

**Prediction of Protein Function with a Probabilistic Model for  
Analysis of Sequence Similarity Networks and Genomic Context**

by

Jeffrey Michael Yunes

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

BIOENGINEERING

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, SAN FRANCISCO

and

UNIVERSITY OF CALIFORNIA, BERKELEY

Prediction of Protein Function with a Probabilistic Model for  
Analysis of Sequence Similarity Networks and Genomic Context

by

Jeffrey Michael Yunes

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

BIOENGINEERING

in the

GRADUATE DIVISION

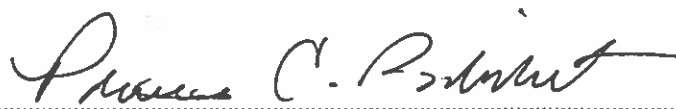
of the

UNIVERSITY OF CALIFORNIA, SAN FRANCISCO

and

UNIVERSITY OF CALIFORNIA, BERKELEY

Approved:



---



---



---

Committee in Charge

Copyright 2018  
by  
Jeffrey Michael Yunes

## Abstract for Nonscientists

A genome is a blueprint for an organism. Written in an alphabet of four biochemical letters, the genome consists of genes and other structural elements. These genes have several purposes, but sometimes they are processed to become proteins. Proteins have many roles, but those that act as molecular machines are called enzymes. Enzymes can have many different functions; for example, one class of enzymes breaks apart molecules, and another class combines molecules.

In 1995, scientists determined the first complete genome of an organism, meaning they determined the sequence of all of the letters in the organism's genome. In 2001, the Human Genome Project produced a draft of the human genome. Scientists used algorithms to delineate the different genes, by looking for patterns or signals that suggest certain parts will be processed proteins and other biomolecules. Scientists have now completed the sequencing of thousands of other genomes, resulting in millions of genes and protein sequences. However, experiments to determine a protein's function are very difficult, and therefore very little is known about most proteins.

The goal of this project is to suggest functions for each of these millions of uncharacterized proteins, so they can be used by scientists and software that depend on the information. The general approach is to describe knowledge as a probabilistic model, and use algorithms to determine which functions for the protein of interest are likely given that model. The knowledge we incorporate is based on the two analytical techniques from computational biology, and can be summarized as (1) proteins that are similar in sequence should have similar molecular functions, and (2) proteins that are functionally associated are probably involved in similar processes.

In addition to describing a complete method for prediction of protein function based on this model, this dissertation evaluates the method as a whole and in part, and discusses insights and avenues for further research.

## Abstract

The number of known protein sequences is growing faster than the number of curated protein functions. To help bridge this gap, bioinformatics scientists have created automated methods for the prediction of protein function. Recently, the focus has been on integrating numerous data sources, and critical evaluation of these methods show that the integrative approach improves predictive performance. However, a basic BLAST-based method is still a top contender.

Computational biologists often use two complimentary approaches to infer functions that are usually more accurate than a BLAST-based method. Analysis of sequence similarity networks can dissect protein functions in a superfamily and infer the function of individual proteins. Briefly, a computational biologist will create a network of proteins in sequence space, which typically shows clusters of similar proteins. She will then highlight which few of these proteins have experimental functional annotations, and paint the network according to other functional features that are broadly available, such as residues in key positions in an alignment. These data are used to identify proteins where a functional change may have occurred, which then can be used to delineate protein families or other protein groups that share a specific function or functional characteristic. However, molecular functional annotation data are very scarce, and there is not enough of it to draw functional boundaries with high confidence.

The second method, analysis of genomic context, is often done in conjunction with sequence similarity network analysis. This approach uses data about the genome neighbors of a protein, or more generally, any functional association data, such protein – protein interaction data, to predict a protein’s molecular function. This technique has been used to refine functional boundaries during sequence similarity network analysis, as well as to generate hypothesis in the absence of characterization of any close homologs.

In this dissertation, I describe *Effusion*, our attempt to automate sequence similarity network analysis and improve on the current methods for the prediction of protein function. *Effusion* modernizes the classical BLAST-based approach while avoiding pitfalls common to state-of-the-art methods. It uses a sequence similarity network to add context for homology transfer, a probabilistic model to account for the uncertainty in labels and function propagation, and the structure of the Gene Ontology (GO) to best utilize sparse input labels and make consistent output predictions. *Effusion*’s model makes it practical to integrate rare experimental data with the abundant primary sequence and sequence similarity data. Our model allows for inference with general purpose, state-of-the-art inference algorithms, makes use of all experimental annotation data, has parameters specific to each GO term, and adds data-derived pseudocounts to predict rare terms.

*Effusion GCA* extends *Effusion* by integrating the chief components necessary for automating genomic context analysis. It performs its analysis over a sequence similarity – functional association network, with a model of protein function that includes a representa-

tion of each protein's biological process, performs simultaneous inference on multiple aspects of protein function, and only propagates functional information where it is appropriate.

We assessed our methods using a critical evaluation method and metrics. The results show that Effusion outperforms standard prediction methods, the most similar prediction methods, and state-of-the-art prediction methods. Effusion GCA does not perform as well as Effusion in aggregate, but offered several other insights. We conclude that these methods represent a significant progress in the field of protein function prediction, and clearly suggest avenues for further advance.

To Mom and Dad, and my brothers, Scott and Jonathan

The more I neglected them, the more they supported me. They should expect a mailbox full of belated birthday cards.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background for protein function prediction . . . . .	2
1.1.1	Proteins . . . . .	2
1.1.2	Functions . . . . .	3
1.1.3	Annotations and predictions . . . . .	4
1.1.4	Challenges for prediction of protein function . . . . .	4
1.1.5	Methods for automated function prediction . . . . .	5
1.1.6	Evaluation of methods for automated function prediction . . . . .	7
1.2	Review of topical mathematics . . . . .	16
1.2.1	Probability . . . . .	16
1.2.2	Parameters and statistics . . . . .	17
1.2.3	Probabilistic graphical models (PGMs) . . . . .	18
<b>2</b>	<b>Effusion: Prediction of Protein Function with a Probabilistic Model for Analysis of Sequence Similarity Networks</b>	<b>20</b>
2.1	Abstract . . . . .	20
2.1.1	Motivation . . . . .	20
2.1.2	Results . . . . .	21
2.2	Introduction . . . . .	21
2.2.1	Sequence similarity . . . . .	22
2.2.2	Sequence similarity networks (SSNs) . . . . .	22
2.2.3	Related work . . . . .	22
2.3	Methods . . . . .	23
2.3.1	Preprocessing . . . . .	24
2.3.2	Building the protein network . . . . .	26
2.3.3	Constructing a tractable probabilistic graphical model . . . . .	27
2.3.4	Parameter learning . . . . .	30
2.3.5	Information content of GO terms . . . . .	33
2.3.6	Inference . . . . .	33
2.3.7	Post-processing . . . . .	34



2.3.8	Evaluation . . . . .	34
2.4	Results . . . . .	37
2.4.1	Comparative analysis . . . . .	38
2.4.2	Creation and use of protein networks . . . . .	43
2.4.3	Embedding GO . . . . .	45
2.4.4	Data-derived alterations of parameters . . . . .	46
2.4.5	Inference on real-world protein data . . . . .	47
2.5	Discussion . . . . .	48
2.5.1	Comparisons to other methods . . . . .	50
2.5.2	Directions for future research . . . . .	51
<b>3</b>	<b>Effusion GCA: Prediction of Protein Function with a Probabilistic Model for Analysis of Genomic Context</b>	<b>53</b>
3.1	Abstract . . . . .	53
3.1.1	Motivation . . . . .	53
3.1.2	Results . . . . .	53
3.2	Introduction . . . . .	54
3.2.1	Genomic context analysis . . . . .	55
3.2.2	Functional associations . . . . .	55
3.2.3	Functional associations networks . . . . .	57
3.2.4	Quantification of association between aspects of function . . . . .	58
3.2.5	Related work . . . . .	58
3.3	Methods . . . . .	59
3.3.1	Preprocessing . . . . .	59
3.3.2	Building the sequence similarity – functional association (SSFA) network	59
3.3.3	Constructing the probabilistic model . . . . .	60
3.3.4	Parameter learning . . . . .	63
3.4	Results . . . . .	64
3.4.1	Comparative analysis . . . . .	64
3.4.2	Creation and use of SSFA networks . . . . .	69
3.4.3	Parameters for functional association edges . . . . .	71
3.4.4	Protein template . . . . .	75
3.5	Discussion . . . . .	80
3.5.1	Comparative analysis . . . . .	80
3.5.2	Creation and use of SSFA networks . . . . .	81
3.5.3	Functional associations . . . . .	82
3.5.4	Inter-ontological parameters . . . . .	82
3.5.5	Alternative representations . . . . .	82
3.5.6	Other contributions . . . . .	84
3.5.7	Directions for future research . . . . .	84

<b>4 Discussion and conclusions</b>	<b>85</b>
4.1 Contributions . . . . .	85
4.2 Failed attempts . . . . .	86
4.3 Limitations . . . . .	88
4.4 Directions for future research . . . . .	89
<b>Bibliography</b>	<b>92</b>
<b>A Supplementary material for Chapter 2</b>	<b>104</b>
<b>B Supplementary material for Chapter 3</b>	<b>132</b>
<b>C Code</b>	<b>138</b>

# List of Figures

1.1	Relative rates of growth of protein sequences and non-electronic annotations . . .	2
2.2	Graphical Summary of Effusion, using Cytochrome P450 CYP1C1 (UniProt Q4ZIL6) as an example . . . . .	26
2.3	Performance plots over all proteins in the test set, regardless of whether any of the methods failed to make predictions . . . . .	39
2.4	Performance plots over treated proteins . . . . .	40
2.5	Performance on catalytic terms . . . . .	42
3.2	GCA case study . . . . .	57
3.4	GCA model . . . . .	61
3.5	Performance plots over all proteins in the test set, regardless of whether any of the methods failed to make predictions . . . . .	65
3.6	Performance plots over treated proteins . . . . .	66
3.7	Performance on catalytic terms . . . . .	67
3.8	SSFA network for UniProt P78334 . . . . .	69
3.9	Reduced SSFA network for UniProt P78334 . . . . .	70
3.10	Distribution of time to build the network . . . . .	72
3.11	Protein template for Effusion GCA . . . . .	76
3.12	Performance plots over treated proteins showing the change in performance due to changes in the protein template for the top-down model . . . . .	77
3.13	Performance plots over treated proteins showing the change in performance due to changes in the protein template for the bottom-up model . . . . .	78
3.14	Histogram of the mean correlation in the protein template . . . . .	79
4.1	Probability of certain enzyme classes given cellular location . . . . .	90
4.2	A finer grained representation of sequence similarity to propagate function more carefully . . . . .	91
A.1	Distribution of time to build the network . . . . .	111
A.2	Distribution of number of nodes in protein network . . . . .	112

A.3	Distribution of number of nodes in protein network after directing it, rooting it, and pruning it . . . . .	113
A.4	Distribution of number of nodes in protein network with positive evidence . . .	114
A.5	Performance curves showing value of adding unannotated proteins . . . . .	115
A.6	Distributions for the maximum number of parents or modeled children in the protein template . . . . .	116
A.7	Performance curves showing value of adding negative evidence . . . . .	117
A.9	Example comparing top-down model, without and with supplementary negative evidence, and bottom-up model . . . . .	119
A.10	Performance curves showing value of weighting counts when computing the parameters . . . . .	126
A.11	A network view of an example, with and without weighting . . . . .	127
A.12	Performance curves showing value of adding pseudocounts to the contingency tables	128
A.13	Network view of predictions for UniProt Q6UWY2 without using pseudocounts, colored by GO:1901681 . . . . .	129
A.14	Legend for Figure A.15 . . . . .	130
A.15	Plots showing the performance of different inference engines on the top-down model and the bottom-up model . . . . .	131
B.1	Performance plot comparing Effusion GCA (top-down, ai_cond) and Effusion (top-down, ai_cond) on UniProt P78334 . . . . .	133

