

UNDERSTANDING PROTEIN FUNCTION THROUGH STATISTICAL INFERENCE AND EVOLUTIONARY ANALYSIS

Wyatt Travis Clark

Submitted to the faculty of the University Graduate School
in partial fulfillment of the requirements
for the degree
Doctor of Philosophy
in the School of Informatics and Computing
Indiana University
June 2013

Accepted by the Graduate Faculty, Indiana University, in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Predrag Radivojac, Ph.D.

Matthew Hahn, Ph.D.

Haixu Tang, Ph.D.

Michael Lynch, Ph.D.

June 4, 2013

This work is dedicated to three people that I love dearly.

To my son, Aldaraa Anderson Clark;
the apple of my eye,
from a doting father.
You bring more joy to my life
than you could ever know.

To my good friend, Jason Chaney;
I think of you often;
of windy country roads leading nowhere;
and the appealing, yet impossible, prospect
of turning back time.

To my grandfather, George Anderson;
the eternal tinkerer and
builder of windmills;
from whom I obtained both
my intellectual curiosity,
and desire to never stop learning.

Acknowledgements

I am deeply grateful to my advisor, Predrag Radivojac, who has been an excellent teacher and mentor. The course of my life was irrevocably changed when I took his machine learning class and was then asked to work on a project as an undergraduate researcher. This resulted in me subsequently being asked to continue working with him as a masters student, and then as a PhD student. Over the past 8 years Pedja has been the catalyst behind the growth I've experienced as a scientist and researcher.

In addition to my advisor I have to thank my committee members, Haixu Tang, Matthew Hahn, and Michael Lynch for their contributions. As well as their input regarding my dissertation, each of these individuals has contributed to my growth through their teaching. From Haixu Tang I have learned much about algorithms and probabilistic models. I have tried to absorb as much as I can from Michael Lynch about population genetics and evolution as possible. Matthew Hahn has not only taught excellent courses on evolution, genomics, and population genetics but has also been directly involved in my research on at least one occasion.

I would like to thank Jim Costello, Brian Eades, and John Colbourne who served as mentors for my undergraduate capstone project that focused on using Hidden Markov Models to identify transposable elements. I'd also like to thank Michael Lynch and his student at that time, Sarah Schaack, for supervising me for one summer in my rather feeble attempt to assist Sarah in quantifying transposable elements in the *Daphnia* genome. Not only did I learn a lot about conducting bioinformatics research, the time I spent working in the laboratory served as an informal crash course in population genetics and evolution.

Faculty members Mehmet Dalkilic and Sun Kim both contributed to my education through their teaching and involvement in my qualifying exams. Mehmet was also a research collaborator on several projects; notably the first paper I published as an undergraduate.

I must acknowledge both Iddo Friedberg and Sean Mooney, with whom I have greatly enjoyed collaborating on the CAFA project. I directly benefitted from the hard work that Iddo and Sean, along with my advisor, put into making CAFA successful.

I have to thank my co-author on the ortholog conjecture paper, Nathan Nehrt, who carried out the brunt of the work for that particular publication.

I am grateful for my friends and classmates Jose Lugo-Martinez, Vikas Rao Pejaver, Heewook Lee, James Denton, Taylor Raborn, Nathan Taylor, Sam Miller, Nathan Husted and Anoop Mayampurath; with whom conversation has always been intellectually stimulating and entertaining. Their camaraderie has made graduate school easier to get through, and they will be missed dearly when I leave Bloomington. I would also like to recognize both past and present lab members who have not already been mentioned; Amrita Mohan, Biao Li, Yong Li, Fuxiao Xin, Nils Schimmelman, Xin He, Sujun Li, Arunima Ram, Shuyan Li, Kymberleigh Pagel, Chao Ji, and Yuxiang Jiang for their involvement in my research and for making the Radivojac Laboratory an enjoyable and productive environment to work in.

The staff in the School of Informatics and Computing have been very helpful; especially Dave Cooley, Linda Hostetter, Lynne Mikolon, Bob Konicek, Rob Henderson, T.J. Jones and Patty Reyes-Cooksey .

I am indebted to all teachers at every level; from kindergarden to graduate school. I received a very high-quality public education that served as the foundation for the rest of my career. I especially have to thank Mark Werking, who was an excellent chemistry and physics teacher. Much of what I learned in the several years I was his student made understanding the abstract concepts and forces that influence protein behavior at the molecular level easy. I had very good math teachers in Chris Olinger and Tom Hamilton, who helped instill excellent analytical skills. My French teacher, Wanetta Cheesman, facilitated my growth culturally, especially through organizing a class trip to England France and Switzerland. In the “shop classes” of Bill Bunger, Don Sturgeon, and Harold Lumpkin I developed pragmatic problem solving skills that I use every day in my research. I am

especially grateful for the personal attention that Harold Lumpikin gave me as a student. In addition to the aforementioned high school teachers, I would like to thank Leah Savion; who in addition to being an excellent teacher, guided me towards informatics as an undergraduate. Dennis Groth has influenced not only me, but hundreds of students who have come through the School of Informatics and Computing as well. At the graduate level, Leonie Moyle, Alessandro Flammini, Esfandiar Haghverdi, and Christopher Raphael all taught excellent classes, and demand my highest respect for their teaching abilities and the high quality research they conduct.

I would like to acknowledge my family. Without both of my parents I would have never made it as far as I have. They have both been equally instrumental in my achievements. My mother, Linda Anderson, who pulled herself up from her bootstraps while raising two kids as a single mom, showed me the importance of hard work and determination. My father, Clifford Clark, who I resemble in so many ways, has always been there rooting me on; from little league to my dissertation defense. He has always placed the well being of my sister and myself as his top priority.

Finally, and most importantly, I have to thank my wife, Sergelen Ariunbaatar, who has shown me infinite patience and love. She has always encouraged me to continue when I've become frustrated, and has been extremely understanding of the long hours demanded of a graduate student.

Wyatt Travis Clark

Understanding Protein Function through Statistical Inference and Evolutionary Analysis

The ability to characterize the functional behavior of an individual protein in a variety of different contexts is one of the cornerstones of carrying out biological investigations. Its importance is especially apparent in the context of understanding life at the molecular level, investigating disease, the development of drugs to cure disease, and the manner in which evolution is modulated by changes in genes.

Although a growing number of tools for the task of interrogating a protein's function are at the disposal of experimental biologists, characterizing the exponentially expanding set of known sequences requires the application of *in silico* techniques.

In this dissertation novel methods for the prediction of protein function from protein sequence and structure data are detailed. Current techniques for the evaluation of function prediction are reviewed and several drawbacks are addressed through the introduction of new information content-based metrics that attempt to provide comprehensive assessment of the performance of a given prediction method. Finally, the influences of speciation and gene duplication on the evolution of protein function is investigated by comparing the functional similarity of different classes of homologous sequences.

Contents

List of Figures	xii
Chapter 1. Prediction and analysis of gene function	1
1. Defining function	2
2. Challenges in predicting function	4
2.1. Multi-functional proteins	4
2.2. Transfer of function by sequence similarity	7
2.3. Quality of non-experimental annotations in Swiss-Prot	10
3. State of the art methods	12
3.1. Community-wide assessment of function prediction	13
4. Analysis of function in evolutionary context	14
5. Conclusion	14
Chapter 2. A multi-output neural network for function prediction	17
1. Methods	17
1.1. Problem formulation	17
1.2. Data sets	18
1.3. Data representation	18
1.4. Classification models	19
1.5. Model selection and evaluation	20
2. Results	21
3. Discussion	23
Chapter 3. Property Kernels based on Vector Quantization	26
1. Problem formulation	28
2. Methods	30

2.1.	Count-based property kernels	31
2.2.	Additive property kernels	32
2.3.	Combined kernel function	32
2.4.	Computational complexity	32
2.5.	String kernel	32
3.	Mapping proteins into property vectors	33
3.1.	Structure based properties	33
3.2.	Sequence based properties	34
3.3.	Tested values of window size and number of centroids	34
4.	Experiments and data	34
4.1.	Cross-validation	35
4.2.	Performance accuracy estimation	36
4.3.	Signal-to-noise ratio	36
5.	Results	37
5.1.	Performance of individual properties	37
5.1.1.	SCOP fold	37
5.1.2.	Catalytic activity	37
5.1.3.	Catalytic activity subclass	37
5.2.	Combined kernel performance	40
5.2.1.	SCOP fold	40
5.2.2.	Catalytic activity	40
5.2.3.	Catalytic activity subclass	40
5.3.	String kernel performance	40
5.4.	Optimal parameter values	41
5.4.1.	SCOP fold	41
5.4.2.	Catalytic activity	41
5.4.3.	Catalytic activity subclass	42
5.5.	Comparing AUC and SNR	42
5.6.	Reduced redundancy data set	42
6.	Discussion	44

Chapter 4. Information-theoretic evaluation of predicted ontological annotations	46
1. Background	48
2. Protein function prediction scenarios	50
3. Methods	51
3.1. Calculating the joint probability of a graph	51
3.2. Calculating the information content of a graph	54
3.3. Comparing two annotation graphs	55
3.4. Measuring the quality of function prediction	57
3.5. Weighted metrics	58
3.6. Semantic distance	59
3.7. Precision and recall	59
3.7.1. Information-theoretic weighted formulation	60
3.8. Supplementary evaluation metrics	61
3.8.1. Basic definitions	61
3.8.2. Information-theoretic metrics between pairs of vertices	62
3.8.3. Information-theoretic metrics between pairs of graphs	63
3.9. Additional topological metrics	64
4. Confusion matrix interpretation of ru and mi	64
5. Analysis of semantic distance	65
6. Annotation models	67
6.1. The Naïve model	68
6.2. The BLAST model	68
6.3. The GOtcha model	68
7. Experiments and results	69
7.1. Average information content of a protein	69
7.2. Two-dimensional plots	70
7.3. Comparisons of single statistics	73
8. Discussion	77
Chapter 5. Testing the ortholog conjecture with comparative functional genomic data	82
1. Results	85

1.1.	Functional similarity between orthologs and all paralogs	85
1.2.	Functional similarity between orthologs and subtypes of paralogs	87
1.3.	Microarray-based measures of functional similarity	89
1.4.	Protein interaction-based measures of functional similarity	91
1.5.	Family-based analyses	92
2.	Discussion	94
2.1.	The evolution of gene function	95
2.2.	Implications for protein function prediction	98
3.	Materials and Methods	100
3.1.	Comparative genomics data	100
3.2.	Protein function data	101
3.3.	Calculation of similarity	102
3.4.	Microarray data	103
3.5.	Protein interaction data	105
Appendix.	Bibliography	107

List of Figures

1.1 Growth of databases	2
1.2 Distribution of sequences with experimental annotations amongst model organisms	3
1.3 The distribution of leaf terms counts per ontology	6
1.4 Accuracy of function transfer using global sequence identity	8
1.5 The precision and recall of function transfer using global pairwise sequence identities	9
1.6 Accuracy of function transfer using global pairwise sequence identities for enzymes vs. non-enzymes	10
1.7 Accuracy of function transfer using global pairwise sequence identities from the same or different organism	11
1.8 Accuracy of function transfer between different versions of the Swiss-Prot database	12
2.1 Precision-recall curves for several GO-term prediction algorithms	22
2.2 The performance of FANN-GO and GOtcha on sets of sequences with varying degrees of maximum similarity to other sequences in the data set	24
3.1 Time Series Representation of Protein Properties	28
3.2 A schematic representation of using VQ to encode a sequence represented as a property vector	29
3.3 Description of angles derived from protein structures	33
3.4 SCOP ROC curves	40
3.5 “catalytic activity” subclass ROC curves	41
3.6 Comparison of SNR and AUC	43
3.7 NR40 catalytic activity results	44

4.1 A graph of terms illustrating different vertices used to calculate a full joint probability	52
4.2 A sample ontology and data set illustrating calculating information content	56
4.3 A graph showing terms used to calculate ru and mi	58
4.4 A Venn diagram	66
4.5 Distribution of information content (in bits) of protein experimental annotations	70
4.6 Plots of $(pr(\tau), rc(\tau))_\tau$, $(wpr(\tau), wrc(\tau))_\tau$, $(ru(\tau), mi(\tau))_\tau$, and $(wru(\tau), wmi(\tau))_\tau$	71
4.7 Detailed figures showing the ru and mi of baseline methods	73
4.8 Performance of information content-based metrics when using the all-pair method of averaging	74
4.9 Performance of iInformation content-based metric when using the max-average method of averaging	75
4.10 Comparison of different methods of calculating precision and recall	76
5.1 Two different gene tree topologies	84
5.2 Comparison of orthologs and paralogs in four pairs of organisms for Molecular Function annotations	86
5.3 Comparison of orthologs and paralogs in four pairs of organisms for Biological Process annotations	87
5.4 Comparison of orthologs and different subtypes of paralogs in four pairs of organisms for Molecular Function annotations	88
5.5 Comparison of orthologs and different subtypes of paralogs in four pairs of organisms for Biological Process annotations	89
5.6 The relationship between gene expression correlation and sequence identity for human-mouse orthologs and paralogs	90
5.7 Interaction based similarity	93
5.8 The relationship between functional similarity and dS calculated using the Goldman and Yang method for inparalogs only.	98

CHAPTER 1

Prediction and analysis of gene function

Characterizing the functional behavior of individual proteins in a variety of different contexts is an important step in understanding life at the molecular level. Endeavors such as understanding biological pathways, investigating disease, and developing drugs to cure those diseases depend on being able to describe the actions of individual proteins, both in terms of their physiochemical molecular function, involvement in biological processes, and the sub-cellular location at which these actions are carried out. In spite of the fact that there are increasingly more tools for the interrogation of protein behavior, there are still a large number of functionally uncharacterized proteins, with the gap between known sequences and experimentally characterized ones growing at an exponential pace (Figure 1.1). Currently, there are about 50,000 proteins with at least one experimentally annotated Gene Ontology (GO) term in Swiss-Prot [Bairoch et al., 2005]. However, owing to the numerous sequencing projects [Lioliou et al., 2008], the gap between annotated and non-annotated proteins has exceeded two orders of magnitude, and will only become wider. There is also a large amount of disparity in the distribution of experimental annotations among organisms, with most annotations occurring in model organisms (Figure 1.2). Furthermore, while most experimentally annotated proteins come from model organisms, with the exception of yeast, less than half of the genome of any model organism has yet to be assigned experimentally characterized GO functions. The numbers in Figure 1.2 do not illustrate the fact that many annotations are quite vague. Because of the multitude of ways a protein's function can be characterized, having a single annotation does not mean that protein has been fully annotated. For example, 26% of proteins with experimentally characterized molecular function annotations in the January 2011 version of Swiss-Prot have protein binding as their most specific molecular function term.

Because of the growing disparity between known and experimentally annotated proteins, and the financial and ethical restrictions placed on experimentally interrogating function, *in silico* methods

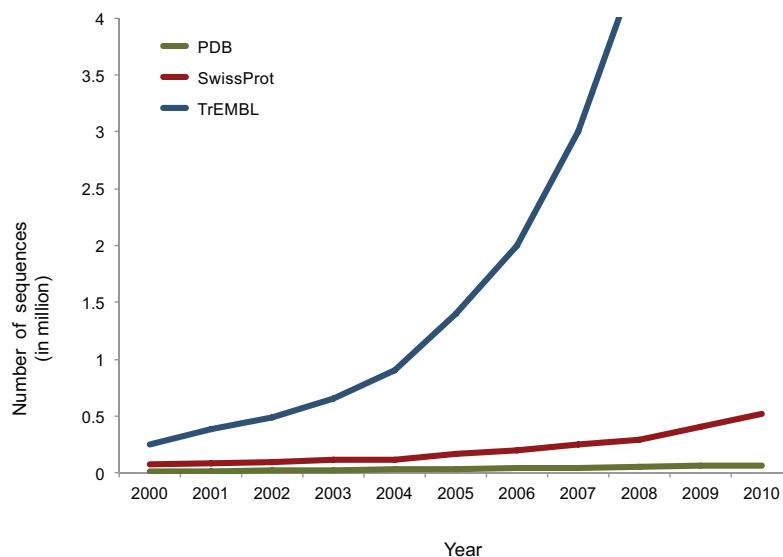


Figure 1.1. The growth in the number of entries in three major databases is shown in millions of sequences. TrEMBL represents the compendium of all known sequences, regardless of how well characterized they are. PDB represents all sequences for which a structure has been experimentally determined. Swiss-Prot is a database of confirmed and relatively well characterized proteins, although not all proteins have been assigned experimentally verified GO annotations.

hold the greatest promise in making progress towards functionally characterizing the large compendium of known proteins. This dissertation is largely focused on *in silico* efforts to infer protein function. Because our ability to draw reliable conclusions about the performance of one technique over another is grounded in having reliable benchmarking methods a great deal of attention is also paid to this topic (Chapter 4).

1. Defining function

In light of the fact that the central unifying theme to this dissertation is protein function, it is perhaps important to take some time to define what it means when we talk about “function”. This dissertation addresses protein function, as defined by the GO consortium [Ashburner et al., 2000]. Although alternative definitions exist [Millikan, 1989], in practice function is generally considered from the causal role perspective [Amundson and Lauder, 1994]. In a very broad sense protein

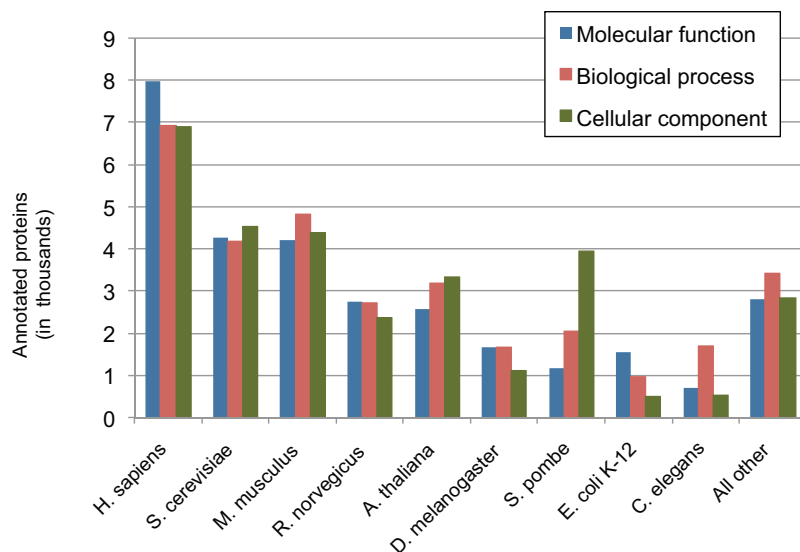


Figure 1.2. The distribution of experimentally annotated (EXP, TAS and IC evidence codes) proteins amongst well characterized organisms and all other organisms for the three branches of the Gene Ontology. Annotations were taken from the January 2012 version of Swiss-Prot.

function is a multi-scale concept that reflects “everything that happens to or through a protein” [Rost et al., 2003], and is typically considered from the biochemical, biological, and phenotypic perspectives [Bartlett et al., 2003]. How the function of a protein is defined is largely dependent on the perspective from which it is being studied. From the biochemical, or molecular, perspective a protein may be a kinase, whereas in terms of its biological function this kinase can be involved in numerous processes, such as cell cycle regulation or cell-cell signaling. Two proteins with the same molecular function may be involved in drastically different biological processes, and conversely, the set of proteins associated with a particular biological process will generally be drawn from a wide range of molecular functions. From the phenotypic viewpoint, a protein is generally associated with variation in observable physical or behavioral traits. For example, kinase variants or mutants may be responsible for disease. Adding another level of complexity to the study of function is the fact that a protein’s molecular function, while generally considered to be a static notion, is modulated by a particular cellular context, e.g. the presence of other molecules, or properties of the physical environment, e.g. temperature [Mohan et al., 2009].

Several classification systems have been proposed to standardize functional annotation and to facilitate computation. With few exceptions, these classification systems usually take on the structure of hierarchical ontologies. Enzyme Commission (EC) numbers [NCotIUoBaMB, 1992] and the MIPS functional catalogue [Ruepp et al., 2004] are two well-accepted schemes; however, the most commonly used functional classification is the Gene Ontology (GO). GO provides three hierarchical classifications as directed acyclic graphs: molecular function ontology, biological process ontology, and cellular component [Ashburner et al., 2000]. With respect to defining a particular gene’s phenotype, the Human Phenotype Ontology [Robinson and Mundlos, 2010], Unified Medical Language System [Bodenreider, 2004] and Disease Ontology [Schriml et al., 2012] are human specific and predominantly constructed to address human disease. In the case of the Unified Medical Language System the focus is on defining associations between genes and medical disorders. Several ontologies provide organism independent terminology for defining phenotype such as the Vertebrate Skeletal Anatomy Ontology [Dahdul et al., 2012].

2. Challenges in predicting function

2.1. Multi-functional proteins. In order to properly design a predictor of function it is important to consider how the classification task should be framed. It is well-known that some proteins have the ability to carry out more than one function, or “moonlight” [Jeffery, 1999, 2009]. Such abilities can be facilitated by either having multiple functional domains, a single domain which binds multiple partners, or by different behavior upon post-translational modifications or change in physiological conditions [Jeffery, 2004, 2009]. Furthermore, there are also multiple terms that could be associated with what could be considered a single function. For example, a protein could be involved in RNA transcription, but be annotated with both “nucleic acid binding” and “RNA polymerase.”

We systematically analyze multi-functionality with respect to molecular function and biological process terms in GO. As an approximation of distinct functions we only consider the number of experimentally determined leaf GO terms associated with each protein. A GO term g was included in the count if no other term associated with the protein had g as its more general term in the ontology. For example, if a protein is associated with the term “protein binding”, the term “binding” is not counted as distinct function because it is a generalization of “protein binding”; however, the

term “nucleic acid binding” would be counted because neither of the two terms is a generalization of the other.

Figure 1.3 shows the distribution of the number of leaf terms associated with each functionally annotated sequence for both ontologies. In total, 26,707 sequences were included in the molecular function analysis and 29,118 sequences were included in the biological process analysis. Only experimental evidence, traceable author’s statement, or curator’s inference were taken from Swiss-Prot v15.15 (exclusion of traceable author’s statement and curator’s inference resulted in very similar distributions; data not shown). The plots show greater diversity in a protein’s participation in a biological process than its ability to carry out distinct molecular functions. About 66% of the proteins experimentally annotated by molecular function terms have only 1 leaf term, with no protein having more than 14. On the other hand, only 44% of proteins have only 1 biological process leaf term, with 6 proteins having 50 terms or more. This is consistent with the expectation that biological processes are governed more by the context in which a protein is utilized, and less by the physicochemical abilities of the protein. Interestingly, both the molecular function and biological process ontologies show a negative exponential decrease in the probability that a protein is associated with an increasing number of functional terms.

Given the incompleteness of experimental annotations and the propensity of researchers to focus on individual genes it is expected that the number of multi-functional proteins will only increase. In the context of protein function prediction, these distributions suggest there is a level of sophistication that should be required from a computational method with respect to its outputs. Function prediction should clearly not be approached as a binary, or even a multi-class classification problem; but as a multi-label classification, or structured learning problem where it should be expected that data points will be more than one out of many labels.

We also analyzed the relationship between the number of molecular functions and biological processes a protein is associated with. It seems intuitive to postulate that proteins with the ability to carry out multiple molecular functions should be more easily utilized in multiple different contexts, giving rise to an association with more biological processes. Out of 19,240 proteins in the intersection of data sets for molecular function and biological process, we found a Pearson correlation coefficient of 0.261 between the numbers of associated leaf terms in the two ontologies. While this correlation may seem weak, we determined that this value is statistically significant by using a permutation test

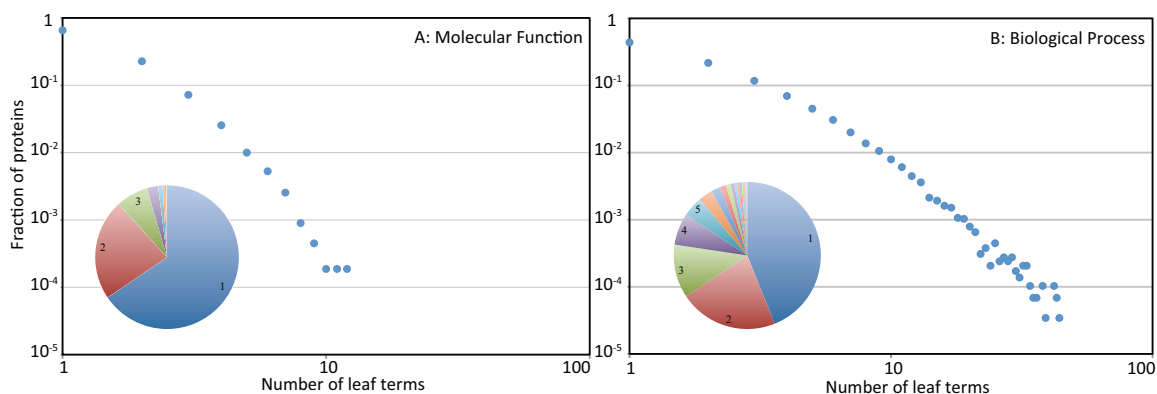


Figure 1.3. The distribution of the number of leaf terms in (A) Molecular Function and (B) Biological Process ontologies. The x-axis represents the number of leaf terms associated with a protein; the y-axis represents the fraction of proteins in the data set with the given number of leaf terms. Both axes are in $\log_{10}$ scale. The inset in each panel provides a pie chart that corresponds to each plot.

where the numbers of molecular function and biological process terms were permuted in the data set of proteins. We carried out 100,000 such permutations and did not find any cases in which the correlation coefficient was 0.261 or greater (the mean correlation coefficient for permuted data was 6.1×10^{-6} and standard deviation was 7.2×10^{-3}).

While, in general, a protein performing multiple molecular functions is associated with multiple biological processes, we conducted further analysis of proteins associated with a single term from one ontology and multiple terms from the other. For example, among the proteins that have only one molecular function leaf term, but multiple biological process terms, we found that receptor binding terms such as “chemokine receptor binding” and “cytokine receptor binding” are the most enriched ($P < 1.0 \times 10^{-7}$; Binomial test). On the other hand, there are also cases in which proteins that are associated with only one leaf term in biological process are related to multiple molecular function terms. Such biological process terms are almost all related to metabolic processes, e.g. “cellular metabolic process”, “primary metabolic process”, etc. ($P < 1.0 \times 10^{-7}$; Binomial test). When analyzing this class of proteins we also found that some terms did not occur as often as expected. When looking at the class of proteins with one molecular function term and multiple biological process terms we found that “catalytic activity” was depleted. While sequences are annotated with

this term 28% of the time in the whole data set (29% in the set of all proteins with single molecular function leaf term), it only occurs 17% of the time when we only consider sequences with exactly one molecular function term but 3 or more biological process terms associated with them. Similarly, proteins having 1 biological process leaf term, but few molecular function terms are usually involved in reproduction. We note that this data needs to be interpreted with caution, because molecular function and biological process terms are incomplete for most proteins and also because there may exist biases in ways current functions are acquired.

2.2. Transfer of function by sequence similarity. Because transferring function based on sequence similarity is one of the most intuitive, and easy to implement methods for inferring function it is important to consider its effectiveness. We evaluated the performance of simple function transfer between similar sequences. The following steps were taken: for each range of pairwise global sequence identities, a target protein received all functional terms from each experimentally annotated protein within the given sequence identity range (regardless of the species). For each target sequence with more than one match in the given identity range the precision and recall (see Section 1.5) were calculated as averages over all pairs. Then, the precision and recall for the entire data set are reported (Figure 1.4) as averages of the averaged precisions and recalls calculated for each sequence covered (a leave-one-out estimation). We also report the coverage for each range of sequence identity as the fraction of proteins with at least one other annotated sequence whose pairwise sequence identity falls within the defined range. Finally, for all covered sequences we report the percentage of perfect annotations, i.e. the percentage of all pairwise annotations in a given bin where both precision and recall were 1.

The results shown in Figure 1.4 suggest that using global sequence identity for the transfer of functional annotations is only moderately accurate (pairwise local alignments performed similarly; data not shown). Surprisingly, even for 100% identity transfer of function does not achieve either precision or recall of 1 (only 17% of identical sequence pairs had perfect transfer of molecular function and 12% of biological process terms). This relatively low rate of perfect annotations among perfect matches, and similarly in the remaining identity bins, in our data set is caused by three factors: (i) sparsity of database annotations, where proteins are incompletely annotated with respect to functionality and also specificity of annotation; (ii) database errors, caused by incorrect interpretation of experiments or by curation errors, [Brenner, 1999, Schnoes et al., 2009] and (iii)

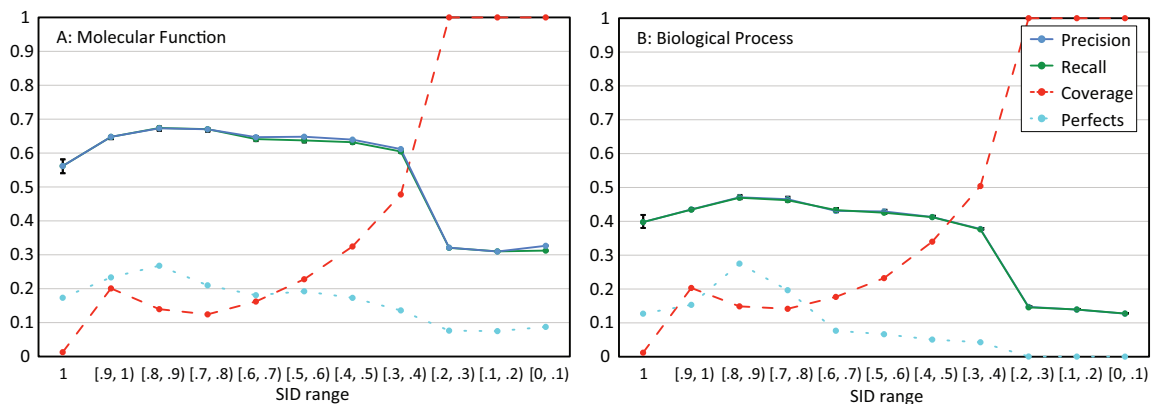


Figure 1.4. Accuracy of function transfer using global pairwise sequence identities (A: Molecular Function; B: Biological Process). For each sequence identity range (x-axis), shown are the average precision (blue solid line) and recall (green solid line) of function transfer by pairwise similarity. The teal dotted line represents the percentage of pairs with perfect annotations (e.g. both precision and recall equal to 1). The red dashed curve represents the percent of proteins that have pairwise matches (annotated with GO terms; experimental evidence code) in a given range. The error bars represent 95% confidence intervals.

organismal context, where the difference between two organisms influences a particular functional role of individual proteins, even at 100% sequence identity.

In general, transferring molecular function annotations is more accurate than transferring biological process annotations, which has previously also been observed by Rogers and Ben-Hur in a different prediction scenario [Rogers and Ben-Hur, 2009]. This is probably a result of the fact that molecular function terms are less dependent upon cellular, tissue, or organismal context, but also that the topological properties, including the average branching factor, the number of terms, and the average depth of a leaf node, between the two ontologies differ (data not shown). For both ontologies the precision of transferring predictions rapidly decreases once the 20 – 30% identity is reached. This range of sequence identity has been termed the “twilight zone” for the inference of protein structure from sequence [Rost, 1999]. In the context of function transfer such a twilight zone cannot be clearly defined (or rather should be extended to the entire range 30 – 100%), with the range below 30% being one where function transfer breaks down completely (“midnight zone”).

We also evaluated an alternative approach where functional terms from all matches in a given identity range were transferred to the target protein and found a very similar level of precision but an increased recall (Figure 1.5).

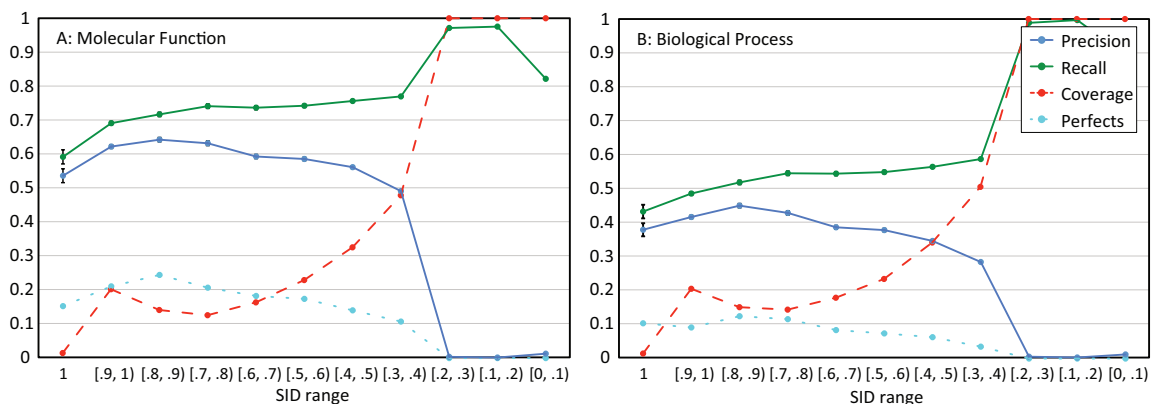


Figure 1.5. The precision and recall of function transfer using global pairwise sequence identities (A: Molecular Function; B: Biological Process). For each sequence identity range (x-axis), shown the average precision (blue solid line) and recall (green solid line) of function transfer were calculated by annotating a target sequence with the annotations of all hits within a particular SID range. The precision and recall curves in this figure differ from those in Figure 2 in that precision and recall for an individual data-point here do not represent an average for each hit, but instead the agglomeration of all terms from hits for a particular range of SID. The teal dotted line represents the percentage of pairs with perfect annotations (e.g. both precision and recall equal to 1). The red dashed curve represents the percent of proteins that have pairwise matches (annotated with GO terms; experimental evidence code) in a given range. The error bars represent 95% confidence intervals.

The quality of function transfer is highly dependent on the particular class of protein. We show this by splitting the molecular function/biological process data sets into enzymes (proteins annotated with term “catalytic activity”), and non-enzymes. As seen in Figure 1.6, the ability to transfer molecular functions to these two classes of proteins was noticeably different and most likely points to a higher quality of functional annotations for enzymes.

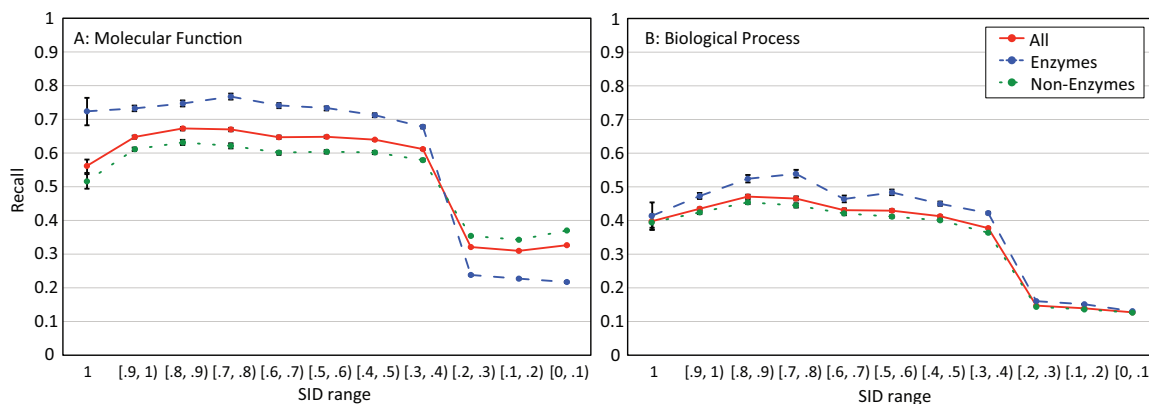


Figure 1.6. Accuracy of function transfer using global pairwise sequence identities for enzymes vs. non-enzymes (A: Molecular Function; B: Biological Process). For each sequence identity range (x-axis), shown is the average precision of function transfer by pairwise similarity (blue dashed line: enzymes, green dotted line: non-enzymes; red solid line: combined). The error bars represent 95% confidence intervals

Another interesting trend in the data is the fact that the quality of annotations transferred between sequences from the same species is higher than that obtained between different species. This can be seen by comparing the precision/recall curves in Figure 1.7 obtained by only considering pairs of sequences from the same species (“within” curve, blue line), and only considering pairs of sequences from different organisms (“between” curve, green line). A similar trend has been previously observed on protein-protein interaction data by [Mika and Rost, 2006].

