of truncation (including no truncation), averaged over 10 runs. The numbers inside the bars in figure (c) indicate the number of rows and columns of the matrix Q.

tors share a sequence motif, where two of the four conserved residues are also found in the adenosine and adenine deaminase proteins (Maier et al., 2005). This does not differentiate the evolutionary appearance of growth factor activity from substrate evolution in this family. It is possible that the parameter estimates imply that growth factor activity should not be modeled as arising spontaneously, but instead be modeled as evolving from a particular deaminase activity (in this family, adenosine). The scale factors σ_{spe} and σ_{dup} did not provide any interpretable evolutionary insight, as they both converged quickly to the boundary 0.01. On the one hand, this suggests that the role of gene duplication in phylogeny-based function prediction may be overemphasized relative to the evolutionary history of actual function mutations, particularly as early studies focused on families with an atypically low degree of gene duplication (Eisen and Hanawalt, 1999). On the other hand, the large number of false positive gene duplication events in the reconciled trees produced through automated pipelines appears to substantially diminish their signal.

2.1.2 Power Set Truncation Approximation Results

We used the AMP/adenosine deaminase family to test the power set truncation approximation. We computed posterior probabilities based on the parameters previously estimated with no truncation from the complete experimental data set, truncating the number of possible functions predicted for a single protein at 1, 2 and 3. Supplemental Figure $4a$ shows the number of predictions for all 251 proteins that differed (regardless of correctness) from the algorithm with no truncation (i.e., truncation level 4), for each of the three possible levels of truncation. Supplemental Figure $4b$ shows the mean difference and variance in posterior probabilities for the leaf proteins at each level of truncation, as compared to the posterior probabilities computed without truncation at the leaf proteins. Supplemental Figure $4c$ shows the average running time for all of the four possible levels of truncation, with the number of rows and columns of the transition rate matrix embedded in the bars. The impact on the posterior probabilities and corresponding functional predictions for a fixed set of parameters at all but level 1 appears modest.

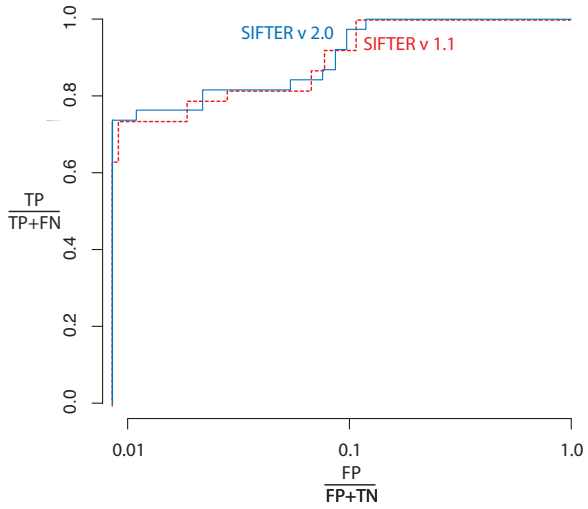
An alternative test of the truncation approximation is to run leave-one-out cross-validation, estimating the parameters with the truncated algorithm at each iteration, for each of the truncation levels. Truncation levels 4, 3 and 2 all achieved 93.9% accuracy (31 of 33), whereas truncation level 1 achieved 90.9% accuracy (30 of 33), missing the additional prediction for protein Q26642_SARPE (predicting adenosine deaminase activity when the experimental annotation is growth factor activity). The ROC-like analysis comparing the different truncation approximations is illustrated in Supplemental Figure 3. As with the results from the previous analysis, the impact of the truncation on all but level 1 appears minimal. Even at level 1 the results are comparable, and the quality of the results is superior to traditional pairwise approaches such as BLAST.

2.1.3 Comparison with previous SIFTER

We compared the new version of SIFTER (version 2.0) with the previous version of SIFTER (version 1.1) (Engelhardt et al., 2006) on the AMP/adenosine deaminase protein family. We computed the accuracy for leave-one-out cross-validation on the deaminase protein family (running GEM for each iteration, with no truncation), finding that SIFTER version 1.1 had 93.9% accuracy (31 of 33) and SIFTER version

2.0 also had 93.9% accuracy (31 of 33), missing the same two proteins. The performance of the two methods are almost identical and show no relevant differences in the ROC-like analysis (Supplemental Figure 5).

In terms of computation speed, SIFTER version 1.1 averaged 296.2ms with 41.6ms standard deviation for 10 iterations of exact computation on the deaminase family, whereas SIFTER version 2.0 averaged 455.3ms with 55.3ms standard deviation for identical 10 runs on the same computer. The maximization step for GEM averaged 11.4ms for SIFTER version 1.1, and 13.8ms for SIFTER version 2.0.



Supplemental Figure 5. ROC-like comparison of SIFTER version 1.1 and SIFTER version 2.0 on AMP/adenosine deaminase family. A comparison of SIFTER version 2.0 with SIFTER version 1.1 on the AMP/adenosine deaminase family of proteins. The curve for SIFTER version 1.1, as described in (Engelhardt et al., 2006), is almost identical to that of SIFTER version 2.0, as described here. Note that the x-axis is on a log scale.