# List of Tables

2.1	Raw contingency table for GO:0016790 . . . . .	30
2.2	Weighted contingency table for GO:0016790 . . . . .	31
2.3	Contingency table for GO:0016790, with pseudocounts . . . . .	32
2.4	Statistics for the time to build the sequence similarity networks . . . . .	44
3.1	Statistics for the time to build the SSFA networks . . . . .	70
3.2	Statistics for the number of functionally associated proteins in the full SSFA network and in the reduced SSFA network . . . . .	70
3.3	Statistics for the number of functionally associated proteins in the full SSFA network and in the reduced SSFA network . . . . .	73
3.4	Mitrofanova parameters used for problem instance based on query UniProt P78334	74
3.5	inter-ontological associations modeled for problem instance based on UniProt P78334 . . . . .	79
A.1	Predictions for Q4ZIL6 made by Effusion . . . . .	105
A.2	Predictions for Q4ZIL6 made by BLAST . . . . .	108
A.3	Predictions for Q921C5 made by Effusion with the top-down model, but without supplementary negative evidence . . . . .	120
A.4	Predictions for Q921C5 made by Effusion with the top-down model, including supplementary negative evidence . . . . .	122
A.5	Predictions for Q921C5 made by Effusion with the bottom-up model . . . . .	124
A.6	Table of inference algorithms . . . . .	130
B.1	Predictions for P78334 made by Effusion GCA (top_down, ai_cond) . . . . .	134
B.2	Predictions for P78334 made by Effusion (top-down, ai_cond) . . . . .	136

# Abbreviations, Etc.

**AUC** area under the curve

**BP** biological process

**CAFA** Critical Assessment of Function Annotation

**CC** cellular component

**DAG** directed acyclic graph

**FAMF** functionally-associated molecular functions

**GO** Gene Ontology

**GO:0003735** structural constituent of ribosome

**GO:0005231** excitatory extracellular ligand-gated ion channel activity

**GO:0005488** binding

**GO:0005515** protein binding

**GO:0006801** superoxide metabolic process

**GO:0008152** metabolic process

**GO:0008395** steroid hydroxylase activity

**GO:0008404** arachidonic acid 14,15-epoxygenase activity

**GO:0009987** cellular process

**GO:0015108** chloride transmembrane transporter activity

**GO:0016051** carbohydrate biosynthetic process

**GO:0016740** transferase activity

**GO:0016787** hydrolase activity

**GO:0016788** hydrolase activity, acting on ester bonds

**GO:0016790** thiolester hydrolase activity

**GO:0016813** hydrolase activity, acting on carbon-nitrogen (but not peptide) bonds, in linear amidines

**GO:0017137** Rab GTPase binding

**GO:0019899** enzyme binding

**GO:0030276** clathrin binding

**GO:0044877** macromolecular complex binding

**GO:0047465** N-acylmannosamine-6-phosphate 2-epimerase

**GO:0050649** testosterone 6-beta-hydroxylase activity

**GO:1901681** sulfur compound binding

**GO:1905741** calcium export from the mitochondrion involved in positive regulation of presynaptic cytosolic calcium concentration

**GOA** Gene Ontology Annotation

**GOC** Gene Ontology Consortium

**IBE** ignore BP evidence

**IC** information content

**IDA** Inferred from Direct Assay

**IEA** Inferred from Electronic Annotation

**MF** molecular function

**MRCA** most recent common ancestor

**MRF** Markov random field

**MST** minimum spanning tree

**PGM** probabilistic graphical model

**PGN** Pfams in genome neighborhood

**SSFA** sequence similarity – functional association

**SSN** sequence similarity network

**SW-WRc** sample-weighted weighted recall

**SW-WPr** sample-weighted weighted precision

**UniProt Q96Q83** Alpha-ketoglutarate-dependent dioxygenase alkB homolog 3

**UniProt P78334** gamma-aminobutyric acid receptor subunit epsilon

**UniProt Q6UWY2** Serine protease 57

**UniProt P34945** Serine-tRNA ligase

**UniProt Q4ZIL6** Cytochrome P450 CYP1C1

**WFP** weighted false positive

**WPr** weighted precision

**WRc** weighted recall

**WTP** weighted true positive



## Acknowledgments

This dissertation was supervised by Prof. Patricia Babbitt. Patsy was an outstanding scientific advisor and role model. She prioritized my educational goals and research interests, and helped me navigate several difficult situations. She had a good eye for identifying people that would work well together, fostered a lab environment where I was comfortable asking questions and making mistakes, and gave me exceptional mentoring. It was an honor to be in her laboratory.

I thank my labmates: Dr. Eyal Akiva, Dr. Benjamin Polacco, Dr. Shoshana Brown, Mr. Doug Stryke, Dr. Elaine Meng, Dr. Michael Hicks, Dr. Susan Mashiyama, Dr. Alan Barber II, Dr. Alexandra Schnoes, and Mr. David Mischel. Their diverse areas of expertise and different approaches to scientific exploration contributed greatly to my experience and this work.

I thank Prof. Hao Li for serving on my dissertation committee and chairing my qualifying examination, and for hosting the weekly Biology of Aging seminar. I thank Prof. John Huelsenbeck for his wonderful course on statistical phylogenetics, the knowledge of which I have applied in my projects and shared with many others. I also thank him for serving on my dissertation and qualifying examination committees. I learned a lot as Graduate Student Instructor for Prof. Kimmen Sjölander's class on protein informatics. She recommended I take a course on probabilistic graphical models, which became the essential component of this dissertation. She also served on my qualifying examination. I am grateful to Prof. Ian Holmes for the role he played in my academic advising, and his excellent course on evolutionary models. I want to thank Prof. Steven Brenner for shaping my early scientific mindset. He often challenged my assumptions, and was usually correct.

I thank Dr. Ariel Jaimovich, Dr. Yutian Chen, Prof. Jennifer Neville, and Dr. Lei Li

for helpful discussions on probabilistic graphical models.

I thank Mr. Matisse Hack, Mr. William Skinner, Ms. Carly Weiss, Greg Michael, Esq., and Mr. Byron Lee for interesting discussions, technical help, and encouragement.

I recognize Mrs. SarahJane Taylor, Mrs. Karen Denton, Mrs. Laurie Roach, and Doug Carlson, Esq. for exceeding their administrative duties.

This dissertation is dedicated to my family, but they also deserve to be acknowledged for their contributions. Mrs. Amy Yunes, Dr. Sheldon Yunes, Dr. Scott Yunes, Jonathan Yunes, Esq., and Scarface were the first ones I turned to every time I was stuck on a problem, and they always listened intently and offered suggestions and support. I am glad we stayed in such frequent contact while I was away from home.

# Chapter 1

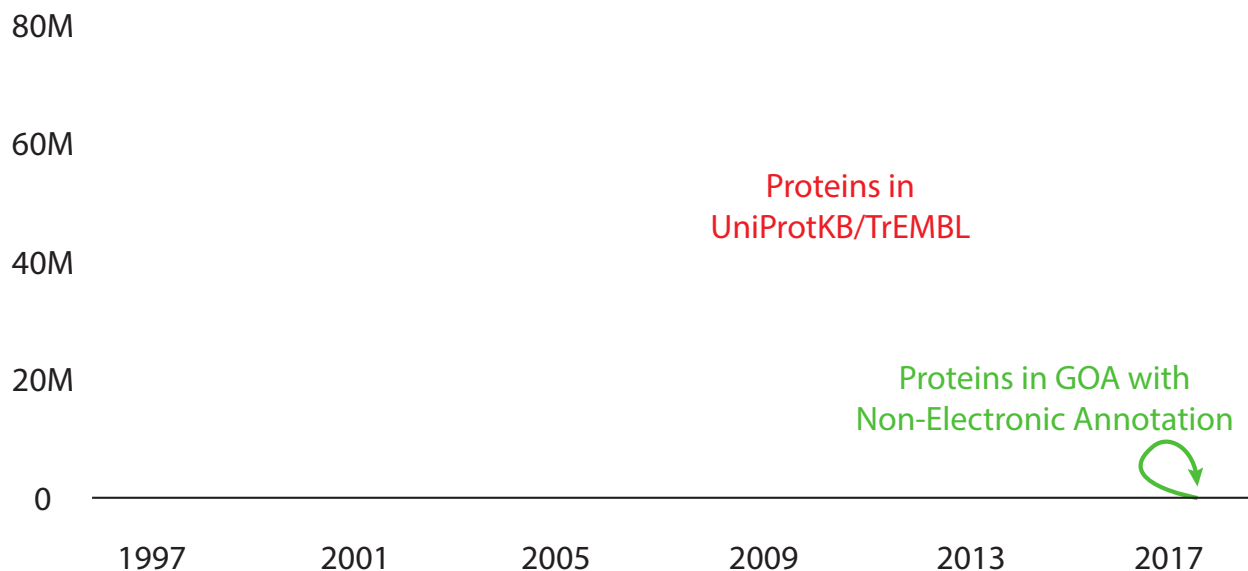
## Introduction

Determining the function of gene products is necessary for understanding life, valuable for applications in health and industry, and a foundational problem in bioinformatics. The number of protein sequences, now greatly amplified by metagenomic sequencing projects, continues to grow at a much faster rate than the number of proteins with experimentally determined or manually curated functions (Figure 1.1). As a result, computational prediction of protein function is needed more than ever.

The task of predicting protein function is, given no constraints on the data that can be used, to identify, rank, or score functional terms that have been withheld or can be experimentally validated. In this work, we are more interested in distinguishing specific molecular activities, rather than predicting general categories of protein function.

This dissertation presents two methods for the prediction of protein function. Both of these methods share the following principles of design. First of all, each method is motivated by an analytical technique that is performed manually by computational biologists. The first technique is the analysis of sequence similarity networks, and is described in Chapter 2. The second technique is the analysis of genomic context, and is described in Chapter 3.

Second, we design our models to treat the rare, valuable data very carefully, and we try



**Figure 1.1:** Relative rates of growth of protein sequences and non-electronic annotations. The drop of proteins in TrEMBL is due to a switch to reference proteomes. The shape of the green arrow is for visibility.

to avoid dilution when incorporating large quantities of less informative data.

Third, we model aspects of protein function probabilistically and in depth, because they are plagued with uncertainty, rich with information, and the objects of interest. We embed our knowledge about how proteins and functions interrelate, and then use our model to infer functions that conform to our model.

## 1.1 Background for protein function prediction

### 1.1.1 Proteins

As of 17 March 2017, UniProtKB [88] includes 80,758,400 protein sequences. The growth of UniProtKB is shown in Figure 1.1.

### 1.1.2 Functions

GO represents protein function using a controlled vocabulary of terms. It models three distinct aspects of protein function: molecular function, biological process, and cellular component. Each GO term has a name and an identifier. For example, GO term N-acylmannosamine-6-phosphate 2-epimerase is identified by GO:0047465. There are currently 48,532 terms defined by GO.

GO terms are organized hierarchically; more general activities are ancestors of more specific activities. For example, isomerase activity is a child of catalytic activity, and is the parent of racemase and epimerase activity. *Root* GO terms have no parents, and are the most general, such as molecular function. *Leaf* GO terms have no children, and are the most specific, such as GO:0047465. It is more common for an internal node to have more children than parents, but GO terms often have multiple parents. For example, cellular metabolic process has two parents: cellular process and metabolic process. The level of a GO term, defined as the minimum path length from root to the GO term, ranges from 0 – 12. The depth of a GO term, defined as the maximum path length from root to the GO term, ranges from 0 – 17. There is only one GO term at depth 17: calcium export from the mitochondrion involved in positive regulation of presynaptic cytosolic calcium concentration (GO:1905741).

We focus on predicting *molecular function*, which is defined as “the biochemical activity (including specific binding to ligands or structures) of a gene product” [5]. There are 10,885 GO terms in the molecular function ontology, 8,799 of which are leaves.

There are additional features of GO, and caveats for using it. It is continuously changing. There are other types of relationships besides that between a parent and a child. Please visit the resource for more information.

A multi-label is a mapping from GO term to boolean value for every GO term in an

ontology or given subontology. A multi-label may contain values that are unknown.