The disappointing performance of transferring function based on sequence similarity highlights the importance of developing advanced methods for inferring function. Although many data bases provide electronic annotations for proteins, we are not aware the quality of these methods had not been assessed until our publication [Clark and Radivojac, 2011]. Below we consider the quality electronic annotations in Swiss-Prot.

2.3. Quality of non-experimental annotations in Swiss-Prot. Protein databases such as GO or Swiss-Prot contain a number of functional annotations supported by non-experimental evidence codes. Because of the ability of these annotations to preclude prediction efforts it is important to consider their quality. We aimed to assess the quality of such annotations by analyzing non-experimental annotations for the proteins in Swiss-Prot (v10.0-v15.0) that in a later release

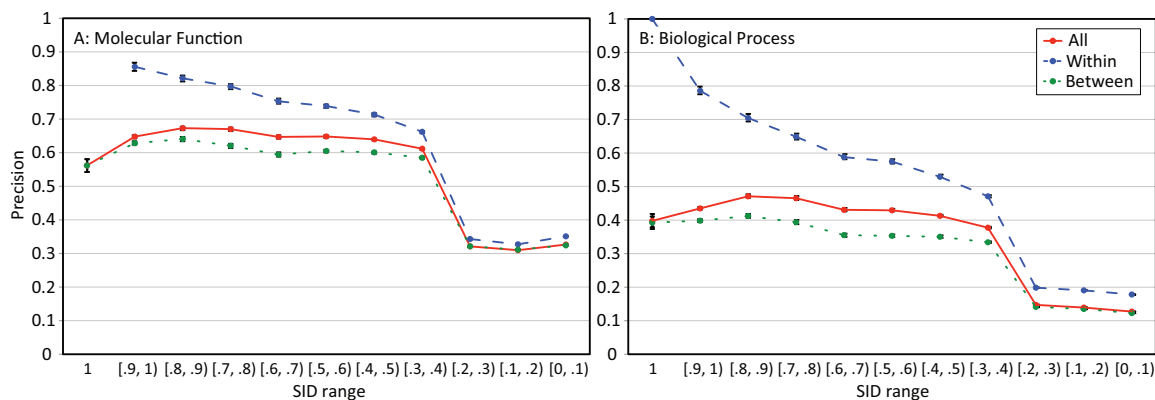


Figure 1.7. Accuracy of function transfer using global pairwise sequence identities from the same or different organism (A: Molecular Function; B: Biological Process). For each sequence identity range (x-axis), shown is the average precision of function transfer by pairwise similarity (blue dashed line: same organism, green dotted line: different organism; red solid line: combined). The error bars represent 95% confidence intervals.

(v15.15) accumulated experimental annotations. Figure 1.8 shows the quality of annotations by non-traceable author statement (NAS) or inferred from electronic annotation (IEA) evidence codes (the remaining non-experimental codes did not contain enough sequences). Using the current annotations (v15.15) as true function, the precision and recall of each protein's annotation were calculated (Figure 1.8).

As shown in Figure 1.8, the quality of electronic annotations is consistently better than that of non-traceable author statements for molecular function, while the trend is reversed for biological process. Interestingly, the precision of the electronic annotations in the Swiss-Prot database for molecular function exceeds the level achieved (Figure 1.4), while the NAS evidence suggest lower confidence levels compared to that of sequence transfer. A historical analysis of biological process annotations suggests that neither electronic annotation, nor non-traceable author statements, have been at the level of simple transfer of annotation; however the latest major release of Swiss-Prot (v15.0) provides more accurate electronic inference than transfer by sequence similarity. Similar results were found by a later paper [Škunca et al., 2012], which serves to reinforce our initial findings that there is room for improvement in deposited electronic annotations.

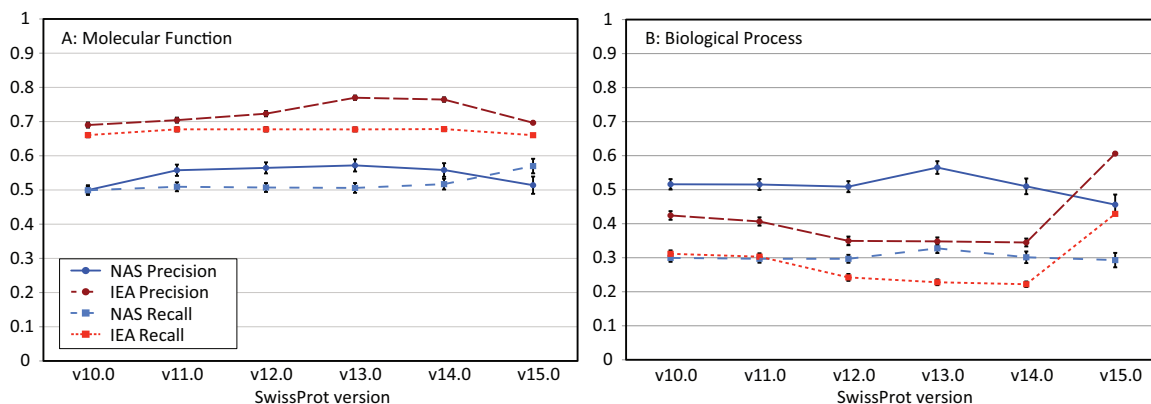


Figure 1.8. Accuracy of function transfer for the Swiss-Prot database (A: Molecular Function; B: Biological Process). X-axis represents a different version of Swiss-Prot. The dark dashed red line represents the precision of IEA annotations and the dark blue solid line represents the precision of NAS annotations. The light red dotted line represents the recall of IEA annotations, and the light blue dashed line represents the recall of NAS annotations. The error bars represent 95% confidence intervals.

3. State of the art methods

Historically, sequence-based inference was the first strategy used to predict protein function, even if most studies at the time avoided explicitly relating homology and function [Doolittle, 1986]. Global and local sequence alignments were used to query sequence databases for similarities with a target protein. With the accumulation of experimentally determined protein functions, the most similar annotated sequences have traditionally been used to infer function [Rost et al., 2003]. Several methods have developed novel techniques for utilizing sequence alignment information as input to supervised learning methods [Clark and Radivojac, 2011, Wass and Sternberg, 2008, Kourmpetis et al., 2013, Minneci et al., 2013, Sokolov and Ben-Hur, 2010]. More advanced methods exploited predicted physicochemical properties [Jensen et al., 2002, 2003, Minneci et al., 2013, Cozzetto and Jones, 2013], evolutionary relationships [Enault et al., 2005, Engelhardt et al., 2005, Gaudet et al., 2011, Marcotte et al., 1999, Pellegrini et al., 1999, Bandyopadhyay et al., 2006], or the structure of functional ontologies in order to achieve different confidence levels for different ontological terms [Barutcuoglu et al., 2006, Hawkins et al., 2006, Martin et al., 2004]. Microarrays [Huttenhower et al.,

2006], protein-protein interaction networks [Deng et al., 2003, Letovsky and Kasif, 2003, Vazquez et al., 2003, Nabieva et al., 2005], protein structures [Laskowski and Thornton, 2008, Pazos and Sternberg, 2004, Pal and Eisenberg, 2005, Hermann et al., 2007] or a combination of data types [Costello et al., 2009, Kourmpetis et al., 2010, Lee et al., 2004, Troyanskaya et al., 2003, Sokolov and Ben-Hur, 2010] have also been exploited. However, most of these methods are limited to a few organisms where such data are available. One way or another, sequence alignment-based inference is the cornerstone of functional inference [Radivojac et al., 2013, Hamp et al., 2013].

Sequence alignment-based transfer of function has been thoroughly studied in the last decade, predominantly for enzymes [Addou et al., 2009, Devos and Valencia, 2000, Rost et al., 2003, Tian and Skolnick, 2003, Todd et al., 2001, Wilson et al., 2000]. The results of these studies indicate that at least 60% sequence identity, and more likely closer to 80%, is required for the accurate transfer of the third level of EC classification. More sophisticated approaches were proposed as well: the GOtcha method was developed in order to take sequence alignment scores between a query protein and a functionally annotated database and overlay them on the functional ontology, cumulatively propagating such scores [Martin et al., 2004]. PFP refined this technique by incorporating PSI-BLAST alignments at very low significance levels and conditional probabilities that a protein is associated with pairs of functional terms [Hawkins et al., 2006]. Other methods such as ProtFun [Jensen et al., 2003], ConFunc [Wass and Sternberg, 2008], GOSling [Jones et al., 2008], and GOstruct [Sokolov and Ben-Hur, 2010] were developed for high-throughput prediction tasks. Finally, phylogenetic methods attempt to exploit particular evolutionary relationships within a gene family [Brown and Sjolander, 2006, Eisen, 1998]. Methods such as SIFTER [Engelhardt et al., 2005] or ortholog identification methods [Remm et al., 2001] belong in this category. Several recent reviews provide good perspectives on protein function prediction at all scales [Dalkilic et al., 2008, Friedberg, 2006, Kann, 2007, Laskowski and Thornton, 2008, Lee et al., 2007, Punta and Ofran, 2008, Rentzsch and Orengo, 2009, Rost et al., 2003].

3.1. Community-wide assessment of function prediction. Many of the previously listed state of the art methods participated in the community-wide assessment of function prediction (CAFA) in which I took a prominent role [Radivojac et al., 2013]. The CAFA experiment was conducted by first creating a data set of 48,298 proteins that lacked experimentally verified GO annotations in Swiss-Prot. Participating groups were then asked to submit annotations for this

set of proteins before a predetermined deadline. During the annotation accumulation phase experimental annotations were allowed to accumulate for approximately a year. This resulted in an evaluation data set of 866 target sequences, 531 with Molecular Function annotations, and 587 with Biological Process annotations. Reinforcing the need for advanced methods for function prediction, we consistently found that state of the art methods outperformed inferring function using sequence similarity (in this case BLAST). Furthermore, BLAST barely outperformed simply using the background distribution of functions as a predictor (Naïve). The BLAST and Naïve models are described in Section 6.2 and Section 6.1 respectively.

4. Analysis of function in evolutionary context

The previous chapters focused largely on the inference of function. While these endeavors are important in their own right, it is desirable that the data produced by these efforts will at some point; perhaps this point has already arrived, serve as more than an encyclopedic cataloging of the known gene/protein universe. In Chapter 5 we use several different types of data, including GO annotations, to compare the functional similarity of orthologous and paralogous sequences [Nehrt et al., 2011]. While the results of our findings have implications for the inference of function prediction, especially when phylogenetic data is used [Eisen, 1998, Škunca et al., 2013]; it also represents a transition from utilizing GO annotations simply to inform a user about a gene, to using both curated and predicted functional annotations to generate and test theories about biology and evolution.

5. Conclusion

With the growing gap between known sequences and experimentally annotated proteins, it is clear that functional annotation of all proteins can only be accomplished by combining experimental and computational methods. Targeted wet lab experiments have been predominantly focused on model organisms with an expectation that results will provide a detailed understanding of these organisms and that the gap between species can be accurately filled by computational methods. Indeed, model organisms provide a large fraction of the genes with experimentally verified functional annotations. In Swiss-Prot v15.15, we found that approximately 90% of annotated proteins in molecular function and biological process belong to 9 model organisms only (*H. sapiens*, *S. cerevisiae*, *M. musculus*, *R. norvegicus*, *A. thaliana*, *D. melanogaster*, *S. pombe*, *E. coli K-12*, and *C. elegans*). However, nearly 60% of the proteins from these model organisms still do not have any experimentally

determined molecular function or biological process terms. Thus, the development and assessment of computational methods is critical for not only filling the gap between model and non-model organisms, but also for completing the annotation of model organisms and driving experimental analyses.

A frequent interpretation of the sequence-structure-function paradigm is that a protein must adopt a single structure (minimum energy state, kinetically reachable) in order to be functional, with such conformation usually called the native state. However, such an understanding has been challenged from both structural and functional perspectives. Many proteins have been characterized as intrinsically disordered. In such proteins, no single structure is seen as being dominant (i.e. high probability conformation with deep energy minimum) and a presence of conformational ensembles (i.e. macro states [Dill, 1999]) is probably even required for function [Dunker et al., 2002, Dyson and Wright, 2005, Radivojac et al., 2007]. At the same time, it is now recognized that multifunctional proteins are also common [Jeffery, 2009]. We find that at least 34% of functionally characterized proteins (by experimental studies) are already assigned more than one distinct molecular function term and that at least 56% of proteins participate in more than one distinct biological process. We believe that the ability of a protein to be multifunctional in terms of its biochemical function needs to be achieved by developing new structural conformations and physicochemical interfaces (including the addition of new domains), whereas its involvement in multiple biological processes does not. This is because an organism need only utilize the given protein in a different context, excluding the necessity to change the actual mechanism through which the protein functions.

We also analyzed the quality of molecular function and biological process term transfer by simple sequence similarity and found that inference by similarity shows flat accuracy in the entire range from 30 – 100% of pairwise sequence identity (unless within the same organism). This leads to the conclusion that more sophisticated computational methods are necessary. To date, much attention has specifically been paid to the quality of function transfer for enzymes [Addou et al., 2009, Rost et al., 2003]. Here, we extended such analyses to non-enzymatic proteins and observed that transfer of function to non-enzymatic proteins is less accurate than that achieved for proteins annotated with any function from the catalytic activity portion of the ontology. While the underlying reasons may simply lie in the sparseness of these parts of the ontology, annotating a protein with functions without knowledge of the associated mechanics (information that is often known for enzymes) can

result in less accurate assignment of proteins with such terms. This may also be true for other classes of terms in the ontology. For example, terms which group together proteins that carry out similar tasks in the cell, but do so through different molecular mechanisms, will be less likely to be defined in terms of sequence similarity among member proteins. Conversely, terms which define a function carried out by a specific mechanism (e.g. zinc finger binding) will be more likely to be inferred by sequence similarity.

CHAPTER 2

A multi-output neural network for function prediction

As shown in Chapter 1, the prediction of protein function is a difficult task. The multi-faceted aspect of function results in proteins, after being thoroughly investigated, almost always being annotated with more than one function. Terms in a hierarchically structured ontology are also not independent, as discussed in detail in Chapter 4. Because of these challenges we chose to utilize a multi-output neural network to predict protein function. One advantage of multi-output neural networks is their ability to learn dependencies between outputs. This is particularly important given the structure of the ontology. The use of a multi-label classifier also negates the need to *post hoc* processing in order to combine multiple one-versus-all, or one-versus-one binary classifiers into a multi-label classifier.

In this chapter we develop a method for probabilistic inference of GO terms using supervised learning. Our algorithm, Functional ANNotator (FANN), employs multi-output artificial neural networks. We show that in the GO annotation task, FANN-GO outperforms standard sequence alignment methods and GOtcha [Martin et al., 2004]. We predict function at the molecular function and biological process levels from protein sequence alone.

1. Methods

1.1. Problem formulation. We consider a set of protein sequences $S = \{s_1, s_2 \dots\}$, a set of protein molecular function terms $F = \{f_1, f_2, \dots\}$ and a set of biological process terms $P = \{p_1, p_2, \dots\}$, where each sequence in S is annotated by at least one element from F or P . Additionally, each function $f \in F$ and process $p \in P$ is associated with at least one sequence from S . Our objective is to construct a classification model which, given an amino acid sequence $s \in S$, assigns posterior probability that the sequence has the ability to carry out each particular function from F , or is involved in each particular biological process in P . Similarly, given a functional term f , or a biological process p , our objective is to find the most likely sequences associated with that function or process.

Table 2.1. The number of proteins per species in each dataset.

Organism	MFO	BPO
<i>H. sapiens</i>	7,253	6,515
<i>S. cerevisiae</i>	4,062	4,071
<i>M. musculus</i>	3,945	4,672
<i>R. norvegicus</i>	2,624	2,696
<i>A. thaliana</i>	2,042	2,813
<i>D. melanogaster</i>	1,627	1,591
<i>E. coli K-12</i>	1,498	1,009
<i>S. pombe</i>	1,079	1,935
<i>C. elegans</i>	569	1,599
All other	2,008	2,217
Total	26,707	29,118

1.2. Data sets. We used the Swiss-Prot database from May 2010 (v.15.15) [Bairoch et al., 2005]. A data set D_{MFO} of 26,707 protein sequences was generated by selecting sequences with molecular functions that were supported by at least one of the following experimental evidence codes: EXP, IDA, IPI, IMP, IGI, IEP, TAS, IC. The 26,707 sequences in D_{MFO} consisted of a total of 4,276 molecular function terms associated with them. Similarly, a separate set D_{BPO} consisting of 29,118 sequences with a total of 11,300 associated biological process terms was generated using the same criterion. We note that $|D_{MFO} \cap D_{BPO}| = 19,240$ and $|D_{MFO} \cup D_{BPO}| = 36,585$.

To arrive at the final data sets, 1,429 proteins with known function in Swiss-Prot v15.15 were removed. Such sequences were either shorter than 50 amino acids or were associated with the same gene name as some other protein in the same organisms. Data sets are summarized in Table 2.1.

1.3. Data representation. Sequence alignments were used to represent each protein sequence as a fixed-length vector in a feature space. Each dimension in the feature space was selected to correspond to one term from a set of available functional terms F (or P for the biological process ontology). While we tested several ways of encoding alignment data into features we found that

using i-scores, as proposed by the GOtcha algorithm [Martin et al., 2004], as features worked the best. For the completeness of this work, we briefly summarize this representation.

First, let $e(s, s_i)$ be the E-value obtained by aligning target sequence s to the i -th sequence in database of proteins with experimentally determined functions, i.e. $s_i \in S$. The r-score for functional term f is then generated as

$$r_f(s) = - \sum_{s_i \in S_f} \log(e(s, s_i)) + c$$

where S_f is a subset of S containing all proteins with functional term f and c is a constant value added to the sum to ensure non-negative r-scores (here we used $c = 2$ and E-value threshold of 10). The i-scores for each function were then calculated by normalizing r_f by the r-score of the root node in a given ontology (term $f_{root} \in F$) as

$$i_f = \frac{r_f}{r_{root}}$$

Finally, a feature vector is obtained by concatenating i-scores for each of the $|F|$ functional terms. We note that the i-score for the root term always equals 1; thus, it was excluded from the vector representation. For the molecular function ontology, the i-score vector representation consists of $|F|$ dimensions, while for the biological process ontology, the i-score representation consists of $|P|$ dimensions. These features were used for the basic version of Functional ANNotator (FANN-GO).

In order to take advantage of the fact that the quality of transferring functions from sequences within the same species was higher than that achieved when transferring functions only from sequences in different species, we made two additional sets of i-score features, one based on the i-scores using the proteins from the same species and another using the i-scores from the proteins from different species only. These features were used for the version of the predictor referred to FANN-GO^{species}.

1.4. Classification models. In order to address the multi-label classification problem of protein function we utilized a multi-output feed-forward neural network framework. Multi-output networks have the ability to simultaneously learn multiple dependent target variables, a property that is well suited to the problem of predicting mutually non-exclusive terms of protein function ontologies.

Before training a multi-output neural network we performed several data preprocessing steps. All features were first normalized using the z-score method. Feature selection filtering was then

performed by using the t-test. Finally, principal component analysis was performed in order to combine highly correlated features (retained variance = 99%).

Owing to the high-memory requirement of a multi-output neural network, it was not practically possible to train a model with more than 1000 outputs on a data set of size $|D_{MFO}|$ or $|D_{BPO}|$. To overcome this limitation, we created ensembles of 100 networks such that in each network only 100 randomly selected outputs were considered. Prediction values for a test sequence for a particular function were finally calculated as an average over the output scores generated from the subset of networks that included the given term in their output layers. All neural networks had 100 hidden neurons, employing the resilient propagation algorithm [Riedmiller and Braun, 1993] in training (with at most 1,000 epochs). The networks were implemented using MATLAB.

1.5. Model selection and evaluation. The accuracy of the model was estimated using 10-fold cross-validation. Parameter selection for each neural network was performed on a separate validation set, such that only the best performing parameter set was used on the test partition. Furthermore, separate sets of features were generated for each fold using only alignments with sequences in the training portion of the data. An individual BLAST database was built for each fold's set of training sequences in order to ensure that alignment E-values were not influenced by sequences in the test portion of the data. All methods were evaluated by plotting precision-recall curves. FANN-GO was evaluated against three different strategies. The Global-SID and Local-SID strategies represent transfer by sequence similarity in which each prediction was generated by transferring functional terms directly from sequences with sequence identity to the query sequence greater than the threshold. In addition, the performance of FANN was compared to the GOtcha classifier [Martin et al., 2004] as well as the Naïve classifier. The Naïve classifier predicts the terms according to their descending prior probabilities in the training data, i.e. the term occurring in 75% of training sequences will be predicted with score 0.75 for all target proteins.

Precision-recall curves were generated as follows. For each query sequence, a set of predictions over all $|F|$ functions was generated. A decision threshold t value above which all predictions were taken was incrementally reduced from 1 to 0, in steps of 0.01. Terms with predictions scores above a particular threshold t_i were selected, and each term was propagated towards the root of the ontology. This resulted in a set of predicted terms P . The precision (pr) and recall (rc) between the predicted terms P and true terms T associated with sequence s were then calculated as

$$pr = \frac{|T \cap P|}{|P|}$$

and

$$rc = \frac{|T \cap P|}{|T|}$$

The final precision and recall were averaged over all test sequences to create a point in the precision-recall space.

2. Results

In this work we present a simple classification method, FANN-GO, for predicting GO terms from MFO and BPO. The model is based on aligning a target sequence to a database of experimentally annotated proteins and calculating the i-score [Martin et al., 2004] that the protein is associated with each functional term. These scores were then used as inputs to an ensemble of multi-output neural networks that were trained to predict the probability that the protein is associated with each function. A variant of FANN-GO, referred to as FANN-GOspecies, is also presented. This model is based on two groups of inputs, one containing inputs where a target protein is aligned only to proteins from the same species and another where the protein is aligned to proteins from different species.

We compared FANN-GO models with GOTcha as well as transfer by global and local sequence identity. We also implemented a naive method that uses prior probabilities of functional terms as prediction scores for all target proteins. That is, the score for each function, for all proteins, is simply the relative frequency of that term occurring in the (training) data set. Performance of all classification models was assessed using 10-fold cross-validation on all functional terms associated with 50 proteins or more (mainly for the purpose of stable accuracy estimation). In doing so, we reduced the number of terms associated with the sequences in the data set for molecular function to 344 and biological process to 1,788. For each ontology all methods were evaluated using the same data sets.

Figure 2.1 shows the performance of our two different supervised methods, FANN-GO (blue line), and FANN-GOspecies (green line) compared to an in-house implementation of the GOTcha method (red line), and two methods utilizing transfer of annotations based on pairwise sequence identity: Global-SID (purple line) based on global sequence alignments, and Local-SID (teal line)

based on local BLAST alignments. These methods were benchmarked against a Naïve classifier (orange line).

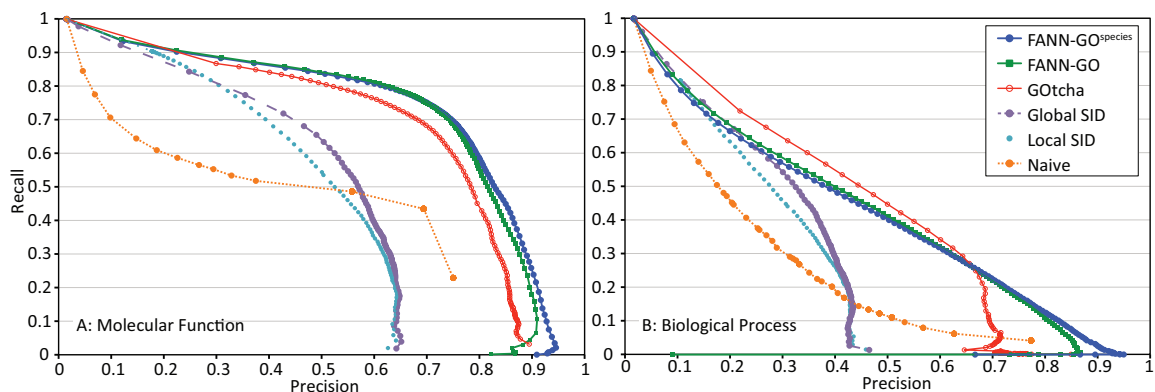


Figure 2.1. Precision-recall curves for several GO-term prediction algorithms (A: Molecular Function; B: Biological Process). The curves were generated by shifting the decision threshold t from 0 to 1 and considering all terms with scores t as predicted functions. The decision threshold for the sequence alignment curves (Global-SID, Local-SID) was the pairwise sequence identity (SID). The Naïve curve (orange dotted) was generated by assigning all proteins the most common functional term, then the second most common, etc. FANN-GO^{species} is represented by the solid blue line. FANN-GO is represented by the solid green line with square markers. GOTcha is represented by the solid red line with red circular markers. Global-SID is represented by the purple dashed line. Finally, Local-SID is represented by small teal markers only.

As shown in Figure 2.1A, FANN-GO outperformed GOTcha in both precision and recall when annotating sequences with MFO terms. Both methods outperformed simple functional transfer based on sequence identity, and nave predictions. Interestingly, annotating sequences with the most probable functional term, “binding”, causes the Naïve model to outperform predictors based on sequence similarity. Only after sequences are annotated with the third most probable function, “catalytic activity,” does the Naïve curve cross the Global-SID curve at the precision and recall obtained at approximately 21% sequence identity. The unusual shape of the Naïve curve is indicative of the highly non-uniform nature of the relative frequencies of terms in MFO. With regard to the top

three most probable terms in the ontology, 74% of sequences are annotated with the term “binding”, 63% are annotated with the term “protein binding”, and 28% are annotated with the term “catalytic activity”. Figure 2.1B shows the precision-recall curves for BPO. Here, FANN-GO outperforms GOtcha in the high-precision/low-recall part of the curve, while the trend reverses for FANN-GO thresholds below 0.49 and GOtcha i-scores below 0.17 (we note that FANN-GO thresholds of < 0.5 do not represent confident predictions). The Naïve predictor again outperforms the sequence identity-based transfer of function for the 8 most probable terms in BPO. It even outperforms GOtcha when annotating sequences with the single most probable term, “cellular process”. We found that the performance advantage of FANN-GO and FANN-GOspecies over GOtcha was similar regardless of whether a protein was annotated with one or multiple leaf terms (data not shown).

To assess whether the prediction was more accurate for proteins more similar to the sequences in the experimentally annotated database, we also performed accuracy estimation in which test proteins were split into two sets, those with at least one other sequence in the data set which shared at least 50% sequence identity with any of the training sequences, and those without one. When comparing FANN-GO with GOtcha we found that the difference in performance between these two methods was even greater for the $> 50\%$ sequence identity set (Figure 2.2). Therefore, FANN-GO may be well-suited to annotating proteins that do not share significant sequence identity with sequences that are experimentally annotated.

3. Discussion

In this work, we investigated the accuracy of function prediction by simple methods such as transfer by sequence similarity contrasted against our new computational method for predicting protein function, Functional ANNotator (FANN). Our new algorithm, Functional ANNotator (FANN) was developed as a supervised method for the prediction of protein function based on sequence alignments only. While diverse types of data may be available in some model organisms, function transfer between model and non-model organisms is ultimately and critically dependent upon sequence-based predictions. Our attempt is a simple algorithm that encodes every protein as a vector of similarities in a space of protein functions, with neural networks used to learn posterior probabilities of sequence-function relationships. Neural networks are well-suited for this classification task due to their natural ability to learn multiple outputs that are dependent and mutually non-exclusive. The

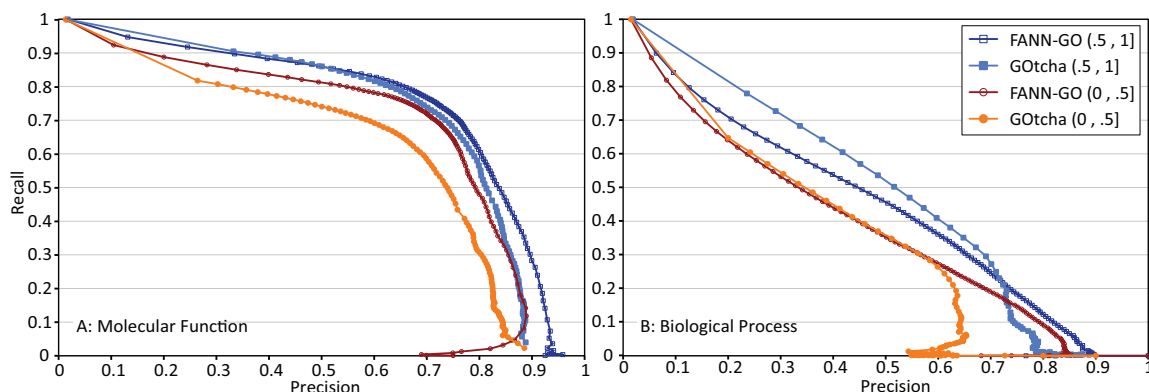


Figure 2.2. The performance of FANN-GO and GOTcha on sets of sequences with varying degrees of maximum similarity to other sequences in the data set (A: Molecular Function; B: Biological Process). Sequences were split into two sets, those with at least one other sequence in the data set with which they shared at least 50% sequence identity, and those without one. The dark blue line with square hollow markers represents the performance of FANN-GO on the $\geq 50\%$ set of sequences. The light blue line with square filled markers represents the performance of GOTcha on the $\geq 50\%$ set of sequences. The red line with hollow circular markers represents the performance of FANN-GO on the $< 50\%$ set of sequences. Finally, the orange line with filled circular markers represents the performance of GOTcha on the $< 50\%$ set of sequences.

methodology best-suited for such prediction tasks is an open problem, with examples including one-versus-all training of binary classifiers, or their combination with post-processing methods for multi-label classification. Compared to previous attempts to incorporate the structure of the ontology in a supervised manner, our method does not require post-processing [Barutcuoglu et al., 2006] or constraints on the kernel functions between input and output spaces [Sokolov and Ben-Hur, 2010]. On the other hand, neural networks, as implemented here, do not necessarily produce a consistent set of predictions, i.e. a parent term may have a lower score than any of its children’s terms. While from a machine learning perspective this is undesired, we believe this feature can be useful for detecting problems with the ontology itself, as it should not be assumed that either molecular function

or biological process are error-free and complete. It should be pointed out that this inconsistency is also a drawback of one-versus-all methods.

Currently, it is believed that the most accurate computational models for sequence-based function prediction are GOtcha-based unsupervised algorithms [Friedberg et al., 2006, Izarzugaza et al., 2007, Rentzsch and Orengo, 2009]. While these models are very good in their performance, we show that supervised methods outperform them on the same data. One drawback, however, is that supervised methods are limited to only functional terms that are associated with a sufficient number of sequences. Therefore, a function prediction task may be well suited for the methods combining supervised (for terms where enough proteins are available) and unsupervised (when only a small number of sequences are available) approaches. Finally, it should be kept in mind that a separate category of *de novo* prediction methods is needed for proteins with novel or organism-specific functions. Such algorithms, however, are beyond the scope of this work.

Property Kernels based on Vector Quantization

The wealth and diversity of experimental data in the life sciences has been a strong catalyst in the increasingly ubiquitous nature of classification methods in computational biology. Over the past couple of decades the scope and sophistication of these methods has significantly increased; growing from relatively simple procedures based on naïve Bayes or logistic regression into fully fledged Bayesian and structured-output learning methods. One of the important factors leading to the increased adoption of modern classification models in computational biology is their ability to not only integrate various types of data, be it sequence, structure, graphs or text; but also their robustness when dealing with data of varying degrees of quality. Methods attuned to producing quality predictions on a particular type of biological data; or that can facilitate the integration of data from disparate sources, have been quickly adopted by the community.

Among the various classification strategies, kernel-based methods have recently become prominent [Shawe-Taylor and Cristianini, 2004]. In general, kernels can be described as similarity functions that operate on pairs of objects and maintain certain mathematical properties. One important characteristic of kernel-based methods that distinguishes them from traditional machine learning approaches is individual objects need not be explicitly represented in a vector, or feature, space $\mathcal{F}$ (even if they often are); an inner product space is obtained through the similarities generated by applying the kernel function to pairs of objects,

$$k(\mathbf{x}_1, \mathbf{x}_2) = \langle \phi(\mathbf{x}_1), \phi(\mathbf{x}_2) \rangle,$$

where $\phi(\mathbf{x})$ represents the embedding of object $\mathbf{x}$ in $\mathcal{F}$. Through the kernel function, $k(\mathbf{x}_1, \mathbf{x}_2)$, objects are implicitly embedded in $\mathcal{F}$, where linear relations (i.e. an optimal separating hyperplane between two classes) are searched for. Kernel approaches are therefore capable of operating on a broad class of input spaces by effectively mapping objects into unknown and potentially infinite dimensional feature spaces. When coupled with support vector machines, they also guarantee that

a globally optimal solution to the optimization problem can be found. Although most kernel-based approaches are in practice formed by vectorizing input objects; not fully exploiting their theoretical potential, they still enable a practitioner to incorporate domain knowledge into modeling the relationship between objects, rather than simply encoding properties of the objects into a potentially high dimension vector space and providing them to a standard classifier.

Kernel approaches have been widely used in computational biology [Schölkopf et al., 2004]; being applied in a range of contexts such as the prediction of remote homology [Leslie et al., 2002], protein structure [Qiu et al., 2007] and function [Vert, 2002, Vacic et al., 2010, Xin and Radivojac, 2011], protein-protein interactions [Ben-Hur and Noble, 2005], disease-gene prioritization [De Bie et al., 2007], the activity of chemical compounds [Ralaivola et al., 2005], etc. An excellent summary and history of early approaches has been provided by Noble [2004]. Most of these approaches have been used to predict properties of proteins, either in terms of entire sequences or individual residues. For example, the most influential kernel methods were proposed to provide inferences regarding remote homology and function from amino acid sequences [Leslie et al., 2002, Jaakkola et al., 1999, 2000, Kuang et al., 2004]. Graph kernels, first introduced by Kondor and Lafferty [2002], have gained significant attention in computational biology due to the potential for modeling a variety of data types through graphs [Borgwardt et al., 2005]. A number of approaches also considered integration of kernels built on different types of data, including scientific text [Sokolov and Ben-Hur, 2010].

For the purposes of this work, we focus our investigation on the viability of kernel approaches in terms of type of data used when forming a kernel representation of proteins. We develop novel methodology for the non-alignment kernel-based supervised classification of protein sequences and structures into distinct categories. In contrast to most previously implemented kernel approaches, we represent a proteins sequence or structure, if available, as a set of time series properties such as hydrophobicity profile or predicted propensity of secondary structure types, calculated by considering an individual amino acid and its flanking regions.

An important problem in calculating the similarity between differing length real-valued vectors is providing a fixed-length representation to be used as input for a given prediction method. To do this, we use the ideas of vector quantization. Vector quantization (VQ) was initially developed as a lossy method for compressing signals [Linde et al., 1980]. We evaluated methods on two distinct

and relevant problems: (i) the classification of protein structures into structural classes and (ii) the prediction of protein function from amino acid sequence.