2.2 SIFTER 2.0 compared with SIFTER 1.1 on one hundred Pfam families

To perform a more thorough comparison of the old version of SIFTER (version 1.1) with the new version of SIFTER (version 2.0), we built 100 Pfam families from Pfam release 24.0, and compared leave-one-out cross-validation prediction accuracy for the two SIFTERs on the proteins with experimental evidence from the GOA UniProt 80.0 database. The SIFTER files, including both the annotation file and the reconstructed phylogeny for each of the 100 families, are available for download at

http://sifter.berkeley.edu, and will work for both versions of SIFTER. Note that we did not reconcile these trees, setting each of the internal nodes to be a speciation event rather than a duplication event, for two reasons: first, the reconciliation methods produced so many false positive duplication events, the actual signal is apparently overwhelmed by noise; second, Pfam no longer releases species trees for each of their families, so these species trees are no longer readily available. These families were chosen to have between two and eight candidate functions, with no limit on their family size. We used the Pfam-A alignments, and reconstructed the phylogenies using FastTree 2 (Price et al., 2010) with the default settings.

SIFTER version 1.1 and SIFTER version 2.0 made predictions for 1632 proteins with experimental annotations across the 100 families in cross-validation runs. SIFTER 2.0 achieved 72.5% accuracy (1183 of 1632), whereas SIFTER 1.1 achieved 70.0% accuracy (1142 of 1632); the two versions agreed on 95.3% of the predictions. We found that SIFTER 2.0 using exact computation took approximately twice as long as SIFTER 1.1, where the bulk of the difference in time SIFTER spent on the most functionally diverse families; if we limit the families to ones with 6 or fewer candidate functions rather than 11, SIFTER 2.0 takes only 2% longer than SIFTER 1.1. These results illustrate that the new model for SIFTER produces equivalent predictions based on exact computation as compared to the specialized model in SIFTER 1.1. From this we can safely conclude that they produce generally comparable results, perhaps with a slight accuracy improvement in the more general model.

2.3 Sulfotransferases: additional results

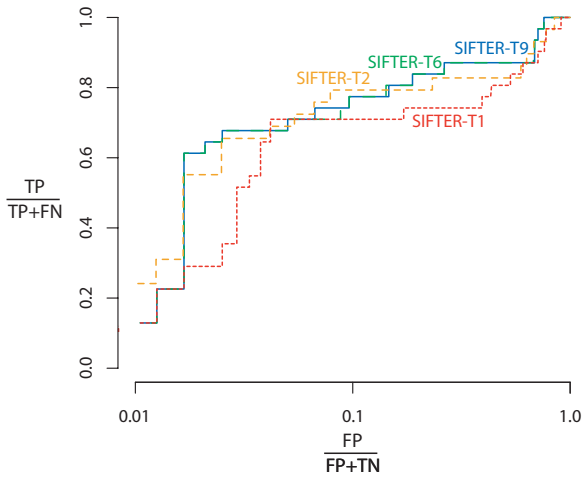
We first applied SIFTER 2.0 to the sulfotransferase family (PF00685) from Pfam 20.0. Our gold-standard data set included 539 proteins and 9 candidate functions in SIFTER. We include here the names of the sulfotransferase proteins for which SIFTER made incorrect predictions using exact computation. The SWISS-PROT identification numbers for the incorrectly annotated proteins from exact computation are ST1E1_HUMAN, ST2B1_HUMAN, CHST1_HUMAN, CHST3_HUMAN, CHST3_RAT, ST1A3_HUMAN, Q91W19_MOUSE, ST1A1_MOUSE, and Q8BT67_MOUSE, which include the five proteins with unique annotations (the first five on this list) as anticipated. Of the proteins that SIFTER

could plausibly annotate correctly given the set of candidate functions in the leave-one-out type analyses, 84.0% (21 of 25) were correct. BLAST made correct predictions for six proteins that were missed by SIFTER, including ST1A3_HUMAN, ST1E1_HUMAN, Q8BT67_MOUSE, ST2B1_HUMAN, CHST1_HUMAN, and CHST3_HUMAN, four of which are proteins with unique function annotations.

The ROC-like analysis for this family (shown in Supplemental Figure 4) at different levels of truncation shows that the SIFTER results do not degrade quickly when truncation is increased. Even at $T = 1$, the ROC-like analysis shows good results on this diverse family. Furthermore, the approximation improved the run time by a significant margin—500-fold in the case of of $T = 2$—with minimal reduction in results (Supplemental Figure 5).