### 1.1.3 Annotations and predictions

The Gene Ontology Annotation (GOA) database [16] contains a list of associations between UniProtKB identifiers and GO terms. Each association is complemented with metadata, including the date the association was made and an evidence code indicating whether the annotation was assigned by a curator using either experimental or computational analysis, or assigned automatically. A positive annotation to a specific GO term implies a positive annotation to any of its more general ancestor GO terms. Only 562,971 protein sequences in UniProtKB have an annotation obtained experimentally. These are provided by member databases and post-processed by the Gene Ontology Consortium (GOC). There are also electronic annotations in Gene Ontology Annotation (GOA), generated by algorithm. Some of these algorithms may simply be the mapping of a manually applied keyword to a GO term. As these annotations come from various labs and genome annotation consortia, neither the proteins nor the GO terms are studied uniformly.

There are caveats in using GOA. For example, a protein can be both positively and negatively annotated to the same GO term. Please visit the resource for more information.

A prediction is an electronic annotation with a confidence score. For example, one of the methods we will use as a baseline uses  $\frac{\text{bitscore}(\text{query}, \text{subject})}{\text{bitscore}(\text{query}, \text{query})}$  for the score. Probabilistic methods, such as the ones proposed here, usually use a probability for the score.

### 1.1.4 Challenges for prediction of protein function

An effective model for protein function prediction must take into account several peculiarities of protein function and the data available to use for its prediction. There are a tremendous

number of protein sequences available, but very few are functionally characterized. Experimental annotations, which usually describe a protein's function in part or at a high level, are expensive to obtain, rare, and collected with bias. Negative annotations, which indicate that a given protein does not have a given activity, are nearly non-existent. The space is large, and there is not a single protein that has a complete multi-label in GOA, with a positive or negative annotation for every functional term in GO. Some GO terms also have few, or no, associated proteins, thwarting typical classification algorithms that require many samples per class.

Although these characteristics complicate function prediction, a method can be constructed to benefit from the constraints they impose. For example, a semi-supervised model can use the multitude of uncharacterized protein sequences, and the calculated pairwise similarity between them, to guide the drawing of functional boundaries between protein clusters. As another example, a method that uses a separate classifier for each GO term will likely have too few training samples for each one, and the resulting predictions may be inconsistent with respect to the GO hierarchy. However, viewing the problem as a structured output prediction problem will not only result in consistent outputs, but will be able to take advantage of annotations throughout GO.

### 1.1.5 Methods for automated function prediction

There are many published methods for the prediction of protein function. We mention classical or imaginative works here, and describe highly relevant works in the appropriate chapter.

A common approach for predicting the molecular function of a given query protein is to search a sequence database of annotated proteins and derive the predicted function from the

annotations of the most similar sequences found [57, 21]. Despite a number of caveats [86, 75, 85, 77], this approach remains popular due to the wide availability of sequence data, the speed at which a similarity search can be conducted [3, 15], and the simplicity and robustness of the method.

The biological rationale supporting this method is that since sequence implies structure, structure implies function, and the sequence of the query protein is similar to the sequence of the database hit, then the function of the query protein is probably similar to the function of the target protein. However, BLAST-based function transfer has limitations: its near-sightedness can give results inconsistent with the evolutionary history of the protein, and it does not provide a mechanism for incorporating non-sequence data.

Many methods use networks to predict protein function. Networks for protein function prediction might represent sequence similarity networks, functional association networks [58, 83], including protein-protein interaction networks, hybrids of the two, or networks with heterogeneous nodes that may represent proteins, substrates, function terms, or phenotypes [48].

Phylogenetic methods also use sequence data, but try to automate phylogenetic analysis [27, 30, 59]. These methods take as input, or infer, a gene tree that shows how the proteins are evolutionarily related. Then, using a parsimonious or probabilistic model of function evolution, these methods compute likely assignments of functions to extant, homologous proteins and ancestral proteins, therefore predicting functions that are consistent with its evolutionary history.

Text-mining methods try to extract experimental evidence from published articles and other text in order to exploit data that is not accessible in databases like GOA.

Some methods infer valuable features to integrate [49], while others incorporate raw data by automatic feature extraction and selection or use of a deep learning algorithm. Some



methods can incorporate structured data using kernels that represent functional similarity or network kernels [80, 43, 44]. Most methods that make use of experimental annotations can also make use of electronic annotations or predictions from other methods [81]. Several high-performing methods integrate large numbers of sequence or structural features in an SVM [22], while others integrate data with carefully built probabilistic models [87]. For a review of data integration methods for automated function prediction, please see Rentzsch and Orengo [73].

The proposed methods are based on addressing limitations in the BLAST-based method, and then extending that method to incorporate a distinct, but related piece of information. Namely this work proposes a model for probabilistic prediction of protein function, and then attempts to automate genomic context analysis with careful extensions to the model. Both methods can be visualized as belief about function propagating around their networks in a statistically rigorous way.

### 1.1.6 Evaluation of methods for automated function prediction

Evaluation of protein function is complex. The samples used for training and for evaluation are derived from GOA, so they are also structured, incomplete, and collected unevenly. The classes are imbalanced, and it is easy to guess correctly that a protein does not have any given function, or guess correctly that a protein is involved in a very general activity. The use of basic evaluation methods and metrics is not always appropriate, and it was difficult to compare results from more involved evaluation protocols.

An *evaluation method* defines the procedure for combining prediction methods, functional data, and metrics to generate performance results. I describe two kinds of evaluation methods in the next two sections.

## Cross-validation

Cross-validation is a technique where data is repeatedly partitioned into a training dataset and an evaluation dataset. The prediction methods then use the data in the training dataset to make predictions for the proteins with data that are in the evaluation dataset and have been withheld. Then, the predictions are compared to the withheld data, and the performance of the method is scored.

For the performance given by the evaluation method to be close to the real-world performance of the classifier, the characteristics of the data in the evaluation dataset should resemble the real-world data on which the classifier will run. For example, assuming that an image classifier is going to be used on new images that are similar to the ones in the dataset, then it is important that the evaluation dataset has images that are similar, but not the same as images that were used for its training.

There are several types of cross-validations that vary by how they partition the data.  $k$ -fold cross validation partitions the data into  $k$  subsamples. For  $k$  iterations, one of the partitions is withheld as the evaluation data, and the other  $k - 1$  partitions are used for training. Leave-one-out cross-validation is another common type of cross-validation. Where there are  $n$  samples, leave-one-out cross-validation is  $k$ -fold cross validation with  $k = n$ . As the simplest cross validation method, the holdout method randomly partitions the data once, and only does one run of training, prediction, and scoring.

For many machine learning problems, randomly partitioning the data is sufficient. However, these methods are poorly suited for evaluation of methods for protein function prediction. Consider the case where our dataset includes all annotations in GOA, including the electronic annotations. So far, this is a reasonable setup, because experimentalists and computational biologists also have access to experimental annotations when annotating the

function of proteins. However, if we then decide to use leave-one-out cross validation, we will overestimate the predictive performance of the method being evaluated. This is because following the characterization of a single protein, electronic annotation methods propagate its function to other proteins that are similar in sequence. Removing the multi- for one single protein, and recovering it from the multi-s of the most similar proteins, is easy, and does not resemble the real world task of function annotation.

This problem is not limited to electronic annotations. Proteins are not studied uniformly. It is common for a discovery in one protein to attract others to study similar proteins. The annotations are correlated in time.

Greene and Troyanskaya [37] studies other biases that effect evaluation of methods for protein function prediction.

### **Critical Assessment of Function Annotation (CAFA)**

Published methods tend to overstate their performance. While this problem is not limited to the field of protein function prediction, the large number of pitfalls make accidental errors in evaluation common, and partial treatment easy to obfuscate. It is common for methods for protein function prediction to claim near perfect accuracy [82]. To address these issues, the community in protein function prediction began using critical assessments with a common protocol and datasets.

Following competition-style assessments in other fields, the first critical assessment for protein annotation, MouseFunc, ran from July 17, 2006 – October 13, 2006 [51]. MouseFunc assembled a standard collection of functional data for mouse, had several teams predict GO terms for mouse proteins, and compared and investigated the predictions.

CAFA [72, 39] is a community experiment that was conceived in late 2009. In its first run, the organizers announced a set of 48,298 target sequences on September 15, 2010, and

asked participants to submit predictions for those targets by January 18, 2011. Compared to cross validation, the participants were not expected to self-limit their use of data. Instead, the participants were free to use all available data and methods, including text mining of the literature and experimental assay. The organizers then evaluated the prediction methods on the set of annotations that were submitted from the submission deadline until December 14, 2011. CAFA has identified methods, metrics, useful sources of information, areas for improvement, and other insights for the field. In this work, we make use of the best practices identified by CAFA.

It is now common for methods to emulate the CAFA experiment for evaluating their performance. This method, which is called *time-stamped evaluation* and other names, is a variation of the holdout method where the data are partitioned by date rather than randomly.

## Metrics

The most basic metric for evaluating a classification method is to withhold the true multi-label for a sample, predict the multi-label for that sample, and then compute the accuracy, or the % correct, where a prediction is correct if it is identical to the true multi-label. In the field of protein function prediction, this metric would not provide valuable information. A multi-label is combination of many GO terms. Although only a subset of multi-labels are valid with respect to the structure of GO, there are still a combinatorial number of them. Even a highly accurate method would rarely get exactly the right combination of GO terms to match the multi-label in the gold standard. This is made definite by the fact that, in this field, the true multi-labels are known to be incomplete. The values reported by this metric would be uninformatively low.

However, this is a common problem faced by machine learning researchers, for example when predicting multi-labels (e.g., cat, outside) for an image recognition task. This problem

is addressed by computing metrics at the label level, such as the number of correctly predicted labels, the number of incorrectly predicted labels, and the total number of labels predicted. One caveat, however, is that a bioinformatics scientist cannot easily verify if a label should have been applied to a particular protein.

The task of protein function prediction involves imbalanced classes; a given protein most likely does not perform a given function. For a method to achieve wonderful looking performance, it need only predict every label false. This is addressed by using the metrics precision and recall, which do not give favorable consideration to predictions of negative terms. Precision is the number of true positives over the total number of predicted positives. Recall is the number of true positives over the total number of positives in the true label.

While, this is effective in a binary classification scenario, it does not address problems in a multi-label classification scenario. Precision and recall consider all positive labels to be equally important. A cowardly method could simply predict common terms and avoid predicting rare terms. However, rare terms are usually more specific than common terms, and are of more interest to consumers of protein function predictions. A realistic, incomplete gold standard makes the situation more dire; the performance of the method would be strongly affected by which of the evaluation proteins were labeled with only the vacuous root annotation! Many papers address this issue simply by not ignoring annotations and predictions to the root term during evaluation, but the underlying problem remains; not all GO terms should be treated equally during evaluation.

This concern was elegantly addressed by Clark. The idea is a version of precision and recall that use a weighted count based on the information content of the terms.

$$\begin{aligned} \text{wpr}(P, T) &= \frac{\sum_{t \in P \cap T} \text{IC}(t)}{\sum_{t \in P} \text{IC}(t)} \\ \text{wrc}(P, T) &= \frac{\sum_{t \in P \cap T} \text{IC}(t)}{\sum_{t \in T} \text{IC}(t)} \end{aligned}$$

While we now have a good metric for comparing a predicted multi-label to a “true” multi-label, we have not yet discussed how to score a method that predicts a multi-label for several proteins, given an evaluation set of “true” multi-labels. The simplest approach is to average the single-protein metric across all evaluation proteins. However, we know that the gold standard dataset is incomplete, and doing this would give equal weight to a protein in the evaluation set with only the root annotation, and a protein with detailed experimental data. Many papers address this issue simply by only evaluating against proteins with the most detailed experimental evidence, or excluding the proteins with the least detailed experimental evidence. But again, the main issue is not resolved; proteins with higher quality annotations should be given more weight than proteins with low quality annotations. The proxy that Clark used for quality of an annotation is the information content of the true multi-label. The metric for the method is then the weighted average of the metrics over the individual proteins above, where they are weighted by information content of the true annotation.

$$\begin{aligned} \text{sw-wpr}(\tau) &= \sum_{i=1}^{N_e} \frac{\text{IC}(T_i)}{\sum_{j=1}^{N_e} \text{IC}(T_j)} \frac{\sum_{t \in P_i(\tau) \cap T_i} \text{IC}(t)}{\sum_{t \in P_i(\tau)} \text{IC}(t)} \\ \text{sw-wrc}(\tau) &= \sum_{i=1}^{N_e} \frac{\text{IC}(T_i)}{\sum_{j=1}^{N_e} \text{IC}(T_j)} \frac{\sum_{t \in P_i(\tau) \cap T_i} \text{IC}(t)}{\sum_{t \in T_i} \text{IC}(t)} \end{aligned}$$

There is a tradeoff between a methods accuracy and its recall. However, different consumers of protein function predictions have different requirements; some are looking for clues into any possible function for the protein, and others are looking for only the most certain protein function predictions. Since a method’s scores or probabilities are unlikely to be correct / calibrated with another’s, evaluators can plot one metric that is dependent on the

method's score against another metric that is dependent on the method's score, and come up with a curve in terms of objective evaluation metrics that allows consumers to choose a method that performs best at the recall they need.