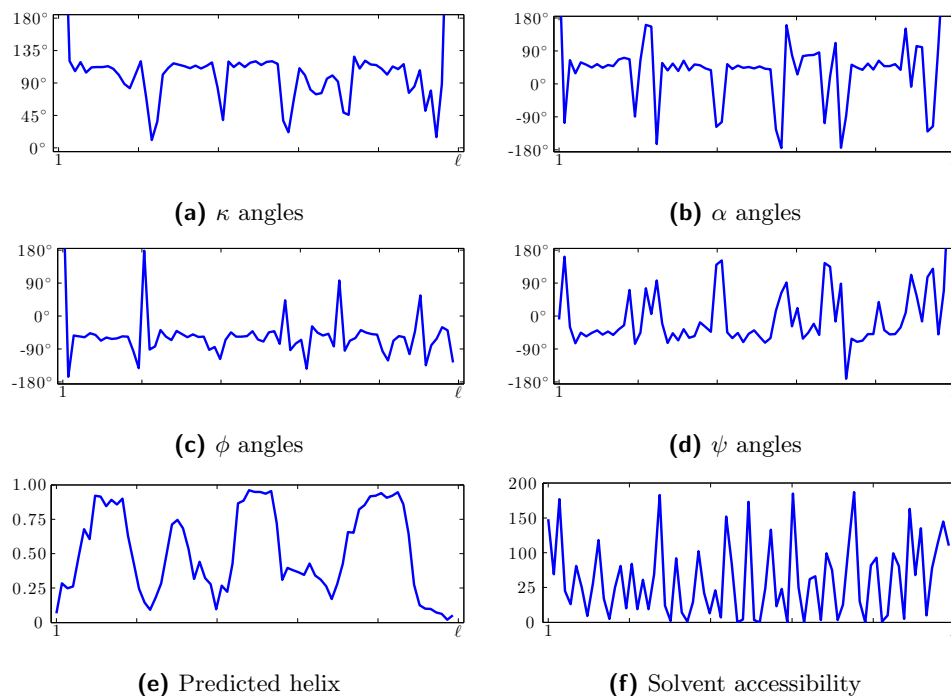


Figure 3.1. Time series representation of various protein properties for the 73 amino acid long DNA helicase RuvA subunit **d1ixra1**.

1. Problem formulation

In this chapter we represent a protein as a set of time series properties such as hydrophobicity profiles, predictions of intrinsically disordered regions, or dihedral angle values if its structure is available (Figure 3.1). One way to augment the differing length time series signals for each sequence into fixed length features to be used as input for a given prediction method is to apply a vector quantization (VQ) kernel. The basic idea behind this method (outlined in Figure 3.2) is to first break each signal up into overlapping fixed n -length segments (Figure 3.2a). These small segments are then clustered in n -dimensional space, generating a set of m centroids representing each cluster (Figure 3.2b). We chose to use k-means clustering as opposed to simply creating a lattice in n -dimensional space, as done by Tuytelaars and Schmid [2007], because sampled property vectors do not fill the space evenly, but instead cluster around evolutionarily conserved or sterically preferred

“regions”. After clustering, each segment can then be represented by which cluster it belongs to, and counts (or relative frequencies) that each subsegment for a property belongs to a cluster can be used to represent a sequence (Figure 3.2c).

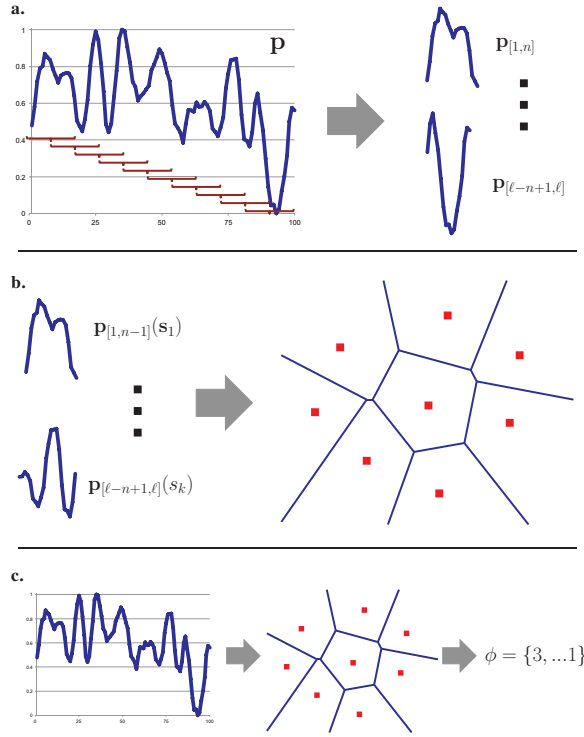


Figure 3.2. A schematic representation of using VQ to encode a sequence represented as a property vector. In 3.2(a) an individual property vector is broken up into n length subsamples. In this example the amount of overlap is $\frac{n}{2}$, although differing values can be used, including randomly choosing samples in instances with a large number of sequences. 3.2(b), subsampled vectors from a database of sequences are used to create a clustering in n -dimensional space, with centroids represented as red squares. Finally, in 3.2(c) an original property vector is encoded using the derived set of centroids by counting the number of sub-vectors which are the closest to each centroid. In this case, sub sampled vectors are usually shifted by a single amino acid.

2. Methods

Let $\mathcal{S} = \{\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \dots\}$ be a universe of protein sequences, where each $\mathbf{s} \in \mathcal{S}$ is a string of symbols from an alphabet of amino acids $\mathcal{A} = \{A, C, D, \dots, Y\}$. Let also $\mathcal{S}_L \subset \mathcal{S}$ be a set of labeled sequences, e.g. those with known structural class or function, that is provided as training data. The objective is to use supervised framework to probabilistically annotate the remaining sequences, i.e. those from the unlabeled set $\mathcal{S}_U = \mathcal{S} - \mathcal{S}_L$.

To map protein sequences into a real-valued vector representation, let $\mathbf{s} = s_1 s_2 \dots s_\ell$ be a length- ℓ protein sequence in $\mathcal{S}$ and $\mathbf{p} = (p_1, p_2, \dots, p_\ell)$ some property vector defined by any particular mapping from $\mathcal{A}^\ell$ to $\mathbb{R}^\ell$. For example, $\mathbf{p}$ may be provided as a vector of hydrophobicity indices corresponding to amino acids in $\mathbf{s}$. Alternatively, it can be represented as predicted helical propensities as outputted by some predictor of secondary structure. For those sequences with available structures, $\mathbf{p}$ may correspond to a sequence of dihedral angles calculated from the protein structure model of $\mathbf{s}$.

Consider now a single property vector $\mathbf{p}$, such as hydrophobicity profile, corresponding to a particular sequence $\mathbf{s} \in \mathcal{S}$. We split $\mathbf{p}$ into n -dimensional sub-vectors $\mathbf{p}_{[1,n]}, \mathbf{p}_{[2,n+1]}, \dots, \mathbf{p}_{[\ell-n+1,\ell]}$, where $\mathbf{p}_{[i,i+j]} = (p_i, p_{i+1}, \dots, p_{i+j})$, and $n \ll \ell$ is a small integer. For example, $\mathbf{p}_{[1,n]}$ corresponds to the first n elements of $\mathbf{p}$ as shown in Figure 3.2(a). For a property vector (amino acid sequence) of length ℓ , there are $\ell - n + 1$ length- n sub-vectors.

Given a set of length- n property sub-vectors $\mathcal{P}$ derived from the sequence universe $\mathcal{S}$, we generate a partition of $\mathbb{R}^n$ into m regions $\mathcal{R} = \{R_1, R_2, \dots, R_m\}$. These regions are represented by a set of n -dimensional vectors, or centroids, $\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_m\}$. Each region R_i represents a Voronoi region such that

$$R_i = \{\mathbf{x} : d(\mathbf{x}, \mathbf{c}_i) \leq d(\mathbf{x}, \mathbf{c}_j), j \neq i\},$$

where $d(\mathbf{x}, \mathbf{c})$ is the Euclidean distance between vector $\mathbf{x}$ and centroid $\mathbf{c}$. We determine $\mathcal{C}$ using k-means clustering, where the initial set of clusters is generated by the splitting method [Gersho and Gray, 1992].

2.1. Count-based property kernels. Variable length property vectors can be transformed into fixed m -length vectors using the partition of $\mathbb{R}^n$ defined by $\mathcal{R}$. Specifically, a property vector $\mathbf{p}$ is mapped into a vector of length m as

$$\mathbf{x} = (\varphi_1(\mathbf{p}), \varphi_2(\mathbf{p}), \dots, \varphi_m(\mathbf{p})),$$

where $\varphi_i(\mathbf{p})$ is the number of n -dimensional vectors $\mathbf{p}_{[i]}$ in $\mathbf{p}$ that belong to partition R_i . Given two property vectors $\mathbf{p}$ and $\mathbf{q}$ and their respective count vectors $\mathbf{x}$ and $\mathbf{y}$, a *count-based property kernel* is defined as

$$k(\mathbf{p}, \mathbf{q}) = \mathbf{x}^T \mathbf{y},$$

where T is the transpose operator. Note that in this notation each count vector is assumed to be a column vector, i.e. $(a, b, c) = [a \ b \ c]^T$, as in Strang [2003]. Since the function $k(\mathbf{p}, \mathbf{q})$ is defined as an inner product between two count vectors, it is a kernel function [Haussler, 1999].

The count-based property kernels can be generalized by utilizing similarities between centroids to provide partial counts to each of the regions in $\mathcal{R}$. First, a similarity function between any two centroids is defined as

$$s(\mathbf{c}_i, \mathbf{c}_j) = e^{-\alpha \cdot \frac{d(\mathbf{c}_i, \mathbf{c}_j)}{d_{\max}}},$$

where $d(\mathbf{c}_i, \mathbf{c}_j)$ is the Euclidean distance between $\mathbf{c}_i$ and $\mathbf{c}_j$ and $d_{\max}$ is the maximum distance between any two centroids in $\mathcal{C}$. The parameter α is a positive constant that defines an extent to which large distances are penalized by the non-linear mapping above, and is optimized in the model evaluation step. A generalized property kernel can now be defined as

$$k_g(\mathbf{p}, \mathbf{q}) = (\mathbf{S}\mathbf{x})^T (\mathbf{S}\mathbf{y}) = \mathbf{x}^T \mathbf{S}^T \mathbf{S} \mathbf{y} = \mathbf{x}^T \mathbf{Q} \mathbf{y},$$

where elements of the similarity matrix $\mathbf{S}$ are defined as $s_{ij} = s(\mathbf{c}_i, \mathbf{c}_j)$. Clearly, for $\mathbf{S} = \mathbf{I}$ the generalized kernel is equivalent to the basic count-based property kernel. If $\mathbf{S}$ is a non-singular matrix, it is easy to show that $\mathbf{Q}$ is a positive definite matrix [Strang, 2003]. Therefore, $k_g(\mathbf{p}, \mathbf{q})$ is a kernel function. Note that the generalized count-based kernel does not lead to a sparse representation, even for large m .

2.2. Additive property kernels. Let us again consider vector $\mathbf{p}$ and partition $\mathcal{R}$. We map $\mathbf{p}$ into an m -dimensional as

$$\mathbf{x} = (\psi_1(\mathbf{p}), \psi_2(\mathbf{p}), \dots, \psi_m(\mathbf{p})),$$

where

$$\psi_i(\mathbf{p}) = \sum_{j=1}^{\ell-n+1} s(\mathbf{p}_{[j, j+n-1]}, \mathbf{c}_i),$$

is the cumulative similarity between vectors $\mathbf{p}_{[j]}$ in $\mathbf{p}$ and centroid $\mathbf{c}_i$, and $s(\cdot)$ is the vector similarity function defined above. Given two property vectors $\mathbf{p}$ and $\mathbf{q}$ and their respective count vectors $\mathbf{x}$ and $\mathbf{y}$, we now define an *additive property kernel* as $k(\mathbf{p}, \mathbf{q}) = \mathbf{x}^T \mathbf{y}$.

2.3. Combined kernel function. Given a set of property kernels $\{k_i(\mathbf{x}, \mathbf{y})\}$, we construct the composite property kernel as a linear combination

$$k(\mathbf{x}, \mathbf{y}) = \sum_i k_i(\mathbf{x}, \mathbf{y})$$

where before and after combining, each kernel is normalized using

$$k(\mathbf{x}, \mathbf{y}) \leftarrow \frac{k(\mathbf{x}, \mathbf{y})}{\sqrt{k(\mathbf{x}, \mathbf{x})k(\mathbf{y}, \mathbf{y})}}.$$

2.4. Computational complexity. The computation of the count vector can be accomplished in $O(\ell mn)$ time if each n -dimensional vector from $\mathbf{p}$ is compared with all centroids in $\mathcal{C}$. Approximation algorithms are available through a decision tree like organization of the centroids. In such a case, only $\log m$ operations are needed resulting in $O(\ell n \log m)$ time [Gersho and Gray, 1992]; however, there is no guarantee that the closest centroid will be found. Using a full-search scheme, the basic and generalized property kernels can be computed in $O(\ell nm)$ and $O(m^2 + \ell nm)$ times, respectively. The storage requirements include $O(mn)$ for storing $\mathcal{C}$ and $O(m^2)$ for storing $\mathbf{Q}$. The additive kernel has complexity $O(\ell nm)$.

2.5. String kernel. As a baseline method with which we could compare results from our VQ model we utilized a string kernel as described by Leslie et al. [2002] for a wide range of word sizes ($n \in \{1 \dots 10\}$). For a given word size n , a sparse 20^n -length vector was created for a protein, where each value represented the number of times a potential n length substring that could be generated

using the 20 amino acid alphabet occurred. An ℓ -length sequence, $\mathbf{s}$, would contain $\ell - n + 1$ such overlapping strings.

3. Mapping proteins into property vectors

3.1. Structure based properties. Structure based properties were generated by converting the 3D coordinates of amino acids into backbone angles. These angles can be used to position each amino acid in relation to surrounding atoms in the protein chain. The usefulness of representing a structure in this manner is that backbone angles are the same regardless of how the overall protein is oriented in space, whereas 3D coordinates can change for the exact same structure simply by rotating or shifting it.

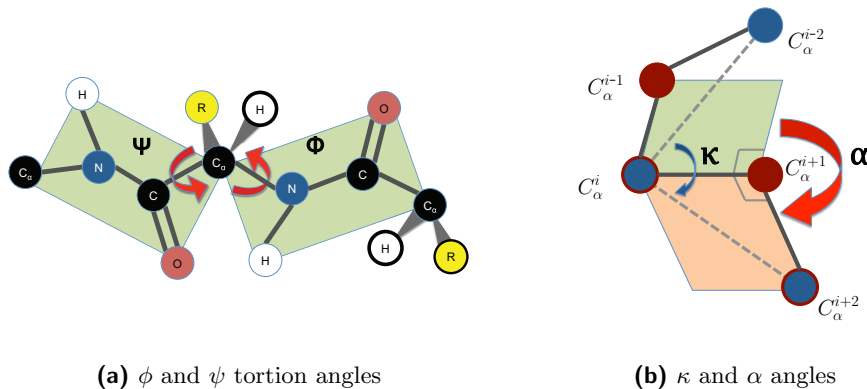


Figure 3.3. The different types of angles that can be measured from protein structures are shown. Section 3.1 illustrates the backbone torsion angles, ϕ and ψ , for a C_α atom that can be measured between its adjacent nitrogen and C' atoms respectively. Figure 3.3a illustrates the angles that can be measured between successive C_α atoms: C_α^{i-1} and C_α^{i+2} in the case of κ angles; and C_α^{i-2} and C_α^{i+2} in the case of α angles.

We utilized four types of backbone angles: κ , α , ϕ , and ψ . Figure 3.3 illustrates how these angles are defined using a protein chain. As illustrated by Section 3.1, ϕ and ψ angles represent the torsion angle, or amount of rotation, of the bond between C_α side-chain attached carbon atom and the adjacent nitrogen (ϕ angle) and C' carbon (ψ angle) atoms. While ϕ and ψ represent torsion angles between adjacent atoms in a protein's backbone, κ and α angles represent the relationship between more distant C_α atoms. An ℓ amino acid length sequence, $\mathbf{s} = s_1, s_2 \dots s_\ell$, will have ℓ

associate C_α carbon atoms: $C_\alpha^1, C_\alpha^2 \dots C_\alpha^\ell$. Using Figure 3.3a as a reference, if one designates a C_α^i atom as a vertex (in the angular sense), the κ angle represents the angle obtained by extending a line from the C_α^i atom’s position to the C_α^{i-2} and C_α^{i+2} atoms. The α angle represents the dihedral angle measured between the C_α^{i-1} and C_α^{i+2} atoms when two planes are generated using the C_α^i atom to the C_α^{i+1} to define the line where the two planes meet.

Angles κ , α , ϕ , and ψ were obtained using DSSP [Kabsch and Sander, 1983]. In addition to generating backbone angles, DSSP also outputs solvent accessibility values which we used as the fifth structure based property.

3.2. Sequence based properties. We generated seven sequence based properties for both the task of categorizing structures into SCOP folds and predicting GO function. These features were generated in order to represent a biologically relevant property associated with a region of a protein sequence: (i) hydrophobicity, calculated using the the Kyte-Doolittle scale [Kyte and Doolittle, 1982] in a sliding window with $w = 11$; (ii) flexibility, calculated as predicted B-factors using our previous model [Radivojac et al., 2004]; secondary structure predictions of (iii) helix, (iv) loop and (v) sheet propensities using our in-house predictor; and intrinsic disorder, (vi) using the previously published VSL2B model [Peng et al., 2006] as well as (vii) predictions from the same in-house predictor used for secondary structures.

3.3. Tested values of window size and number of centroids. We tested a range of combinations of values for n (window size), and m (number of centroids) for each property. For values of n , we tested $n \in \{2^i : i = 2 \dots 5\}$. Similarly, for the number of centroids, m , we tested $m \in \{2^i : i = 6, 8, 10, 12\}$. For all values of m and n we performed k-means clustering using 10^6 randomly sampled vectors from all data points.

4. Experiments and data

In the first experiment prediction was performed as a one-versus-all classification at the SCOP fold level for proteins categorized as α , β , $\alpha + \beta$, or α/β folds. We utilized Astral 1.75A, which represents a database of proteins classified according to tertiary structures as defined by SCOP [Murzin et al., 1995]. In order to ensure that redundancy in the data set did not lead to inflated assessment of performance we chose the 40% non-redundant version of the database. Table 3.1 summarizes the positive and negative data points used for each category of SCOP fold.

SCOP Fold	positives	negatives
all α	1,901	7,486
all β	2,175	7,212
$\alpha + \beta$	2,665	6,722
α/β	2,646	6,741

Table 3.1. Summary of data used for SCOP fold classification documenting the number of positives and negatives used for the classification of protein structures as α , β , $\alpha + \beta$, or α/β folds.

In the second experiment we attempted to distinguish enzymes, or those proteins annotated with the term “catalytic activity”, from all other proteins. GO annotations were obtained from the April 2012 release of Swiss-Prot [Bairoch et al., 2005] in conjunction with the May 4, 2012 version of the gene ontology [Ashburner et al., 2000]. Only annotations supported by evidence codes EXP, IDA, IPI, IMP, IGI, IEP, TAS, IC were used. This resulted in a total of 24,882 proteins with experimentally verified annotations, 9,506 of which were annotated with the term “catalytic activity” and 18,936 of which represented putative negative data points.

4.1. Cross-validation. We performed 10-fold cross validation on both data sets. For each fold non-overlapping test and training sets, consisting of $\frac{1}{10}$ -ths and $\frac{9}{10}$ -ths of the data respectively, were generated. All 10 testing sets were mutually exclusive and collectively exhaustive. We created a single set of testing folds for all classification sub-tasks: one set for SCOP fold classification and a separate for the prediction of catalytic activity.

We took steps in order to ensure that positive data points for each class were evenly distributed among folds. Because protein structures are only assigned to a single SCOP class ensuring equal distribution among testing folds was very simple: we merely assigned a random $\frac{1}{10}$ of all proteins for a tested SCOP class to each testing fold. The same was done when creating testing folds for “catalytic activity” classification. We equally distributed terms belonging to each subclass of “catalytic activity” ensuring that proteins that may have belonged to more than one class only appear in one testing fold and did not appear more than once in each fold.

4.2. Performance accuracy estimation. For each binary classification experiment we calculated the area under the Receiver Operating Characteristic curve (AUC). Predictor scores were first normalized to fall between the range of $[0, 1]$. A decision threshold, τ , was varied from 0 to 1 in increments of .01. At each threshold, sensitivity (sn_τ) and specificity (sp_τ) were calculated. The AUC of a model was then calculated using the trapezoid rule as

$$AUC = \sum_{\tau \in \mathcal{T}} sp_\tau - sp_{(\tau-.01)} \frac{sn_\tau - sn_{(\tau-.01)}}{2}$$

where $\mathcal{T}$ defines the set of thresholds.

While we evaluated each feature type for a combination of window size and number of clusters on each classification task separately we also desired to obtain a single value that could be used to benchmark each combination of parameters on all evaluated SCOP classes. In order to do this we averaged AUC across multiple one-vs-all classification tasks. While it would be simple to merely average the AUC values for each SCOP fold; if, for example, a fold class has few positive examples relative to others it will disproportionately influence the average.

In order to account for class imbalances between each SCOP fold we utilized a weighted method for calculating AUC : AUC^w . If we have a set of classes $C = \{c_1, c_2 \dots c_n\}$ on which binary predictions were carried out, with associated counts of positive data points $N = \{n_1, n_2 \dots n_n\}$ and associated AUC values $AUC = \{auc_1, auc_2 \dots auc_n\}$ we calculate the weighted average AUC , as

$$AUC^w = \frac{\sum_{i=1..n} n_i \times auc_i}{\sum_{i=1..n} n_i}$$

4.3. Signal-to-noise ratio. We also calculated the signal-to-noise ration (SNR) obtained when encoding and decoding-property based representations of proteins using VQ. Given an original property vector $\mathbf{p}$ and the reconstructed version of this vector $\hat{\mathbf{p}}$ the signal to noise ratio was calculated using the logarithmic decibel scale as

$$SNR_{dB}(\mathbf{p}, \hat{\mathbf{p}}) = \log_{10} \left(\frac{\sum_{i=1}^{\ell} p_i^2}{\sum_{i=1}^{\ell} (p_i - \hat{p}_i)^2} \right)$$

On this scale one decibel signifies that the noise (or squared difference between the original and reconstructed signal) represents $1/10$ -th of the signal.

5. Results

5.1. Performance of individual properties.

5.1.1. *SCOP fold.* Table 3.2 shows the performance of each property when predicting SCOP folds. Among structure-based properties we found that κ backbone angles had the best performance, both for individual SCOP folds and in terms of its AUC^w (0.961). Solvent accessibility (SolAcc) values performed the worst out of structure-based properties, obtaining the lowest AUC for all SCOP folds and in terms of AUC^w (0.868).

In general all structure-based properties outperformed sequence-based properties. Among sequence-based properties predicted secondary structures performed the best, especially predictions that a residue is in a helix ($AUC^w = 0.788$), a loop ($AUC^w = 0.771$), or a sheet ($AUC^w = 0.787$). Calculated hydrophobicity and VSL2B based predictions of disorder propensity performed the worst ($AUC^w = 0.707$ and $AUC^w = 0.682$ respectively). As a predictor of disorder PDBDisorder-based features performed better than VSL2B ($AUC^w = 0.764$), although this may be based on information leak due to this predictor being trained on PDB data.

5.1.2. *Catalytic activity.* The first column of results in Table 3.3 shows the performance of each property when predicting whether a protein is annotated with the catalytic activity term. We found that disorder-based predictions (VSL2B $AUC = 0.742$ and PDBDisorder $AUC = 0.718$) and predicted B-factors ($AUC = 0.722$) performed the best, whereas, contrary to their performance in distinguishing between SCOP folds, predicted secondary structures performed the worst (Helix $AUC = 0.687$, Loop $AUC = 0.698$, and Sheet $AUC = 0.681$).

5.1.3. *Catalytic activity subclass.* Table 3.3 shows the performance of each property when predicting the tested subclasses of catalytic activity. As with predicting catalytic activity, predicted B-factors also performed well in predicting catalytic activity subclass ($AUC^w = 0.659$). Predicted secondary structures had mixed performance, with predicted loops obtaining a comparatively high AUC^w of 0.647, whereas helix and sheet predictions only achieved a AUC^w of 0.511 and 0.621 respectively.

Property\Category	All α			All β			α/β			$\alpha + \beta$			AUC^w		
	m	n	AUC	m	n	AUC	m	n	AUC	m	n	AUC	m	n	AUC^w
Alpha	4,096	8	0.994	4,096	8	0.982	4,096	16	0.967	4,096	32	0.829	4,096	8	0.930
Kappa	4,096	16	0.995	4,096	16	0.986	4,096	16	0.978	4,096	32	0.903	4,096	32	0.961
Phi	4,096	16	0.990	4,096	16	0.975	4,096	16	0.964	4,096	32	0.809	4,096	32	0.920
Psi	4,096	8	0.991	4,096	16	0.981	4,096	16	0.970	4,096	32	0.850	4,096	32	0.938
SolAcc	4,096	8	0.960	4,096	8	0.951	4,096	32	0.914	4,096	32	0.710	4,096	32	0.868
B-factors	256	8	0.790	256	8	0.709	64	8	0.809	16	4	0.587	64	4	0.717
Helix	16	4	0.868	16	8	0.871	64	16	0.842	64	32	0.637	64	32	0.788
Hydrophobicity	256	4	0.711	1,024	4	0.728	64	2	0.834	64	2	0.585	256	4	0.707
Loop	256	16	0.872	64	8	0.848	16	32	0.821	1,024	32	0.607	64	16	0.771
PDBDisorder	1,024	8	0.842	64	4	0.849	64	4	0.819	256	32	0.591	64	4	0.764
Sheet	64	2	0.904	64	8	0.853	256	16	0.836	64	32	0.641	64	32	0.787
VSL2B	16	2	0.699	64	32	0.628	64	8	0.812	64	2	0.589	64	8	0.682
String Kernel	–	2	0.863	–	3	0.878	–	4	0.860	–	5	0.634	–	4	0.794
Sequence	64	16	0.915	64	16	0.876	64	16	0.880	64	16	0.621	64	16	0.813
Structure	4,096	32	0.992	4,096	32	0.983	4,096	32	0.979	4,096	32	0.903	4,096	32	0.961
Sequence + Structure	–	–	0.989	–	–	0.967	–	–	0.970	–	–	0.851	–	–	0.939
Sequence + Structure + SK	–	–	0.989	–	–	0.967	–	–	0.970	–	–	0.851	–	–	0.939

Table 3.2. Optimal performance, according to AUC , of each property based feature when predicting SCOP fold. For each property feature the combination of m and n values that obtained the highest AUC for an individual SCOP fold are reported. The last three columns show the weighted AUC , AUC^w , obtained across all SCOP folds. Results when combining all sequence and property based features are shown in the bottom section of the table. Because different combinations of m and n are used when combining properties these values are not shown for the Sequence + Structure and Sequence + Structure + SK combination of features.

Property \ Category	catalytic activity			oxidoreductase activity			transferase activity			hydrolase activity			lyase activity			isomerase activity			ligase activity			AUC^w		
	m	n	AUC	m	n	AUC	m	n	AUC	m	n	AUC	m	n	AUC	m	n	AUC	m	n	AUC	m	n	AUC^w
B-factors Helix Hydrophobicity Loop PDBDisorder Sheet VSL2B	256	32	0.722	4,096	32	0.709	4,096	32	0.644	4,096	32	0.630	4,096	4	0.688	4,096	32	0.738	4,096	32	0.671	-	-	0.659
	256	8	0.687	4,096	8	0.634	4,096	32	0.603	4,096	32	0.600	4,096	32	0.644	4,096	32	0.665	4,096	8	0.595	-	-	0.611
	4,096	2	0.701	4,096	16	0.695	4,096	16	0.622	4,096	32	0.653	4,096	4	0.692	4,096	16	0.722	4,096	32	0.638	-	-	0.653
	256	32	0.698	4,096	32	0.678	4,096	32	0.642	4,096	32	0.628	4,096	32	0.664	4,096	32	0.712	4,096	32	0.638	-	-	0.647
	256	32	0.718	4,096	32	0.683	4,096	32	0.645	4,096	16	0.640	4,096	32	0.676	4,096	32	0.717	4,096	16	0.635	-	-	0.65
String Kernel Combined Combined + SK	256	4	0.681	4,096	32	0.651	4,096	32	0.612	4,096	32	0.606	4,096	32	0.649	4,096	32	0.683	1,024	32	0.611	-	-	0.621
	256	16	0.742	1,024	32	0.661	4,096	16	0.594	4,096	32	0.614	4,096	32	0.638	4,096	32	0.680	1,024	8	0.630	-	-	0.620
	-	5	0.857	-	5	0.949	-	6	0.928	-	6	0.934	-	6	0.929	-	5	0.920	-	5	0.910	-	-	0.930
NR_{40} String Kernel NR_{40} Combined NR_{40} Combined + SK	256	16	0.776	4,096	32	0.805	4,096	32	0.753	4,096	32	0.754	4,096	32	0.779	4,096	32	0.816	4,096	32	0.763	4,096	32	0.767
	-	-	0.781	-	-	0.805	-	-	0.753	-	-	0.754	-	-	0.779	-	-	0.816	-	-	0.763	-	-	0.767
	-	5	0.733	-	-	0.733	-	-	0.619	-	-	0.612	-	-	0.725	-	-	0.669	-	-	0.668	-	-	0.649
NR_{40} String Kernel NR_{40} Combined NR_{40} Combined + SK	256	16	0.775	4,096	32	0.681	4,096	32	0.560	4,096	32	0.545	4,096	32	0.617	4,096	32	0.641	4,096	32	0.570	4,096	32	0.583
	-	-	0.775	-	-	0.681	-	-	0.559	-	-	0.548	-	-	0.629	-	-	0.649	-	-	0.574	-	-	0.585

Table 3.3. Optimal performance, according to AUC , of each property based feature when predicting “catalytic activity” subclass. For each property feature the combination of m and n values that obtained the highest AUC for an individual “catalytic activity” subclass are reported.

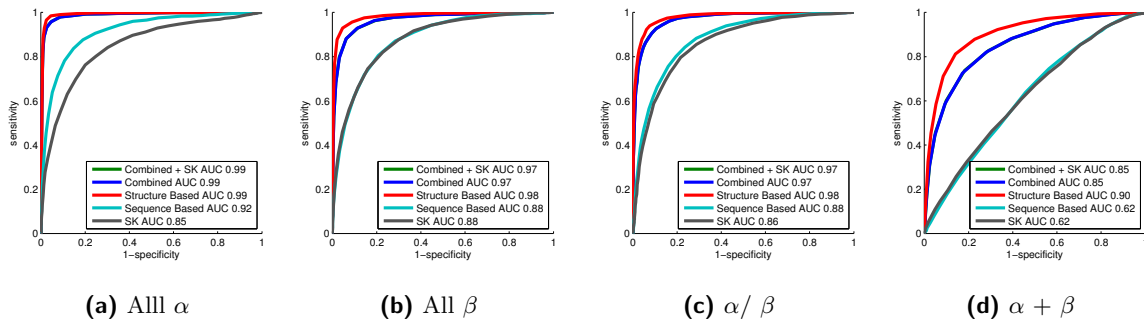


Figure 3.4. SCOP ROC curves showing the performance of each combination of combined features and the string kernel when predicting each category of SCOP fold tested.

5.2. Combined kernel performance. In order to test the predictive ability of combining multiple properties we implemented a combined kernel that combined multiple properties. In order to reduce the computational complexity of this task we only combined properties given utilizing m and n values that achieved the highest AUC^w for each property type.

5.2.1. SCOP fold. We observed marginal improvements in performance when combining sequence based properties, achieving a AUC^w of 0.81 compared to the best performance of an individual sequence-based property of 0.78 (helix predictions). The combined kernel for structure-based properties saw no improvement over the maximum performance obtained for an individual structure property ($AUC^w = 0.961$ for both the combined kernel and κ angles), and actually observed decreased performance when combining both sequence and structure-based properties ($AUC^w = 0.939$).

5.2.2. Catalytic activity. We obtained marginal improvements when combining sequence based properties, achieving a AUC^w of 0.776 compared to an AUC^w of 0.742 for VSL2B.

5.2.3. Catalytic activity subclass. The most marked improvement in performance obtained by the combined kernel was when predicting catalytic activity subclass, achieving a AUC^w of 0.767 compared to the highest AUC^w obtained by an individual sequence property of 0.659 (predicted B-factors).

5.3. String kernel performance. We found that the string kernel outperformed sequence based properties in both the task of predicting catalytic activity and its subclass (AUC^w of 0.857

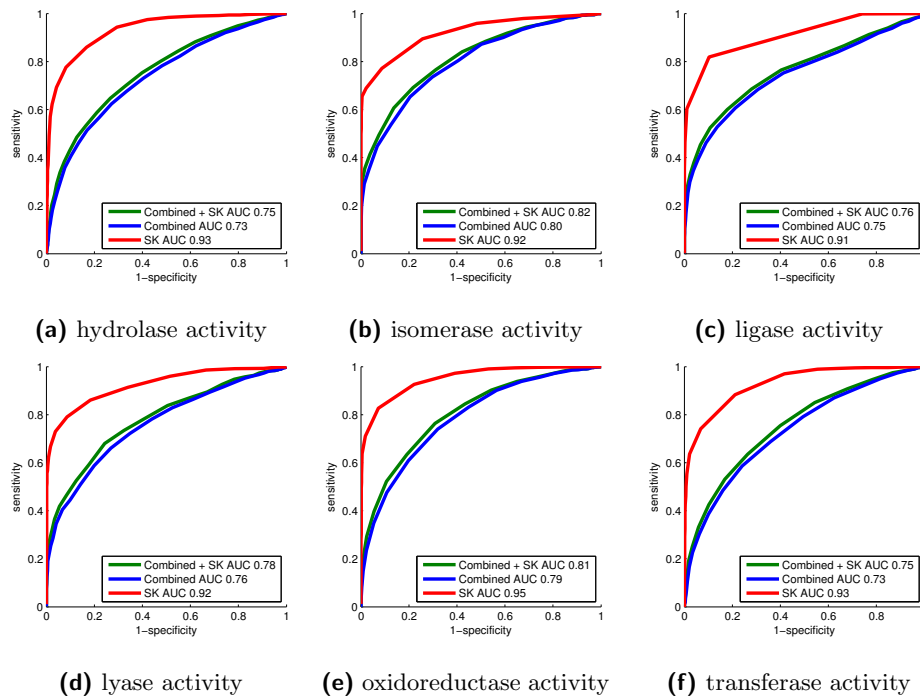


Figure 3.5. ROC curves when showing the performance of each combination of combined features and the string kernel when predicting each subclass of "catalytic activity".

and 0.930) respectively. On the other hand, the string kernel did not show superior performance when predicting SCOP fold, only obtaining a AUC^w of 0.794 compared to an AUC^w of 0.961 obtained by the combined structure kernel.

5.4. Optimal parameter values.

5.4.1. *SCOP fold.* We found that structure-based properties consistently preferred large numbers of centroids, obtaining maximum AUC at $m = 4,096$ for all structure-based properties and all classification tasks. Optimal window sizes were 8 or 16 amino acids for all SCOP folds besides $\alpha + \beta$ which favored wider window size, obtaining a maximum AUC at $n = 32$ for all folds.

Sequence-based properties were less consistent in the optimal values of m and n , covering a range of values for each feature and SCOP fold.

5.4.2. *Catalytic activity.* There was very little variation in preferred values of m when predicting catalytic activity with all features aside from predicted hydrophobicity obtaining maximum AUC

values at $m = 256$. There was much more variation in preferred window sizes with hydrophobicity obtaining smallest optimal window size of 2, and B-factors and PDB Disorder preferring longer window sizes ($n = 32$).

5.4.3. *Catalytic activity subclass*. Sequence based properties were much more consistent in the values of m and n preferred when predicting catalytic activity subclass, almost always preferring large values of m (4,096)

5.5. Comparing AUC and SNR. Figure 3.6 shows a scatterplot of SNR and AUC^w values. We found that derived backbone angles obtained the lowest SNR values. Similarly, properties that obtained low AUC^w values seemed to have a higher range of SNR values. Although backbone angles, as a class, obtained lower values of SNR , these values did not seem to correlate with higher AUC values; and we found very little correlation with SNR obtained at particular values of AUC for this class of features Table 3.4, although they were the only property class to obtain a positive correlation between SNR and AUC . The opposite was the case for VSL2B based disorder predictions and predicted secondary structure. These features, which obtained the worst performance when predicting SCOP fold had the strongest negative correlation between SNR and AUC . While the trend of positive correlation (although weak) between SNR and AUC was observed for derived backbone angles, the overall trend for other features went against this trend.