2.3.1 Non-experimental annotations

As discussed above, five proteins could not possibly be correctly predicted by SIFTER in the leave-one-out cross-validation, because each is the only protein with its particular experimental annotation. We investigated whether including non-experimental annotations might enable these to be predicted correctly in these experiments. Including non-experimental annotations as observations does not yield significant improvement in the results. We ran leave-one-out cross-validation on the set of proteins with experimental annotations and electronic (i.e., *IEA*, with a probability of correctness set to 0.2) annotations at truncation level 2, obtaining 73.3% accuracy (22 of 30). This experiment predicted proteins ST1A3_HUMAN and ST2B1_HUMAN correctly, and CHST7_HUMAN incorrectly, as compared to the non-truncated experiments using only experimental evidence. Although one would hope that including electronic annotations would mitigate the problems associated with unique experimental annotations by including some of the same electronic annotations for the same functions in this diverse protein family this was the case for only one of the five proteins with unique experimental annotations (ST2B1_HUMAN). This may be because, in certain families such as this one, GO experimental evidence is often for a more specific term in the GO hierarchy than the non-experimental evidence, thus there are still few or no examples of the appropriately specific term.

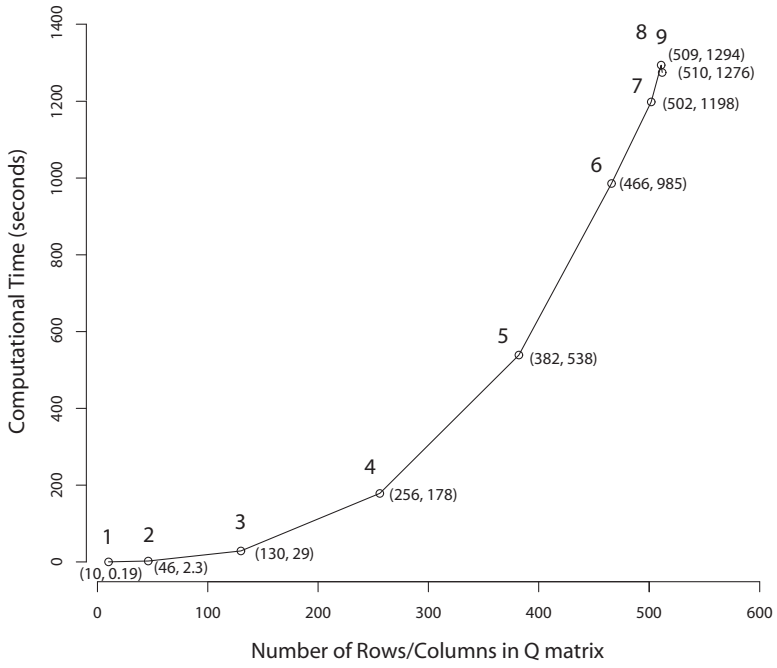


Supplemental Figure 6. SIFTER **truncation approximation comparisons for the sulfotransferase family.** This figure shows a comparison of different levels of truncation in the SIFTER approximation for the sulfotransferase family of proteins. Truncation level is indicated by T followed by the truncation level. Note that the x-axis is on a log scale.

2.3.2 Pfam/GOA versions

We can examine differences in the sulfotransferase results between Pfam release 20.0 and Pfam release 24.0 to try to infer how SIFTER's results are impacted by different versions of the databases. For both versions, we used the SIFTER default parameters; however for the runs on release 24.0 we did not reconcile the gene and species trees and instead set all internal nodes to be speciation events (we do not believe this difference meaningfully impacted the comparison). Furthermore, for the latter release, we had to use SIFTER's truncation approximation ($T = 1$) because of the prohibitively large number of candidate functions.

For the sulfotransferases, release 20.0 has 539 proteins, and release 24.0 has 2317. The more recent version of the GOA database includes new experimental annotations for previous members of the family, including for proteins from *Drosophila melanogaster*, zebrafish, slime mold, and *Arabidopsis thaliana*. The number of candidate functions in SIFTER is increased from 9 to 15, making exact computation in SIFTER infeasible in the latter version of the data set. The total number of proteins with experimental evidence rose from 48 to 80. Overall, the prediction accuracy increased from 43.8% (21 of 48 with



Supplemental Figure 7. SIFTER **truncation approximation performance for the sulfotransferase family.** This graph illustrates how the time to compute posterior probabilities scales relative to the size of the transition rate matrix Q for the sulfotransferase family. There is a 50–500 times speedup in going from the complete matrix Q to a matrix truncated at $T = 3$ or $T = 2$, with no meaningful loss in accuracy (see previous figure). The truncation level (1-4) and (x, y) coordinates are included at each point for clarity.

experimental evidence, since we did not remove the 18 proteins with only the more general term *sulfotransferase activity* in this experiment) to 52.5% (42 of 80, 22 of which have only the general term *sulfotransferase activity*, a far lower proportion). Overall, the prediction accuracy remained fairly stable for this family.