However, concerns remain. A paper showed, by looking at how reported performance changes with respect to completeness of the dataset over time, that it is effected by missing data.

### “Gold Standard” Datasets

There are a few standard datasets for training and evaluation of methods for protein function prediction.

The SwissProt section of UniProt contains manually reviewed entries, with a focus on model organisms.

GOA, described in Annotations and predictions contains evidence codes that are manually assigned and *Inferred from Electronic Annotation (IEA)*. These annotations are submitted by UniProt, the Institute for Genomic Research, GeneDB, and organism-specific GOC members, such as FlyBase. Annotations in GOA are well structured, and are always associated with GO terms and an evidence code, and often with a literature reference or other documentation for its source.

The Structure – Function Linkage Database presents a distinctive alternative to the aforementioned datasets. Rather than focus on annotations within model organisms, SFLD curators focus on functionally diverse enzyme superfamilies. These curators perform sequence similarity network analysis, collect all experimental evidence from the literature, and use characteristics of the sequence and structures of the proteins to identify isofunctional clusters.

## Baselines

Baseline methods are simple methods for protein function prediction that are useful for giving a frame of reference for the performance metrics and charts.

One of the simplest useful baselines is the Naive method. The Naive method ignores the particular query protein, and predicts the same GO terms for every query, based on the background frequencies of the GO term in the dataset. Whereas in a balanced binary classification setting it would be easy to intuit the value of a performance metric or shape of a performance chart for a Naive method, this is much more difficult in the case of multi-classification problem with unbalanced classes.

In this work, we use a modified Naive baseline, *Naive+*, which is more useful when we are evaluating proteins which were partially annotated in the training data. For a given query protein  $i$ , this method first predicts with certainty the GO terms for which there was evidence in the training data. Then, the method predicts GO term  $g$  with probability

$$P(X_g^i = \text{True}) = P(X_g^i = \text{True} \mid X_{\text{GO parents}}^i) \\ \times \prod_{t \in \text{GO parents}} P(X_t^i = \text{True})$$

Therefore, if there is a direct annotation in the training data for the query protein having catalytic activity, then the method will predict catalytic activity with certainty, but also predict a relatively high probability for transferase activity, based on  $P(X_{\text{transferase activity}}^i = \text{True} \mid X_{\text{catalytic activity}}^i = \text{True})$  being high in the background data. Notice that, in the case of a protein lacking any annotation in the training data, this method reduces to the Naive method.

This probabilistic propagation of GO terms to more specific GO terms can be applied to the raw predictions of any method to increase its recall.



A *BLAST-based method* is simple enough to be considered a baseline, but offers very good performance. There are several ways to implement a BLAST-based method that would result in methods with very different characteristics. We require high query coverage so that our function transfer is limited to sequences with the same domain architecture. We also use a relatively strict threshold for sequence similarity. Whereas some BLAST-based methods only consider the top BLAST hit, we consider all BLAST hits that are more similar than our thresholds, so that our BLAST-based method has improved coverage. For each hit, for each preprocessed annotation, our BLAST-based method predicts the GO term with a probability equal to the bitscore of the hit, divided by the greater of the self alignment scores for the query and the subject. Therefore, a query protein hitting itself or an identical sequence will transfer all of its training annotations at 100%.

We also implemented a *Random BLAST* method. This method uses the same BLAST hits and training annotations as our BLAST-based method, but the probability for function transfer, which is shared by all the GO terms for each hit, is chosen uniformly at random. The quality of the Random BLAST predictions are good for three reasons. First of all, by using the strict BLAST parameters mentioned above, the predicted GO terms are all reasonable. Second, the predictions are consistent with the GO hierarchy, because of the way that preprocessing the annotations propagates direct annotations to more general terms, and the way that we keep the maximum probability prediction for each GO term. Third, GO terms that occurred multiple times in the BLAST hits, either because of their generality or their relevance, have more chances to be transferred to the query with a high probability. By comparing the performance of this method to the performance of the BLAST-based method, we hope to give a sense of scale for the differences in performance between the methods that we evaluate.

The details of our implementation of these baselines, and the other methods that we use

in our comparisons, are described in the relevant Methods sections.

## 1.2 Review of topical mathematics

Only a few elementary terms and operations from probability are needed to understand the bulk of this dissertation. The following reviews these concepts.

### 1.2.1 Probability

A *discrete probability* is a function that maps an event to a real positive number less than one, e.g.,  $P(\text{Rain} = \text{yes}) = .2$  such that the sum of the probabilities of all events is one:  $P(\text{Rain} = \text{yes}) + P(\text{Rain} = \text{no}) = 1$ . From now on, let  $1 := \text{yes}$  and  $0 := \text{no}$ .

#### Joint Probability

A *joint probability* is the probability of multiple events happening at once, e.g.,  $P(\text{Rain} = 1, \text{Clouds} = 1, \text{Grass Wet} = 1)$ . Joint probabilities must also sum to unity:

$$\sum_{r \in \{0,1\}} \sum_{c \in \{0,1\}} \sum_{w \in \{0,1\}} P(\text{Rain} = r, \text{Clouds} = c, \text{Grass Wet} = w) = 1$$

The table defining a discrete joint probability distribution is  $\mathcal{O}(k^N)$ , where there are  $N$  variables at most cardinality  $k$ .

Many important probabilities can be computed from a joint probability.

#### Marginal Probability

A *marginal probability* is a probability over a subset  $X$  of variables in a joint distribution. It can be calculated from a joint probability by summing over the variables not in  $X$ :

$$P(\text{Rain} = r) = \sum_{c \in \{0,1\}} \sum_{w \in \{0,1\}} P(\text{Rain} = r, \text{Clouds} = c, \text{Grass Wet} = w)$$

### Conditional Probability

A *conditional probability*  $P(X = x \mid Y = y)$  is a probability of an event  $X = x$  given that another event has occurred  $Y = y$ . It can be calculated from a joint distribution, or a marginal distribution, by normalizing the joint distribution  $P(X = x, Y = y)$  by the probability of the given event  $P(Y = y)$ . Note that the denominator is a marginal probability of the numerator. For example,

$$\begin{aligned} P(X = x \mid Y = y) &= P(\text{Rain} = r \mid \text{Clouds} = c, \text{Grass Wet} = w) \\ &= \frac{P(\text{Rain} = r, \text{Clouds} = c, \text{Grass Wet} = w)}{P(\text{Clouds} = c, \text{Grass Wet} = w)} \\ &= \frac{P(\text{Rain} = r, \text{Clouds} = c, \text{Grass Wet} = w)}{\sum_{r \in \{0,1\}} P(\text{Rain} = r, \text{Clouds} = c, \text{Grass Wet} = w)} \end{aligned}$$

### 1.2.2 Parameters and statistics

There are representations of probability distributions that are more concise than a table. They can also be specified in an analytical form that depend on parameters. Take, for example, a probability model for the number of heads observed when flipping a coin ten times. We can represent this as table of one probability for each of the 10 possible outcomes. Alternatively, we can specify that  $P(\text{heads observed} = k) = \binom{10}{k} \pi^k (1 - \pi)^{10-k}$ , where  $\pi$  is a parameter that describes how likely our coin is to land on heads.

If  $\pi = .5$ , then we are working with a typical fair coin. However, parameters are paramount. A model that says  $\pi = .99$  would imply that our coin is loaded and flipping ten heads in a row is the most likely event. The field of statistics deals with estimating parameters

from data. One of the most common paradigms for doing this is to attempt to find the parameter(s) that maximize the probability of the data, known as the likelihood. For example, if we observe 5 heads and 5 tails, then  $P_{\pi=.5}(\text{heads observed} = 5) > P_{\pi=.99}(\text{heads observed} = 5)$ . In fact, the maximum likelihood estimate of  $\pi$  would be

$$\begin{aligned}\hat{\pi} &= \operatorname{argmax}_{\pi \in [0,1]} P_{\pi}(\text{heads observed} = 5) \\ &= .5\end{aligned}$$

### 1.2.3 Probabilistic graphical models (PGMs)

In joint distributions with many random variables, the table defining the distribution can be too large to use. Probabilistic graphical models (PGMs) are a framework for representing complex distributions in factored form. Algorithms exist for performing inference, specifically for computing marginal probabilities, on general PGMs [66]. We represent our model as a directed probabilistic graphical model, known as a Bayesian network. With missing edges in a PGM corresponding to conditional independence assertions, a directed PGM factorizes a joint distribution into local conditional probabilities

$$P(X = x) = \prod_{x_i \in x} P(x_i | \text{pa}(x_i))$$

where  $x := x_1, x_2, \dots, x_N$  and  $\text{pa}(x_i)$  represents the parents of variable  $x_i$ .

As for any discrete joint distribution, computing marginal probabilities from a simple PGM consists of summing out the nuisance variables. In the unfactored form of the joint distribution, this summation would require a number of terms exponential in the number of variables. In the factored form, however, many computations can be reused by distributing

summation inward. For example, if our joint distribution  $P(A, B, C)$  factors into  $P(A)P(B | A)P(C | B)$  because  $C$  is independent of  $A$  given  $B$ , then we could compute  $P(C)$  as follows

$$\begin{aligned} P(C) &= \sum_{a \in A} \sum_{b \in B} P(A, B, C) \\ &= \sum_{a \in A} \sum_{b \in B} P(C | B)P(A)P(B | A) \\ &= \sum_{b \in B} P(C | B)P(B | A) \sum_{a \in A} P(A) \end{aligned}$$

This algorithm is called the elimination algorithm. It is an exact algorithm for any graphical model, but it can result in intermediate factors with too many terms. Algorithms for approximate inference are either based on sampling or the use of a tractable variant of the distribution.

## Chapter 2

# Effusion: Prediction of Protein Function with a Probabilistic Model for Analysis of Sequence Similarity Networks

## 2.1 Abstract

### 2.1.1 Motivation

Sequence similarity networks are one of the primary tools used to investigate sequence-function relationships in large sets of homologous sequences [6, 13, 8]. Although it is well known that proteins with similar functions typically cluster together, in practice, experimental annotations are extremely scarce and scattered unevenly across the protein network. Moreover, most of these annotations describe a protein's function only in part or at a high level. Automating the classification of protein functions via clustering is further complicated by the fact that proteins often have a molecular function that is represented by a whole hierarchy of terms. In other words, some specific functions should be isolated to a specific cluster, some general functions might span the entire network of proteins, and these clusters

often overlap.

### 2.1.2 Results

We present a method for predicting protein function, *Effusion*, that uses a sequence similarity network (SSN) to add context for homology transfer, a probabilistic model to account for the uncertainty in labels and function propagation, and the structure of the GO to best utilize sparse input labels and make consistent output predictions. *Effusion*'s model admits a simple parameterization that makes it practical to integrate rare experimental data and abundant primary sequence and sequence similarity. We demonstrate *Effusion*'s performance using a critical evaluation method and provide an in-depth analysis. We also dissect the design decisions we used to address challenges for predicting protein function. Finally, we propose directions in which the framework of the method can be modified for additional predictive power.