Property Class	Correlation Coefficient
Backbone angles	0.07
Secondary structure	-0.52
PDB Disorder	-0.21
VSL2B	-0.58
Hydro	-0.07
B-Factor	-0.21

Table 3.4. The correlation between SNR and AUC^w for various properties.

5.6. Reduced redundancy data set. We also generated a reduced redundancy data set of proteins with GO annotations in which the maximum pairwise sequence identity output by BLAST between any two sequences was 40%. This non-redundant 40% data (NR40) set was generated to

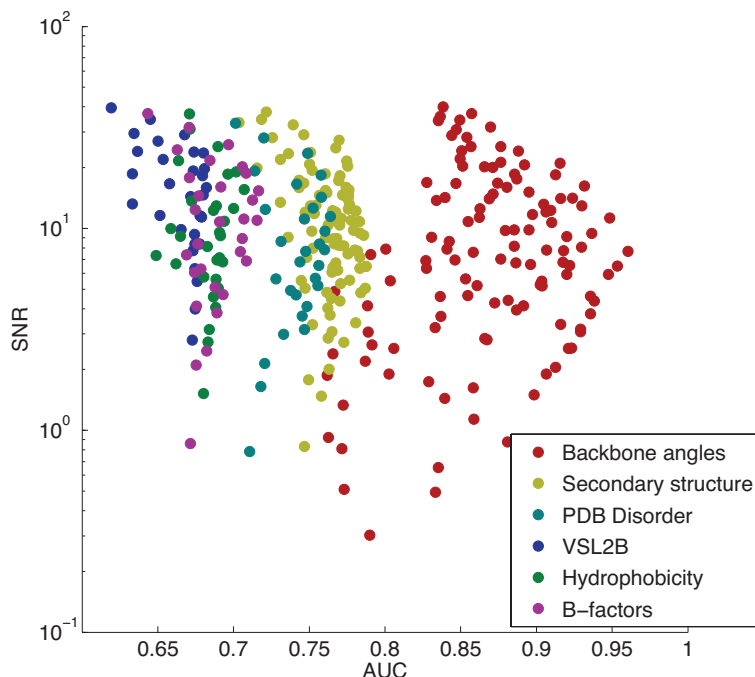


Figure 3.6. Comparison of AUC and SNR . For each feature type obtained SNR values were plotted as a function of AUC^w values for the prediction of SCOP fold.

simulate the performance of each property when, for a given query protein, there is no sequence that is both annotated and of a reasonable level of sequence similarity.

We found that although the string kernel outperformed sequence properties when predicting catalytic activity, the combined sequence property kernel outperformed the string kernel when predicting catalytic activity. As shown by Figure 3.7 the combined sequence property kernel obtains slightly better performance than the string kernel, obtaining an AUC of 0.78 compared to 0.73 respectively. Interestingly, the combined string kernel and sequence property kernel did not achieve improved performance, obtaining the same AUC as the combined sequence property kernel (0.78). This is similar to the performance of the same set of features when applied to the non reduced redundancy data set, although in that case the combined string kernel and sequence property kernel grouped with the sequence property features.

When predicting catalytic activity subclass we observed similar results as on the unfiltered data set; the string kernel obtained higher AUC values than both the sequence property kernel and the combined string kernel and sequence property kernel for all tested subclasses of catalytic activity.

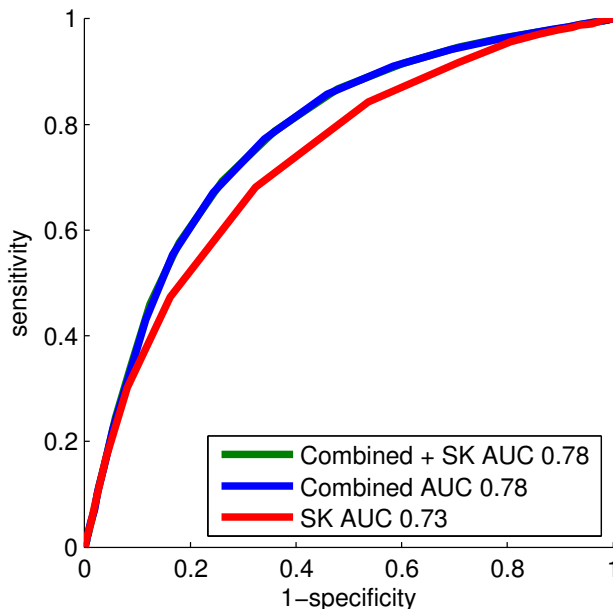


Figure 3.7. ROC curves obtained when combining the string kernel with sequence properties (green line), combining all sequence properties (blue curve), and the string kernel (red curve) when predicting catalytic activity on the 40% non redundant data set of proteins.

6. Discussion

Several aspects of results show that the VQ kernel holds much potential. The marked improvement in performance compared to the string kernel in predicting SCOP fold indicates that our novel structure properties and VQ kernel are a viable framework for the task of categorizing protein structure. While structure-based properties exhibited superior performance when compared to the string kernel, the same could not be said for sequence-based properties; which underperformed both the string kernel and structure properties in all classification tasks. There are several explanations for why this was observed. First, in the task of SCOP fold prediction, it is intuitive that structure based features would outperform all other features. The general preference of sequence-based properties to prefer lower values of m in the task of SCOP fold prediction may be an indicator that sampled vectors sparsely filled n dimensional space. Sequence properties did perform well when applied to the task of predicting catalytic activity on the NR40 data set, obtaining an *AUC* of 0.78 compared to the string kernels *AUC* of 0.73 (Figure 3.7). This indicates that when attempting to classify a novel

sequence predicted properties may be a better way to encode a protein than simply representing it as a sequence of amino acids.

Traditionally, protein sequences are represented as sequences of letters representing amino acids, or as the coordinates of atoms in 3-dimensional space. Alignment algorithms such as BLAST [Altschul et al., 1997] do not explicitly consider the physiochemical properties associated with each amino acid; it would be a stretch to argue that through substitution matrices such as PAM [Dayhoff et al., 1978] and BLOSUM [Henikoff and Henikoff, 1992] alignment algorithms implicitly encode physiochemical similarities between amino acids. It could be presumed that much of what these substitution matrices capture is the propensity of one amino acid to transition into another due to physiochemical similarity. Evolutionarily related sequences can diverge at the sequence level yet still retain the same functions. Conversely, similar structures and functions can be carried out by unrelated, and potentially very distant at the sequence level, proteins due to convergent evolution. In this paper we made a conscious choice to encode proteins; not as a sequence of letters, or 3D coordinates, but explicitly as the physiochemical and structural characteristics associated with those amino acids. We found that this representation when combined the the VQ framework enable fast and accurate classification of protein sequences, especially when categorizing tertiary structures.

In addition to proposing a fundamental shift in the way protein sequences and structures are represented, our VQ kernel is robust and can easily be extended to any type of data that can be represented or transformed into a time-series representation. While the focus of this project was on the development of novel representations of proteins and the testing of the basic VQ concept, potentially large improvements in performance might be obtained by fine tuning parameters and implementing more advanced clustering and learning algorithms. Specifically, structured learning methods should improve performance when performing multi-label classification as in the case of predicting protein function.

CHAPTER 4

Information-theoretic evaluation of predicted ontological annotations

Ontological representations have been widely used in biomedical sciences to standardize knowledge representation and exchange [Robinson and Bauer, 2011]. Modern ontologies are typically viewed as graphs in which vertices represent terms or concepts in the domain of interest and edges represent relational ties between terms (e.g. is-a, part-of). While, in theory, there are no restrictions on the types of graphs used to implement ontologies, hierarchical organizations, such as trees or directed acyclic graphs, have been frequently used in the systematization of biological experiments, organismal phenotypes, or structural and functional descriptions of biological macromolecules.

In molecular biology, one of the most frequently used ontologies is the Gene Ontology [Ashburner et al., 2000], which standardizes the functional annotation of genes and gene products. The development of the Gene Ontology (GO) was based on the premise that the genomes of all living organisms are composed of genes whose products perform functions derived from a finite molecular repertoire. In addition to knowledge representation, GO has also facilitated large-scale analyses and automated annotation of gene product function [Radivojac et al., 2013]. As the rate of accumulation of uncharacterized sequences far outpaces the rate at which biological experiments can be carried out to characterize those sequences, computational function prediction has become increasingly useful for the global characterization of genomes and proteomes as well as for guiding biological experiments via prioritization [Sharan et al., 2007, Rentzsch and Orengo, 2009].

The growing importance of tools for the prediction of GO annotations, especially for proteins, presents the problem of how to accurately evaluate such tools. Because terms can automatically be associated with their ancestors in the GO graph, the task of an evaluation procedure can be framed as comparing the predicted graph with the graph represented by the true, experimentally verified annotations. It should be explicitly noted that the structure of the ontology introduces dependence between terms, dependencies that must be appropriately considered when comparing two graphs.

This is obviously true for terms that share an ancestor/descendant relationship, but also for terms that are not related through direct ancestry. For example, although the terms “H3 histone acetyltransferase activity” and “H4 histone acetyltransferase activity” do not share an ancestor/descendant relationship and describe different behaviors of histone acetyltransferases, they nonetheless are not independent.

Protein function is also complex and context dependent. A single biological experiment rarely results in complete characterization of a protein’s function. Furthermore, annotations might not always be meaningful; a point that is particularly evident in cases where only high-throughput experiments are used for functional characterization, leading to shallow annotation graphs. These types of annotations pose a problem in evaluation as the ground truth is incomplete and noisy.

Furthermore, different computational models produce different outputs that must be accounted for. Some models simply assign binary yes-or-no values to terms; while others might assign a different score to potentially each node in the ontology, with an expectation that a good decision threshold would be applied to provide useful annotations. Finally, a complicating factor is posed by the fact that GO, as most current ontologies, is generally unfinished and contains a range of specificities of functional descriptions at the same depth of the ontology [Alterovitz et al., 2010].

There are two important factors related to the development of evaluation metrics. First, because both the experimental and predicted annotation of genes can be represented as subgraphs of the generally much larger GO graph, it is unlikely that a given computational method will provide an exact prediction of the experimental annotation. It is therefore necessary to develop metrics that facilitate calculating degrees of similarity between pairs of graphs and appropriately address dependency between nodes. Ideally, such a measure of similarity would be able to characterize both the level of correct prediction of the true (albeit incomplete) annotation but also the level of misannotation. The second important factor related to the evaluation metric is its interpretability. Characterizing the predictor’s performance should be meaningful to a downstream user. Ideally, an evaluation metric would have a simple probabilistic interpretation.

In this chapter, we develop an information-theoretic framework for evaluating the prediction accuracy of computer-generated ontological annotations. We first use the structure of the ontology to probabilistically model, via a Bayesian network, the prior distribution of protein experimental annotation. We then apply our metric to three protein function prediction algorithms selected

to highlight the limitations of typically-considered evaluation metrics. We show that our metrics provide added value to the current analyses of the strengths and weaknesses of computational tools. Finally, we argue that our framework is probabilistically well founded and show that it can also be used to augment already existing evaluation metrics.

1. Background

The issue of performance evaluation is closely related to the problems of measuring similarity between pairs of graphs or sets. First, we note that a protein’s annotation (experimental or predicted) is a graph containing a subset of nodes in the ontology together with edges connecting them. We use the term *leaf node* to describe a node that has no descendants in the annotation graph, although it is allowed to have descendants in the ontology. A set of leaf terms completely describes the annotation graph.

We roughly group both graph similarity and performance evaluation metrics into topological and probabilistic categories, and note that a particular metric may combine aspects from both. More elaborate distinctions are provided by Pesquita et al. [2009] and Guzzi et al. [2012]. Topological metrics rely on the structure of the ontology to perform evaluation and typically employ metrics that operate on sets of nodes and/or edges. A number of topological measures have been utilized, including the Jaccard and cosine similarity coefficients (the cosine function maps the binary term designations into a vector space), the shortest path-based distances [Rada et al., 1989], etc. In the context of classifier performance analysis, two common 2D metrics are the precision/recall curve and the Receiver Operating Characteristic (ROC) curve. Both curves are constructed based on the overlap in either edges or nodes between true and predicted terms and have been widely used to evaluate the performance of tools for the inference of GO annotations. They can also be used to provide a single statistic to rank classifiers through the maximum F-measure in the case of precision/recall curve or the area under the ROC curve. The area under the ROC curve has a limitation arising from the fact that the ontology is relatively large, but that the number of terms associated with a typical protein is relatively small. In practice, this results in specificities close to one regardless of the prediction, as long as the number of predicted terms is relatively small.

Although these statistics provide good feedback regarding multiple aspects of a predictor’s performance, they do not always address node dependency or the problem of unequal specificity of

functional annotations found at the same depth of the graph. Coupled with a large bias in the distribution of terms among proteins, prediction methods that simply learn the prior distribution of terms in the ontology could appear to have better performance than they actually do.

The second class of similarity/performance measures are probabilistic or information-theoretic metrics. Such measures assume an underlying probabilistic model over the ontology and use a database of proteins to learn the model. Similarity is then assessed by measuring the information content of the shared terms in the ontology but can also take into account the information content of the individual annotations. Unlike with topological measures where updates to the ontology affect similarity between objects, information theoretic measures are also affected by changes in the underlying probabilistic model even if the structure of the ontology remains the same.

Probabilistic metrics closely follow and extend the methodology laid out by Resnik [1995], which is based on the notion of information content between a pair of individual terms. These measures overcome biases related to the structure of the ontology; however, they have several drawbacks of their own. One that is especially important in the context of analyzing the performance of a predictor is that they only report a single statistic, namely the similarity or distance between two terms or sets of terms. This ignores the tradeoff between precision and recall that any predictor has to make. In the case of Resnik’s metric, a prediction by any descendant of the true term will be scored as if it is an exact prediction. Similarly, a shallow prediction will be scored the same as a prediction that deviates from the true path at the same point, regardless of how deep the erroneous prediction might be. Although some of these weaknesses have been corrected in subsequent work [Jiang and Conrath, 1997, Lin, 1998, Schlicker et al., 2006], there remains the issue that the available probabilistic measures of semantic similarity resort to ad hoc solutions to address the common situation where proteins are annotated by graphs that contain multiple leaf terms [Clark and Radivojac, 2011]. Various approaches have been taken, including averaging between all pairs of leaf terms [Lord et al., 2003], finding the maximum among all pairs [Resnik, 1999], or finding the best-match average, but each such solution lacks strong justification in general. For example, all-pair averaging leads to anomalies where the exact prediction of an annotation containing a single leaf term u would be scored higher than the exact prediction of an annotation containing two distinct leaf terms u and v of equal information content, when it is more natural to think that the latter prediction should be scored higher. Finally, all semantic similarity metrics that incorporate some

form of pairwise matching between leaf terms tacitly assume that the objects to be compared are annotated by similar numbers of leaf terms. As such, they could produce undesirable solutions when applied to a wide range of prediction algorithms such as those outputting a very large number of predicted terms.

2. Protein function prediction scenarios

In protein function prediction, there are two scenarios in which a computational model can be constructed [Radivojac et al., 2013]: (i) given a new protein, the task of a classifier is to find all functional terms that the protein is associated with (“what is the function of this protein?”); and (ii) given a functional term, the task of a classifier is to find all proteins associated with this term (“what are the proteins associated with this function?”). Undoubtably, the two prediction scenarios are related because a perfect predictor of protein function would solve both questions at the same time. However, imperfect predictors may address one question better than the other. For example, a classifier built to address the first question is expected to be accurate in predicting all (or many) functional terms and that prediction scores over all functional terms are comparable. On the other hand, a predictor developed to consider only one functional term at a time need not consider any other term. Such models can perform well even if they just rank all test proteins. In addition to these differences, the two types of models are evaluated in different ways (yet they use the same terminology).

Models that are concerned with predicting function on a previously unseen protein (scenario 1, above) need to devise evaluation metrics to estimate the expected accuracy of a predicted consistent graph P when the experimental (true) annotation of the protein is graph T . Alternatively, the models that are concerned with ranking the proteins according to their likelihood to be associated with a particular functional term v (scenario 2, above) need to be evaluated based on the expectation that a particular protein is associated with a functional term v . Here, the models are usually evaluated for each functional term v , one at a time. Evaluation metrics corresponding to the former problem are significantly more challenging than the metrics corresponding to the latter problem. In the latter case, one can simply consider a particular decision threshold for predicting whether a protein is associated with function v and then calculate the fraction of positive predictions that are correct (precision) as well as the fraction of proteins known to be associated with functional term v that

have been retrieved (recall). Such evaluation has been discussed by Sharan et al. [2007], among others. Calculating precision and recall for the former scenario is the topic of our study.

3. Methods

Our objective here is to introduce information-theoretic metrics for evaluating classification performance in protein function prediction. In this learning scenario, the input space $\mathcal{X}$ represents proteins whereas the output space $\mathcal{Y}$ contains directed acyclic graphs describing protein function according to the Gene Ontology (GO).

Because of the hierarchical nature of GO, both experimental and computational annotations need to satisfy the *consistency requirements*:

- i.* If vertex (term) v from the ontology is true, then all of its ancestors must also be true.
- ii.* If vertex (term) v from the ontology is false, then all of its descendants must also be false.

By enforcing these requirements we frame the task of a classifier as assigning the best consistent subgraph of the ontology to each new protein and output a prediction score for this subgraph and/or each predicted term.

We simplify the exposition by referring to such graphs as prediction or annotation graphs. In addition, we frequently treat consistent graphs as sets of nodes or functional terms and use set operations to manipulate them.

We now proceed to provide a definition for the information content of a (consistent) subgraph in the ontology. Then, using this definition, we derive information-theoretic performance evaluation metrics for comparing pairs of graphs.

3.1. Calculating the joint probability of a graph. Let each term in the ontology be a binary random variable and consider a fixed but unknown probability distribution over $\mathcal{X}$ and $\mathcal{Y}$ according to which the quality of a prediction process will be evaluated. We shall assume that the prior distribution of a target can be factorized according to the structure of the ontology, i.e. we assume a Bayesian network as the underlying data generating process for the target variable. According to this assumption, each term is independent of its ancestors given its parents and, thus, the full joint probability can be factorized as a product of individual terms obtained from the set of conditional probability tables associated with each term [Koller and Friedman, 2009],

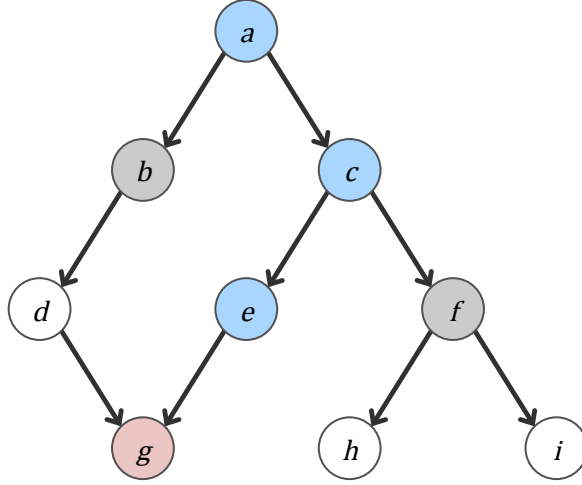


Figure 4.1. A directed acyclic graph representing the relationships between terms in an ontology. True term, or T , are represented by blue colored vertices: $\{a, c, e\}$. The children of true terms, $\mathcal{C}(T)$, are denoted by grey and light-red vertices: $\{b, f, g\}$. Terms which do not contribute anything to the full joint probability are shown in white and light-red: $\{d, g, h, i\}$. Term g is colored light red to point out a special case when calculating the full joint probability of a graph. Although one of term g 's parents are true, because it has one false parent it will not contribute to the calculation of the full joint probability because the probability it is false is always equal to one (Table 4.1). All blue and grey terms taken together comprise the *markov blanket* of T and are necessary when calculating the full joint probability of a configuration, or combination of true and false values for all terms, in $\mathbf{V}$.

$$(1) \quad \Pr(\mathbf{V}) = \prod_{v \in \mathbf{V}} \Pr(v | \mathcal{P}(v)),$$

where $\mathbf{V}$ denotes all vertices the ontology, v denotes a vertex in $\mathbf{V}$ and $\mathcal{P}(v)$ is the set of parent nodes of v . Here we use $\mathbf{V}$ (all terms) instead of T (only true terms) to illustrate the fact that we are calculating the probability of a given configuration of the full ontology with the inclusion of all true and false terms. In practice the true terms, or the subgraph T , are generally all that is considered both in the context of this chapter and in other papers addressing similar topics.

Table 4.1. A table showing a conditional probability distribution for term g from Figure 4.1. The purpose of this table is to illustrate how the consistency requirement affects joint probability tables in the bayesian network. If at least one of a term's parents are false it is guaranteed that term is also false. This is shown by considering the last three rows in the table.

		$\Pr(g)$	
d	e	True	False
True	True	.7	.3
True	False	0	1
False	True	0	1
False	False	0	1

The scope of terms considered can be reduced when calculating the joint probability of a configuration of $\mathbf{V}$ without affecting the final probability value. Because of the enforced consistency requirement (i.e. all ancestors of true terms are true; all descendants of false terms are false) the full joint probability of a configuration of the ontology, $\mathbf{V}$, can be calculated by considering only terms whose parents are all true. Equation (1) can be rewritten as

$$(2) \quad \Pr(\mathbf{V}) = \prod_{v \in T \cup \mathcal{C}(T)} \Pr(v | \mathcal{P}(v)),$$

where T denotes all terms in $\mathbf{V}$ that are true, $\mathcal{C}(T)$ defines the children of all terms in T (specifically, false terms all of whose parents are true), and $T \cup \mathcal{C}(T)$ defines the unique union of the two sets.

The derivation of Equation (2) can be obtained by considering Figure 4.1 which illustrates a sample ontology with 9 terms and Table 4.1 which represents the conditional probability distribution, or table, for vertex g . If we write the joint probability of $\mathbf{V}$ using Equation (1) we obtain

$$\begin{aligned} \Pr(\mathbf{V}) &= \Pr(a = 1) \times \Pr(b = 0 | a = 1) \times \Pr(c = 1 | a = 1) \times \Pr(d = 0 | b = 0) \\ &\quad \times \Pr(e = 1 | c = 1) \times \Pr(f = 0 | c = 1) \times \Pr(g = 0 | d = 0, e = 1) \\ &\quad \times \Pr(h = 0 | f = 0) \times \Pr(i = 0 | f = 0). \end{aligned}$$

First, we see that terms h and i contribute nothing to the joint probability because the probability a term is false given its parents are all false is always equal to 1. Although b , and f are false, because one of their parents are true the resulting probability is not 1. Term g can also be ignored because one of its parents is false. For example, consider Table 4.1, which shows the conditional probability table of term g . If we consider instances where one of g 's parents are false we see that the resulting values are guaranteed to be 1. The resulting joint probability can subsequently be rewritten as

$$\begin{aligned} \Pr(\mathbf{V}) &= \Pr(a = 1) \times \Pr(b = 0|a = 1) \times \Pr(c = 1|a = 1) \\ &\quad \times \Pr(e = 1|c = 1) \times \Pr(f = 0|c = 1). \end{aligned}$$

Because in practice most annotations only draw upon a small fraction of terms, most terms and their descendants, will be negative and will not contribute to the calculation of a given joint probability. For this reason, although calculating the full joint probability has an asymptotic upper bound defined by the number of terms in the ontology, in practice it should have much lower complexity.

In this context, we are only interested in marginal probabilities that a protein is experimentally associated with a consistent subgraph T in the ontology. This probability can be expressed as

$$(3) \quad \Pr(T) = \prod_{v \in T} \Pr(v|\mathcal{P}(v)).$$

Equation (3) can be derived from the full joint factorization by first marginalizing over the leaves of the ontology and then moving towards the root(s) for all nodes not in T .

Although marginalizing over all negative terms gives a formulation that deviates from calculating the full joint probability it both simplifies calculating the information content of a sub-graph and can be philosophically justified. Unobserved annotations are only putative negative observations because explicit negative annotations are rarely deposited in biological databases. In ignoring all negative terms we are implicitly recognizing that they are simply unobserved variables and not actual negative annotations.

3.2. Calculating the information content of a graph. Now that we have properly defined the joint probability of a set of terms, or subgraph of the ontology, calculating the information content of those terms is relatively straightforward. The information content of a subgraph can be thought of as the number of bits of information one would receive about a protein if it were

annotated with that particular subgraph. We calculate the information content of a subgraph T in a straightforward manner as

$$i(T) = \log \frac{1}{\Pr(T)}$$

and use a base 2 logarithm as a matter of convention. The information content of a subgraph T can now be expressed by combining the previous two equations as

$$\begin{aligned} i(T) &= \sum_{v \in T} \log \frac{1}{\Pr(v|\mathcal{P}(v))} \\ &= \sum_{v \in T} ia(v), \end{aligned}$$

where, to simplify the notation, we use $ia(v)$ to represent the negative logarithm of $\Pr(v|\mathcal{P}(v))$. Term $ia(v)$ can be thought of as the increase, or accretion, of information obtained by adding a child term to a parent term, or set of parent terms, in an annotation. We will refer to $ia(v)$ as *information accretion* (perhaps information gain would be a better term, but because it is frequently used in other applications to describe an expected reduction in entropy, we avoid it in this situation).

A simple ontology containing five terms together with a conditional probability table associated with each node is shown in Figure 4.2A. Note that because of the graph consistency requirement, each conditional probability table is limited to a single number. For example, at node b in the graph, the probability $\Pr(b = 1|a = 1)$ is the only one necessary because $\Pr(b = 0|a = 1) = 1 - \Pr(b = 1|a = 1)$ and because $\Pr(b = 1|a = 0)$ is guaranteed to be 0. In Figure 4.2B we show a sample data set of four proteins functionally annotated according to the distribution defined by the Bayesian network. In Figure 4.2C, we show the total information content for each of the four annotation graphs.

3.3. Comparing two annotation graphs. We now consider a situation in which a protein's true and predicted function are represented by graphs T and P , respectively. We define two metrics that can be thought of as the information-theoretic analogs of recall and precision, and refer to them as remaining uncertainty and misinformation, respectively.

Definition 1. The *remaining uncertainty* about the protein's true annotation corresponds to the information about the protein that is not yet provided by the graph P . More formally, we express the remaining uncertainty (ru) as

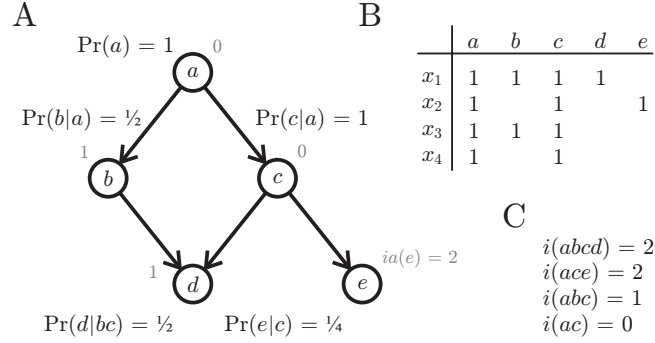


Figure 4.2. An example of an ontology, data set, and calculation of information content. (A) An ontology viewed as a Bayesian network together with a conditional probability table assigned to each node. Each conditional probability table is limited to a single number due to the consistency requirement in assignments of protein function. Information accretion calculated for each node, e.g. $ia(e) = -\log \Pr(e|c) = 2$, are shown in grey next to each node. (B) A data set containing four proteins whose functional annotations are generated according to the probability distribution from the Bayesian network. (C) The total information content associated with each protein found in panel B; e.g. $i(ace) = ia(a) + ia(c) + ia(e) = 2$. Note that $i(ab) = 1$ and $i(abcde) = 4$ although proteins with such annotation have not been observed in part B.

$$(4) \quad ru(T, P) = \sum_{v \in T \setminus P} ia(v),$$

which is simply the total information content of the nodes in the ontology that are contained in true annotation T but not in the predicted annotation P . Note that, in a slight abuse of notation, we apply set operations to graphs to manipulate only the vertices of these graphs.

Definition 2. The *misinformation* introduced by the classifier corresponds to the total information content of the nodes along incorrect paths in the prediction graph P . More formally, the misinformation is expressed as

$$(5) \quad mi(T, P) = \sum_{v \in P \setminus T} ia(v),$$

which quantifies how misleading a predicted annotation is.

Here, a perfect prediction (one that achieves $P = T$) leads to $ru(T, P) = mi(T, P) = 0$. However, both $ru(T, P)$ and $mi(T, P)$ can be infinite in the limit. In practice, though, $ru(T, P)$ is bounded by the information content of the particular annotation, while $mi(T, P)$ is only limited by the number of annotations a predictor chooses to return.

To illustrate calculation of remaining uncertainty and misinformation, in Figure 2 we show a sample ontology where the true annotation of a protein T is determined by the two leaf terms t_1 and t_2 , whereas the predicted subgraph P is determined by the leaf terms p_1 and p_2 . The remaining uncertainty $ru(T, P)$ and misinformation $mi(T, P)$ can now be calculated by adding the information accretion corresponding to the nodes circled in grey.

Finally, this framework can be used to define the similarity between the protein’s true annotation and the predicted annotation without relying on identifying an individual common ancestor between pairs of leaves (this node is usually referred to as the maximum informative common ancestor; Guzzi et al. 2012). The information content of the subgraph shared by T and P is one such possibility; i.e.

$$s(T, P) = \sum_{v \in T \cap P} ia(v).$$

3.4. Measuring the quality of function prediction. A typical predictor of protein function usually outputs scores that indicate the strength (e.g. posterior probabilities) of predictions for each term in the ontology. To address this situation, the concepts of remaining uncertainty and misinformation need to be considered as a function of a decision threshold τ . In such a scenario, predictions with scores greater than or equal to τ are considered positive predictions, while the remaining associations are considered negative (if the strength of a prediction is expressed via P-values or e-values, values lower than the threshold would indicate positive predictions). Regardless of the situation, every decision threshold results in a separate pair of values corresponding to the remaining uncertainty $ru(T, P(\tau))$ and misinformation $mi(T, P(\tau))$.

The remaining uncertainty and misinformation for a previously unseen protein can be calculated as expectations over the data generating probability distribution. Practically, this can be performed by averaging over the entire set of proteins used in evaluation, i.e.

$$(6) \quad ru(\tau) = \frac{1}{n} \sum_{i=1}^n ru(T_i, P_i(\tau))$$

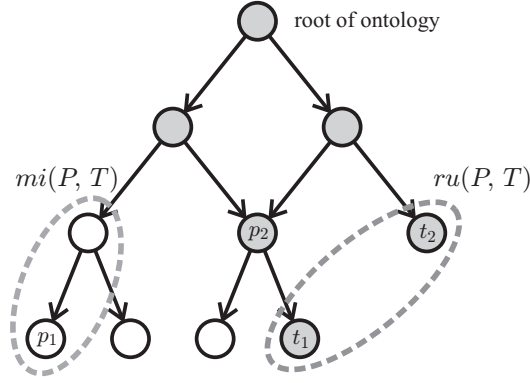


Figure 4.3. Illustration of calculating remaining uncertainty and misinformation given a predicted annotation graph P and a graph of true annotations T . Graphs P and T are uniquely determined by the leaf nodes p_1 , p_2 , t_1 , and t_2 , respectively. Nodes colored in grey represent graph T . Nodes circled in grey are used to determine remaining uncertainty (ru ; right side) and misinformation (mi ; left side) between T and P

and

$$(7) \quad mi(\tau) = \frac{1}{n} \sum_{i=1}^n mi(T_i, P_i(\tau))$$

where n is the number of proteins in the data set, T_i is the true set of terms for protein x_i , and $P_i(\tau)$ is the set of predicted terms for protein x_i given decision threshold τ . Note that once the set of terms with scores greater than or equal to τ is determined, the set $P_i(\tau)$ is composed of the unique union of the ancestors of all predicted terms. As the decision threshold is moved from its minimum to its maximum value, the pairs of $(ru(\tau), mi(\tau))$ will result in a curve in 2D space. We refer to such a curve using $(ru(\tau), mi(\tau))_\tau$. Removing the normalizing constant $(\frac{1}{n})$ from the above equations would result in the total remaining uncertainty and misinformation associated with a database of proteins and a set of predictions.

3.5. Weighted metrics. One disadvantage of definitions in Equations 6 and 7 is that an equal weight is given to proteins with low and high information content annotations when averaging. To

address this we assign a weight to each protein according to the information content of its experimental annotation. This formulation naturally downweights proteins with less informative annotations compared to proteins with rare, and therefore more informative (surprising), annotations. In biological data sets, frequently seen annotations have a tendency to be incomplete or shallow annotation graphs and arise due to the limitations or high-throughput nature of some experimental protocols. We define *weighted remaining uncertainty* as

$$(8) \quad wru(\tau) = \frac{\sum_{i=1}^n i(T_i) \cdot ru(T_i, P_i(\tau))}{\sum_{i=1}^n i(T_i)}$$

and *weighted misinformation* as

$$(9) \quad wmi(\tau) = \frac{\sum_{i=1}^n i(T_i) \cdot mi(T_i, P_i(\tau))}{\sum_{i=1}^n i(T_i)}$$

3.6. Semantic distance. Finally, to provide a single performance measure which can be used to rank and evaluate protein function prediction algorithms, we introduce *semantic distance* as the minimum distance from the origin to the curve $(ru(\tau), mi(\tau))_\tau$. More formally, the semantic distance S_k is defined as

$$(10) \quad S_k = \min_{\tau} (ru^k(\tau) + mi^k(\tau))^{\frac{1}{k}},$$

where k is a real number ≥ 1 . Setting $k = 2$ results in the minimum Euclidean distance from the origin. The preference for Euclidean distance ($k = 2$) over say Manhattan distance ($k = 1$) is to penalize unbalanced predictions with respect to the depth of predicted and experimental annotations.

3.7. Precision and recall. In order to contrast the semantic distance-based evaluation with more conventional performance measures, in this section we briefly introduce precision and recall for measuring functional similarity. As before, we consider a set of propagated experimental terms T and predicted terms $P(\tau)$, and define precision as the fraction of terms predicted correctly. More specifically,

$$pr(T, P(\tau)) = \frac{|T \cap P(\tau)|}{|P(\tau)|},$$

where $|\cdot|$ is the set cardinality operator. Only proteins for which the prediction set is non-empty can be used to calculate average precision. To address this issue the root term is counted as a prediction for all proteins. Similarly, recall is defined as the fraction of experimental (true) terms which were correctly predicted, i.e.

$$rc(T, P(\tau)) = \frac{|T \cap P(\tau)|}{|T|}.$$

As before, precision $pr(\tau)$ and recall $rc(\tau)$ for the entire data set are calculated as averages over the entire set of proteins (note that an alternative definition of precision and recall given by Verspoor et al. 2006 is described in Section 3.9). Finally, to provide a single evaluation measure we use the maximum F-measure over all decision thresholds. For a particular set of terms T and $P(\tau)$, F-measure is calculated as the harmonic mean of precision and recall. More formally, the final evaluation metric is calculated as

$$(11) \quad F_{\max} = \max_{\tau} \left\{ 2 \cdot \frac{pr(\tau) \cdot rc(\tau)}{pr(\tau) + rc(\tau)} \right\}$$

where $pr(\tau)$ and $rc(\tau)$ are calculated by averaging over the data set.

3.7.1. Information-theoretic weighted formulation. The definition of information accretion and the use of a probabilistic framework defined by the Bayesian network enables the straightforward application of information accretion to weight each term in the ontology. Therefore, it is easy to generalize the definitions of precision and recall from the previous section into a weighted formulation. Here, weighted precision and weighted recall can be expressed as

$$wpr(T, P(\tau)) = \frac{\sum_{v \in T \cap P(\tau)} ia(v)}{\sum_{v \in P(\tau)} ia(v)}$$

and

$$wrc(T, P(\tau)) = \frac{\sum_{v \in T \cap P(\tau)} ia(v)}{\sum_{v \in T} ia(v)}.$$

Weighted precision $wpr(\tau)$ and recall $wrc(\tau)$ can then be calculated as weighted averages over the database of proteins, as in Equations 8 and 9.