In a second example, the AMP/adenosine deaminase set of proteins has 251 members in release 18.0 and 1607 members in release 24.0. With six candidate functions now instead of four, and considering only the annotations available in the GOA database instead of the complete collection in our gold-standard data set, this family had 19 of 20 proteins correctly predicted for release 24.0, missing only AMPD1_RAT, for which SIFTER predicted AMP deaminase function, but the sequence did not have this (probable) experimental annotation in the GOA database. As with the sulfotransferases, the over-

all prediction accuracy did not change substantially between releases. However, because of signficant changes between versions in Pfam membership, and the difficulty of generalizing from two examples, there is not sufficient evidence that the Pfam or GOA versions will not substantially impact SIFTER's results across all families.

2.4 Nudix family: additional results

The Nudix hydrolase family (PF00293) includes 3703 proteins in Pfam release 20.0. The 66 candidate functions and large family size (compared to the other families studied here) produced a rich phylogeny with intriguing possibilities for further investigation. One observation is that many proteins with identical or similar functions cluster tightly in certain areas in the tree, in particular nucleotide-sugar diphosphatase (pink terms), diphosphoinositol polyphosphate diphosphatase (aqua terms), coenzyme A diphosphatase (gray terms), and diadenosine polyphosphate hydrolase activities (forest green terms). NAD diphosphatase activities are interestingly split into two clades, one of which is composed of proteins that are predominantly specific only for NAD-related compounds, while the other is made up of hydrolases that are also active on ADP-ribose and other dinucleoside polyphosphates. A grouping of mostly ADP-ribose diphosphatases in the middle of the tree is unique in that it clusters tightly, it is distant from other nucleotide sugar diphosphatases, and, moreover, within this clade the eukaryotic and bacterial/viral hydrolases are in two distinct groupings. In addition, most non-ADP-ribose diphosphatases cluster distantly from ADP-ribose diphosphatases.

A few particular proteins are worth noting. DIPP_ASFB7 is the only diphosphoinositol polyphosphate diphosphatase that does not cluster with other proteins of the same function, but instead is closely aligned with another viral hydrolase demonstrating quite different functions (Y06L_BPT4). Another protein of note is Q9RVP7_DEIRA, a nucleoside *di*phosphate diphosphatase that is closely related to three nucleoside *tri*phosphate diphosphatases, perhaps pointing to a similar catalytic mechanism for these four proteins.

In our Nudix family results for SIFTER, the average time for computing the posterior probabilities for all nodes in this tree was 146.78 seconds with a standard deviation of 0.62 second, as averaged over

the 97 runs involved in leave-one-out cross-validation.

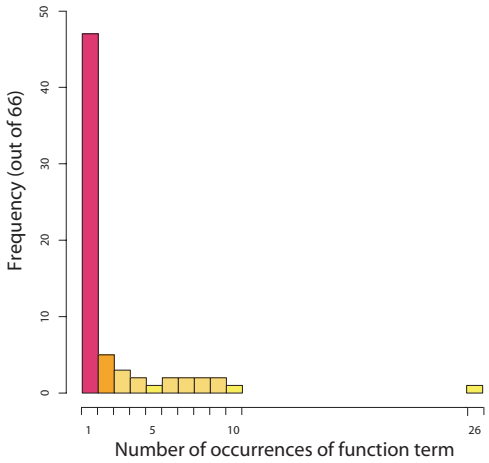
2.4.1 Functional diversity in the Nudix family

The large functional diversity in the Nudix family is the main reason for difficulty in inferring molecular function. In this family, our data set labels five proteins (Q4U4W6, Q53738, Q81EE8, P32056, and O35013) with single, unique functions (i.e., they are the only protein in that tree to have that experimental annotation). In the case of protein O35013, it is labeled with four functions that are all unique in the Nudix family. Furthermore, there are 47 function terms that only appear once in the annotated proteins, most of which co-occur in proteins with more common annotations (Supplemental Figure 8). Most functional terms occur experimentally in this family once or twice, with the single extreme example of ADP-ribose diphosphatase activity occurring experimentally in 26 proteins in the family. The small number of proteins with common functional activity indicates that methods that predict molecular function via annotation transfer will encounter difficulty. It may also reflect limitations of the experimental studies performed in this family to date.

2.4.2 Generalizing Functional Annotations

We wanted to examine the tradeoff between predicting molecular function at a more general level of the GO hierarchy and sensitivity. Within a family, we can selectively generalize some of the functional terms to improve sensitivity when, for example, there exist characterization assays that provide a general screen for particular types of hydrolases. Although developing a method to automatically determine the appropriate level of generalization is beyond the scope of this paper, we manually generalized the candidate functions for a single family to examine the impact on SIFTER's performance. We generalized the leaf terms in the Nudix family candidate functional terms that grouped biochemically in the natural way, only collapsing branches of the tree that were descended at least two branches from the most recent common ancestor term. After generalization there were 15 candidate molecular functions, 10 of which are generalized terms and the rest of which are original functional terms.