## 2.2 Introduction

Here, we propose and evaluate a new method, *Effusion*, that uses a network of partially characterized sequences to suggest accurate function predictions. The use of SSNs, the incorporation of GO, and the application of PGMs has been previously reported (see Related work). However, our method, its model, and its parameters are the first to integrate these features in a way that is accurate, practical, and extensible. Specifically, our model, inspired by network analysis in computational biology [6, 13, 8, 14], admits a highly interpretable set of parameters, which we can learn for each GO term and from all experimental annotations, augment them with pseudocounts, and submit to general-purpose inference algorithms. Evaluation of the predictions shows that our method can accurately discern the molecular

functions (MFs) of a protein, even when faced with partial, autocorrelated samples and classes that are imbalanced and related hierarchically.

### 2.2.1 Sequence similarity

BLAST [3] and DIAMOND [15] are two methods that can quickly search a protein sequence database for proteins that are similar in sequence to a given query protein sequence and return scores representing their similarity. Two proteins with high sequence similarity and statistical significance are assumed to be homologous.

### 2.2.2 Sequence similarity networks (SSNs)

SSNs often use unannotated proteins to provide context for predicting molecular functions [6, 8, 35]. Visually, they show putative clusters of conserved function, space between clusters with few proteins where there may be a change in function, and unexplored regions of the sequence space [6, 47, 56, 24].

### 2.2.3 Related work

#### Network-based methods

Algorithms that use both labeled and unlabeled data are called *semi-supervised*.

For a review of network-based methods, please see Sharan, Ulitsky, and Shamir [79].

#### Methods that use GO

Several methods, including sequence similarity-based methods, use the structure of GO to improve prediction quality, for example by ranking more general terms higher than more specific terms, or ensuring predicted functions are consistent [28, 9, 64, 38].



## Probabilistic methods

PGMs are especially useful for the prediction of protein function because the topology of the model is data generated, rather than user generated, and PGMs can model random variables with complex relationships [46, 25]. For example, SIFTER constructs a probabilistic model that has the topology of a phylogenetic tree [31].

Most protein network-based PGMs focus on protein-protein interaction networks [42], but some authors present, or suggest, a PGM based on a sequence similarity network. Carroll and Pavlovic [17] and Mitrofanova, Pavlovic, and Mishra [61] additionally incorporate the structure of GO into their PGM. We build on the ideas of these methods, but use a model, parameters, and algorithm that are better suited to the problem of predicting protein function.

Methods that try to automate SSN analysis report severe limitations, including predicting on very few functional terms, necessitating ad-hoc inference algorithms and post-processing steps, and forgiving evaluation methods. Our method, Effusion addresses the heart of what made the other methods impractical.

## 2.3 Methods

A graphical summary of the method is shown in Figure 2.2.

First, we build a protein network with edges of sequence similarity. This network is quickly constructed by broadly BLASTing [3] sequence queries to collect homologs (Figure 2.2 (a)) and using DIAMOND [15] to fill in the all-by-all sequence similarity edges of the network (Figure 2.2 (b)).

Second, we construct a tractable PGM based on this network. The network is first

converted to a minimum spanning tree (MST), pruned to query proteins or proteins with non-electronic annotation, and directed outward from the query (Figure 2.2 (c)). For each edge in the MST, we link the corresponding MF terms from GO (Figure 2.2 (d)). Since the model is directed, the parameters have a global probabilistic interpretation: each variable in the probabilistic model adds a factor representing the conditional probability of that variable given its parents. We can then learn these parameters from looking at all pairs of neighboring proteins that have experimental annotations.

Finally, we perform inference with a general-purpose, state-of-the-art inference software and output the predictions. A network view of the output predictions for two GO terms are shown in Figure 2.2 (e). Since the method has a GO-structured model of protein function, detailed predictions are given for every protein in the network. We show the detailed predictions for the query in Figure 2.2 (f).

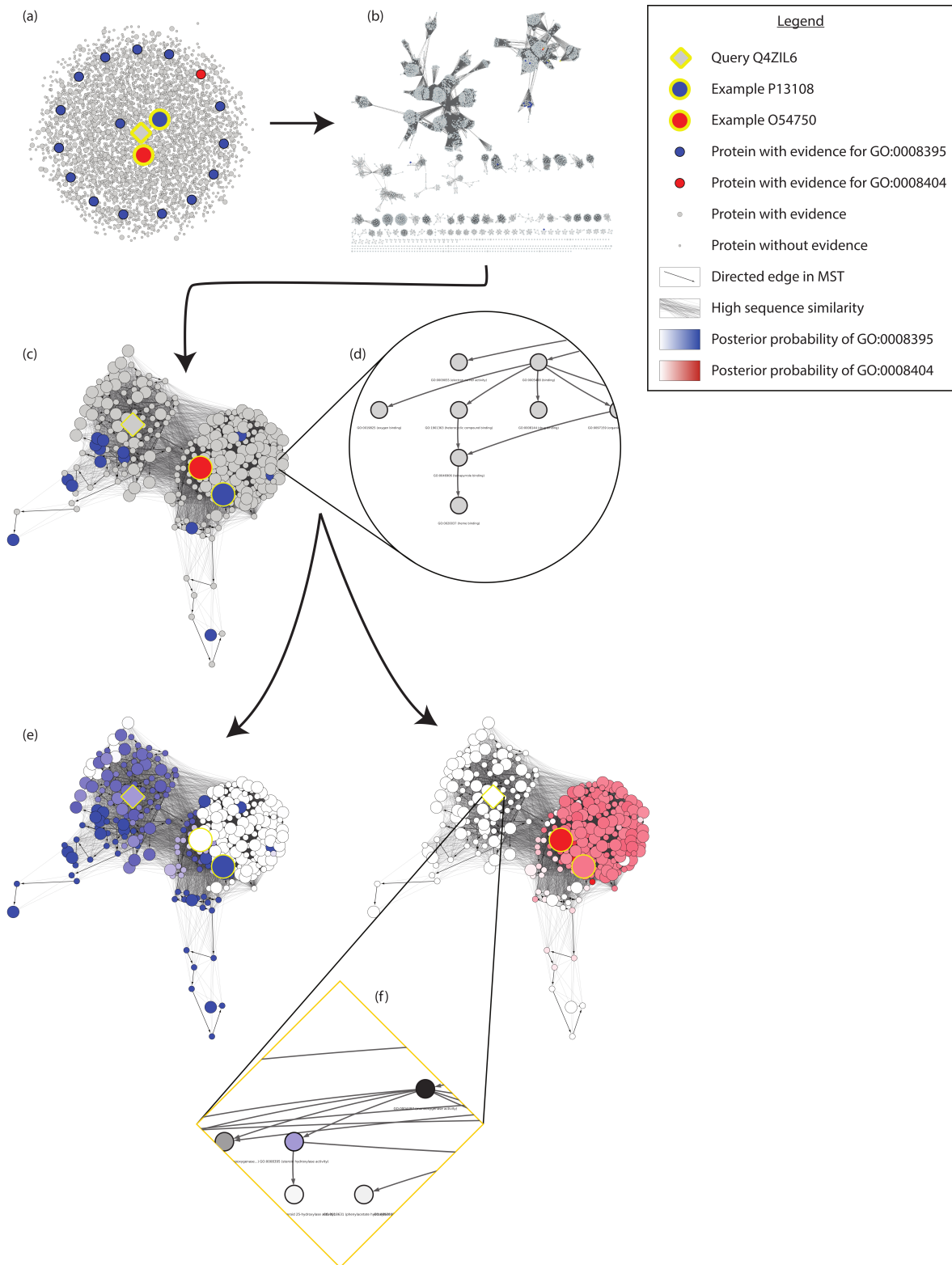
The details of the method follow.

### 2.3.1 Preprocessing

We downloaded the datasets for our analysis in early April, 2017.

- protein sequences: UniProtKB, compiled from SwissProt and TrEMBL, version 2017\_03
- experimental annotations: Gene Ontology Annotation Database, version 2017-03-11
- controlled vocabulary: Gene Ontology, version 2017-03-31
- sequence similarity: Computed by BLAST version 2.6.0+ for single query searches, or via DIAMOND version 0.9.10 for all-by-all calculations

We only included annotations to UniProtKB identifiers, excluding a smaller number of annotations linked to gene products in other databases. As is standard in the field,



**Figure 2.2:** Graphical Summary of Effusion, using UniProt Q4ZIL6 as an example. (a) Homologs of UniProt Q4ZIL6 collected by BLAST. UniProt Q4ZIL6 has been experimentally annotated to a descendent of steroid hydroxylase activity (GO:0008395), but this annotation was one of the ones withheld during the test phase, and it has no annotation to any arachidonic acid 14,15-epoxygenase activity (GO:0008404), an annotation for a homolog of the query reported by BLAST. (b) A SSN is built from the all-by-all edges computed via DIAMOND. The network is visualized with Cytoscape [78] using Organic Layout. (c) The reduced network. The layout is applied to all edges, but only the MST edges are used in the model. The resulting network is used as the topology of a PGM. (d) The protein function of each node is represented by a subset of GO, with each GO term represented by Bernoulli random variable. (e) Two views of the network following inference. The left figure is shaded according to the probability of GO:0008395. The right figure is shaded according to the probability of GO:0008404. (f) Probabilities for a subset of terms for query UniProt Q4ZIL6. Probabilities are calculated for each candidate GO term for each node. Nodes are shaded from white being 0% to black being 100%, except for the node representing GO:0008395, which is colored a shade of blue based on its posterior probability.

we excluded IEA annotations. For each positive annotation, we explicitly added positive annotations for each ancestor GO term for the same protein and annotation metadata (i.e., date and evidence code). Similarly, a negative annotation for a protein to a GO term was considered a negative annotation for the same protein to every descendent GO term. We refer to the resulting set of annotations as *preprocessed* annotations.

The dataset for the discovery phase included all preprocessed annotations through 2015. Annotations from 1 January 2016 onward were withheld for all purposes until final evaluation and analysis.

### 2.3.2 Building the protein network

For a given query protein, we build a SSN, where nodes represent proteins and edges represent pairwise sequence similarity. This network is quickly constructed by collecting homologs with BLAST, and then using DIAMOND to fill in the all-by-all sequence similarity edges of the

network. BLAST is run with a permissive  $E$ -value threshold ( $1e-8$ ), but limited to sequences that cover 90% of the query, and filtered to sequences with a bit score per residue of at least 0.25. We use the resulting sequences to format a DIAMOND DB, and search each resulting sequence against this DB using DIAMOND (evalue =  $1e-8$ , query-cover = 90%, subject-cover = 90%), and more restrictive parameters (max-target-seqs = 1000, minimum bit score per residue of  $1.4 \times 0.25$ ). The all-by-all calculation uses the more restrictive parameters to limit the number of edges for reasons of computational practicability. This heuristic usually provides the MST edges we need for building the model (see [Creation and use of protein networks]) without wasting space and time.

### 2.3.3 Constructing a tractable probabilistic graphical model

The protein network is first reduced to make it more amenable to learning and inference. To get the reduced network, we convert the protein network to a MST (edge weight =  $-\text{bit score per residue}$ ), direct it outward from the query, and prune it to query proteins or proteins with non-electronic annotation. The goal of this network reduction was to imbue local factors with a global probabilistic interpretation, which facilitates parameter learning. Another benefit of this reduction is that it results in inference running more quickly, by removing tight loops in SSNs, and allowing pruning. However, we note that the PGM is still not a tree because variables generally still have multiple parents, coming from GO and from the corresponding term of the parent protein in the reduced network. Therefore, iterative, approximation algorithms are needed for inference.

## Protein template

Each protein in the reduced network generates a subgraph in the PGM, by instantiating a copy of the *protein template*. The protein template models the molecular function of a protein. It has the topology of a subgraph of the Gene Ontology Consortium’s molecular function ontology. Each node in the subgraph is a *candidate GO term*. Every GO term for which there is a positive or negative preprocessed annotation in the SSN is a candidate GO term. Each candidate GO term, for each protein, is modeled as a Bernoulli random variable, and we ultimately calculate marginal posterior probabilities for each of them.

We implemented and evaluated two models for relating the GO terms within a protein template: a top-down model and a bottom-up model. Each model has its own advantages and disadvantages.