In addition to weighted precision and recall our framework also facilitates calculating weighted specificity

$$wsp(T, P(\tau)) = \frac{\sum_{v \in T^c \cap P^c(\tau)} ia(v)}{\sum_{v \in T^c} ia(v)}$$

where T^c represents the complement of set T ($G \setminus T$).

3.8. Supplementary evaluation metrics. When calculating remaining uncertainty, misinformation, precision and recall only consistent subgraphs of the Gene Ontology were considered. Under this framework, if a protein is annotated with multiple terms (either experimentally determined or predicted), as in Figure 4.2, we determine consistent graphs T (true) or P (predicted) by recursively propagating annotations towards the root(s) of the ontology and taking a union of all terms encountered along the way. In each of these measures, it is sufficient to only consider vertices (terms) in the annotation graphs and calculate the similarity measure by manipulating vertices in an additive fashion. For example, each vertex in T or P counts equally in the precision/recall-based evaluation while the information accretion is used to weight the vertices in the ru-mi-based evaluation.

A distinctly different approach can be taken by considering, on an individual basis, each leaf term that comprises a set T or P . This is the approach taken to calculate various information-theoretic metrics [Resnik, 1995, Jiang and Conrath, 1997, Lin, 1998, Lord et al., 2003, Schlicker et al., 2006] as well as to provide an alternative definition of precision and recall [Verspoor et al., 2006]. In this context the sets of leaf terms that define T and P (which we refer to as $\mathcal{L}(T)$ and $\mathcal{L}(P)$ respectively, and formally introduce below) are used to calculate a given metric. After calculating all pairwise metrics between the leaf terms, several different methods for averaging these scores can be applied to create a single similarity (or distance) value between T and P . We discuss these approaches below.

3.8.1. Basic definitions. Suppose we are given an ontology in the form of directed acyclic graph $G = (V, E)$, where V is a set of vertices and $E \subset V \times V$ is the set of edges. In this graph, given an edge $(u, v) \in E$, we refer to vertex u as a parent of v and, alternatively, to vertex v as a child of u . We also consider a set of all ancestors of v , $\mathcal{A}(v)$, and find this set by recursively identifying parents of all discovered nodes starting with v until the root(s) of the ontology is (are) reached. For mathematical convenience, we consider vertex v to be a member of $\mathcal{A}(v)$. Finally, given two vertices

u and v , we define a set of common ancestor nodes between these two vertices as $\mathcal{A}(u, v)$. Thus, $\mathcal{A}(u, v) = \mathcal{A}(u) \cap \mathcal{A}(v)$.

Consider now a consistent annotation graph T , where the set of vertices in T is a subset of vertices in G . We define $\mathcal{L}(T)$, or the set of leaf terms represented by T , as

$$(12) \quad \mathcal{L}(T) = \{u : u \in T \wedge \neg \exists ((u, v) \in E \wedge v \in T)\}.$$

In other words, $\mathcal{L}(T)$ contains only those vertices (terms) from T that do not have children in T . Thus, the leaf terms are defined with respect to a particular annotation graph T and generally differ from the leaf nodes in the ontology.

3.8.2. Information-theoretic metrics between pairs of vertices. When calculating the information-theoretic metrics of Resnik [1995], Jiang and Conrath [1997], Lin [1998], Lord et al. [2003], and Schlicker et al. [2006], we calculate the information content of an individual term $v \in V$ as

$$(13) \quad i(v) = \log \frac{1}{\text{Pr}(v)}$$

where $\text{Pr}(v)$ can be calculated as the relative frequency of term v among experimentally annotated proteins. The similarity between two distinct terms u and v as defined by Resnik [1995] was calculated as

$$(14) \quad s_R(u, v) = \max_{w \in \mathcal{A}(u, v)} \{i(w)\},$$

where $\mathcal{A}(u, v)$, as mentioned above, defines the set of common ancestors of terms u and v . Similarity as defined by Lin [1998] was calculated as

$$(15) \quad s(u, v) = \frac{s_R(u, v)}{i(u) + i(v)},$$

and as defined by Schlicker et al. [2006] as

$$(16) \quad s(u, v) = \frac{s_R(u, v)}{i(u) + i(v)} \cdot (1 - \min_{w \in \mathcal{A}(u, v)} \{\text{Pr}(w)\}).$$

Finally, the distance metric defined by Jiang and Conrath [1997] was calculated as

$$(17) \quad d(u, v) = i(u) + i(v) - 2 \cdot s_R(u, v).$$

3.8.3. *Information-theoretic metrics between pairs of graphs.* Since the above metrics are only defined for two distinct terms, it is necessary to provide a mechanism to utilize these metrics in instances where a protein is annotated with graphs containing multiple leaf terms. Given two non-empty consistent annotation graphs of true and predicted terms, T and P , and the set of leaf terms that define each set, $\mathcal{L}(T)$ and $\mathcal{L}(P)$, we employed two strategies for averaging. In the first case values were averaged between all possible pairs of terms in $\mathcal{L}(T)$ and $\mathcal{L}(P)$. Specifically, we calculated $s(T, P)$ as

$$(18) \quad s(T, P) = \frac{1}{|\mathcal{L}(T)| \cdot |\mathcal{L}(P)|} \sum_{t \in \mathcal{L}(T)} \sum_{p \in \mathcal{L}(P)} s(t, p).$$

We refer to this form of averaging as *all-pair* averaging. This method of averaging was applied by Lord et al. [2003] in calculating similarity between two functional annotations.

In the second case we calculated the similarity between the two sets as the average of the maximum similarity between a term from one set and all terms in the other. Specifically, we calculated $s(T, P)$ as

$$(19) \quad s(T, P) = \frac{1}{2|\mathcal{L}(T)|} \sum_{t \in \mathcal{L}(T)} \max_{p \in \mathcal{L}(P)} \{s(t, p)\} + \frac{1}{2|\mathcal{L}(P)|} \sum_{p \in \mathcal{L}(P)} \max_{t \in \mathcal{L}(T)} \{s(t, p)\}.$$

This measure represents the technique of averaging employed by Verspoor et al. [2006] when calculating precision and recall (originally referred to as hierarchical precision and recall). There, the authors separately calculate precision as

$$(20) \quad pr(T, P) = \frac{1}{|\mathcal{L}(P)|} \sum_{p \in \mathcal{L}(P)} \max_{t \in \mathcal{L}(T)} \frac{|\mathcal{A}(t, p)|}{|\mathcal{A}(p)|}$$

and recall as

$$(21) \quad rc(T, P) = \frac{1}{|\mathcal{L}(T)|} \sum_{t \in \mathcal{L}(T)} \max_{p \in \mathcal{L}(P)} \frac{|\mathcal{A}(t, p)|}{|\mathcal{A}(t)|}.$$

We refer to this method of averaging as *max-average*. Although not implemented here, Schlicker et al. [2006] employ a technique for averaging that is similar to Equation (19), but takes the maximum average similarity for one set as opposed to the average between the two. Specifically,

$$(22) \quad s(T, P) = \max \left\{ \frac{1}{|\mathcal{L}(T)|} \sum_{t \in \mathcal{L}(T)} \max_{p \in \mathcal{L}(P)} \{s(t, p)\}, \frac{1}{|\mathcal{L}(P)|} \sum_{p \in \mathcal{L}(P)} \max_{t \in \mathcal{L}(T)} \{s(t, p)\} \right\}$$

When averaging pairwise comparisons for distance metrics, the above averages are calculated as the average of minimum pairwise distances instead of maximum pairwise similarities.

3.9. Additional topological metrics. In addition to information-theoretic metrics, we also used Jaccard’s similarity coefficient [Jaccard, 1901] when calculating the similarity between the two consistent annotation graphs T and P . The Jaccard similarity coefficient is defined as

$$(23) \quad s(T, P) = \frac{|T \cap P|}{|T \cup P|}.$$

We note that cosine similarity as well as Maryland bridge coefficient [Glazko et al., 2005] usually result in values correlated with the Jaccard similarity coefficient. For that reason, these two similarity measures were not presented.

4. Confusion matrix interpretation of ru and mi

While we introduce remaining uncertainty and misinformation in the context of comparing sub graphs of a larger ontology these two terms can also be intuitively interpreted as the information theoretic analogs of false positives and false negatives or Type I and Type II errors. If we consider the confusion matrix in Table 4.2 we see the four positive outcomes that can occur when attempting to perform binary classification. False positives (Type I errors) are instances where a data point is a negative example but is incorrectly classified as a positive. In the context of hypothesis testing these are cases where the null hypothesis is true, but has incorrectly been rejected. False negatives (Type II errors) represent cases where the data point is a positive, but has incorrectly been labeled as a negative (false negatives). These are cases where the null hypothesis is false, but a model has failed to reject it.

Given a set of true terms, T , and predicted terms, P , then $P \setminus T$ would represent the terms that would fall in the false positive or Type I error portion of the confusion matrix. As defined in

Table 4.2. A sample confusion matrix showing the four potential outcomes when performing binary classification.

		True label	
		Positive	Negative
Predicted label	Positive	True positives	False positives
	Negative	False negatives	True negatives

Equation (5), this is the same set of terms whose joint probability, and subsequently information content, are measured when calculating misinformation. In this context, misinformation can be thought of as the number of bits of information, instead of traditional counts, the false positives represent. Conversely, the if we consider false negatives, or the set of terms defined by $T \setminus P$ we are referring to the same set of terms used to calculate remaining uncertainty in Equation (4).

5. Analysis of semantic distance

It is desirable to show that $S_k(T_1, T_2)$ is a metric. This is well known for $k = 1$ [Deza and Deza, 2013], but we wish to extend the proof to larger values of k .

In order to rove that $S_k(T_1, T_2)$ is a metric we first represent the terms associated with a set of sequences using a venn diagram (Figure 4.4) where p , q , s , and t each represent a mutually exclusive and collective exhaustive subsets of terms in T_1 , (where any of these sets may be the empty set) , and similarly define $T_2 = \{q, r, t, u\}$, we can rewrite Equation (10) as

$$S_k(T_1, T_2) = \left(\left[\sum_{t \in p \cup s} ia(t) \right]^k + \left[\sum_{t \in r \cup u} ia(t) \right]^k \right)^{1/k}$$

In an abuse of notation where we simply refer to the sum of ia values for terms in each set as the set itself, the above equation can be further simplified and written as

$$S_k(T_1, T_2) = \left([p + s]^k + [r + u]^k \right)^{1/k}$$

THEOREM 1. S_k between two sets of terms, T_1 and T_2 , is a metric for any real number $k \geq 1$.

PROOF.

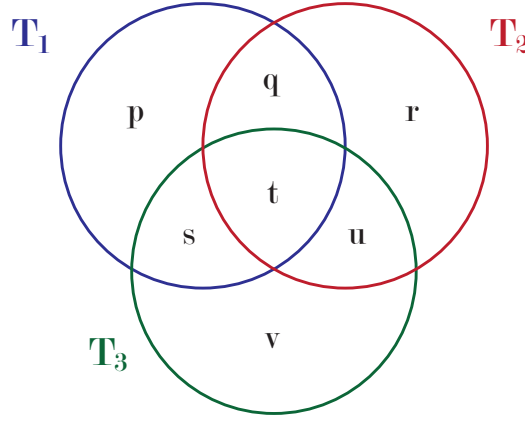


Figure 4.4. A Venn diagram showing the subsets of terms used to calculate the semantic distance between three pairs of annotations, T_1 , T_2 , and T_3 .

- *Non-negativity*

The non-negativity of $S_k(T_1, T_2)$ is ensured by the fact that the information accretion, $ia(t)$, of an individual term is non-negative. Because the conditional probability of a term given its parents, $\Pr(t|parents)$ is always ≥ 0 and ≤ 1 , by virtue its information accretion, $-\log(t|parents)$, is always non-negative.

- *Identity and indiscernibility*

$T_1 = T_2$ if and only if $S_k(T_1, T_2) = 0$. First, if $T_1 = T_2$, then $p, s, r, u = \emptyset$, and the resulting $ru(T_1, T_2)$, $mi(T_1, T_2)$, and $S_k(T_1, T_2)$ values will be zero. Second, if $S_k(T_1, T_2) = 0$ we know that $T_1 = T_2$. This is because the sum of information accretion values for any combination of terms will always be greater than zero, the only way that $ru(T_1, T_2)$ and $mi(T_1, T_2)$ can be equal to zero is when $p, s, r, u = \emptyset$. When these sets are empty, it must be the case that $T_1 = T_2$; and all terms in T_1 and T_2 are the same.

- *Symmetry*

$S_k(T_1, T_2) = S_k(T_2, T_1)$ because of the commutative property of addition, and the symmetric nature of ru and mi . For example, it can be shown that

$$ru(T_1, T_2) = mi(T_2, T_1),$$

because $(p + s) = (p + s)$.

- *Triangle inequality*

Here we prove that $S_k(T_1, T_2) + S_k(T_1, T_3) \geq S_k(T_2, T_3)$ by showing that

$$\left[(p+s)^k + (r+u)^k\right]^{\frac{1}{k}} + \left[(p+q)^k + (v+u)^k\right]^{\frac{1}{k}} \geq \left[(q+r)^k + (s+v)^k\right]^{\frac{1}{k}}$$

where all elements are non-negative. In order to prove this we rely on the Minkowski inequality, which can be expressed as

$$\left(\sum_{i=1}^n a_i^k\right)^{1/k} + \left(\sum_{i=1}^n b_i^k\right)^{1/k} \geq \left(\sum_{i=1}^n [a_i + b_i]^k\right)^{1/k},$$

where, for the sake of simplicity absolute value symbols have been omitted. We first modify the left-hand side, using the Mikowski inequality,

$$\left[(p+s)^k + (r+u)^k\right]^{\frac{1}{k}} + \left[(p+q)^k + (v+u)^k\right]^{\frac{1}{k}} \geq \left[(r+u+p+q)^k + (p+s+v+u)^k\right]^{\frac{1}{k}},$$

by setting the set $a = (r+u, p+s)$ and $b = (p+q, v+u)$.

Since we know that inequalities are transitive (i.e. if $a > b$ and $b > c$ then $a > c$), and it can be shown that

$$\left[(r+u+p+q)^k + (p+s+v+u)^k\right]^{\frac{1}{k}} \geq \left[(q+r)^k + (s+v)^k\right]^{\frac{1}{k}},$$

the original formula has been proven. □

6. Annotation models

We first collected all proteins with GO annotations supported by experimental evidence codes (EXP, IDA, IPI, IMP, IGI, IEP, TAS, IC) from the January 2011 version of the Swiss-Prot database (29,699 proteins in MFO; 31,608 in BPO; and 30,486 in CCO). We then generated three simple function annotation models: Naïve, BLAST and GOtcha, in order to assess the ability of performance metrics to accurately reflect the quality of a predicted set of annotations. In addition to these three methods, we generated another set of “predictions” by collecting experimental annotations for the same set of proteins from a database generated by the GO Consortium released at about the same time as our version of Swiss-Prot. This was done to quantify the variability of experimental annotation across different databases using the same set of metrics. In addition, this comparison can be used to estimate the empirical upper limit of prediction accuracy because the observed

performance is limited by the noise in experimental data. All computational methods were evaluated using 10-fold cross-validation.

6.1. The Naïve model. The Naïve model was designed to reflect biases in the distribution of terms in the data set and was the simplest annotation model we employed. It was generated by first calculating the relative frequency of each term in the training data set. This value was then used as the prediction score for every protein in the test set; thus, every protein in the test partition was assigned an identical set of predictions over all functional terms. The performance of the Naïve model reflects what one could expect when annotating a protein with no knowledge about that protein.

6.2. The BLAST model. The BLAST model was generated using local sequence identity scores to annotate proteins. Given a target protein sequence x , a particular functional term v in the ontology, and a set of sequences $S_v = \{s_1, s_2, \dots\}$ annotated with term v , we determine the BLAST predictor score for function v as

$$\max_{s \in S_v} \{sid(x, s)\},$$

where $sid(x, s)$ is the maximum sequence identity returned by the BLAST package [Altschul et al., 1997] when the two sequences are aligned. We chose this method to mimic the performance one would expect if they simply used BLAST to transfer annotations between similar sequences.

6.3. The GOfcha model. The third method, GOfcha [Martin et al., 2004], was selected to incorporate not only sequence identity between protein sequences, but also the structure of the ontology (technically, BLAST also incorporates structure of the ontology but in a relatively trivial manner). Specifically, given a target protein x , a particular functional term v , and a set of sequences $S_v = \{s_1, s_2, \dots\}$ annotated with function v , one first determines the r-score for function v as

$$r_v = c - \sum_{s \in S_v} \log(e(x, s)),$$

where $e(x, s)$ represents the E-value of the alignment between the target sequence x and sequence s , and $c = 2$ is a constant added to the given quantity to ensure all scores were above 0. Given the r-score for function v , i-scores were then calculated by dividing the r-score of each function by the

score for the root term $i_v = r_v/r_{root}$. As such, GOTcha is an inexpensive and robust predictor of function.

7. Experiments and results

In this section we first analyze the average information content in a data set of experimentally annotated proteins and then evaluate performance accuracy of different function prediction methods using both topological and probabilistic metrics. Each experiment was conducted on all three categories of the Gene Ontology: Molecular Function (MFO), Biological Process (BPO), and Cellular Component (CCO) ontologies. In order to avoid cases where the information content of a term is infinite a pseudo-count of 1 was added to each term, and the total number of proteins in the data set was incremented when calculating term frequencies.

7.1. Average information content of a protein. We first examined the distribution of the information content per protein for each of the three ontologies (Figure 4.5). We observe a wide range of information contents in all ontologies, reaching over 128 bits in case of BPO (which corresponds to a factor of 128 in the probability of observing particular annotation graphs). The distributions for MFO and CCO show unusual peaks for very low information contents, suggesting that a large fraction of annotation graphs in these ontologies are low quality. One such anomaly is created by the term 'binding' in MFO that is associated with 72% of proteins. Furthermore, 41% of proteins are annotated with its child 'protein binding' as a leaf term, and 26% are annotated with it as their sole leaf term. Such annotations, which are clearly a consequence of high-throughput experiments, present a significant difficulty in method evaluation.

Previously, we showed that the distribution of leaf terms in protein annotation graphs exhibits scale-free tendencies [Clark and Radivojac, 2011]. Here, we also analyzed the average number of leaf terms per protein and compared it with the information content of that protein. We estimate the average number of leaf terms to be 1.6 (std. 1.0), 3.0 (std. 3.6), and 1.6 (std. 1.0) for MFO, BPO, and CCO, respectively, and calculate Pearson correlation between the information content and the number of leaf terms for a protein (0.80, 0.92, and 0.71). Such high level of correlation suggests that proteins annotated with a small number of leaf terms are generally annotated by shallow graphs. This is particularly evident in the case of 'protein binding' annotations that can be derived from yeast-2-hybrid experiments, but provide little insight into the functional aspects of these complexes

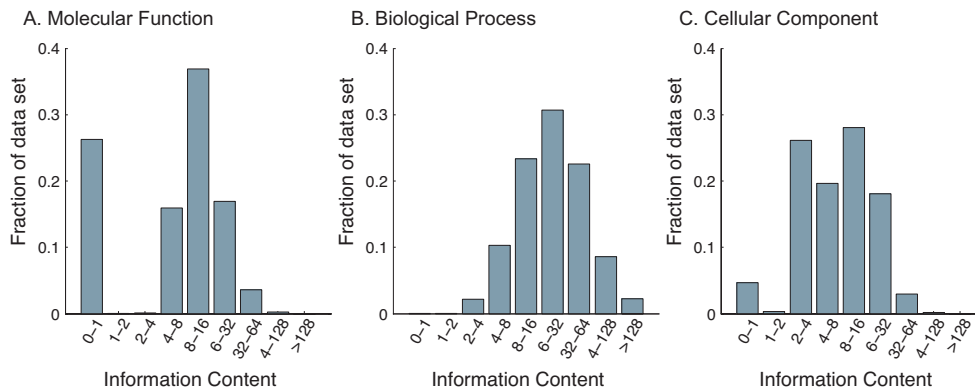


Figure 4.5. Distribution of information content (in bits) over proteins annotated by terms for each of the three ontologies. The average information content of a protein was estimated at 10.9 (std. 10.2), 32.0 (std. 33.6), and 10.4 (std. 9.2) bits for MFO, BPO, and CCO, respectively.

when only viewed as GO annotations. We believe the wide range of information contents coupled with the fact that a large fraction of proteins were essentially uninformative, justifies the weighting proposed in this work.

7.2. Two-dimensional plots. In order to assess how each metric evaluated the performance of the four prediction methods we generated two-dimensional plots. Figure 4.6 shows the performance of each predictor using precision/recall and ru-mi curves, as well as their weighted variants. The performance of the GO/Swiss-Prot annotation is represented as a single point because it compares two databases of experimental annotations where predictions are all binary and do not have associated scores.

When looking at the precision/recall curves, we first observe an unusually high area under the curve associated with the Naive model. This is a result of a significant fraction of low information content annotations that are relatively easy to predict by simply using prior probabilities of terms as prediction values. In addition, these biases lead to a biologically unexpected result where the predictor based on the BLAST algorithm performs on par with the Naive model, e.g. $F_{\max}(\text{BLAST}, \text{MFO}) = 0.65$ and $F_{\max}(\text{Naive}, \text{MFO}) = 0.60$, while $F_{\max}(\text{BLAST}, \text{CCO}) = 0.63$; $F_{\max}(\text{Naive}, \text{CCO}) = 0.64$. The largest difference between the BLAST and Naive models was observed for BPO, which has a Gaussian-like distribution of information contents in the logarithmic scale (Figure 4.5).

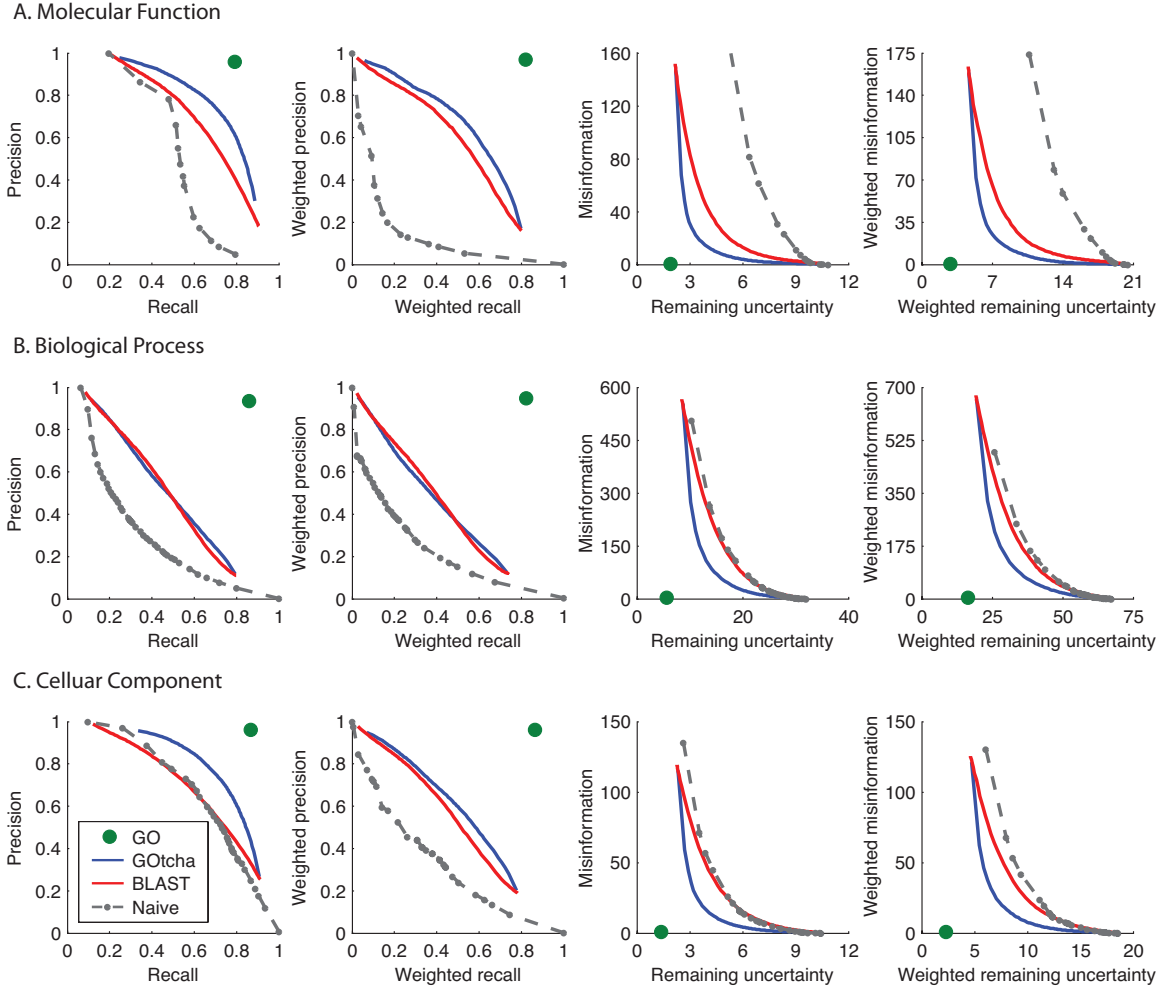


Figure 4.6. Two-dimensional evaluation plots. Each plot shows three prediction methods: Naive (grey, dashed), BLAST (red, solid), and GOTcha (blue, solid) constructed using cross-validation. Green point labeled GO shows the performance evaluation between two databases of experimental annotations, downloaded at the same time. The rows show the performance for different ontologies (MFO, BPO, CCO). The columns show different evaluation metrics: $(pr(\tau), rc(\tau))_{\tau}$, $(wpr(\tau), wrc(\tau))_{\tau}$, $(ru(\tau), mi(\tau))_{\tau}$, and $(wru(\tau), wmi(\tau))_{\tau}$.

The second column of plots in Figure 4.6 shows the weighted precision/recall curves. Here, we observe large changes in the performance accuracy, especially for the Naive model, in MFO and

CCO categories, whereas the BPO category was, for the most part, not impacted. We believe that the information-theoretic weighting of precision and recall resulted in more meaningful evaluation.

In Figure 4.7, we present more detailed results related to the ru-mi curves presented in Figure 4.6. In the top row, we show the same ru-mi curves as in the third column of Figure 4.6, with yellow dots providing information where the maximum values of semantic distance S_2 were reached for each method. Interestingly, because the predictors generally associate scores to all nodes in the Gene Ontology, the amount of over-prediction can be very large for low decision thresholds, which consequently results in large values of misinformation. To provide better insight into the balance between remaining uncertainty and misinformation achieved by the semantic distance S_2 , in the bottom row we present the same curves for small values of misinformation only.

The information-theoretic measures are shown in the last two columns of Figure 4.6 and in Figure 4.7. One useful property of ru-mi plots is that they explicitly illustrate how many bits of information are yet to be revealed about a protein (on average) as a function of misinformation that is introduced by over-prediction or misannotation. In all three categories, the amount of misinformation being introduced increases rapidly; quickly obtaining a rate that is twice the amount of expected information for an average protein. We believe these plots shed new light into how much information overload a researcher can be presented with by drawing predictions at a particular threshold. Looking from right to left in each plot, we observe an elbow in each of the curves (at about 3 bits for MFO and CCO and 12 bits for BPO; Figure 4.6) after which the remaining uncertainty barely decreases, while misinformation grows out of control.

Figure 4.8 and Figure 4.9 show results when implementing the supplemental information theoretic metrics using all-pair and max-average methods of averaging, respectively. It is important to mention that in a direct application of Resnik’s similarity function, we refer to it as *Lord* in Figure 4.8 (all-pair averaging) and as *Resnik* in Figure 4.9 (max-average method of averaging). This is because, to the best of our knowledge, in the context of comparing functional annotations of proteins the all-pair averaging was first proposed by Lord et al. [2003].

In Figure 4.10 we contrast two different types of precision-recall curves. In the top row, we present the same results as in Figure 4.5 of the main manuscript. In the bottom row, we use the max-averaging method for calculating precision-recall outlined by Verspoor et al. [2006]. The methods provided similar results although the max-average formulation had generally larger values

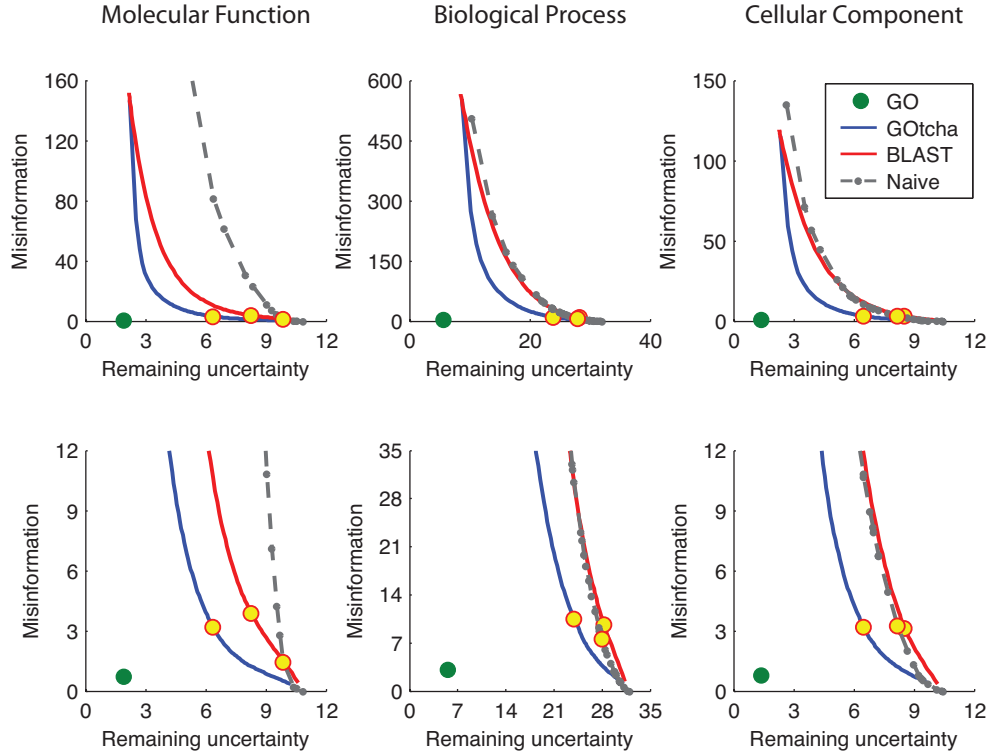
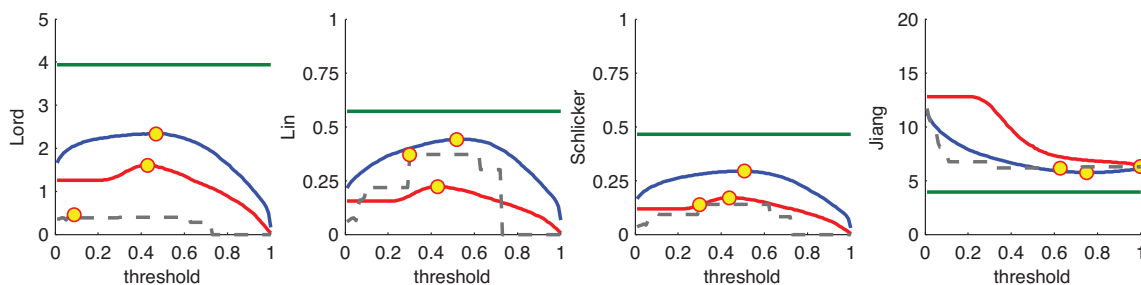


Figure 4.7. Figures showing the remaining uncertainty and misinformation of baseline methods with yellow dots denoting values at which each method obtains its S_2 value. For better interpretation of the values at which each method achieves its S_2 value the bottom row of figures show the same plots as the top row, but with adjusted y-axis limits.

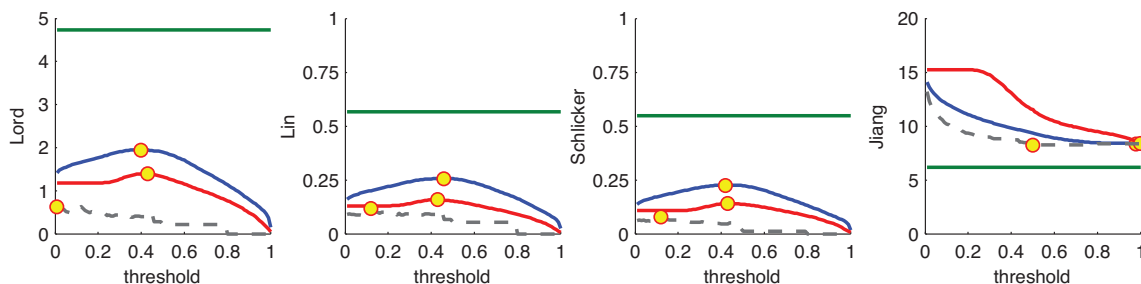
of $F_{\max}$ than the standard formulation of precision and recall (as defined in the main manuscript). However, these larger values of $F_{\max}$ occurred at lower decision thresholds.

7.3. Comparisons of single statistics. Here we analyze the ability of single measures to rank predictors and lead to useful evaluation insights. We compare the performance of semantic distance to several other methods that calculate either topological or semantic similarities. For each evaluation method, the decision threshold was varied for each of the prediction methods and the threshold providing the best performance was selected as optimal. We then analyze and discuss the performance of these metrics at those optimal thresholds.

A. Molecular Function



B. Biological Process



C. Cellular Component

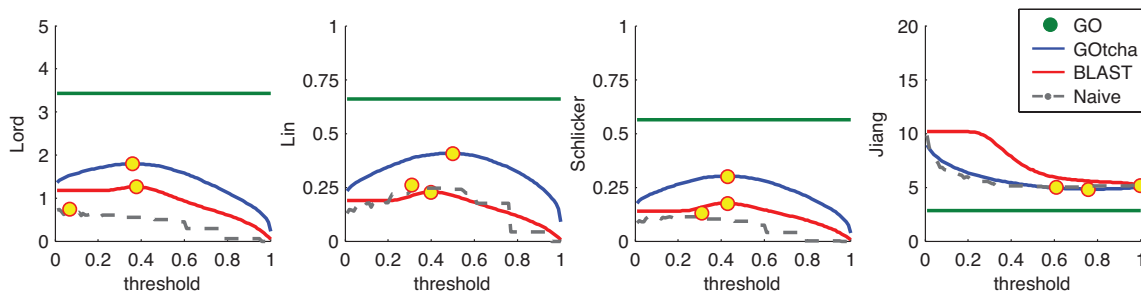
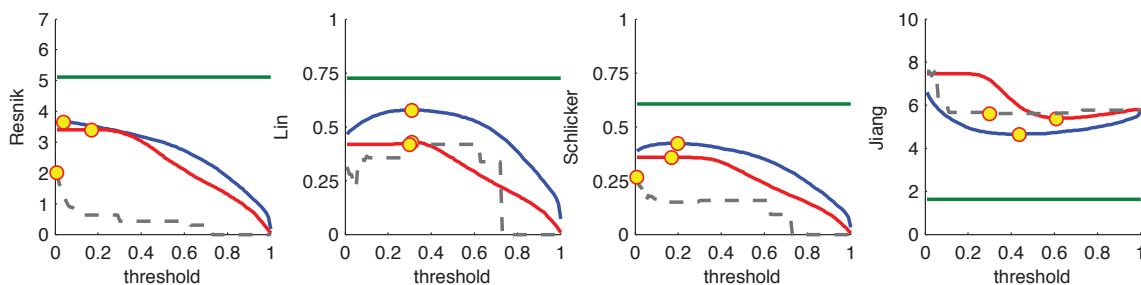


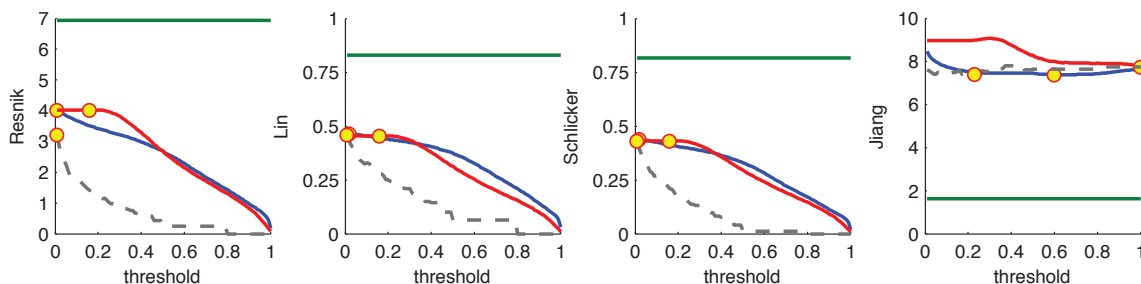
Figure 4.8. Two-dimensional evaluation plots of information content-based metric performances when using the all-pair method of averaging. Yellow dots denote the maximum similarity or, in the case of Jiang and Conrath [1997], the minimum distance, that each method obtains.