We ran leave-one-out cross-validation at truncation level 1 on these data, achieving 78.4% accuracy



Supplemental Figure 8. Functional diversity in the Nudix family. This histogram illustrates the number of occurrences of each of the 66 different candidate functional terms in the 97 experimentally characterized proteins. Many of them occur only once; ADP-ribose diphosphatase occurs 26 times. This histogram represents the available characterizations and should not be used to interpret the relative counts of the functions in the entire family, as these counts may be skewed significantly by protein choice, assay difficulty, etc. Protein functions encountered only once cannot be predicted correctly in the leave-one-out experiments.

(76 of 97). Because the generalization reduced the diversity of this family extensively, we also ran leave-one-out cross-validation at truncation level 2, also obtaining 78.4% accuracy (76 of 97). Performing the same generalization for BLAST functional predictions achieves 42.3% accuracy (41 of 97). The ROC-like analysis for this experiment is shown in Figure 7, where SIFTER predicts 43.6% of the annotations correctly at 99% specificity, and BLAST predicts 1.7% of the annotations correctly at 99% specificity. For comparison, the non-generalized version of SIFTER predicts 24.4% of the annotations correctly at 99% specificity, and the non-generalized version of BLAST predicts 2.4% of the annotations correctly at 99% specificity.

The reason that the generalized BLAST performs poorly relative to the non-generalized BLAST at high specificity is that a large number of general but incorrect hydrolase predictions are made in the data set with low corresponding E-values; although these general terms were ignored when the candidate functions were specific terms, they were counted as incorrect when the candidate functions were generalized. Thus the generalized results from BLAST have a large number of false positives with low corresponding

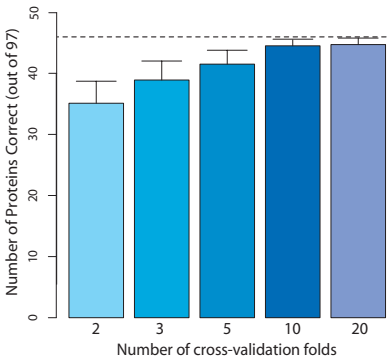
E-values. Looking at the overall graph, though, there appears to be a trade-off between prediction accuracy and the level of specificity of the functional term. These experiments tell us that biologists who need general function predictions for a particular set of proteins can sacrifice function term specificity in return for more accurate predictions.

2.4.3 Evaluating the Value of Observations

To evaluate SIFTER's sensitivity to data sampling, we left out multiple characterized proteins' annotations at each round of cross-validation. Specifically, we ran 2-, 3-, 5-, 10- and 20-fold cross-validation on this data set. In this type of cross-validation experiment, the data are randomly split into K disjoint sets (or *folds*), and the experiment is performed K times, leaving out one of the K subsets on each iteration during the posterior probability computation, and testing the accuracy of predictions on the held-out set. For 2-fold cross-validation, in which one half of the experimental annotations are removed for each run, SIFTER achieved 36.2% accuracy (35.1 of 97), as averaged over ten runs. For 20-fold cross-validation, in which approximately 5 of the experimental annotations are removed at random for each run, SIFTER achieved 46.1% accuracy (44.7 out of 97), as averaged over ten runs. As expected, as more evidence becomes available to SIFTER, the annotations improve up to a certain point (Supplemental Figure 9). At 20-fold cross-validation, the accuracy is slightly less than the leave-one-out cross-validation accuracy, quantifying the value of four additional observations out of the 97 total.

2.5 *S. pombe*: Additional information

The fungal data set included 2800 phylogenies representing as many different Pfam-A domains within the fungal genomes. Of the original $427,324$ proteins from the 46 fungal genomes, $236,854$ proteins contained at least one Pfam-A domain and a family with greater than four members. We include for completeness the set of 46 fungal species used in this analysis, and the phylogeny we used to reconcile each of the protein families against (Supplemental Figure 10).



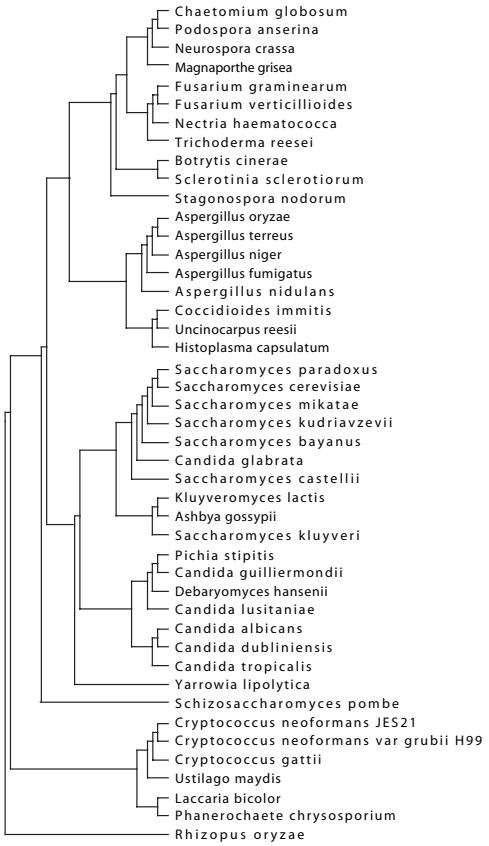
Supplemental Figure 9. Number of correct annotations for SIFTER on the Nudix family of proteins across different numbers of folds. The x-axis of this figure represents five different partitions for cross-validation, from 2-fold to 20-fold cross-validation. The y-axis represents the average number of proteins for which SIFTER correctly predicted the function for each of the different cross-validation tests. The bars shown are the standard deviation for each partition. The dotted line at $y = 46$ represents the performance of leave-one-out cross-validation. All of the different partitions were run ten times.