In a top-down model, the parent(s) of a variable representing a GO term  $t$  for protein  $i$  include the parents (more general terms) in GO, so the factors  $\psi(x_t^i, \text{pa}(x_t^i))$  are  $P(x_t^i | \text{pa}(x_t^i)) = P(x_t^i | x_{\text{GO parents}(t), \text{other model parents}(x_t^i)}^i)$ . It is more common to model a PGM in a top-down fashion, because PGMs that limit factor sizes, in particular those that have only a single parent per node, admit more tractable PGM inference.

However, a top-down model has limitations for modeling protein function. If, for example, a protein is annotated as a DNA polymerase, then that protein has an implied annotation to polymerase in general, and that will give the protein a high probability of any type of polymerase, such as an RNA polymerase. In this scenario, the posterior probability for RNA polymerase could be higher than the posterior probability for any specific DNA polymerase.

To address this, we added supplementary negative evidence as follows. For each protein with evidence, if the (weighted, see Parameter learning) contingency table shows that a particular unobserved term is unlikely ( $< 50\%$ ) given the observed values of its sibling

terms, then we infer a negative annotation for that sibling. For the example above, since an annotation for RNA polymerase is unlikely to co-occur with an annotation for DNA polymerase, then a protein with a positive annotation for DNA polymerase and no annotation for RNA polymerase would have a supplementary negative annotation for RNA polymerase.

These supplementary negative annotations are considered as evidence and deemed certain, and since they are not always correct, the necessity of negative annotations is a limitation of the top-down model. For example, a query which is known to have transferase activity (GO:0016740) in its training data will be given no chance of having hydrolase activity (GO:0016787), because the two terms are siblings, and  $P(\text{hydrolase}|\text{transferase}) = 25.3\% < 50\%$ . However, our data show that the terms were co-annotated to the same protein 834 times.

In our example above, we would prefer that our model were powerful enough to allow evidence for DNA polymerase to *explain-away* our belief in RNA polymerase. Therefore, we also experimented with a bottom-up model, where the parents of a variable representing GO term  $t$  for protein  $i$  include the children (more specific terms) in GO:  $\psi(x_t^i, \text{pa}(x_t^i)) := P(x_t^i|\text{pa}(x_t^i)) = P(x_t^i|x_{\text{GO children}(t), \text{other model parents}(x_t^i)}^i)$ .

We use *GO kin* to refer to GO parents in the top-down model, and GO children in the bottom-up model. As we continue the description of our method, *model* refers generally to both the top-down and bottom-up model, except as specified.

### **Incorporating sequence similarity edges**

When two proteins are similar in sequence and have an MST edge between them, we connect the corresponding molecular function terms. Assuming reasonable parameters, the factor associated with this edge induces two proteins that are similar in sequence to have similar molecular functions.

### 2.3.4 Parameter learning

Since the model is directed, the parameters have a global probabilistic interpretation: each variable in the probabilistic model adds a factor representing the conditional probability of that variable given its parents.

For the model just presented, the parameters are  $P(x_t^i \mid x_{\text{GO kin}(t)}^i, x_t^{\text{BLAST parent}(i)})$ , the probability of protein  $i$  having term  $t$ , given the GO kin of GO term  $t$  for protein  $i$ , and the corresponding GO term for the BLAST parent.

We can learn these parameters from all available experimental data, not just the data in a network for a specific query. To do so, we compare the label for each protein with a preprocessed annotation to the label of the of the most similar protein with a preprocessed annotation. The similarity of the most similar protein must be below the similarity thresholds specified above. We count the number of times there was a gain of function, loss of function, or other such events. We use thiolester hydrolase activity (GO:0016790) as an example to show the contingency tables we use for calculating the parameters. Table 2.1 shows the contingency table of raw counts.

**Table 2.1:** Raw contingency table for GO:0016790

Protein's annotation to GO:0016788 (GO parent)	BLAST neighbor's annotation to GO:0016790	Count protein annotation to GO:0016790 is negative or unknown	Count protein is positively annotated to GO:0016790
-/?	-/?	41446	0
-/?	+	5	0
+	-/?	1453	5
+	+	3	19



Protein's annotation to GO:0016788 (GO parent)	BLAST neighbor's annotation to GO:0016790	Count protein annotation to GO:0016790 is negative or unknown	Count protein is positively annotated to GO:0016790
---	--	--	--

We note also that we learn parameters for each GO term. So, for example, the probability of a hydrolase losing its ability to hydrolyze ester bonds GO:0016788 over adjacent proteins is low (1.5%); the probability of a hydrolase losing its ability to hydrolyze a thiolester bond GO:0016790 is higher (4.3%).

Our contingency tables are based on preprocessed annotations rather than inferred functions, so incomplete annotations result in model that make a gain or loss of function between neighboring proteins very likely. We address this by weighting the count contributed by each protein by the information content of the protein's label  $IC(\text{label}) = \sum_{g \in \text{label}} IC(g)$ . An example contingency table of weighted counts is shown in Table 2.2.

**Table 2.2:** Weighted contingency table for GO:0016790

Protein's annotation to GO:0016788 (GO parent)	BLAST neighbor's annotation to GO:0016790	Count protein annotation to GO:0016790 is negative or unknown	Count protein is positively annotated to GO:0016790
-/?	-/?	2630000	0
-/?	+	421	0
+	-/?	116000	678
+	+	119	1420

In order for our model to have a chance at predicting rare GO terms, we added pseudocounts to our contingency tables. Our aim was to add counts from contingency tables for similar terms, but with more experimental evidence. Therefore, we transformed each raw (or information content (IC) weighted) contingency table for GO term  $g$ ,  $C_g^{\text{raw} / \text{weighted}}$ , with the recursion

$$C_g^{\text{pseudo}} := 0.10 \times C_{\text{MRCA}(\text{GO parents}(g))}$$

$$C_g = C_g^{\text{raw} / \text{weighted}} + C_g^{\text{pseudo}}$$

where most recent common ancestor (MRCA) is the recent common ancestor. A contingency table with pseudocounts added to the raw counts are shown in Table 2.3.

**Table 2.3:** Contingency table for GO:0016790, with pseudocounts. In practice, pseudocounts are added to the weighted contingency table, but here they are added to the raw table for illustration purposes.

Protein's annotation to GO:0016788 (GO parent)	BLAST neighbor's annotation to GO:0016790	Count protein annotation to GO:0016790 is negative or unknown	Count protein is positively annotated to GO:0016790
-/?	-/?	45500	0
-/?	+	16.6	0
+	-/?	1930	17.3
+	+	5.07	219

### 2.3.5 Information content of GO terms

The information content of a GO term  $g$  is calculated by

$$\text{IC}(g) = -\log_2(P(g \mid \text{GO parents}(g)))$$

### 2.3.6 Inference

Our models were complex enough that it was intractable to use exact inference algorithms, and standard approximate algorithms, such as belief propagation (BP, acronym in this context only), tree-reweighted belief propagation (TRWBP), generalized loop corrected BP (GLC), Gibbs sampling, and mean field could not reliably compute reasonable probabilities (e.g., not  $p = 0.5$  for all terms), for the top-down model and especially for the bottom-up model, due to its higher tree width. However, we found the performance to be similar from various software implementations of state-of-the-art inference algorithms that performed well at the recent Uncertainty in Artificial Intelligence (UAI) inference competition, namely variations on adaptive inference [1] and SampleSearch [36]. By default, we used adaptive inference with conditioning (ai\_cond) when evaluating our test predictions.

Runtime and required memory depends on the number of proteins in the pruned SSN, the number and topology of the candidate GO terms, and the parameters given to the inference engine. Since these numbers varied greatly per query, we selected an algorithm that uses the maximum amount of time and memory given to it. Specifically, we set a per query limit of 40 minutes of CPU time and 8 GB memory.

### 2.3.7 Post-processing

Effusion’s predictions could be directly evaluated, but the BLAST-based method’s predictions needed to be post-processed for them to be competitive. We applied the following post-processing uniformly to the raw predictions of all the methods that we evaluated.

Specifically, methods that do not necessarily use the structure of GO may perform poorly as a result of predicting a very general term with the same probability as a specific term. We break ties in favor of more general terms by applying the following transformation:

$$P^{\text{new}} := P^{\text{raw}} \times P(\text{Depth} = \text{GO term depth})$$

$$P(\text{Depth} = \text{GO term depth}) := 1 - \epsilon \times \text{GO term depth}$$

with  $\epsilon := 0.0001$ .

Any duplicate predictions that may arise, perhaps as a result of hits to multiple subjects with annotations to the same GO term, are resolved by keeping only the prediction with the highest probability.

### 2.3.8 Evaluation

We performed evaluation via temporal holdout [37]. The testing phase used annotations through 2015 for training, and withheld annotations from the start of 2016. This reflects the methodology of the CAFA [72, 39], is reflective of the true task of automating the manual process of characterization of protein function, and is widely recommended [37].

All 2757 proteins with a new annotation to a GO term in the molecular function ontology inferred by direct assay (evidence code IDA) were used as the evaluation set. We evaluated all proteins and all terms with this criterion. We did not limit our evaluation to proteins that had no annotations in the training set. We included all GO terms in the molecular

function ontology, and we did not exclude GO terms that are rarely observed, nor did we exclude proteins annotated only to rare GO terms.

We used performance metrics that are revealing and critical, proposed or suggested by [20], with minor modifications. *weighted true positive (WTP)* and *weighted false positive (WFP)* are similar to the true positive count and the false positive count, respectively, but weight the counts by the information content of the GO terms to account for the imbalanced, hierarchically structured label space. Dividing WTP by the IC of the predicted terms gives *weighted precision (WPr)*. Similarly, dividing WTP by the IC of the terms in the standard gives *weighted recall (WRc)*. In an attempt to upweight high quality samples and down-weight low quality samples, *sample-weighted weighted precision (SW-WPr)* and *sample-weighted weighted recall (SW-WRc)* additionally use a weighted average over the evaluation proteins, where the weight is the information content of the true annotation. Neither recall nor its weighted variants are expected to go to 100%, since, for some evaluation proteins, withheld GO terms may not exist in any of the preprocessed training annotations.  $N_e$  represents the number of proteins being evaluated.

$$\begin{aligned}
\text{wtp}(\tau) &= \frac{1}{N_e} \sum_{i=1}^{N_e} \sum_{t \in P_i(\tau) \cap T_i} \text{IC}(t) \\
\text{wfp}(\tau) &= \frac{1}{N_e} \sum_{i=1}^{N_e} \sum_{t \in P_i(\tau) \setminus T_i} \text{IC}(t) \\
\text{wpr}(\tau) &= \frac{1}{N_e} \sum_{i=1}^{N_e} \frac{\sum_{t \in P_i(\tau) \cap T_i} \text{IC}(t)}{\sum_{t \in P_i(\tau)} \text{IC}(t)} \\
\text{wrc}(\tau) &= \frac{1}{N_e} \sum_{i=1}^{N_e} \frac{\sum_{t \in P_i(\tau) \cap T_i} \text{IC}(t)}{\sum_{t \in T_i} \text{IC}(t)} \\
\text{sw-wpr}(\tau) &= \sum_{i=1}^{N_e} \frac{\text{IC}(T_i)}{\sum_{j=1}^{N_e} \text{IC}(T_j)} \frac{\sum_{t \in P_i(\tau) \cap T_i} \text{IC}(t)}{\sum_{t \in P_i(\tau)} \text{IC}(t)} \\
\text{sw-wrc}(\tau) &= \sum_{i=1}^{N_e} \frac{\text{IC}(T_i)}{\sum_{j=1}^{N_e} \text{IC}(T_j)} \frac{\sum_{t \in P_i(\tau) \cap T_i} \text{IC}(t)}{\sum_{t \in T_i} \text{IC}(t)}
\end{aligned}$$

We compare our method against the most similar method described in the literature [17]. *Carroll2006* is a close implementation of their method; it does not use our method for adding supplementary negative evidence, it uses the full network, instead of the reduced network, and it uses the parameters that they describe. However, it predicts on the same candidate ontology, rather than a limited one; it uses a general interference algorithm, rather than an ad-hoc one; it uses the resulting probabilities, rather than thresholded ones; and it uses our critical evaluation method. *Carroll2006Params* is a close implementation of Effusion, in that it uses our reduced network and other modifications, but it uses their parameters, namely, the normalized BLAST scores.