We implemented the semantic similarity metrics of Resnik [1995], Jiang and Conrath [1997], Lin [1998], and Schlicker et al. [2006], as detailed in Section 3.8. Because each of these measures is only defined for a single pair of terms in the ontology, scores between two protein annotation graphs (true graph T vs. predicted graph P) we implemented both all pair averaging (Tables 4.3 and 4.4) and max-averaging (Table 4.5) for these metrics as detailed in Section 3.8.3. We note that the all-pair averaging using Resnik’s term similarity was implemented by Lord et al. [2003]

A. Molecular Function



B. Biological Process



C. Cellular Component

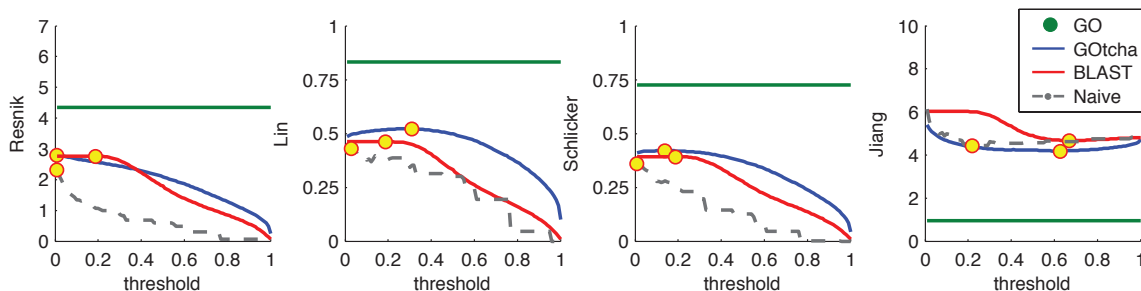
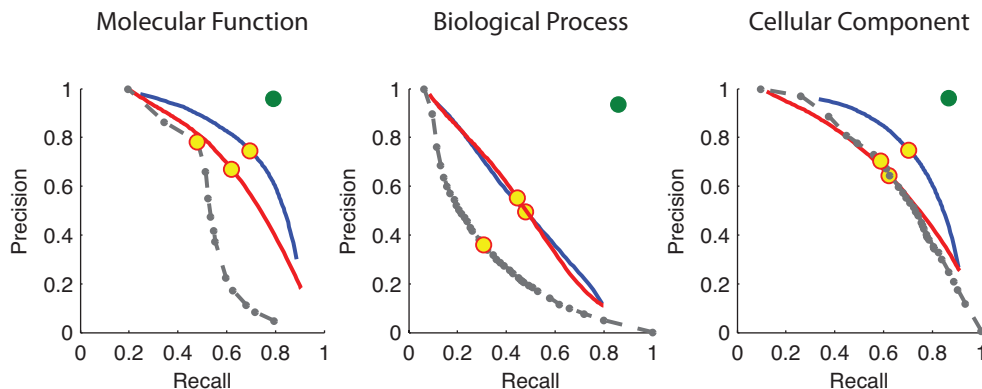


Figure 4.9. Two-dimensional evaluation plots of information content-based metric performances when using the max-average method of averaging. Yellow dots denote the maximum similarity or, in the case of Jiang and Conrath [1997], the minimum distance, that each method obtains.

in the context of GO annotations. In addition to these semantic measures, we also implemented the Jaccard similarity coefficient between the sets of vertices in the two annotation graphs (Section 3.9). For precision/recall curves and ru-mi curves, we used $F_{\max}$ and S_2 measures as single values to obtain optimal thresholds.

Tables 4.3, 4.3, and 4.5 show the maximum similarity, or minimum distance in the case of Jiang and Conrath’s and semantic distance, that each metric obtained for each of our classification models.

A. Precision/Recall



B. Precision/Recall (max-average)

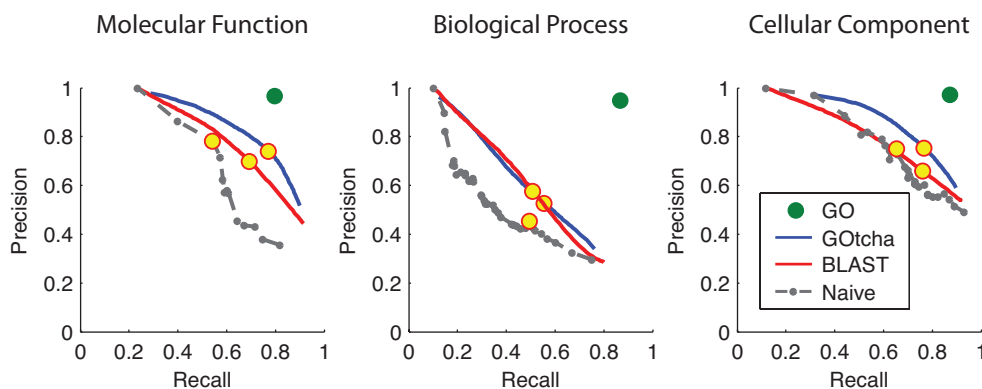


Figure 4.10. Plots showing results using the standard method of calculating precision and recall (top row) compared to using the max-average method of calculating precision and recall as detailed by Verspoor et al. [2006] (bottom row). Yellow dots denote the values of precision and recall at which each method obtains its $F_{\max}$ value.

In addition to reporting the maximum similarity we also report the decision threshold at which that value was obtained along with the associated level of remaining uncertainty and misinformation at that threshold.

Considering Tables 4.3 and 4.4 the first interesting observation is that all metrics, aside from that of Jiang and Conrath, obtain optimal thresholds that result in relatively similar levels of remaining uncertainty and misinformation for the GOTcha model. However, all metrics, aside from semantic distance and Jiang and Conrath's distance seem to favor extremely high levels of misinformation at the reported decision thresholds for the BLAST model. For MFO and CCO the semantic similarity

measures of Lord *et al.*, Lin and Schlicker *et al.* report misinformation levels that are more than twice the information content of the average protein in that ontology for the BLAST model. In BPO those are even more extreme. We believe this is a direct consequence of the pairwise term averaging applied in these methods.

In Table 4.5, we provide results analogous to those from Table 4.3, but using max-average method instead of all-pair averaging. We generally observe similar trends as before, but note that the values of BLAST thresholds used for functional transfer are even lower than when all-pair averaging was used (except for the measure by Jiang and Conrath 1997, where max-average method seems to be beneficial).

It is particularly interesting to analyze the optimal thresholds obtained for the BLAST model. These thresholds can be interpreted as the level of sequence identity above which each metric reports functional transfer can be made. For example, because their optimal BLAST thresholds are relatively low, the levels of misinformation provided by the similarities of Lord *et al.*, Lin, and Schlicker *et al.* are rather large in both tables. $F_{\max}$ and Jaccard approaches also report low threshold values for all ontologies, while Jiang and Conrath’s distance selects the optimal threshold at an overly restrictive 100% sequence identity. We believe that the semantic distance S_2 provides more reasonable values for functional transfer, finding an optimal distance at 77%, 88% and 78% for MFO, BPO, and CCO, respectively.

We believe that, generally, all-pair averaging provides better results regarding functional similarity than does max-average. This is predominantly based on the even lower optimal thresholds obtained for the BLAST model in Table 4.5.

8. Discussion

Here we introduce an information-theoretic framework for evaluating the performance of computational protein function prediction. We frame protein function prediction as a structured-output learning problem in which the output space is represented by consistent subgraphs of the GO graph. We argue that our approach directly addresses evaluation in cases where there are multiple true and predicted (leaf) terms associated with a protein by taking the structure of the ontology and the dependencies between terms induced by a hierarchical ontology into account. Our method also

Table 4.3. Performance evaluation of common information-theoretic metrics. For each measure, the decision threshold was varied across the entire range of predictions to obtain the maximum or minimum value (shown in column 1). The threshold at which each method reached the best value is shown in column 2. Columns 3 and 4 show the remaining uncertainty (*ru*) and misinformation (*mi*) calculated according to the Bayesian network. Each semantic similarity metric was calculated according to the relative frequencies of observing each term in the database.

Lord	Molecular Function				Biological Process				Cellular Component			
	Max	Threshold	ru	mi	Max	Threshold	ru	mi	Max	Threshold	ru	mi
GOtcha	2.34	0.47	6.34	3.20	1.95	0.40	23.36	11.90	1.80	0.36	5.88	4.58
BLAST	1.61	0.43	4.69	27.90	1.40	0.43	16.73	139.57	1.27	0.38	4.42	37.24
Naive	0.46	0.09	9.56	4.23	0.63	0.01	10.35	504.88	0.75	0.07	5.81	16.34
Lin	Max	Threshold	ru	mi	Max	Threshold	ru	mi	Max	Threshold	ru	mi
	Max	Threshold	ru	mi	Max	Threshold	ru	mi	Max	Threshold	ru	mi
GOtcha	0.44	0.52	6.67	2.67	0.26	0.46	24.43	9.40	0.41	0.50	6.71	2.76
BLAST	0.22	0.43	4.69	27.90	0.16	0.43	16.73	139.57	0.23	0.40	4.78	30.45
Naive	0.37	0.30	10.39	0.21	0.12	0.12	24.92	23.14	0.26	0.31	8.98	1.32
Schlicker	Max	Threshold	ru	mi	Max	Threshold	ru	mi	Max	Threshold	ru	mi
	Max	Threshold	ru	mi	Max	Threshold	ru	mi	Max	Threshold	ru	mi
GOtcha	0.29	0.51	6.60	2.76	0.23	0.42	23.73	10.99	0.30	0.43	6.31	3.56
BLAST	0.17	0.44	4.83	25.39	0.14	0.43	16.73	139.57	0.18	0.43	5.26	23.26
Naive	0.14	0.30	10.39	0.21	0.08	0.12	24.92	23.14	0.13	0.31	8.98	1.32
Jiang	Min	Threshold	ru	mi	Min	Threshold	ru	mi	Min	Threshold	ru	mi
	Min	Threshold	ru	mi	Min	Threshold	ru	mi	Min	Threshold	ru	mi
GOtcha	5.74	0.75	8.20	1.27	8.38	0.98	30.88	1.22	4.83	0.76	8.21	1.19
BLAST	6.34	1.00	10.62	0.43	8.39	1.00	31.31	1.40	5.20	1.00	10.16	0.35
Naive	6.19	0.63	10.53	0.13	8.24	0.50	31.75	0.07	5.01	0.61	10.13	0.08

facilitates accounting for the high level of biased and incomplete experimental annotations of proteins by allowing for the weighting of proteins based on the information content of their annotations. Because we maintain an information-theoretic foundation, our approach is relatively immune to the potential dissociation between the depth of a term and its information content; a weakness of often-used topological metrics in this domain such as precision/recall or ROC based evaluation. At the

Table 4.4. Performance evaluation of topological metrics and S_2 . For each measure, the decision threshold was varied across the entire range of predictions to obtain the maximum or minimum value (shown in column 1). The threshold at which each method reached the best value is shown in column 2. Columns 3 and 4 show the remaining uncertainty (ru) and misinformation (mi) calculated according to the Bayesian network. Each semantic similarity metric was calculated according to the relative frequencies of observing each term in the database.

Jaccard	Molecular Function				Biological Process				Cellular Component			
	Max	Threshold	ru	mi	Max	Threshold	ru	mi	Max	Threshold	ru	mi
GOtcha	0.57	0.46	6.29	3.32	0.31	0.34	22.24	15.24	0.56	0.43	6.31	3.56
BLAST	0.37	0.50	5.74	14.72	0.19	0.50	19.68	76.98	0.34	0.43	5.26	23.26
Naive	0.46	0.30	10.39	0.21	0.17	0.19	27.53	9.22	0.47	0.31	8.98	1.32

F_{max}	Molecular Function				Biological Process				Cellular Component			
	Max	Threshold	ru	mi	Max	Threshold	ru	mi	Max	Threshold	ru	mi
GOtcha	0.72	0.43	6.12	3.68	0.49	0.32	21.84	16.69	0.73	0.43	6.31	3.56
BLAST	0.64	0.48	5.42	17.89	0.49	0.50	19.68	76.98	0.63	0.45	5.57	19.42
Naive	0.60	0.29	9.87	1.44	0.33	0.19	27.53	9.22	0.64	0.33	9.22	0.80

S₂	Molecular Function				Biological Process				Cellular Component			
	Min	Threshold	ru	mi	Min	Threshold	ru	mi	Min	Threshold	ru	mi
GOtcha	7.11	0.47	6.34	3.20	26.14	0.43	23.91	10.56	7.23	0.46	6.48	3.21
BLAST	9.13	0.77	8.25	3.90	29.89	0.88	28.28	9.69	9.08	0.78	8.51	3.15
Naive	9.98	0.10	9.72	2.80	29.00	0.22	27.67	8.72	8.79	0.21	7.71	4.95

same time, because we take a holistic approach to considering a protein’s potentially large set of true or predicted functional associations we resolve many of the problems introduced by the practice of aggregating multiple pairwise similarity comparisons common to existing semantic similarity measures.

Although there is a long history [Resnik, 1999] and a significant body of work in the literature regarding the use of semantic similarity measures [Pesquita et al., 2009, Guzzi et al., 2012], to the best of our knowledge all such metrics are based on single statistics and are unable to provide insight into the levels of remaining uncertainty and misinformation that every predictor is expected

to balance. Therefore, the methods proposed in this work extend, modify, and formalize several useful information-theoretic metrics introduced over the past decades. In addition, both remaining uncertainty and misinformation have natural information-theoretic interpretations and can provide meaningful information to the users of computational tools. At the same time, the semantic distance based on these concepts facilitates not only the use of a single performance measure to evaluate and rank predictors, but can also be exploited as a loss function during training.

One limitation of the proposed approach is grounded in the assumption that a Bayesian network, structured according to the underlying ontology, will perfectly model the prior probability distribution of a target variable. An interesting anomaly with this approach is that the marginal probability, and subsequently the information content, of a single term (i.e. consistent graph with a single leaf term) calculated by considering the conditional probabilities between child and parent terms in the graph does not necessarily match the relative term frequency in the database. *Ad hoc* solutions that maintain the term information content are possible but would result in sacrificed interpretability of the metric itself. One such solution can be obtained via a recursive definition

$$ia(v) = i(v) - \sum_{u \in \mathcal{P}(v)} ia(u)$$

where $i(v)$ is estimated directly from the database and $ia(root) = 0$.

Finally, rationalizing between evaluation metrics is a difficult task. The literature presents several strategies where protein sequence similarity, protein-protein interactions or other data are used to assess whether a performance metric behaves according to expectations [Guzzi et al., 2012]. In this work, we took a somewhat different approach and showed that the demonstrably biased protein function data can be shown to provide surprising results with well-understood prediction algorithms and conventional evaluation metrics. Thus, we believe that our experiments provide evidence of the usefulness of the new evaluation metric.

Table 4.5. Performance of information-theoretic methods when calculating performance as the average of maximum similarity (or distance) between each true and predicted term as described in Section 3.8.3. $F_{\max}$ values were calculated using precision and recall as detailed by Verspoor et al. [2006]. The decision threshold was varied across the entire range of predictions to obtain the maximum or minimum value (shown in column 1) for each method. The threshold at which each method reached the best value is shown in column 2. Columns 3 and 4 show the remaining uncertainty and misinformation calculated according to the Bayesian network.

Resnik	Molecular Function				Biological Process				Cellular Component			
	Max	Threshold	ru	mi	Max	Threshold	ru	mi	Max	Threshold	ru	mi
GOtcha	3.65	0.04	2.96	32.17	4.02	0.01	10.19	273.90	2.80	0.01	2.70	59.02
BLAST	3.41	0.17	2.17	151.77	4.03	0.16	8.48	566.88	2.77	0.19	2.26	119.34
Naive	2.02	0.01	5.07	177.91	3.22	0.01	10.35	504.88	2.32	0.01	2.65	135.04

Lin	Max	Threshold	ru	mi	Max	Threshold	ru	mi	Max	Threshold	ru	mi
	Max	Threshold	ru	mi	Max	Threshold	ru	mi	Max	Threshold	ru	mi
GOtcha	0.58	0.31	5.32	5.86	0.46	0.02	11.10	192.89	0.52	0.31	5.55	5.58
BLAST	0.43	0.31	2.90	90.72	0.45	0.16	8.48	566.88	0.46	0.19	2.26	119.34
Naive	0.42	0.30	10.39	0.21	0.46	0.01	10.35	504.88	0.43	0.03	3.90	56.83

Jiang	Min	Threshold	ru	mi	Min	Threshold	ru	mi	Min	Threshold	ru	mi
	Min	Threshold	ru	mi	Min	Threshold	ru	mi	Min	Threshold	ru	mi
GOtcha	4.65	0.44	6.18	3.56	7.39	0.60	26.65	5.30	4.19	0.63	7.45	1.80
BLAST	5.38	0.61	6.97	7.58	7.75	1.00	31.31	1.40	4.67	0.67	7.84	5.13
Naive	5.61	0.30	10.39	0.21	7.41	0.23	27.73	8.49	4.45	0.22	8.17	3.25

Schlicker	Max	Threshold	ru	mi	Max	Threshold	ru	mi	Max	Threshold	ru	mi
	Max	Threshold	ru	mi	Max	Threshold	ru	mi	Max	Threshold	ru	mi
GOtcha	0.42	0.20	4.56	9.32	0.44	0.02	11.10	192.89	0.42	0.14	4.29	12.76
BLAST	0.36	0.17	2.17	151.77	0.43	0.16	8.48	566.88	0.39	0.19	2.26	119.34
Naive	0.27	0.01	5.07	177.91	0.43	0.01	10.35	504.88	0.36	0.01	2.65	135.04

$F_{\max}$	Max	Threshold	ru	mi	Max	Threshold	ru	mi	Max	Threshold	ru	mi
	Max	Threshold	ru	mi	Max	Threshold	ru	mi	Max	Threshold	ru	mi
GOtcha	0.76	0.33	5.45	5.41	0.54	0.23	19.79	26.04	0.76	0.31	5.55	5.58
BLAST	0.70	0.44	4.83	25.39	0.54	0.46	18.06	108.73	0.71	0.37	4.23	41.17
Naive	0.64	0.29	9.87	1.44	0.47	0.07	22.06	51.55	0.70	0.20	7.23	6.75

Testing the ortholog conjecture with comparative functional genomic data

The potential for gene duplication to generate evolutionary novelty was first noted in 1918 by Calvin Bridges (cited in [Bridges, 1935]), and the idea quickly found many supporters [Bridges, 1936, Muller, 1936, Stephens, 1951]. The advent of protein-sequencing technologies in the 1950s and ‘60s confirmed the presence of many gene duplicates, and once again researchers championed the importance of duplication in evolution [Ohno, 1970]. Today, the sequencing of hundreds of whole genomes has revealed the ubiquity of gene duplicates in all domains of life, and a growing number of empirical and computational studies have provided direct evidence for the role of gene duplication in adaptation [Hahn, 2009].

As the first protein-sequence data became available, Zuckerkandl and Pauling [1965] made the distinction between “duplication-independent homology” and “duplication-dependent homology,” what we now refer to as orthology and paralogy, respectively [Fitch, 1970, 2000]. They recognized that the paralogous α -, β -, and γ -hemoglobin chains present in all jawed vertebrates were less functionally similar to each other than were orthologous copies between closely related species, largely because they had been diverged for a very long period of time. Despite the fact that this and a small handful of other examples were confined to cases with very deep divergences between paralogs, the idea that orthologs were more similar in function than paralogs continued to be a basic tenet of comparative studies. As the first large genome sequencing projects were completed and thousands of previously unknown genes had to be annotated, this idea reappeared in the seminal papers of the field now known as phylogenomics: “Normally, orthologs retain the same function in the course of evolution, whereas paralogs evolve new functions, even if related to the original one. Thus, identification of orthologs is critical for reliable prediction of gene functions in newly sequenced genomes” [Tatusov et al., 1997]. Similar statements can be found in many papers (e.g. [Chen and Jeong, 2000, Dolinski and Botstein, 2007, Eisen, 1998, Hulsen et al., 2006, Koonin, 2005, Lee et al.,

2007, Zmasek and Eddy, 2002, Engelhardt et al., 2005]), and, as pointed out by Studer et al. [2008] can even be found in the primer on phylogenetics at the National Center for Biotechnology Information (NCBI) website (<http://www.ncbi.nlm.nih.gov/About/primer/phylo.html>).

We refer to the hypothesis that orthologs are more likely to be functionally similar than are paralogs as the “ortholog conjecture” (cf. [Koonin, 2005]). In fact, only rarely has it even been noted that this idea is a hypothesis about functional similarity [Koonin, 2005, Studer and Robinson-Rechavi, 2009] in most studies it is either assumed to be true or is supported by evidence from a small number of genes. It is certainly the case that increased rates of sequence evolution often follow gene duplication [Goodman et al., 1975, Kimura, 1981, Li and Gojobori, 1983, Lynch and Conery, 2000], but rarely are these changes connected to functional differences (e.g. [Des Marais and Rausher, 2008]). Moreover, one of the three major hypotheses for the maintenance of gene duplicates (subfunctionalization) does not require any functional change, and another (gene dosage) even prohibits such changes from occurring [Hahn, 2009]. There have been studies comparing rates of adaptive evolution in duplicates versus single-copy genes, but these have provided conflicting results [Han et al., 2009, Studer et al., 2008]; rates of adaptive evolution may also be poor predictors of overall functional similarity [Tennessen, 2008]. We do not know of any study that has systematically tested the ortholog conjecture.

A large number of methods have been developed to identify orthologous relationships among proteins. These methods range from simple pairwise comparisons, to standard phylogenetic tree-building, to probabilistic assignment using Bayesian analyses [Gabaldon, 2008, Kuzniar et al., 2008, Sennblad and Lagergren, 2009]. Several databases provide predicted orthologs [Alexeyenko et al., 2006, Datta et al., 2009, Vilella et al., 2009], and whole scientific meetings are devoted to their study [Gabaldon et al., 2009]. While the identification of orthologs is certainly highly relevant to many evolutionary questions, especially in systematics, many of these methods are explicitly made for functional inference. Note also that in most cases these methods are only distinguishing between orthologs and outparalogs [Sonnhammer and Koonin, 2002]: that is, between an ortholog and a paralog that duplicated before the speciation event separating the orthologs, and that is therefore almost always more diverged at the sequence level (Figure 5.1). Paralogs more closely related to each other than either is to an ortholog (“inparalogs”; Figure 5.1) are by definition co-orthologous

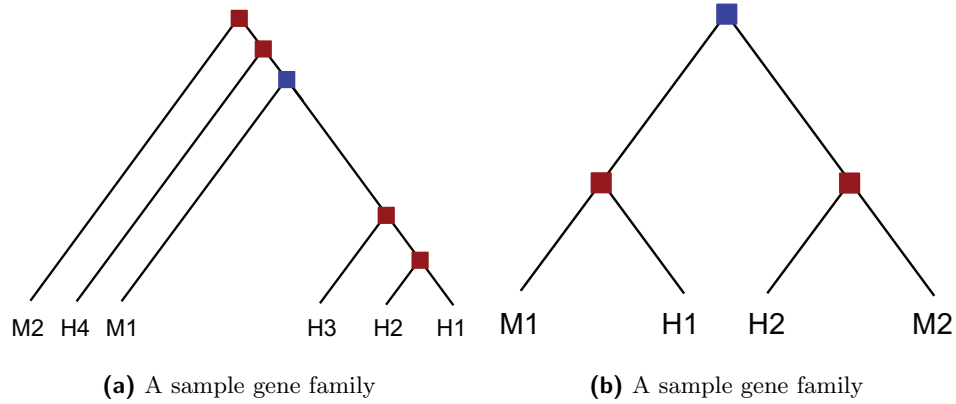


Figure 5.1. Different types of homology relationships among genes. Duplication nodes are shown in red and speciation nodes are shown in blue. Figure 5.1a) The figure shows four hypothetical genes in humans (H1-H4) and two in mouse (M1-M2). There are four types of homologs shown: 1) M1 is an ortholog of H1, H2, and H3 because their last common ancestor is a speciation event (one-to-many orthology). 2) H1 is an inparalog of H2 and H3, with respect to the human-mouse split, because their last common ancestor is a duplication event more recent than the human-mouse split. 3) M1 is a within-species outparalog of M2 because they are related by a duplication event that occurred before the human-mouse split. 4) M1 is also a between-species outparalog of H4 because they are related by a duplication event before the human-mouse split (and in different genomes). Figure 5.1b) The figure shows two hypothetical genes in humans (H1 and H2) and two in mouse (M1 and M2). There are three types of homologs shown: 1) M1 and H1 are one-to-one orthologs, as are M2 and H2. 2) M1 is a within-species outparalog of M2 because they are related by a duplication event that occurred before the human-mouse split, as are H1 and H2. 3) M1 is a between-species outparalog of H2 because they are related by a duplication event before the human-mouse split (and in different genomes), as are H1 and M2.

to a single-copy gene from another species, and neither represents the “true” ortholog (though see [Han and Hahn, 2009] for more complex sets of relationships between inparalogs).

Here we have extended our analysis of the functional similarity of orthologous and paralogous sequences; originally measured for pairs of genes from human and mouse [Nehrt et al., 2011], to

additional pairs of organisms and types of data. We also add additional controls for biases in GO data implemented in subsequent publications [Altenhoff et al., 2012].

Surprisingly, we find that multi-copy genes are equally or more likely to provide accurate functional information than are single-copy genes. Our results suggest that the organism itself plays at least as large a role in determining the function of genes as does the particular sequence of the gene alone. This insight will benefit the assignment of function to genes whose roles are not yet known by widening the pool of appropriate genes from which function can be inferred. Because paralogs are almost always either more- or less-related to a focal gene than an ortholog (for inparalogs or outparalogs, respectively), it is meaningless to compare the predictive power of all orthologs to all paralogs; it seems obvious that closely related orthologs will be more similar in function than distantly related paralogs, and vice versa. Instead, we focus on the predictive power of both orthologs and paralogs as a function of protein sequence divergence. Our results demonstrate that paralogous genes from the same species are often a much better predictor of functional divergence than are orthologs or paralogs from different species, even at lower sequence identities.

1. Results

1.1. Functional similarity between orthologs and all paralogs. Figures 5.2 and 5.3 shows the relationship between functional similarity and protein global sequence identity for both orthologous and paralogous pairs between all pairs of organisms tested. Functional similarity was calculated between all pairs of homologous proteins (i.e. those in the same gene family) in human and mouse (Figures 5.2a and 5.3a), human and rat (Figures 5.2b and 5.3b), mouse and rat (Figures 5.2c and 5.3c), and worm and fly (Figures 5.2d and 5.3d) for which there is an experimentally defined function for both members of the pair. The number of paralogs and homologs compared when considering each pair of organisms is summarized in Table 5.4.

For the Biological Process category, Figure 5.3a shows that the average functional similarity between human-mouse orthologs was consistently close to 0.4 over the entire range of sequence identities. A similar trend was observed for other pairs of organisms. Similarly, for the Molecular Function category, Figure 5.2a, shows functional similarity was close to 0.6 over the entire range when considering the human and mouse comparison. Again, a similar trend was followed when considering other organism pairs. The relatively low levels of similarity are at least partially influenced by the

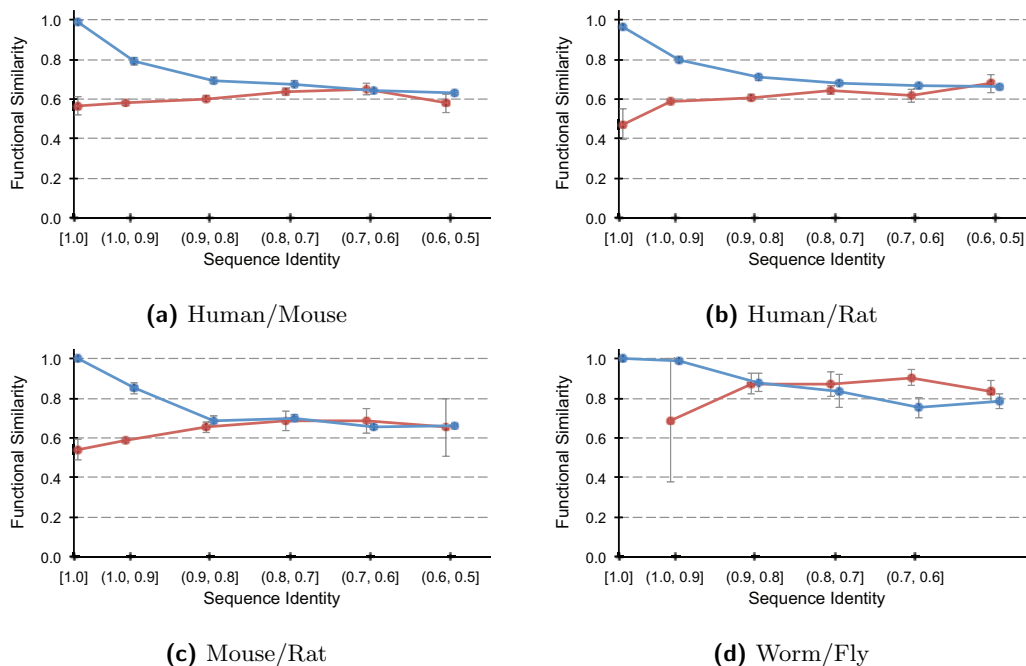


Figure 5.2. Comparison of orthologs (red) and paralog (blue) functional similarity across a range of global sequence identity values in four pairs of organisms when considering Molecular Function annotations. Standard error bars are shown.

sparsity of annotation, but this is unlikely to affect comparisons between classes of homologs. Most strikingly, there is no correlation between functional similarity and protein sequence identity for orthologs (Figures 5.2 and 5.3): two orthologs have the same average functional similarity at 99% as they do at 51% (see Discussion). This relationship held in humans and mouse when using dN/dS as a measure of sequence similarity (results not shown). In contrast, the functional similarity between paralogs is positively correlated with sequence identity for both ontologies, but shows a steeper decline for Biological Process. In Figures 5.2 and 5.3, the protein pairs included in the paralog category consist of both inparalogs and outparalogs; thus, the distributions largely consist of inparalogs in the high sequence identity ranges and outparalogs in the low sequence identity ranges.

Contrary to a common assumption (the “ortholog conjecture”), the functional similarity between paralogs is significantly higher than that between orthologs for high sequence identities and functional similarity is nearly the same for the different types of homologs as sequence identity approaches 50%.

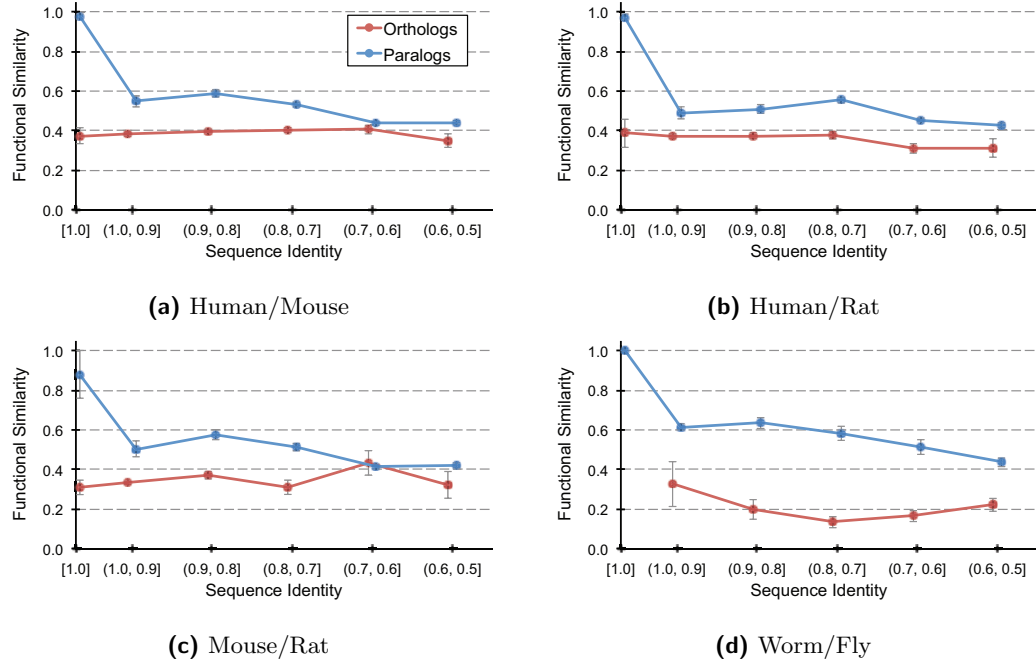


Figure 5.3. Comparison of orthologs (red) and paralog (blue) functional similarity across a range of global sequence identity values in four pairs of organisms when considering Biological Process annotations. Standard error bars are shown.

The curves do not provide comparable information for sequence identities below 50% because of an insufficient number of 1-to-1 orthologous pairs with very low identity.

1.2. Functional similarity between orthologs and subtypes of paralogs. While the ortholog data can be easily understood from Figures 5.2 and 5.3, the combination of several types of paralogs obscures the interpretation of the functional similarity between paralogs. We therefore separated paralogs into three further classes: (i) inparalogs, (ii) within-species outparalogs, and (iii) between-species outparalogs (Figure 5.1). Inparalogs and within-species outparalogs include protein pairs from the same species whereas between-species outparalogs include different organism pairs only. Figures 5.4 and 5.5 presents results for these separate types of paralogs for all four species comparisons; note that the curves for orthologs are identical to those in Figures 5.2 and 5.3. Table 5.4 documents the number of subtypes of paralogs considered when comparing each pair of organisms. Of note is that the majority of paralogs utilized in the worm-fly comparison are inparalogs. Given their more distant time of divergence this is expected.