3 Methods: supplemental information

3.1 Annotations to probabilities

As described in Engelhardt et al. (2005), each protein i is associated with a Boolean random vector X_i, where each Boolean component represents a candidate function that takes value 1 when protein i has that particular molecular function and 0 if that function is not associated with protein i. Because the methods we propose are exponential in the number of *candidate functions*, or the set of molecular functions that represent random variables in the tree, we would like to make this set as small as possible without reducing precision. We can do this using the GO directed acyclic graph DAG structure, by eliminating molecular function terms with deterministic dependencies. For every protein in a family, we associate the experimental annotations with their functional terms in the GO DAG. In the GO DAG, we first prune all ancestors of nodes with annotations (even if the ancestors themselves have annotations), then we prune all non-annotated nodes. This leaves a set of candidate functions that are neither ancestors nor descendants of each other, ensuring there are no deterministic dependencies between them in terms of the semantic network.

We define *children* as immediate descendants of a node, and *parents* as immediate ancestors of a



Supplemental Figure 10. Phylogeny of fully-sequenced fungal genomes. The actual branch lengths were not estimated, as gene-species tree reconciliation does not use branch lengths. This tree was derived from tree reconstruction methods based on concatenating the sequences of 42 genes common to the set of fungal species, and then correcting for an instance of long branch attraction in the Aspergillus clade, as originally in Fitzpatrick et al. (2006). We compared this tree to those found in two other sources (Stajich, 2006; James and et al., 2006) to build this consensus tree and to correctly insert the species in this study that were not in these original phylogenies.

node; we assume that edges between terms are all "is a" edges, as is true most commonly in the molecular function ontology. Thus, more *specific* molecular function terms are descendants of the more *general* terms. Although we are aware of the limitations of GO, here we assume it is both complete and accurate in order to interpret information from the GOA database in a probabilistic way.

For each protein with experimental evidence, the annotations at pruned ancestor terms in the GO DAG are propagated to the set of descendant candidate functions by effectively marginalizing out the ancestor terms. We gave a probability of correctness of 0.9 to *IDA* and *TAS*, and of 0.8 to *IMP*. When there were

multiple annotations at a single term node, the annotations were combined by multiplying the probability of their errors. Annotations are propagated to the candidate terms by assuming that the probability that children terms have a value 1, when a parent term has value 1, has probability $\frac{1}{r^{|S|}}$. In this equation, $|S|$ is the size of the subset S of children terms and r is the solution to the equation $\sum_{S \in \mathbf{S}} \frac{1}{r^{|S|}} = 1$, where $\mathbf{S}$ is the power set of all children terms of a particular term. As a simple example, when a parent node has annotations with the probabilty of correctness equal to 0.98, and has a single child node with no annotations, then propagating the evidence to the child node will yield an annotation at the child node with probability of correctness equal to 0.98. Note that we set the probability of the empty set to zero, effectively assuming that if a protein has a particular function, it must also have at least one of the function's descendant terms related by "is a" edges. Marginalizing out all of the non-candidate function terms eliminates all deterministic dependencies from the random vector for each protein. The random vectors representing observations of molecular function activity are set to the values from this computation for each protein with experimental evidence. These extant proteins with molecular function observations are among the leaves of the phylogeny.

3.2 Transition rate matrix: motivation

We designed the instantaneous transition rate matrix Q to embody the following semantics. In a single instant, the probability of more than one functional change (i.e., loss or gain of a single function) in a protein is zero. Of course, the probability of these transitions will be non-zero when time $t > 0$ has passed, according to the definition of the matrix exponential. Note in particular that the probability of multiple transitions (the creation of a path between the states with more than one functional change) will be non-zero when some finite period of time has passed. Furthermore, some states are the result of one of multiple possible events. For example, if a parent protein in state 01 transitions to state 11 in the child, the appearance of the function 1 could be a result of function 2 mutating into function 1 while retaining function 2 as well (ϕ_{21}) or the spontaneous appearance of function 1 (α_1). The total probability of a transition is an integral over all possible transitions.

This approach thus also takes into account the possibility of a single change in function over a

finite time period. This models the impact of various changes in protein sequence that control and modify function. An additional domain may be added to a protein in a single mutation event (i.e., a gene duplication or exon shuffling event), conferring an additional molecular function. Mutations of individual nucleic acids (coding for this protein or related proteins) or a change in environment may accumulate to confer enzymatic activity for an additional substrate, or yield (over time) a different chemical reaction entirely. All of these possibilities are implicitly modeled by our particular choice of matrix Q.