We also compare our method against SIFTER [32, 76], for which there is full pipeline available, was shown to be a top performer in CAFA [72, 39], and although its goal is automated phylogenetic analysis, it is based on sequence-data like Effusion.

We compare our method against a sequence-similarity-based method, referred to in this paper simply as BLAST, implemented as follows. NCBI BLAST was run with the same parameters as the homology gathering step as Effusion (task = blastp,  $E$ -value = 1e-8,

qcov\_hsp\_perc = 90) to collect sequences that align globally, and filter to sequences with a bit score per residue of at least 0.25. For each hit, for each preprocessed annotation, we predict the GO term for that annotation for the query with probability equal to the normalized bit score per residue =  $\frac{\text{bit score}}{\max(\text{len}(\text{query}), \text{len}(\text{subject}))}$ . The post-processing steps described in Post-processing are applied to the raw predictions.

Although our plots use ratio scales, we plot another baseline to convey relative scale. The *Random BLAST* method is implemented the same as the BLAST method, using the same parameters and thresholds, except that the preprocessed annotations are transferred with a probability equal to a number chosen uniformly at random. Note that this does not merely assign random probabilities to all candidate GO terms—probabilities will remain consistent with GO by construction.

## 2.4 Results

Effusion is a simple sequence-similarity only method that utilizes a probabilistic model to account for the uncertainty in labels and function propagation, unlabeled protein data to add context for homology transfer, and the structure of GO to best utilize sparse input labels and make consistent output predictions. It uses an MST reduction of the network so that parameters can be calculated from all experimental data, weighted by the quality of the annotations, augmented with annotations and pseudocounts derived from the data, and used as input to general purpose PGM inference algorithms.

We provide an implementation of Effusion. The source code, written in Python, is available online. We rely on software written by others in C++ for the parts of Effusion that are computationally intensive, namely the database software, similarity computations, and inference engines.

We first analyze the predictions made by the method as a whole, and then show the effects of the various components of the method.

### 2.4.1 Comparative analysis

The dataset for the test phase contained 2757 proteins that had an Inferred from Direct Assay (IDA) annotation to the molecular function ontology dated in 2016. All of these were included for evaluation, except where indicated otherwise.

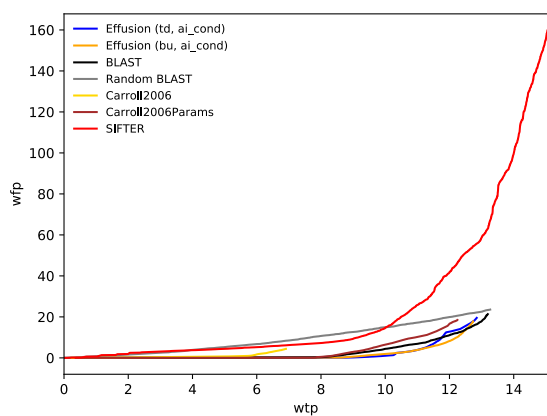
We evaluated Effusion against the methods described in the methods as BLAST remains a high-performing method and is used in CAFA evaluations and by many annotation pipelines. Since both Effusion and BLAST use the same data, the results are highly interpretable.

The BLAST-based method made non-root molecular function predictions for 75.2% (2072 / 2757) of the evaluation proteins. The remaining proteins did not have a protein with positive preprocessed annotations within the thresholds. Effusion (top down, adaptive inference with conditioning) made non-root molecular function predictions for 70.4% (1942 / 2757) of the total, or 93.7% (1942 / 2072) of the proteins for which BLAST made predictions.

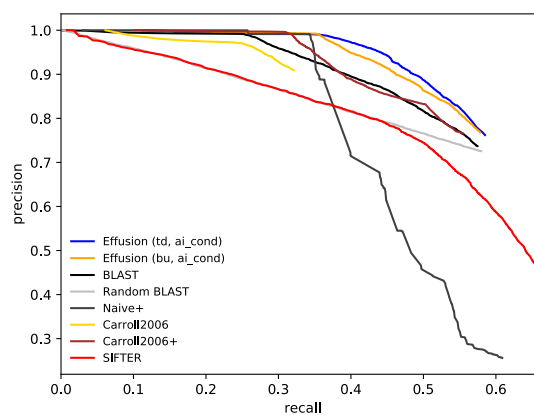
Compared to BLAST, Effusion has additional steps that can fail (e.g., inference) or result in a loss of proteins with evidence (i.e., the MST heuristic). Therefore, we first looked at the performance of Effusion (top-down and bottom-up, ai\_cond) and the baselines over all evaluation proteins, including those proteins for which Effusion failed to make a prediction (Figure 2.3). In all of these plots, Effusion generally performs better than BLAST.

The proteins where Effusion failed to make a prediction, but BLAST was able to make a prediction, were usually due to artificial limitations on runtime (data not shown), and a prediction could still be made by a user with a special interest in a specific protein by removing the limits. Therefore, in order to see how well Effusion typically performs, we

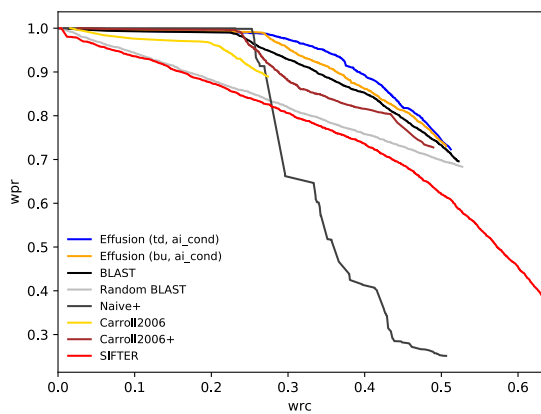




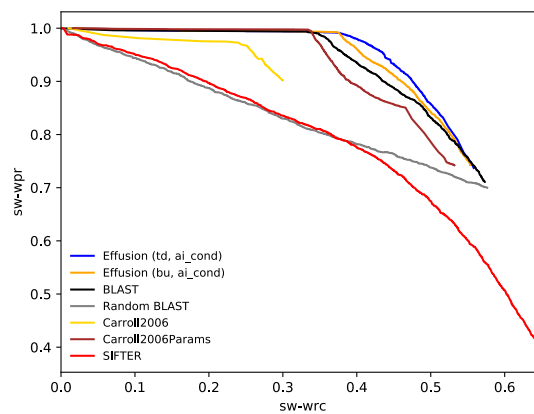
(a) WFP vs. WTP



(b) Precision vs. Recall

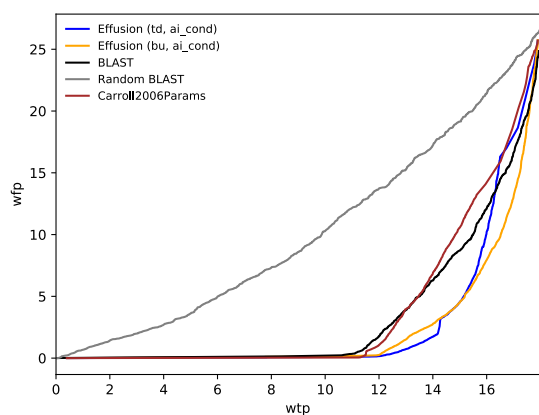


(c) Weighted Precision vs. Weighted Recall

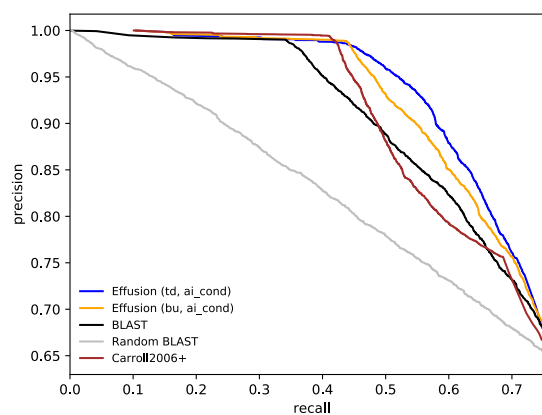


(d) Sample-Weighted Weighted Precision vs. Sample-Weighted Weighted Recall

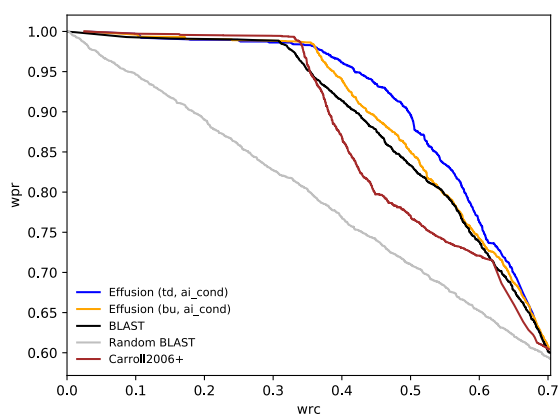
**Figure 2.3:** Performance plots over all proteins in the test set, regardless of whether any of the methods failed to make predictions.



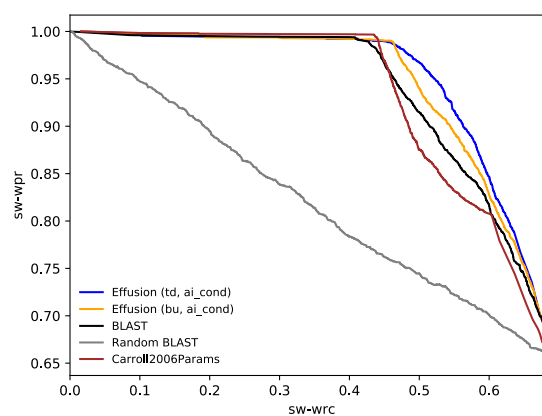
(a) WFP vs. WTP



(b) Precision vs. Recall



(c) Weighted Precision vs. Weighted Recall



(d) Sample-Weighted Weighted Precision vs. Sample-Weighted Weighted Recall

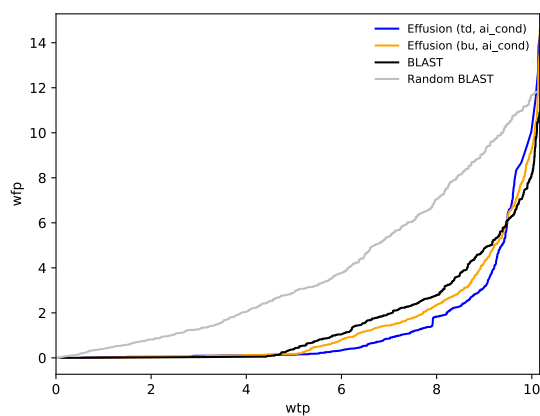
**Figure 2.4:** Performance plots over treated proteins.

looked at the performance of the methods over only those evaluation proteins that had a prediction made by all methods being considered. We call these proteins *treated* proteins. The plots are similar to those of Figure 2.3, and are shown in Figure 2.4. As expected from the evaluation under the full set of evaluation proteins, Effusion generally performed better than BLAST on all metrics.

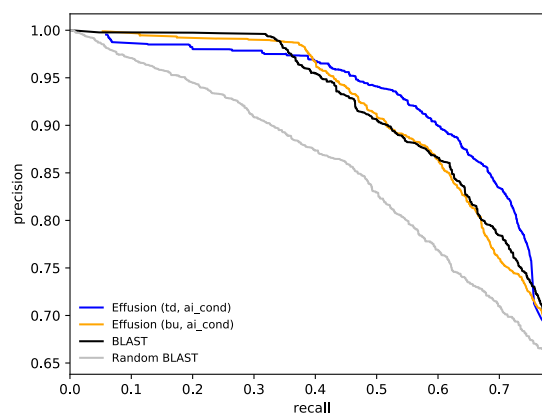
We performed a similar analysis on a per protein basis, essentially comparing 1942 classifiers of Effusion (top-down, ai\_cond) and by BLAST. It is expected that Effusion and BLAST will predict the same GO terms in the same order for many proteins; the methods use the same data, and 538 / 1942 of the queries were associated with protein templates (see [Protein template]) with  $\leq 1$  candidate leaf terms, or networks with  $\leq 1$  proteins with evidence. However, we identified 782 proteins where, for the GO terms predicted by both methods, Effusion reordered the predictions made by BLAST and resulted in a change in performance, according to area under the wfp vs. wtp curve (y vs. x). In general, Effusion accumulated the same bits of true positives, with fewer bits of false positives, for 53% (418 / 782) of the queries ( $p = 0.03$ , binomial test with  $H_0=0.5$ ). The percentage increased to 61.9% (313 / 505) when we limited the analysis to those queries where the query itself did not have evidence.