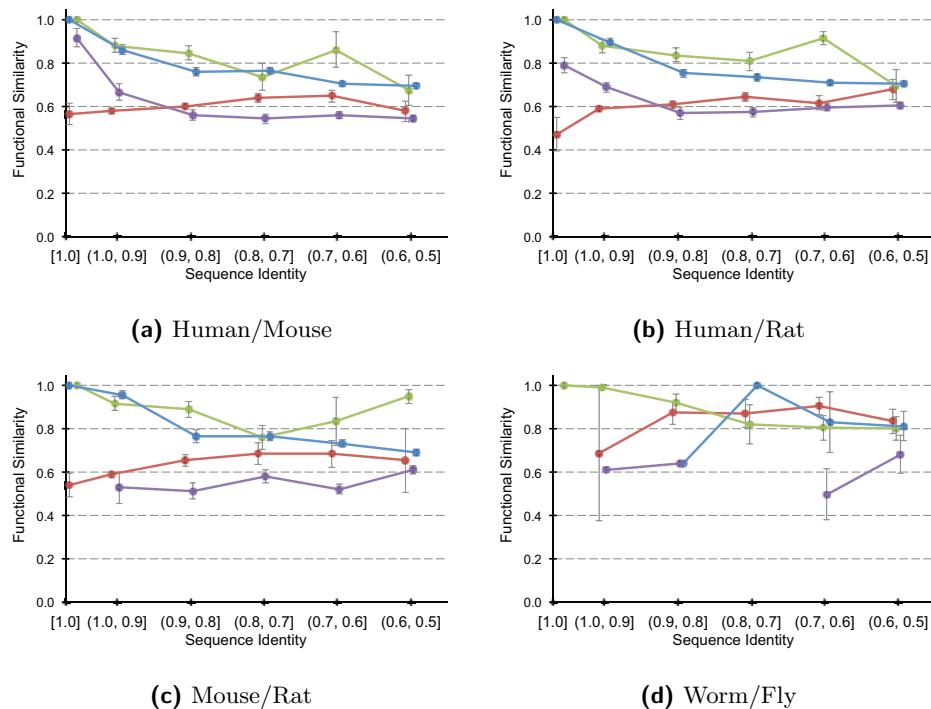


Figure 5.4. Comparison of orthologs and different subtypes of paralogs in four pairs of organisms for Molecular Function annotations. Results are divided into orthologs (red), inparalogs (green), within-species outparalogs (blue), between-species outparalogs (purple) for four pairs of organisms. Standard error bars are shown.

The functional similarity curves show a clear difference between subtypes of paralogs. Inparalogs appear to be most functionally similar to one another, and their functional similarity is positively correlated with sequence identity in both ontologies. Within-species outparalogs have a slightly steeper decline than inparalogs, but are significantly more functionally similar than either between-species outparalogs or orthologs. The between-species outparalogs show trends most similar to orthologs. In fact, in the Biological Process category, these two curves are nearly identical. However, in the Molecular Function category, the more sequence-similar outparalogs have slightly higher functional similarity than do orthologs, while the less-similar outparalogs have lower functional similarity than do orthologs. These results are consistent regardless of which pair of organisms are being compared. In the Discussion we propose an explanation for these relationships.

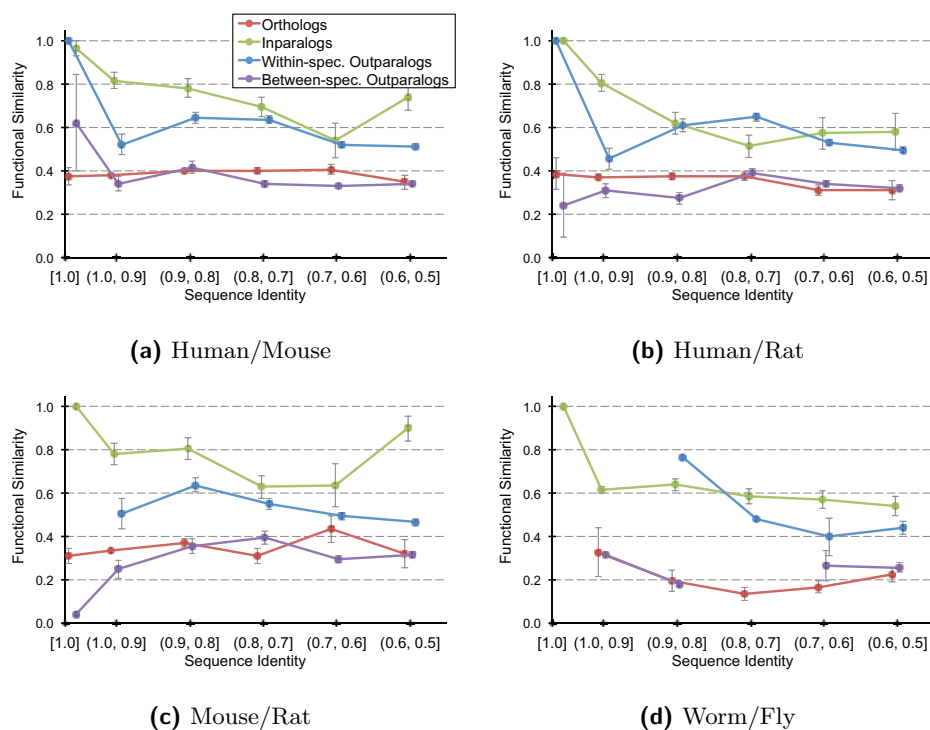


Figure 5.5. Comparison of orthologs and different subtypes of paralogs in four pairs of organisms for Biological Process annotations. Results are divided into orthologs (red), inparalogs (green), within-species outparalogs (blue), between-species outparalogs (purple) for four pairs of organisms. Standard error bars are shown.

1.3. Microarray-based measures of functional similarity. Because all of the above analyses are based on user-reported or curator-based determinations of function, they may still be affected by an individual researcher’s biases that we cannot control for. The only way to avoid this potential problem is to obtain a measure of function that is not dependent on an individual’s interpretation of experiments. Therefore, we conducted a parallel analysis of the relationship between protein similarity and functional similarity using microarray data from 25 homologous tissues in human and mouse [Su et al., 2004].

We used the correlation in levels of normalized gene expression across tissues as our measure of functional similarity (see Materials and Methods). Our final microarray dataset included 10,863 orthologs and 21,780 paralogous comparisons of all types, consisting of 2,014 inparalogs, 10,396 within-species outparalogs, and 9,370 between-species outparalogs. Figure 5.6 shows the relationship

between functional similarity and protein sequence identity for all pairs of genes represented in the microarray dataset. Consistent with all of the results obtained from the GO experimental dataset, microarray-based functional similarity shows a generally higher similarity between paralogs than orthologs ($\geq 70\%$; $P < 0.01$; Wilcoxon test) and a strong positive correlation with the sequence identity of paralogs but not orthologs. Our results were not dependent on the distance measure used to quantify functional similarity (see Materials and Methods). In addition, we again find that within-species paralogs, whether inparalogs or outparalogs, show the strongest relationship between sequence similarity and functional similarity.

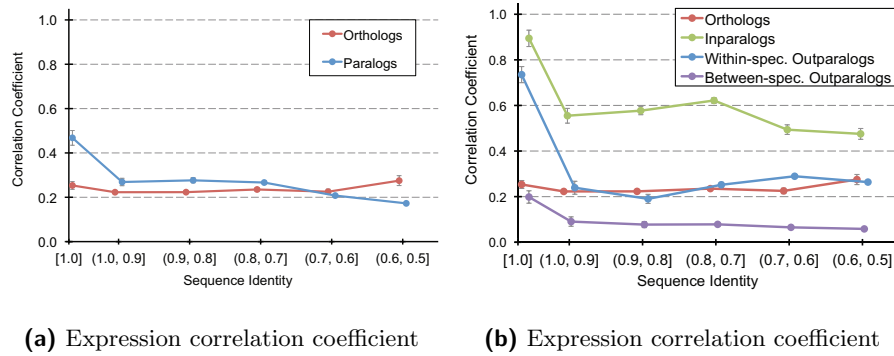


Figure 5.6. The relationship between the correlation in gene expression across 25 tissues (as measured by microarray) and sequence identity for (Figure 5.6a) human-mouse orthologs (red) and all paralogs (blue), and (Figure 5.6a) human-mouse orthologs (red), inparalogs (green), within-species outparalogs (blue), between-species outparalogs (purple). Standard error bars are shown.

The microarray data used here have also been utilized in a number of previous evolutionary studies, though these studies largely focused only on paralogs [Makova and Li, 2003], only on orthologs [Liao and Zhang, 2006], or on comparisons between orthologs with and without lineage-specific paralogs [Huminiacki and Wolfe, 2004]. While these previous studies did not present their analyses in exactly the same way as we have done, we stress that for both paralogs and orthologs our results are in strong quantitative and qualitative agreement with these studies. For both the relationship between protein similarity and functional similarity, and for the average correlation in expression

patterns, our results are consistent with previous results; that is, nothing about the way we have conducted our analysis has biased us toward our finding. We have largely followed the proscriptions of these previous papers for normalizing the microarray data and in controlling for cross-hybridization, which all of these previous papers agree does not appear to be an issue in these data.

Because there is no interpretation or assignment of functional terms needed to obtain these results, we believe they strongly support all of our previous analyses. Therefore, these two datasets are largely non-overlapping and provide independent support for the results.

1.4. Protein interaction-based measures of functional similarity. Protein-protein interaction (PPI) data present a potentially useful high-throughput method for analyzing function. The function of a protein is in large part defined by the type of interactions it takes part in, be they transient and short-lived [Crowley and Ubbink, 2003], or more permanent as in the formation of complexes [Nooren and Thornton, 2003]. Even though only 73% of proteins in the gene ontology are annotated with the term “binding” it could easily be argued that every protein binds something, be it a small molecule, other proteins, or another copy of itself. There are several high-throughput techniques that allow one to detect protein interactions. Perhaps the most common high-throughput method for the detection of protein interactions is the yeast two-hybrid (Y2H) assay [Fields and Song, 1989].

Although the false positive rate for Y2H experiments is estimated to be high [Deane et al., 2002, Hart et al., 2006, Sprinzak et al., 2003], they provide large scale data that are not subject to the potential biases of other smaller-scale hypothesis-driven experiments. Y2H experiments enable the detection of two target proteins’ interaction by first inserting the tested genes into the yeast genome with transcription activating and binding domains attached to each protein respectively. Interactions are then detected through the expression of a reporter gene when the two transcription domains are brought together by the interaction of the target proteins. One caveat is that, while detecting interactions in yeast (specifically the yeast nucleus) may provide a neutral organism in which to conduct experimentation, it also removes many of the species-specific environmental factors such as post-translational modifications and transcription factors that can affect protein behavior.

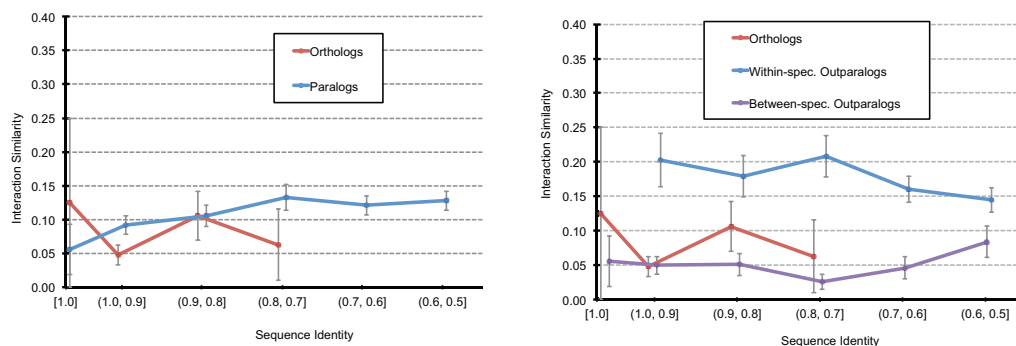
Several studies have assessed the conservation of protein interactions, looking at the ability to infer interactions based on homology; or its common proxy sequence similarity [Aloy et al., 2003, Yu et al., 2004, Mika and Rost, 2006, Saeed and Deane, 2008, Tyagi et al., 2012, Lewis et al.,

2012]. For example, Mika and Rost [2006] compare interactions within and between organisms, but do not distinguish between the different types of homologs. While confusing between-species paralogs with orthologs is not necessarily a problem when one is simply attempting to determine the utility of transferring interactions from one species to another, it is problematic when statements are consequently made about the comparative functional similarity of orthologs and paralogs.

Another potential shortcoming of previous studies is their edge-based focus. As a more direct way of testing ‘the ortholog conjecture we are specifically interested in testing how conserved the interaction partners of homologous sequences are, not how conserved a particular edge, or interaction, is. The focus on edges (interactions), as opposed to vertices (i.e. proteins), of previous studies does not facilitate the direct comparison of homologous sequences. Furthermore, proteins with a large number of interaction partners will disproportionately influence results when performing edge-based analysis. Because of the specific nature of the analysis we are seeking to undertake, we developed a novel metric for comparing the similarity of a pair of genes interaction partners (Section 3.5). This idea is illustrated in Figure 5.9 where the set of interaction partners for two proteins are compared. Instead of asking whether a particular edge, say from H_1 to H_2 is conserved in mouse, we compare the set of interaction partners of two proteins; in this case H_1 and M_1 .

We analyzed interactions for human and mouse genes taken from IntAct version 150 [Hermjakob et al., 2004]. When comparing the interaction similarity of genes in human and mouse we did not find any evidence to support the ortholog conjecture. Figure 5.7 shows the results obtained when applying the metric detailed in Section 3.5 to PPI data. Sequence identity bins with fewer than 15 sequences being compared for any homology type were ignored. Inparalogs were omitted from Figure 5.7b because there were fewer than 15 sequences in all bins compared. Of note is the fact that, although paralogs only show higher interaction similarity for the $[.9 - 1)$ sequence identity range in Figure 5.7a, when comparing within-species outparalogs with orthologs and between-species outparalogs in Figure 5.7b we again find that pairs of sequences from the same organism group together and those in different organisms show lower similarity.

1.5. Family-based analyses. The ortholog conjecture can also be framed in the context in cases where one is given both a paralog and an ortholog and it is desirable to determine which is more functionally similar. It could be argued that protein families that have not experienced any duplications do not actually provide an opportunity to test the ortholog conjecture, regardless of



(a) Interaction similarity between orthologs and paralogs (b) Interactions similarity between orthologs and subtypes of paralogs

Figure 5.7. The relationship between interaction similarity, measured as described in Section 3.5, and global sequence identity for human and mouse genes. Interactions were taken from IntAct version 150. Figure 5.7a shows results for orthologs and all paralogs, whereas Figure 5.7b divides paralogs into different subtypes. In-paralogs were omitted due to insufficient data, as were results for any bin containing fewer than 15 comparisons between pairs of genes. Homolog pairs are divided into orthologs (red), within-species outparalogs (blue), between-species outparalogs (purple). Standard error bars are shown.

their level of conservation across a wide range of organisms, because there are no paralogs with which function can be compared. Testing whether an ortholog or paralog exhibits a higher level of functional similarity is a moot point when there is not both a paralog and an ortholog whose functional similarity can be compared; yet although our analysis presented in Figures 5.2 and 5.4 compares the functional similarity between orthologs and paralogs on a large scale, it allows that these strict 1:1 orthologs still be considered.

In order to test this slightly different formulation of the ortholog conjecture, it is useful to take a family-based view and compare the predictive power of paralogs and orthologs within the same family. In order to do this we asked, for a given family, first whether an ortholog or a paralog was more similar at the sequence level, and then whether an ortholog or the particular paralog was more similar at the functional level.

The counts for the groups were obtained as follows: for each family, only one target protein (functionally annotated) was selected uniformly randomly from all proteins with at least one ortholog and at least one paralog in the family, and all its functionally annotated homologs were collected. We then asked whether at least one of the paralogs had higher sequence similarity than the ortholog, and then whether it had higher or lower functional similarity.

This analysis required functionally annotated triples within gene families (i.e. the target gene, an ortholog, and a paralog of any type); thus 1-to-many and many-to-many orthologous relationships were included in this analysis. In cases where multiple genes were co-orthologous to the target, the ortholog having the highest sequence identity with the selected target protein was used for comparison. Note that each gene family was counted only once in this analysis, preventing families with large numbers of lineage-specific duplications from biasing the results. Finally, to ensure that the choice of target protein did not unduly affect the results, we repeated the analysis 1,000 times, choosing a new target protein from the 1145 unique families containing experimentally annotated triples each time (685 with Biological Process and 711 families with Molecular Function annotation). Table 5.1 summarizes counts in the Biological Process and Molecular Function ontologies for human and mouse.

The family-based analysis showed similar trends to those observed in previous sections. In the Biological Process category, if the orthologous sequence was more similar to the target protein, the ortholog had higher functional similarity to the target protein than all of its paralogs in only $33.4 \pm 0.1\%$ of the cases (mean $\pm$ standard error). In contrast, in $82.9 \pm 0.4\%$ of protein families in which a paralogous sequence was most similar to the target protein, it was also functionally most similar. In the Molecular Function category, the observed difference between orthologs and paralogs was similar: an ortholog had higher functional similarity to the target protein than all of its paralogs in only $49.5 \pm 0.1\%$ of the cases. On the other hand, if the most similar sequence to a target protein was a paralog, the paralog was functionally most similar to the target protein in $71.1 \pm 0.5\%$ of families.

2. Discussion

The accelerating pace of whole-genome sequencing coupled with the rapid, but relatively slower, pace of functional genomics projects has required commensurately fast methods for computational

Table 5.1. Family-based analysis using functional similarity (FS) and sequence identity (SI) for human and mouse. Each field shows the average number of protein families ($\pm$ standard error), out of 1,000 runs with randomly selected target proteins, in which the row and column conditions were satisfied.

	Biological Process		Molecular Function	
	Paralog greater FS	Ortholog greater FS	Paralog greater FS	Ortholog greater FS
Paralog greater SI	17.4 ± 0.2	3.6 ± 0.1	17.7 ± 0.2	7.2 ± 0.1
Ortholog higher SI	442.4 ± 0.8	221.6 ± 0.8	346.8 ± 0.9	339.3 ± 0.9

annotation of genes and proteins. Because functional studies are disproportionately concentrated in only a handful of model organisms, the working model for computational annotation has been transfer-by-similarity [Rentzsch and Orengo, 2009], a principle in which experimentally determined functional annotation from a characterized protein is assigned to an uncharacterized protein if their sequence similarity is greater than some pre-specified threshold (e.g. sequence identity, e-value). With some caveats involving local vs. global sequence alignments (especially for multi-domain proteins), the basic tenet of such function transfer is that proteins that are closely related (and therefore similar in sequence) tend to have similar functions. Several recent papers have discussed the details of such annotation transfer, attempting to find similarity thresholds necessary for accurate inference of enzymatic functions [Addou et al., 2009, Rost et al., 2003]. More sophisticated prediction algorithms, exploiting not only sequence similarity but also the structure of functional ontologies, have also been proposed [Hawkins et al., 2006, Martin et al., 2004]. The field of phylogenomics [Tatusov et al., 1997, Eisen, 1998] uses evolutionary relationships as a guide to function prediction from sequence, preferentially transferring annotations between orthologs over paralogs because they are believed to be more functionally similar (the ortholog conjecture). Our study is the first to address this assumption using experimental evidence from 12,204 unique papers as well as an independent microarray dataset.

2.1. The evolution of gene function. Our results strongly suggest that the ortholog conjecture is not correct between human and mouse: given equivalent levels of protein divergence (or even slightly higher divergence), paralogous genes from the same species (either human or mouse) are better predictors of function than are orthologs from the other species. A similar result was

previously obtained among yeast, fly, and worm when comparing conserved protein-protein interactions between homologs within the same species and homologs from different species (although this study did not distinguish among orthologs, inparalogs, and outparalogs Mika and Rost [2006]). We ensured that our analyses were not affected by a large number of possible biases. We considered biases due to the ontology terms used in human and mouse, the depth of annotation terms used, whether homologs were studied in the same or different publications, biases due to differences in experimental procedures, and even biases in the user-defined interpretation of function. We found several interesting biases in the data; notably, that functions of homologs of any kind reported in the same publication or using the same experimental technique were more likely to be similar than a random pair of homologs of equal protein divergence; but none of these biases affected the qualitative patterns found in our data.

In addition to a general lack of support for the ortholog conjecture, our analyses revealed several surprising patterns. One of the most surprising is the lack of any discernible relationship between protein similarity and functional similarity for orthologs, whether considering Biological Process or Molecular Function annotations (Figures 5.2 and 5.3). Average functional similarity for orthologs is between 60 – 70%, regardless of level of divergence. Even for orthologous proteins approaching 100% identity, there is still relatively little overlap in annotation. While this fact may at first seem surprising, it is important to consider how individual experiments are conducted. Almost never are single genes (or proteins) from both mouse and human isolated and then compared in the same *in vitro* assay. Instead, the vast majority of experiments included in our dataset are conducted *in vivo* (e.g. knockouts, genetic crosses), *in situ* (e.g. tissue-specific expression), or *in vitro* but with species-specific conditions and/or interactors (e.g. yeast two-hybrid). Function is therefore assessed in the context of individual organisms, not in a common laboratory setting.

The importance of cellular and organismal context in defining protein function may go a long way toward explaining many aspects of our results, including the lack of a relationship between functional and sequence similarity for orthologs, the presence of this relationship for paralogs, and the differences between different types of paralogs (in-/outparalogs). We propose that, although how quickly a protein sequence itself evolves is important to understanding the rate at which protein function evolves, the rate at which its cellular context, including directly and indirectly interacting molecules, evolves plays a very important role in the evolution of function. To further explain this

hypothesis, note that all of the orthologous genes between a pair of organisms are the same age: that is, they all share a last common ancestor at the split between those two organisms, regardless of their level of sequence identity. Unlike orthologs, the paralogs studied here shared common ancestors at many different times in the past, with some paralogs having split only a few million years ago while others split > 100 million years ago. We propose that this difference in divergence times is the key to understanding the difference in relationships between functional and sequence similarity. The orthologs all share the same age, and therefore the same average functional similarity; but the paralogous pairs are of many different ages, and therefore different functional similarities.

Why should proteins of the same age share the same level of functional similarity? While there is no direct role for “time” in evolution that is not tied to mutation, we suggest that what time represents here is the evolution of the cellular context: the sum of the evolutionary changes over all of the directly and indirectly interacting molecules. If this context evolves at a steady rate (i.e. the average amount of functional change among all of the interacting molecules remains relatively constant), then protein function will appear to evolve at a steady rate, a rate largely disconnected from the level of an individual proteins sequence divergence. Several pieces of evidence support this conjecture. First, our results above show that even orthologous proteins that are 100% identical have different functions. Since it is obvious that the proteins themselves have not changed, the change must be due to regulation or downstream effects of these molecules. For example, Liao and Zhang [2008] found that $> 20\%$ of genes that are essential for viability in humans are not essential in mouse. It is unlikely that changes to the proteins themselves have made them essential or not, but rather that their context in cellular and organismal networks has evolved [Liao and Zhang, 2008]. Second, we find a weak relationship between synonymous sequence identity a good measure of divergence time [Li, Wen-Hsiung, 1997], and functional similarity for paralogous pairs (Figure 5.8, results produced by Nathan Nehrt). This supports the idea that time is a key factor in the evolution of protein function. Finally, we again note that there is higher functional similarity among inparalogs and within-species outparalogs than there is for either orthologs or between-species outparalogs. Because both inparalogs and within-species outparalogs are present in the same organism, it is highly likely that they share a much more similar cellular context. And because this context is highly similar, the functions of these proteins are also likely to be more similar. Our conclusion is that the most important aspect of functional similarity is not sequence similarity,

but rather contextual similarity. A straightforward experiment to test this proposal would involve collecting functional data for orthologous pairs of different ages to see whether there is the predicted relationship between sequence identity and functional similarity. We would expect to see the same pattern in any pair of orthologs considered, of any age (cf. [Mika and Rost, 2006]).

Some researchers may be concerned that the function being measured here is not independent of the organism, and is therefore not appropriate for testing the ortholog conjecture. Of course it is possible that if measured in a common *in vitro* environment orthologous proteins really would be more functionally similar than paralogous proteins; after all, studies of rates of protein sequence evolution suggest an increased rate of sequence change among paralogs [Goodman et al., 1975, Kimura, 1981, Li and Gojobori, 1983, Lynch and Conery, 2000]. However, this is not the manner in which the vast majority of functional data is collected, and would therefore be little solace in applying the ortholog conjecture to real data.

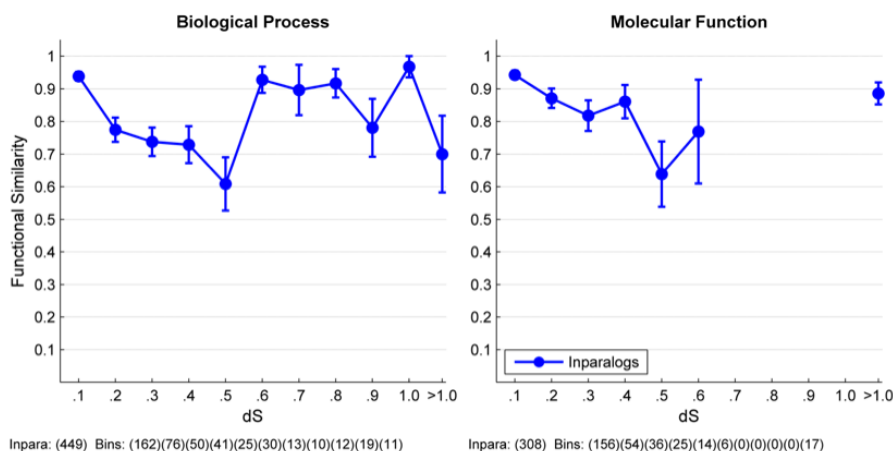


Figure 5.8. The relationship between functional similarity and dS calculated using the Goldman and Yang method for inparalogs only.

2.2. Implications for protein function prediction. The results of our study suggest that neither sequence similarity nor identification of orthologous assignments alone can be considered an accurate predictor of protein function. We find that orthologous proteins between human and mouse share a constant level of functional similarity over a wide range of (global) sequence identities,

while the functional similarity between paralogs is dependent on the type of paralogy, level of sequence identity, relative chromosomal location of duplicated genes, and organismal context. We find that sequence identity thresholds as a means of function transfer are generally applicable only to within-species paralogs. Moreover, these thresholds depend on the type of paralogy and a specific duplication event, with inparalogs typically having lower thresholds for similarly accurate functional transfer than outparalogs. On the other hand, in the absence of within-species paralogs, our data indicate that orthologs and between-species outparalogs are similarly accurate in predicting protein function. In general, however, such relationships cannot be deemed ideal for function transfer of GO terms, as the average accuracy of predictions using orthologs and between-species outparalogs were consistently lower than 0.70 (Figures 5.2 and 5.3). Though many computational methods use only orthologous genes for function prediction, for methods that can be tuned to exploit different types of evolutionary relationships (e.g. SIFTER; [Engelhardt et al., 2005]) our results can be used to improve prediction accuracy. The usefulness of including paralogous genes when inferring function has already been illustrated by Škunca et al. [2013]

Functional annotation of genes with unknown function is often carried out by researchers working on particular proteins. In these cases, far from being an automated process of ortholog identification and functional transfer, individual researchers may examine the function of many closely related homologs before making decisions about functional annotations, or even before designing experiments. If they are available, researchers may be using the functions of both orthologs and paralogs to guide their own functional annotations. When inparalogs are available and happen to have the highest sequence identity, these genes may actually be the ones having the largest influence on the functional annotations in common databases; such a process of individual functional inference would create a pattern much like the one we observe. While our analysis of microarray data is consistent with the high functional similarity of within-species paralogs and is free from individual researcher or curator bias, we cannot rule out the possibility that such bias exists in widely used databases. However, such biases are likely to only apply to organisms already being studied by a large community of researchers in molecular biology. Many new genomes are being sequenced solely for the evolutionary or environmental importance of a species, and are therefore unlikely to have much prior data on gene and protein function. In these cases, our results suggest that functional transfer need not be dependent on the identification of orthologous genes in a model organism.

It is well documented that small number of experimental annotations are transferred to a much larger number of homologous proteins and propagated across biological databases, often with gross inaccuracies [Schnoes et al., 2009]. Inaccurate functional annotation via computational methods can influence a wide variety of biological conclusions: for instance, any analysis looking for enriched or over-represented GO terms. We suggest that such studies should be cautiously interpreted until the prediction of protein function reaches the sensitivity and specificity necessary for accurate functional inference.

Finally, it must be mentioned again that our study has only addressed protein functions in a limited number of organisms. A fuller picture of the accuracy of protein function prediction would include many pairs of species from across the tree of life (see Mika and Rost [2006] for similar results from comparisons among yeast, fly, and worm). However, our study includes comparisons between human and both mouse and rat: if the main purpose of biomedical research into model organisms is to understand the function of genes and proteins in humans, then we might expect these studies to be predictive of function in humans. While our results certainly show that function can be transferred between organisms, they also strongly suggest that the best model organism is ourselves.

3. Materials and Methods

3.1. Comparative genomics data. Ensembl Compara (release 49, March 2008) gene trees were used to identify all homologous human-human, mouse-mouse, and human-mouse gene pairs. Though there are many methods and databases available for identifying homologous relationships, they provide qualitatively similar results [Shi et al., 2010]. Ortholog assignments: Ensembl homology descriptions “ortholog 1:1” and “apparent ortholog 1:1” were used to annotate orthologous pairs. The apparent orthologs were treated as 1-to-1 orthologs since this description can result from a situation where a gene duplication is actually followed by gene losses in both lineages, but more often occurs because of an incorrect tree topology and incorrect duplication node labeling [Vilella et al., 2009]. Paralog assignments: all between-species paralogs were treated as outparalogs. To distinguish inparalogs from outparalogs among the within-species paralogs, we examined the branch on the tree where the gene duplication took place to determine if the duplication occurred subsequent to the human-mouse speciation event. While the Compara dataset does not include bootstrap

values for each node in the gene tree, incorrect trees will only conflate orthologs with between-species outparalogs and inparalogs with within-species outparalogs (because species assignments will never be mistaken). Though we cannot control for each type of error, the fact that within-species gene pairs cannot be confused with between-species gene pairs (of any kind) means that our main results are robust to the exact tree topologies. As well as updating GO annotations the original single organism comparison of human and mouse has been expanded to include three additional comparisons between organisms: human and rat, rat and mouse, and worm and fly Table 5.4 details the number of sequences with at least one paralog or ortholog for each organism comparison.

Table 5.2. Number of sequences with homologs in each organism comparison for four pairs of organisms

Organism pair	Organism 1	Organism 2
Human and Mouse	Human	Mouse
	18380	19863
Human and Rat	Human	Rat
	17614	19215
Mouse and Rat	Mouse	Rat
	20580	20762
Worm and Fly	Worm	Fly
	5539	2310

3.2. Protein function data. Biological Process and Molecular Function protein function information was retrieved from the January 2011 version of the Swiss-Prot database [Bairoch et al., 2005]. Only curated GO term annotations were used in the analysis. These include all experimentally inferred annotations: inferred from direct assay (IDA), expression pattern (IEP), genetic interaction (IGI), mutant phenotype (IMP), and physical interaction (IPI) evidence codes. We also included the traceable author statement (TAS) and inferred by curator (IC) evidence codes. Since both the Biological Process and Molecular Function ontologies are represented by directed acyclic graphs (DAGs), the original functional terms were propagated towards the root of each DAG (with the root node excluded) thus producing a complete set of terms for each protein. The January, 2011

version of the gene ontology was used to conduct term propagation. Table 5.3 details the number of proteins in each organism that had GO annotations supported by experimental or curated evidence in each ontology. Table 5.4 details the number of pairs of sequences of each homology sub-type when looking at the intersection of genes with experimental GO annotations and homology relationships

Table 5.3. Number of sequences with experimental annotations in each ontology

Organism	Taxonomic Identifier	Molecular Function	Biological Process
<i>Caenorhabditis elegans</i>	6239	1035	2509
<i>Drosophila melanogaster</i>	7227	3921	4023
<i>Homo sapiens</i>	9606	11296	9827
<i>Mus musculus</i>	10090	6073	6732
<i>Rattus norvegicus</i>	10116	2607	2587

3.3. Calculation of similarity. We calculated protein sequence identity by using Needleman-Wunsch alignments of protein sequences with the BLOSUM62 scoring matrix (gap opening penalty = 11; gap extension penalty = 1). We divided the number of matching residues by the length of the alignment. For the calculation of dN/dS and dS, we used the Goldman and Yang method [Goldman and Yang, 1994].

To calculate functional similarity for the GO data, let $T(p)$ be a set of propagated GO terms for protein p and $T(q)$ be a set of propagated GO terms for protein q . Then, the functional similarity $fs(p, q)$ between p and q was calculated as:

$$fs(p, q) = \frac{|T(p) \cap T(q)|}{2 \times |T(p)|} + \frac{|T(p) \cap T(q)|}{2 \times |T(q)|}$$

This formula can be interpreted as the average of the fraction of correctly predicted functional terms in p when protein q is used to predict p 's function (by transfer of all its terms), and the fraction of correctly predicted functional terms in q when protein p is used to predict q 's function [Radivojac et al., 2008]. This measure of functional similarity is known as the Maryland bridge coefficient and is highly correlated with the Jaccard coefficient—the size of the intersection over the size of the union between two sets [Glazko et al., 2005]. Clearly, $0 \leq fs(p, q) \leq 1$, with 0 corresponding to proteins

(a) Molecular Function

	trees	orthologs	inparalogs	within-species outparalogs	between-species outparalogs
Human and Mouse	1925	2088	205	2068	1514
Human and Rat	1479	1418	206	1925	1322
Mouse and Rat	957	926	127	1023	668
Worm and Fly	110	45	991	26	20

(b) Biological Process

	trees	orthologs	inparalogs	within-species outparalogs	between-species outparalogs
Human and Mouse	1973	2257	211	1790	1346
Human and Rat	1381	1310	178	1548	1024
Mouse and Rat	1061	1061	144	991	707
Worm and Fly	192	95	1214	56	42

Table 5.4. Counts of the number of trees and subtypes of homology relationships used when performing multi-organism comparison of functional similarity for Molecular Function (Table 5.4a) and Biological Process (Table 5.4b) annotations.

with disjoint sets of functional terms and 1 corresponding to proteins with identical sets of terms. Functional similarity of 0 may occur because we removed the root node from each ontology.

To calculate functional similarity for the microarray data, we used the Pearson correlation coefficient (the Euclidean distance provided similar results). The correlation coefficient $corr(p, q)$ between genes p and q (in a somewhat abused notation where p and q represent both genes and their indices in microarrays) for normalized data was calculated as:

$$corr(p, q) = \frac{\sum_{j \in T} (n_{pj} - \mu_p) \cdot (n_{qj} - \mu_q)}{\sqrt{\sum_{j \in T} (n_{pj} - \mu_p)^2} \sqrt{\sum_{j \in T} (n_{qj} - \mu_q)^2}}$$

where T is the index set of tissues being considered, n_{pj} is the normalized expression for gene p in tissue j , and μ_p is the mean expression level for gene p over all tissues in T .

3.4. Microarray data. Microarray data presented in Su et al. [2004] was retrieved from the Gene Expression Omnibus, accession GSE1133. The data were collected on three different microarray

platforms, two from human and one from mouse. The two platforms from human, GPL96 and GPL1074, consist of expression values in 78 tissues for 22,283 and 11,391 probesets respectively. The mouse platform, GPL1073, consists of expression values in 61 tissues for 31,373 probesets. 25 of these tissues are common between human and mouse and were used here. In order to create an updated mapping between probesets and genes, individual probe sequences (there are 16 per probeset) were searched against Ensembl transcripts using exact matches returned from BLAST. Only probesets that perfectly matched a genes sequence and did not have probes matching any other gene were considered. When multiple probesets uniquely matched to the same gene, the values were averaged after normalization to give a single genic expression value.

Expression data was normalized within each platform individually. Expression values were first normalized within each individual tissue using the z-score method, forcing expression values within a tissue to have a mean of 0 and a standard deviation of 1. After expression values were normalized within a tissue, they were again normalized for individual probesets across tissues, forcing expression values for a single probeset to have a mean of 0 and a standard deviation of 1 across tissues. Specifically, if we represent the expression value of a probeset i in a tissue j as s_{ij} , we can define the tissue-normalized expression value, t_{ij} , as:

$$t_{ij} = \frac{s_{ij} - \mu_j}{\sigma_j}$$

where μ_j and σ_j are the mean and standard deviation of expression values in tissue j . The final normalized expression value for a probeset i in tissue j , n_{ij} , is defined as:

$$n_{ij} = \frac{t_{ij} - \mu_i}{\sigma_i}$$

where μ_i and σ_i are the mean and standard deviation of t_i values for gene i in all tissues. After these two steps of normalization, we averaged probesets that match to the same gene and then averaged duplicate samples for the same tissue. In total, we were able to obtain expression data for 15,907 human genes and 15,552 mouse genes. This reduced the number of gene trees with at least two functionally annotated genes to 7,495; the total number of data pairs used for orthologs is 10,863, for inparalog pairs is 2,014, for within-species outparalogs is 10,396, and for between-species outparalogs is 9,370.