Other evolutionary possibilities are not modeled by our choice of matrix Q. In particular, we have assumed that the instantaneous rate of transition between states with more than one difference, e.g., a 01 state and a 10 state, has probability zero. Of course this does not reflect all biological possibilities. There are examples of single nucleotide mutations, an event that would be considered instantaneous, that change specificity from one substrate to another. We have chosen to allow this case to be subsumed by the transition paths implemented by the matrix exponential, in particular a function gain followed by a complementary function loss.

A more general modeling concern may be the simplification of describing a protein performing a certain function as a binary variable. Alternatively, we could model this using a continuous variable capturing the effectiveness of a particular enzyme to catalyze a particular reaction, such as k_{cat}/K_m. It would be possible to use diffusion theory to model this variable as a continuous one, but we have chosen not to go this route for a number of reasons. The primary reason is one of data: there is simply not enough data available for particular enzymes to model this robustly. A more subtle question is whether this feature of a protein evolves in parallel with protein sequence, which impacts the appropriateness of phylogenetic methods for this modified problem.

For a thorough discussion of continuous-time Markov chain as related to evolutionary processes, see (Felsenstein, 2003), Chapter 13.

3.3 Expectation Maximization to estimate parameters

We use generalized expectation maximization (GEM) to estimate paramaters in this model, when parameter estimation is possible. The E-step is the computation of the posterior probabilities for each

unobserved random variable, using the standard message passing algorithm for trees (Felsenstein, 1989). Because there is no simple analytical expression for the matrix exponential function of this transition rate matrix Q, we compute these values numerically for a given Q using the jLapack library (Blount and Chatterjee, 1998). The M-step is implemented using projected gradient ascent (Bertsekas, 1999) for each of the parameters σ, Φ, and α, derived from the gradient of the expected complete log likelihood of the model with respect to each of the parameters. Each step of the gradient ascent is scaled by step size ρ. The parameter constraints mentioned above define the space onto which the gradient steps are projected. The Φ and α parameters are projected via normalization onto an $M + 1$ sided cone defined by the M simplex, and the scale parameters are projected back to 0.01 when they fall below that value.

In practice, we take a single projected gradient step for each iteration of GEM. We stop GEM iterations when the sum of the absolute value of the total change in parameters is less than some cutoff c. In our experiments, we set the step size ρ of the gradient ascent to 0.01 and the cutoff c to 0.0015, but these may vary based on the size of the family and the number of observations. We initialized the parameters to the defaults with the exception of setting $\sigma_{spe} = 0.5$ and $\sigma_{dup} = 0.8$.

3.4 Methods for comparison

3.4.1 BLAST keyword extraction

To build the BLAST annotations based on the written descriptions of molecular function from the non-redundant (nr) database, we manually built a parser to map the written descriptions to a subset of GO terms. For each GO term that was in the candidate functions for the protein families of interest or appeared often in the BLAST search results for proteins in these families, we investigated the different ways that term was expressed in the BLAST results from the nr database; in our mapping a single GO term may be mapped from a possibly large number of keyword terms. Using this mapping then, for each protein we extracted the list of top BLAST hits by E-value (using BioPerl (Stajich et al., 2002)), and mapped that ranked list to the associated GO molecular function term. We visually inspected all of the results to confirm that there were no important GO terms omitted from the keyword mapping. We found the most significant hit with a candidate function annotation and transferred that molecular function prediction to

the query protein with its corresponding E-value. Because of the time-consuming nature of building this mapping, there are a large number of omissions and errors in mapping; however, these errors are mostly to the benefit of the BLAST method results. We wanted to include a source of annotations other than the GOA database for a greater diversity of comparative methods.

3.4.2 Orthostrapper

We ran the Orthostrapper (Storm and Sonnhammer, 2002) version from February 6, 2002. We reconciled the phylogeny using eukaryotes and non-eukaryotes. We clustered the data using a bootstrap cutoff of 1, resulting in non-statistically supported clusters (but with much better results in our analyses than using a bootstrap cutoff of, say, 750). In each cluster, we transferred all available experimental annotations from member proteins onto the remaining proteins without experimental characterizations. If a protein was present in multiple clusters, we transferred all of the annotations associated with each of those clusters to that protein. This method yields an unranked set of predictions for each protein; multiple annotations were resolved in favor of the correct one. We performed 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-like analysis was performed by determining true positive and false positive annotations for all clusters generated by bootstrap cutoffs between 1000 and 0. Because of the prohibitively long run time for large families, we only ran Orthostrapper on the deaminase family.

3.5 Data set preparation

3.5.1 AMP/Adenosine deaminase family

The GOA Uniprot 28.0 database contained experimental GO annotations for 13 proteins in the AMP/adenosine deaminase Pfam family (PF00962) version 18.0, and our literature search revealed experimental annotations for an additional 20 proteins, including our experimental characterization of a *Plasmodium falciparum* protein (Engelhardt et al., 2005), resulting in 33 proteins with experimental annotations. The alignment for the full phylogeny was from Pfam 18.0. The subset of sequences with experimental annotations were aligned using `hmmalign` (Eddy, 1998) with the deaminase HMM profile from Pfam

release 18.0. The phylogenies were reconstructed using PAUP* version $4.0b10$ maximum parsimony with the BLOSUM50 matrix (Swofford, 2001; Henikoff and Henikoff, 1992). This gold-standard family has been greatly extended relative to the family that we built for our original experiments (Engelhardt et al., 2005).

3.5.2 Fungal genomes data

Gene finding was performed in each genome using a number of different methods, including GeneWise (Birney et al., 2004), FgenesH+ (Salamov and Solovyev, 2000), and GLEAN (Mackey et al., 2008); see (Stajich, 2006) for complete details.

References

Bertsekas, D. P. (1999). *Nonlinear Programming*. Athena Scientific, New Hampshire, USA, 2nd edition.

Birney, E., Clamp, M., and Durbin, R. (2004). GeneWise and genomewise. *Genome Research* **14**:988–995.

Blount, B. and Chatterjee, S. (1998). An evaluation of java for numerical computing. In *ISCOPE*, pages 35–46.

Charlab, R., Rowton, E. D., and Ribeiro, J. M. C. (2000). The salivary adenosine deaminase from the sand fly. *Experimental Parasitology* **95**:45–53.

Eddy, S. R. (1998). Profile hidden markov models. *Bioinformatics* **14**:755–763.

Eisen, J. A. and Hanawalt, P. C. (1999). A phylogenomics study of dna repair genes, proteins, and processes. *Mutation Research* **3**:171–213.

Engelhardt, B. E., Jordan, M. I., and Brenner, S. E. (2006). A graphical model for predicting protein molecular function. In *Proceedings of the 23rd International Conference on Machine Learning*.

Engelhardt, B. E., Jordan, M. I., Muratore, K., and Brenner, S. E. (2005). Protein molecular function prediction by bayesian phylogenomics. *PLoS Computational Biology* **1**:e45.

Felsenstein, J. (1989). Phylip – phylogeny inference package (version 32). *Cladsistics* **5**:164–166.

Felsenstein, J. (2003). *Inferring Phylogenies*. Sinauer Associates, Inc.

Fitzpatrick, D. A., Logue, M. E., Stajich, J. E., and Butler, G. (2006). A fungal phylogeny based on 42 complete genomes derived from supertree and combined gene analysis. *BMC Evolutionary Biology* **6**:99–114.

Henikoff, S. and Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Science U S A* **89**:10915–10919.

Hirschhorn, R. and Ellenbogen, A. (1986). Genetic heterogeneity in adenosine deaminase (ada) deficiency: five different mutations in five new patients with partial ada deficiency. *American Journal of Human Genetics* **38**:13–25.

James, T. Y. and et al. (2006). Reconstructing the early evolution of fungi using a six-genome phylogeny. *BMC Evolutionary Biology* **443**:818–822.

Mackey, A. J., Liu, Q., Pereira, F. C., and Roos, D. S. (2008). Glean: Improved eukaryotic gene prediction by statistical consensus of gene evidence. *in preparation* .

Maier, S. A., Galellis, J. R., and McDermid, H. E. (2005). Phylogenetic analysis reveals a novel protein family closely related to adenosine deaminase. *Journal of Molecular Evolution* **61**:776–794.

Maier, S. A., Podemski, L., Graham, S. W., McDermid, H. E., and Locke, J. (2001). Characterization of the adenosine deaminase-related growth factor (adgf) gene family in *Drosophila*. *Gene* **280**:27–36.

Price, M. N., Dehal, P. S., and Arkin, A. P. (2010). Fasttree 2 approximately maximum-likelihood trees for large alignments. *PLoS ONE* **5**:e9490+.

Ribard, C., Rochet, M., Labedan, B., Daignan-Fornier, B., Alzari, P., Scazzocchio, C., and Oestreicher, N. (2003). Sub-families of alpha/beta barrel enzymes: a new adenine deaminase family. *Journal of Molecular Biology* **334**:1117–1131.

Salamov, A. A. and Solovyev, V. V. (2000). Ab initio gene finding in drosophila genomic dna. *Genome Research* **10**:516.

Stajich, J. E. (2006). *A comparative genomic investigation of fungal genome evolution*. Ph.D. thesis, Duke University.

Stajich, J. E., Block, D., Boulez, K., Brenner, S. E., Chervitz, S. A., Dagdigian, C., Fuellen, G., Gilbert, J. G. R., Korf, I., Lapp, H., et al. (2002). The bioperl toolkit: Perl modules for the life sciences. *Genome Research* **12**:1611–1618.

Storm, C. E. and 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 (*and other methods)*. Sinauer Associates.