3.5. Protein interaction data. We obtained protein interaction data from IntAct version 150 [Hermjakob et al., 2004]. We only utilized results high-throughput experiments that are relatively immune to selection bias. Because IntAct contains results many different types of experiments we also only utilized data from high-throughput experiments (two hybrid, experiment code MI:0018; two hybrid array, experiment code MI:0397; and two hybrid fragment pooling approach, experiment code MI:0399) and physical interactions (MI0915).

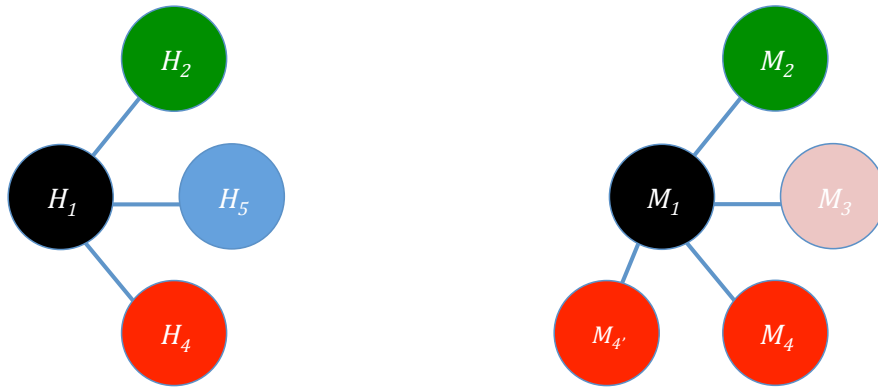


Figure 5.9. Two sample interactions networks for homologous sequences in human (H_1) and mouse (M_1). A common numerical suffix denotes shared ancestry. Paralogous sequences are designated with a prime symbol. In this case, M_4 and $M_{4'}$ are paralogs. H_2 and M_2 as well as H_4 and M_4 are orthologs. H_5 and M_3 share no homologs among the interaction partners of H_1 or M_1 .

Previous studies of the conservation of interactions have all done so from an edge-based perspective. That is, if we represent a PPI network as an undirected graph $G = (V, E)$ where each vertex, $p \in V$, is a protein, and each edge, (p_i, p_j) , represents a pair of proteins that interact $(p_i, p_j) \in E$ these studies have only addressed the conservation of individual edges, (p_i, p_j) , within or between organisms. A different approach, which tests the ortholog conjecture more directly, would be to look at how conserved the set of interactions between pairs of proteins are. Specifically, if we define the set of proteins that interact with protein p_i as $V_i = \{p_k : (p_i, p_k) \in E\}$, we can then define a precision measure between the interaction partners of two proteins p_i and p_j as,

$$pr(p_i, p_j) = \frac{|H(V_j, V_i)|}{|V_j|}$$

and a recall measure as,

$$rc(p_i, p_j) = \frac{|H(V_i, V_j)|}{|V_i|}$$

where we define the set $H(V_i, V_j)$ as the set of proteins from V_i that are either in, or have a homolog in, V_j

$$H(V_i, V_j) = \{p_k : p_k \in V_i \wedge (p_k \in V_j \vee \text{homolog}(p_k) \in V_j)\}.$$

Figure 5.9 can be used to illustrate the basic idea behind our vertex based metric. In this example we are interested in calculating the interaction similarity between two genes: $H1$ and $M1$. The interactions for the two genes are denoted as $V_{H1} = \{H_2, H_5, H_4\}$, and $V_{M1} = \{M_2, M_3, M_4, M_{4'}\}$ respectively. When determining how many vertices from V_{H1} are in V_{M1} we find that 2 out of three proteins are either members of V_{M1} , or have homologs in V_{M1} . This enables us to calculate $pr(H1, M1) = rc(M1, H1) = \frac{3}{4}$. Furthermore, we can calculate $rc(H1, M1) = pr(M1, H1) = \frac{2}{3}$.

Bibliography

- S. Addou, R. Rentzsch, D. Lee, and C. A. Orengo. Domain-based and family-specific sequence identity thresholds increase the levels of reliable protein function transfer. *Journal of Molecular Biology*, 387(2):416–430, 2009.
- A. Alexeyenko, J. Lindberg, R. Perez-Bercoff, and E. L. L. Sonnhammer. Overview and comparison of ortholog databases. *Drug Discovery Today: Technologies*, 3(2):137–143, 2006.
- P. Aloy, H. Ceulemans, A. Stark, and R. B. Russell. The relationship between sequence and interaction divergence in proteins. *Journal of Molecular Biology*, 332(5):989–998, 2003.
- A. M. Altenhoff, R. A. Studer, M. Robinson-Rechavi, and C. Dessimoz. Resolving the Ortholog Conjecture: Orthologs Tend to Be Weakly, but Significantly, More Similar in Function than Paralogs. *PLoS Computational Biology*, 8(5):e1002514, 2012.
- G. Alterovitz, M. Xiang, D. P. Hill, J. Lomax, J. Liu, M. Cherkassky, J. Dreyfuss, C. Mungall, M. A. Harris, M. E. Dolan, et al. Ontology engineering. *Nature Biotechnology*, 28(2):128–130, 2010.
- S. F. Altschul et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402, 1997.
- R. Amundson and G. V. Lauder. Function without purpose. *Biology and Philosophy*, 9(4):443–469, 1994.
- M. Ashburner et al. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.
- A. Bairoch, R. Apweiler, C. H. Wu, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, et al. The universal protein resource (UniProt). *Nucleic Acids Research*, 33(1):D154–D159, 2005.
- D. Bandyopadhyay, J. Huan, J. Liu, J. Prins, J. Snoeyink, W. Wang, and A. Tropsha. A structure-based function inference using protein family-specific fingerprints. *Protein Science*, 15(6):1537–1543, 2006.

- G. J. Bartlett, A. Todd, and J. Thornton. Inferring protein function from structure. In P. E. Bourne and H. Weissig, editors, *Structural Bioinformatics*, volume 44, chapter 19, pages 387–407. Wiley-Liss Inc., 2003.
- Z. Barutcuoglu, R. E. Schapire, and O. G. Troyanskaya. Hierarchical multi-label prediction of gene function. *Bioinformatics*, 22(7):830–836, 2006.
- A. Ben-Hur and W. S. Noble. Kernel methods for predicting protein-protein interactions. *Bioinformatics*, 21(1):i38–i46, 2005.
- O. Bodenreider. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32(Database issue):D267–270, 2004.
- K. M. Borgwardt, C. S. Ong, S. Schönauer, S. Vishwanathan, A. J. Smola, and H.-P. Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21(suppl 1):i47–i56, 2005.
- S. E. Brenner. Errors in genome annotation. *Trends in Genetics*, 15(4):132–133, 1999.
- C. Bridges. The Bar "gene" a duplication. *Science*, 83:210–211, 1936.
- C. B. Bridges. Salivary chromosome maps with a key to the banding of the chromosomes of *Drosophila melanogaster*. *Journal of Heredity*, 26(2):60–64, 1935.
- D. Brown and K. Sjolander. Functional classification using phylogenomic inference. *PLoS Computational Biology*, 2(6):e77, 2006.
- R. D. Chen and S. S. Jeong. Functional prediction: Identification of protein orthologs and paralogs. *Protein Science*, 9(12):2344–2353, 2000.
- W. T. Clark and P. Radivojac. Analysis of protein function and its prediction from amino acid sequence. *Proteins, Structure, Function and Bioinformatics*, 79(7):2086–2096, 2011.
- J. C. Costello, M. M. Dalkilic, S. M. Beason, J. R. Gehlhausen, R. Patwardhan, S. Middha, B. D. Eads, J. R. Andrews, et al. Gene networks in *drosophila melanogaster*: integrating experimental data to predict gene function. *Genome Biology*, 10(9):R97, 2009.
- D. Cozzetto and D. T. Jones. The contribution of intrinsic disorder prediction to the elucidation of protein function. *Current Opinion in Structural Biology*, 2013.
- P. B. Crowley and M. Ubbink. Close encounters of the transient kind: protein interactions in the photosynthetic redox chain investigated by NMR spectroscopy. *Accounts of Chemical Research*, 36(10):723–730, 2003.

- W. M. Dahdul, J. P. Balhoff, D. C. Blackburn, A. D. Diehl, M. A. Haendel, B. K. Hall, H. Lapp, J. G. Lundberg, C. J. Mungall, M. Ringwald, et al. A unified anatomy ontology of the vertebrate skeletal system. *PloS One*, 7(12):e51070, 2012.
- M. M. Dalkilic, J. C. Costello, W. T. Clark, and P. Radivojac. From protein-disease associations to disease informatics. *Frontiers in Bioscience*, 13:3391–3407, 2008.
- R. S. Datta, C. Meacham, B. Samad, C. Neyer, and K. Sjölander. Berkeley PHOG: PhyloFacts orthology group prediction web server. *Nucleic Acids Research*, 37(suppl 2):W84–W89, 2009.
- M. O. Dayhoff, R. M. Schwartz, and B. C. Orcutt. A model of evolutionary change in proteins. *Atlas of Protein Sequence and Structure*, 5:345–352, 1978.
- T. De Bie, L.-C. T. Tranchevent, L. Van Oeffelen, and Y. Moreau. Kernel-based data fusion for gene prioritization. *Bioinformatics*, 23(13):i125–i132, 2007.
- C. M. Deane, L. Salwin, I. Xenarios, and D. Eisenberg. Protein Interactions. *Molecular Cellular Proteomics*, 6(1):349–356, 2002.
- M. Deng, K. Zhang, S. Mehta, T. Chen, and F. Sun. Prediction of protein function using protein-protein interaction data. *Journal of Computational Biology*, 10(6):947–960, 2003.
- D. L. Des Marais and M. D. Rausher. Escape from adaptive conflict after duplication in an anthocyanin pathway gene. *Nature*, 454:762–765, 2008.
- D. Devos and A. Valencia. Practical limits of function prediction. *Proteins*, 41(1):98–107, 2000.
- M. M. Deza and E. Deza. *Encyclopedia of Distances*. Springer, 2 edition, 2013.
- K. A. Dill. Polymer principles and protein folding. *Protein Science*, 8(6):1166–1180, 1999.
- K. Dolinski and D. Botstein. Orthology and functional conservation in eukaryotes. *Annual Review of Genetics*, 41:465–507, 2007.
- R. F. Doolittle. *Of URFS and ORFS: A Primer on How to Analyze Derived Amino Acid Sequences*. University Science Books, 1986.
- A. K. Dunker, C. J. Brown, J. D. Lawson, L. M. Iakoucheva, and Z. Obradovic. Intrinsic disorder and protein function. *Biochemistry*, 41(21):6573–6582, 2002.
- H. J. Dyson and P. E. Wright. Intrinsically unstructured proteins and their functions. *Nature Reviews Molecular Cell Biology*, 6(3):197–208, 2005.
- J. A. Eisen. Phylogenomics: Improving functional predictions for uncharacterized genes by evolutionary analysis. *Genome Research*, 8:163–167, 1998.

- F. Enault, K. Suhre, and J.-M. Claverie. Phymbac “Gene Function Predictor”: a gene annotation tool based on genomic context analysis. *BMC Bioinformatics*, 6(257), 2005.
- B. E. Engelhardt, M. I. Jordan, K. E. Muratore, and S. E. Brenner. Protein molecular function prediction by Bayesian phylogenomics. *PLoS Computational Biology*, 1(5):e45, 2005.
- S. Fields and O. Song. A novel genetic system to detect protein-protein interactions. *Nature*, 340(6230):245–246, 1989.
- W. M. Fitch. Distinguishing homologous from analogous proteins. *Systematic Zoology*, 19(2):99–113, 1970.
- W. M. Fitch. Homology: a personal view on some of the problems. *Trends in Genetics*, 16(5):227–231, 2000.
- I. Friedberg. Automated protein function prediction—the genomic challenge. *Briefings in Bioinformatics*, 7(3):225–242, 2006.
- I. Friedberg, M. Jambon, and A. Godzik. New avenues in protein function prediction. *Protein Science*, 15(6):1527–1529, 2006.
- T. Gabaldon. Large-scale assignment of orthology: back to phylogenetics? *Genome Biology*, 9(10):235, 2008.
- T. Gabaldon, C. Dessimoz, J. Huxley-Jones, A. J. Vilella, E. L. L. Sonnhammer, and S. Lewis. Joining forces in the quest for orthologs. *Genome Biology*, 10(9):403, 2009.
- P. Gaudet, M. S. Livstone, S. E. Lewis, and P. D. Thomas. Phylogenetic-based propagation of functional annotations within the gene ontology consortium. *Briefings in Bioinformatics*, 12(5):449–462, 2011.
- A. Gersho and R. M. Gray. *Vector quantization and signal compression*, volume 159 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, 1992.
- G. Glazko et al. The choice of optimal distance measure in genome-wide datasets. *Bioinformatics*, 21(Suppl 3):iii3–iii11, 2005.
- N. Goldman and Z. H. Yang. Codon-based model of nucleotide substitution for protein-coding DNA sequences. *Molecular Biology and Evolution*, 11(5):725–736, 1994.
- M. Goodman, G. W. Moore, and G. Matsuda. Darwinian evolution in genealogy of hemoglobin. *Nature*, 253(5493):603–608, 1975.

- P. H. Guzzi et al. Semantic similarity analysis of protein data: assessment with biological features and issues. *Briefings in Bioinformatics*, 13(5):569–585, 2012.
- M. W. Hahn. Distinguishing among evolutionary models for the maintenance of gene duplicates. *Journal of Heredity*, 100:605–617, 2009.
- T. Hamp, R. Kassner, S. Seemayer, E. Vicedo, C. Schaefer, D. Achten, F. Auer, A. Boehm, T. Braun, M. Hecht, et al. Homology-based inference sets the bar high for protein function prediction. *BMC Bioinformatics*, 14(Suppl 3):S7, 2013.
- M. V. Han and M. W. Hahn. Identifying parent-daughter relationships among duplicated genes. *Pacific Symposium on Biocomputing*, 14:114–125, 2009.
- M. V. Han, J. P. Demuth, C. L. McGrath, C. Casola, and M. W. Hahn. Adaptive evolution of young duplicated genes in mammals. *Genome Research*, 19:859–867, 2009.
- G. T. Hart, A. K. Ramani, and E. M. Marcotte. How complete are current yeast and human protein-interaction networks? *Genome Biology*, 7(11):120, 2006.
- D. Haussler. Convolution kernels on discrete structures. Technical Report 1, University of California at Santa Cruz, 1999.
- T. Hawkins, S. Luban, and D. Kihara. Enhanced automated function prediction using distantly related sequences and contextual association by PFP. *Protein Science*, 15(6):1550–1556, 2006.
- S. Henikoff and J. G. Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences of the United States of America*, 89(22):10915–10919, 1992.
- J. C. Hermann, R. Marti-Arbona, A. A. Fedorov, E. Fedorov, S. C. Almo, B. K. Shoichet, and F. M. Raushel. Structure-based activity prediction for an enzyme of unknown function. *Nature*, 448(7155):775–779, 2007.
- H. Hermjakob, L. Montecchi-Palazzi, C. Lewington, S. Mudali, S. Kerrien, S. Orchard, M. Vingron, B. Roechert, P. Roepstorff, A. Valencia, H. Margalit, J. Armstrong, A. Bairoch, G. Cesareni, D. Sherman, and R. Apweiler. IntAct: an open source molecular interaction database. *Nucleic Acids Research*, 32(Database issue):D452–D455, 2004.
- T. Hulsen, M. A. Huynen, J. de Vlieg, and P. M. A. Groenen. Benchmarking ortholog identification methods using functional genomics data. *Genome Biology*, 7(4), 2006.
- L. Huminiecki and K. H. Wolfe. Divergence of spatial gene expression profiles following species-specific gene duplications in human and mouse. *Genome Research*, 14(10a):1870–1879, 2004.

- C. Huttenhower, M. Hibbs, C. Myers, and O. G. Troyanskaya. A scalable method for integration and functional analysis of multiple microarray datasets. *Bioinformatics*, 22(23):2890–2897, 2006.
- J. M. Izarzugaza, O. Grana, M. L. Tress, A. Valencia, and N. D. Clarke. Assessment of intramolecular contact predictions for casp7. *Proteins*, 69 Suppl 8:152–158, 2007.
- T. Jaakkola, M. Diekhans, and D. Haussler. Using the fisher kernel method to detect remote protein homologies. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, volume 149, page 158, 1999.
- T. Jaakkola, M. Diekhans, and D. Haussler. A discriminative framework for detecting remote protein homologies. *Journal of Computational Biology*, 7(1-2):95–114, 2000.
- P. Jaccard. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 37:574–579, 1901.
- C. J. Jeffery. Moonlighting proteins. *Trends in Biochemical Sciences*, 24(1):8–11, 1999.
- C. J. Jeffery. Molecular mechanisms for multitasking: recent crystal structures of moonlighting proteins. *Current Opinions in Structural Biology*, 14(6):663–668, 2004.
- C. J. Jeffery. Moonlighting proteins—an update. *Molecular Biosystems*, 5(4):345–350, 2009.
- L. J. Jensen, R. Gupta, N. Blom, D. Devos, J. Tamames, C. Kesmir, H. Nielsen, H. H. Staerfeldt, K. Rapacki, C. Workman, et al. Prediction of human protein function from post-translational modifications and localization features. *Journal of Molecular Biology*, 319(5):1257–1266, 2002.
- L. J. Jensen, R. Gupta, H. H. Staerfeldt, and S. Brunak. Prediction of human protein function according to gene ontology categories. *Bioinformatics*, 19(5):635–642, 2003.
- J. J. Jiang and D. W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *International Conference on Research in Computational Linguistics*, pages 19–33, 1997.
- C. E. Jones, J. Schwerdt, T. A. Bretag, U. Baumann, and A. L. Brown. Gosling: a rule-based protein annotator using blast and go. *Bioinformatics*, 24(22):2628–2629, 2008.
- W. Kabsch and C. Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2637, 1983.
- M. G. Kann. Protein interactions and disease: computational approaches to uncover the etiology of diseases. *Briefings in Bioinformatics*, 2007.
- M. Kimura. Was globin evolution very rapid in its early stages: a dubious case against the rate-constancy hypothesis. *Journal of Molecular Evolution*, 17(2):110–113, 1981.

- D. Koller and N. Friedman. *Probabilistic Graphical Models*. The MIT Press, Cambridge, Massachusetts, 2009.
- R. I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *Machine Learning International Workshop and Conference*, pages 315–322, 2002.
- E. V. Koonin. Orthologs, paralogs, and evolutionary genomics. *Annual Review of Genetics*, 39:309–338, 2005.
- Y. A. Kourmpetis, A. D. van Dijk, M. C. Bink, R. C. van Ham, and C. J. Ter Braak. Bayesian Markov Random Field analysis for protein function prediction based on network data. *PloS One*, 5(2):e9293, 2010.
- Y. A. Kourmpetis, A. D. van Dijk, and C. J. ter Braak. Gene ontology consistent protein function prediction: the falcon algorithm applied to six eukaryotic genomes. *Algorithms for Molecular Biology*, 8(1):10, 2013.
- R. Kuang, E. Ie, K. Wang, K. Wang, M. Siddiqi, Y. Freund, and C. Leslie. Profile-based string kernels for remote homology detection and motif extraction. In *Computational Systems Bioinformatics Conference, 2004. CSB 2004. Proceedings. 2004 IEEE*, pages 152–160. IEEE, 2004.
- A. Kuzniar, R. van Ham, S. Pongor, and J. A. M. Leunissen. The quest for orthologs: finding the corresponding gene across genomes. *Trends in Genetics*, 24(11):539–551, 2008.
- J. Kyte and R. F. Doolittle. A simple method for displaying the hydropathic character of a protein. *Journal of Molecular Biology*, 157(1):105–132, 1982.
- R. A. Laskowski and J. M. Thornton. Understanding the molecular machinery of genetics through 3D structures. *Nature Review Genetics*, 9(2):141–151, 2008.
- D. Lee, O. Redfern, and C. Orengo. Predicting protein function from sequence and structure. *Nature Reviews Molecular Cell Biology*, 8(12):995–1005, 2007.
- I. Lee, S. V. Date, A. T. Adai, and E. M. Marcotte. A probabilistic functional network of yeast genes. *Science*, 306(5701):1555–1558, 2004.
- C. Leslie, E. Eskin, and W. S. Noble. The spectrum kernel: a string kernel for SVM protein classification. *Pacific Symposium on Biocomputing*, 575(50):564–575, 2002.
- S. Letovsky and S. Kasif. Predicting protein function from protein/protein interaction data: a probabilistic approach. *Bioinformatics*, 19 Suppl 1:i197–204, 2003.

- A. C. Lewis, N. S. Jones, M. A. Porter, and C. M. Deane. What evidence is there for the homology of protein-protein interactions? *PLoS Computational Biology*, 8(9):e1002645, 2012.
- W. H. Li and T. Gojobori. Rapid evolution of goat and sheep globin genes following gene duplication. *Molecular Biology and Evolution*, 1(1):94–108, 1983.
- Li, Wen-Hsiung. *Molecular Evolution*. Sinauer Associates, Sunderland, Mass., 1997.
- B. Y. Liao and J. Zhang. Evolutionary conservation of expression profiles between human and mouse orthologous genes. *Molecular Biology and Evolution*, 23(3):530–540, 2006.
- B. Y. Liao and J. Z. Zhang. Null mutations in human and mouse orthologs frequently result in different phenotypes. *Proceedings of the National Academy of Sciences of the United States of America*, 105(19):6987–6992, 2008.
- D. Lin. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, pages 296–304. Morgan Kaufmann, 1998.
- Y. Linde, A. Buzo, and R. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28(1):84–95, 1980.
- K. Liolios, K. Mavromatis, N. Tavernarakis, and N. C. Kyrpides. The genomes on line database (gold) in 2007: status of genomic and metagenomic projects and their associated metadata. *Nucleic Acids Research*, 36(Database issue):D475–479, 2008.
- P. W. Lord et al. Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation. *Bioinformatics*, 19(10):1275–1283, 2003.
- M. Lynch and J. S. Conery. The evolutionary fate and consequences of duplicate genes. *Science*, 290(5494):1151–1155, 2000.
- K. D. Makova and W. H. Li. Divergence in the spatial pattern of gene expression between human duplicate genes. *Genome Research*, 13(7):1638–1645, 2003.
- E. M. Marcotte, M. Pellegrini, H.-L. Ng, D. W. Rice, T. O. Yeates, and D. Eisenberg. Detecting protein function and protein-protein interactions from genome sequences. *Science*, 285(5428):751–753, 1999.
- D. M. Martin et al. GOtcha: a new method for prediction of protein function assessed by the annotation of seven genomes. *BMC Bioinformatics*, 5:178, 2004.
- S. Mika and B. Rost. Protein-protein interactions more conserved within species than across species. *PLoS Computational Biology*, 2(7):e79, 2006.

- R. G. Millikan. In defense of proper functions. *Philosophy of Science*, pages 288–302, 1989.
- F. Minneci, D. Piovesan, D. Cozzetto, and D. T. Jones. Ffpred 2.0: Improved homology-independent prediction of gene ontology terms for eukaryotic protein sequences. *PLoS One*, 8(5):e63754, 2013.
- A. Mohan, V. N. Uversky, and P. Radivojac. Influence of sequence changes and environment on intrinsically disordered proteins. *PLoS Computational Biology*, 5(9):e1000497, 2009.
- H. J. Muller. Bar duplication. *Science*, 83(2161):528–530, 1936.
- A. G. Murzin, S. E. Brenner, T. Hubbard, C. Chothia, and Others. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, 247(4):536–540, 1995.
- E. Nabieva, K. Jim, A. Agarwal, B. Chazelle, and M. Singh. Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinformatics*, 21 Suppl 1:i302–310, 2005.
- N.-I. NCotIUoBaMB. *Enzyme nomenclature*. Academic Press, New York, 1992.
- N. L. Nehrt, W. T. Clark, P. Radivojac, and M. W. Hahn. Testing the ortholog conjecture with comparative functional genomic data from mammals. *PLoS Computational Biology*, 7(6):e1002073, 2011.
- W. S. Noble. Support Vector Machine Applications in Computational Biology. In B. Schölkopf, K. Tsuda, and J.-P. Vert, editors, *Kernel methods in computational biology*, chapter 3, pages 71–92. The MIT Press, 2004.
- I. M. A. Nooren and J. M. Thornton. Diversity of protein-protein interactions. *The European Molecular Biology Organization Journal*, 22(14):3486–3492, 2003.
- S. Ohno. *Evolution by Gene Duplication*. Springer-Verlag, 1970.
- D. Pal and D. Eisenberg. Inference of protein function from protein structure. *Structure*, 13(1):121–130, 2005.
- F. Pazos and M. J. Sternberg. Automated prediction of protein function and detection of functional sites from structure. *Proceedings of the National Academy of Sciences of the United States of America*, 101(41):14754–14759, 2004.
- M. Pellegrini, E. M. Marcotte, M. J. Thompson, D. Eisenberg, and T. O. Yeates. Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *Proceedings of the National Academy of Sciences of the United States of America*, 96(8):4285–4288, 1999.

- K. Peng, P. Radivojac, S. Vucetic, A. K. Dunker, and Z. Obradovic. Length-dependent prediction of protein intrinsic disorder. *BMC Bioinformatics*, 7(1):208, 2006.
- C. Pesquita et al. Semantic similarity in biomedical ontologies. *PLoS Computational Biology*, 5(7):e1000443, 2009.
- M. Punta and Y. Ofran. The rough guide to in silico function prediction, or how to use sequence and structure information to predict protein function. *PLoS Computational Biology*, 4(10):e1000160, 2008.
- J. Qiu, M. Hue, A. Ben-Hur, J.-P. Vert, and W. S. Noble. A structural alignment kernel for protein structures. *Bioinformatics*, 23(9):1090–1098, 2007.
- R. Rada et al. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(1):17–30, 1989.
- P. Radivojac, Z. Obradovic, D. K. Smith, G. Zhu, S. Vucetic, C. J. Brown, J. D. Lawson, and A. K. Dunker. Protein flexibility and intrinsic disorder. *Protein Science*, 13(1):71–80, 2004.
- P. Radivojac, L. M. Iakoucheva, C. J. Oldfield, Z. Obradovic, V. N. Uversky, and A. K. Dunker. Intrinsic disorder and functional proteomics. *Biophysical Journal*, 92(5):1439–1456, 2007.
- P. Radivojac, K. Peng, W. T. Clark, B. J. Peters, A. Mohan, S. M. Boyle, and S. D. Mooney. An integrated approach to inferring gene-disease associations in humans. *Proteins, Structure, Function and Bioinformatics*, 72(3):1030–1037, 2008.
- P. Radivojac, W. T. Clark, et al. A large-scale evaluation of computational protein function prediction. *Nature Methods*, 10(3):221–227, 2013.
- L. Ralaivola, S. J. Swamidass, H. Saigo, P. Baldi, et al. Graph kernels for chemical informatics. *Neural Networks*, 18(8):1093–1110, 2005.
- M. Remm, C. E. Storm, and E. L. Sonnhammer. Automatic clustering of orthologs and in-paralogs from pairwise species comparisons. *Journal of Molecular Biology*, 314(5):1041–1052, 2001.
- R. Rentzsch and C. Orengo. Protein function prediction—the power of multiplicity. *Trends in Biotechnology*, 27(4):210–219, 2009.
- P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 448–453, 1995.
- P. Resnik. Semantic similarity in a taxonomy: an information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130,

- 1999.
- M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: the rprop algorithm. In *International Conference on Neural Networks*, volume 1, pages 586 – 591. IEEE, 1993.
- P. N. Robinson and S. Bauer. *Introduction to bio-ontologies*. CRC Press, Boca Raton, Florida, U.S.A., 2011.
- P. N. Robinson and S. Mundlos. The human phenotype ontology. *Clinical Genetics*, 77(6):525–534, 2010.
- M. F. Rogers and A. Ben-Hur. The use of gene ontology evidence codes in preventing classifier assessment bias. *Bioinformatics*, 25(9):1173–1177, 2009.
- B. Rost. Twilight zone of protein sequence alignments. *Protein Engineering*, 12(2):85–94, 1999.
- B. Rost, J. Liu, R. Nair, K. O. Wrzeszczynski, and Y. Ofra. Automatic prediction of protein function. *Cellular and Molecular Life Sciences*, 60(12):2637–2650, 2003.
- A. Ruepp, A. Zollner, D. Maier, K. Albermann, J. Hani, M. Mokrejs, I. Tetko, U. Guldener, G. Mannhaupt, M. Munsterkotter, and H. W. Mewes. The funcat, a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucleic Acids Research*, 32(18):5539–5545, 2004.
- R. Saeed and C. Deane. An assessment of the uses of homologous interactions. *Bioinformatics*, 24(5):689–95, Mar. 2008.
- A. Schlicker et al. A new measure for functional similarity of gene products based on Gene Ontology. *BMC Bioinformatics*, 7:302, 2006.
- A. M. Schoes, S. D. Brown, I. Dodevski, and P. C. Babbitt. Annotation error in public databases: Misannotation of molecular function in enzyme superfamilies. *PLoS Computational Biology*, 5(12):e1000605, 2009.
- B. Schölkopf, K. Tsuda, and J.-P. Vert, editors. *Kernel methods in computational biology*. The MIT Press, 2004.
- L. M. Schriml, C. Arze, S. Nadendla, Y.-W. W. Chang, M. Mazaitis, V. Felix, G. Feng, and W. A. Kibbe. Disease ontology: a backbone for disease semantic integration. *Nucleic Acids Research*, 40(D1):D940–D946, 2012.

- B. Sennblad and J. Lagergren. Probabilistic orthology analysis. *Systematic Biology*, 58(4):411–424, 2009.
- R. Sharan et al. Network-based prediction of protein function. *Molecular Systems Biology*, 3:88, 2007.
- J. Shawe-Taylor and N. Cristianini, editors. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.
- G. Shi, L. Zhang, and T. Jiang. MSOAR 2.0: Incorporating tandem duplications into ortholog assignment based on genome rearrangement. *BMC Bioinformatics*, 11(1):10, 2010.
- N. Škunca, A. Altenhoff, and C. Dessimoz. Quality of computationally inferred gene ontology annotations. *PLoS Computational Biology*, 8(5):e1002533, 2012.
- N. Škunca, M. Bošnjak, A. Kriško, P. Panov, S. Džeroski, T. Šmuc, and F. Supek. Phyletic profiling with cliques of orthologs is enhanced by signatures of paralogy relationships. *PLoS Computational Biology*, 1553:734X, 2013.
- A. Sokolov and A. Ben-Hur. Hierarchical classification of gene ontology terms using the gostruct method. *Journal of Bioinformatics and Computational Biology*, 8(2):357–376, 2010.
- E. L. L. Sonnhammer and E. V. Koonin. Orthology, paralogy and proposed classification for paralog subtypes. *Trends in Genetics*, 18(12):619–620, 2002.
- E. Sprinzak, S. Sattath, and H. Margalit. How reliable are experimental protein-protein interaction data? *Journal of Molecular Biology*, 327(5):919–923, 2003.
- S. Stephens. Possible significance of duplication in evolution. *Advances in Genetics*, 4:247–265, 1951.
- G. Strang. *Introduction to Linear Algebra*. Wellsley-Cambridge Press, 3 edition, 2003.
- R. A. Studer and M. Robinson-Rechavi. How confident can we be that orthologs are similar, but paralogs differ? *Trends in Genetics*, 25(5):210–216, 2009.
- R. A. Studer, S. Penel, L. Duret, and M. Robinson-Rechavi. Pervasive positive selection on duplicated and nonduplicated vertebrate protein coding genes. *Genome Research*, 18(9):1393–1402, 2008.
- A. I. Su, T. Wiltshire, S. Batalov, H. Lapp, K. A. Ching, D. Block, J. Zhang, R. Soden, M. Hayakawa, G. Kreiman, M. P. Cooke, J. R. Walker, and J. B. Hogenesch. A gene atlas of the mouse and human protein-encoding transcriptomes. *Proceedings of the National Academy of Sciences of the United States of America*, 101(16):6062–6067, 2004.

- R. L. Tatusov, E. V. Koonin, and D. J. Lipman. A genomic perspective on protein families. *Science*, 278(5338):631–637, 1997.
- J. A. Tennessen. Positive selection drives a correlation between non-synonymous/synonymous divergence and functional divergence. *Bioinformatics*, 24(12):1421–1425, 2008.
- W. Tian and J. Skolnick. How well is enzyme function conserved as a function of pairwise sequence identity? *Journal of Molecular Biology*, 333(4):863–882, 2003.
- A. E. Todd, C. A. Orengo, and J. M. Thornton. Evolution of function in protein superfamilies, from a structural perspective. *Journal of Molecular Biology*, 307(4):1113–1143, 2001.
- O. G. Troyanskaya, K. Dolinski, A. B. Owen, R. B. Altman, and D. Botstein. A bayesian framework for combining heterogeneous data sources for gene function prediction (in *saccharomyces cerevisiae*). *Proceedings of the National Academy of Sciences of the United States of America*, 100(14):8348–8353, 2003.
- T. Tuytelaars and C. Schmid. Vector Quantizing Feature Space with a Regular Lattice. In *11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.
- M. Tyagi, R. R. Thangudu, D. Zhang, S. H. Bryant, T. Madej, and A. R. Panchenko. Homology Inference of Protein-Protein Interactions via Conserved Binding Sites. *PLoS ONE*, 7(1):e28896, 2012.
- V. Vacic, L. M. Iakoucheva, S. Lonardi, and P. Radivojac. Graphlet kernels for prediction of functional residues in protein structures. *Journal of Computational Biology*, 17(1):55–72, 2010.
- A. Vazquez, A. Flammini, A. Maritan, and A. Vespignani. Global protein function prediction from protein-protein interaction networks. *Nature Biotechnology*, 21(6):697–700, 2003.
- K. Verspoor et al. A categorization approach to automated ontological function annotation. *Protein Science*, 15(6):1544–1549, 2006.
- J.-P. Vert. A tree kernel to analyse phylogenetic profiles. *Bioinformatics*, 18(1):S276–S284, 2002.
- A. J. Vilella, J. Severin, A. Ureta-Vidal, L. Heng, R. Durbin, and E. Birney. Ensembl Compara GeneTrees: Complete, duplication-aware phylogenetic trees in vertebrates. *Genome Research*, 19(2):327–335, 2009.
- M. N. Wass and M. J. Sternberg. ConFunc—functional annotation in the twilight zone. *Bioinformatics*, 24(6):798–806, 2008.

- C. A. Wilson, J. Kreychman, and M. Gerstein. Assessing annotation transfer for genomics: quantifying the relations between protein sequence, structure and function through traditional and probabilistic scores. *Journal of Molecular Biology*, 297(1):233–249, 2000.
- F. Xin and P. Radivojac. Computational methods for identification of functional residues in protein structures. *Current Protein Peptide Science*, 120(6):456–69, 2011.
- H. Yu, N. M. Luscombe, H. X. Lu, X. Zhu, Y. Xia, J.-D. J. Han, N. Bertin, S. Chung, M. Vidal, and M. Gerstein. Annotation transfer between genomes: protein-protein interologs and protein-DNA regulogs. *Genome Research*, 14(6):1107–1118, 2004.
- C. M. Zmasek and S. R. Eddy. RIO: analyzing proteomes by automated phylogenomics using resampled inference of orthologs. *BMC Bioinformatics*, 3(1):14, 2002.
- E. Zuckerkandl and L. Pauling. Evolutionary divergence and convergence in proteins. In V. Bryson and H. J. Vogel, editors, *Evolving Genes and Proteins*, pages 97–166. Academic Press, New York, 1965.