We were especially interested about the ability of our method to differentiate catalytic activities. This is an important and difficult problem [2]; in functionally diverse enzyme superfamilies, homologous members have evolved to catalyze many different chemical reactions [34], and these proteins are often misannotated in public databases [77]. We performed an additional evaluation constrained to the subset of GO representing these GO terms. Figure 2.5 plots the performance of our methods on the catalytic subset. The performance of Effusion (top-down, ai\_cond) and BLAST differed on 142 enzymes, according to area under the the wfp vs. wtp curve. Effusion outperformed BLAST on 91 / 142 64.1% of the queries ( $p = 0.0005$ ).

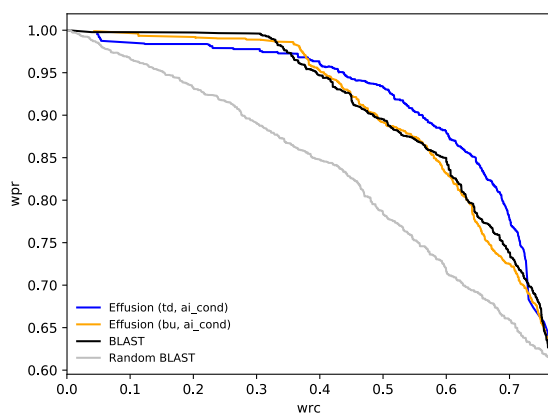
An example from the test dataset that demonstrates Effusion's utility is the protein identified by UniProt Q4ZIL6 in Zebrafish (Figure 2.2). One of the experimental annotations withheld from the test dataset was for testosterone 6-beta-hydroxylase activity (GO:0050649), which is a type of GO:0008395. This protein is not experimentally annotated



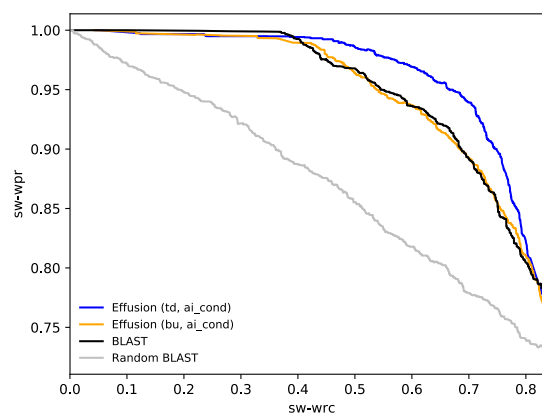
(a) WFP vs. WTP



(b) Precision vs. Recall



(c) Weighted Precision vs. Weighted Recall



(d) Sample-Weighted Weighted Precision vs. Sample-Weighted Weighted Recall

**Figure 2.5:** Performance on catalytic terms. Performance plots over treated proteins. Annotations and predictions were filtered to catalytic terms

to GO:0008404.

Effusion was able to combine evidence for GO:0008395 from the surrounding network context of UniProt Q4ZIL6 (see Figure 2.2 (c)). It correctly predicted GO:0008395 at 36.3% (Figure 2.2 (e)), at which point Effusion predicted no false positives. Effusion also predicted GO:0008404 lower at <1% (Figure 2.2 (f)). Table A.1 details the predictions for UniProt Q4ZIL6 made with Effusion.

BLAST, however, predicted this protein to have GO:0008404 with a probability of 20.0%, based on its proximity to O54750 ( $E$ -value =  $1.13045e-86$ , total bit score 278.87, bit score per residue = 0.53, alignment length = 488, query length = 523, subject length = 488, identities = 161, positives = 259). Compared to Effusion, BLAST accumulated 19 false positives (82.38 bits of information), before predicting GO:0008395 at 17%, based on hitting P13108 ( $E$ -value =  $2.62e-78$ , total bit score 256.914, bit score per residue = 0.49, alignment length = 502, subject length = 502, identities = 167, positives = 256). Predictions for UniProt Q4ZIL6 made with BLAST are shown in Table A.2.

## 2.4.2 Creation and use of protein networks

Effusion networks were generated quickly using BLAST to collect the homologs and DIAMOND to fill in the all-by-all edges that could be used in the MST (Table 2.4, Figure A.1). Effusion succeeded in making networks for 98.3% (2710 / 2757) of the queries. The median number of nodes in the protein network was 268.5, the maximum was 71,286, and 31 networks had only the query. The distributions for the number of nodes in the protein network are shown in Figure A.2.

**Table 2.4:** Statistics for the time to build the sequence similarity networks. Times are rounded to the nearest second

Statistic	Time to Build Network (s)
Mean	1233
Std	3463
Min	10
25%	69
50%	173
75%	705
Max	33607

After reducing the network to the MST and pruning the protein network to proteins that either had evidence or were queries, 899 networks had only the query remaining. This was due either to an absence of proteins with evidence in the original network, or to the more stringent threshold used for collecting the MST edges. The mean number of nodes in the reduced network was 57, if we exclude networks that contained only the query, with a maximum network size of 3284 nodes. 1987 networks had at least one protein with non-root positive preprocessed evidence. The distribution of the number of proteins in the reduced network are shown in Figure A.3, and the distribution of the number of proteins with positive, non-root evidence is shown in Figure A.4.

We compared Effusion to several baselines that do not use a semi-supervised approach to measure the value of adding the network context, shown in Figure A.5. The BLAST-like method used Effusion’s probabilistic framework, but only included network edges from the query to proteins with evidence. The supervised method also used Effusions’s probabilistic

framework, but completely unlabeled proteins were deleted from the protein network and excluded from the subsequent model. The full network version of Effusion, which used the most unlabeled proteins, performed the best, and the BLAST-like version of Effusion, which used the fewest unlabeled proteins, performed the worst.

### 2.4.3 Embedding GO

When viewed as a subontology of GO, the true label of each query protein usually had multiple leaf molecular functions 58.0% (1,598 / 2,757), with a median of 10 GO terms and a median of 2 leaf GO terms. Effusion's subontology of candidate GO terms, derived from the training annotations in the reduced network, also usually had multiple leaf molecular functions 55.8% (1,513 / 2,710), with a median of 9.5 GO terms, and a median of 2 leaf GO terms.

The candidate GO terms overlapped significantly with the terms for the query that were in the standard, but they did not overlap exactly. Methods that predicted all of the candidate terms would have incurred, on average, 8.25 true positives (13.08 bits of information), but also have 7.80 false positives (20.34 bits of information) and 4.79 false negatives (9.86 bits of information).

The speed at which inference can be run is dominated by the number of parents of a variable in the graphical model, which are GO parents in the top-down model, or modeled GO children in the bottom-up model. The distributions for the maximum of these per query are shown in Figure A.6.

We compared the standard top-down method with supplementary negative evidence, the standard bottom-up, which does not require negative evidence, and the top-down method without the supplementary evidence. The performance curves are in Figure A.7. While the

improvement seen here is modest, it was crucial when experimenting with larger networks that included non-homologous proteins (e.g., those found by searching protein interaction databases, data not shown).

Q921C5 provides an example that exemplifies the difference between the models (Figure A.9). Without supplementary negative evidence, the top-down model incorrectly predicted Q921C5 as having clathrin binding (GO:0030276) with a probability of 26.4% that ranked it above a few correct GO terms (Table A.3). This is because the query Q921C5 had evidence for Rab GTPase binding (GO:0017137) and therefore implied evidence for ancestor term protein binding (GO:0005515), and  $P(\text{GO:0030276} \mid \text{GO:0005515})$  is relatively high (11.0%). However, we also have implied evidence for enzyme binding (GO:0019899), and since GO:0030276 is unlikely (35.7%) to co-occur with GO:0019899, we assume this protein does not have GO:0030276. The full table of predictions is shown in Table A.4. Notice that there were some correct GO terms, such as macromolecular complex binding (GO:0044877), that were also predicted at 0% due to incorrect supplementary negative evidence for the query. On the other hand, because the bottom-up model has factors over all the child GO terms, it does not require negative evidence. The bottom-up model predicted GO:0030276 at only 19.6%. A table of predictions for Q921C5 using the bottom-up model is shown in Table A.5.

#### 2.4.4 Data-derived alterations of parameters

The added value of weighting counts by information content of the sample is shown in Figure A.10. An example illustrating the effect of weighting is Serine-tRNA ligase (UniProt P34945), shown in Figure A.11. Without weighting, the probability of a protein being involved in binding (GO:0005488) given its protein network parent has GO:0005488 is 90.7%.



In the network, the probability of correct function decreases quickly, and the query is predicted at 32.6%. After weighting,  $P(x_i^{\text{GO:0005488}} | x_{\text{pa}(i)}^{\text{GO:0005488}})$  increased to 95.2% and the probability for the query having the term increased to 59.3%.

The value of adding pseudocounts to the contingency table is shown in Figure A.12. An example from the top-down model is Serine protease 57 (UniProt Q6UWY2), which had withheld experimental annotation of sulfur compound binding (GO:1901681), is shown in Figure A.13. This GO term is rarely observed experimentally, so the calculated statistic for  $P(x_i^{\text{GO:1901681}} | x_{\text{pa}(i)}^{\text{GO:1901681}})$  was only 73.8%, the probability for this term decayed quickly from the protein with evidence to the query protein, and the query protein was predicted to have GO:1901681 at only 20.6%. After the addition of pseudocounts by the method described above, the calculated statistic for  $P(x_i^{\text{GO:1901681}} | x_{\text{pa}(i)}^{\text{GO:1901681}})$  increased to 92%, and the prediction for the query increased to 56.2%. Pseudocounts were especially beneficial for the bottom-up model, due to its sensitivity to probabilities at the leaves.

### 2.4.5 Inference on real-world protein data

During the discovery phase, we evaluated many of the inference algorithms implemented in OpenGM [4], libdai [62], and top contenders from the UAI Inference Challenge in 2010 [29], 2014, and 2016 (Figure A.15). Many of the algorithms, particularly the standard algorithms, crashed, failed to converge, or converged to obviously incorrect probabilities for a large number of our queries, particularly when running the bottom-up model, which has much larger factors. We selected adaptive inference with conditioning (ai\_cond) [1] for running the test set predictions, based on its consistent performance during the discovery phase. Although this method succeeded in making predictions for most 97.7% (1942 / 1987) proteins with evidence in the reduced network, the software still gave poor results on some

queries; for example, for 8 queries, the software predicted low probabilities for the root molecular function term.

Overall, the performance plots in Figure A.15 show two clusters of methods with similar performance. These clusters correspond to the top-down model and the bottom-up model.

## 2.5 Discussion

By engineering a method and model that accounts for the essential idiosyncrasies of protein function prediction, we were able to make predictions that are practical and more accurate than the standard homology transfer method, using the same primary source of data. Our method, and our evaluation of it, are focused on identifying specific activities, rather than transferring the general function of remote homologs. We believe this is particularly useful to the broad scientific community.

BLAST can be viewed as making predictions on a star shaped network of annotated proteins. Effusion, however, makes a network from data that is discarded by BLAST, and uses it to make predictions that consider a query protein's relative position in a sequence similarity network. Computing the all-by-all similarity matrix for each of many thousand queries is now practical with DIAMOND.

Effusion uses PGMs, which are a natural fit for the problem of predicting protein function for several important reasons. PGMs account for the random error introduced by propagation of functional information. Second, inference on PGMs results in confidence scores (i.e., the probabilities) that are useful to consumers of protein function predictions. Third, PGMs can take advantage of the large amount of unlabeled and partially unlabeled protein data, while most other methods cannot handle such high degrees of sparseness. Fourth, PGMs can seamlessly do structured output prediction, and therefore produce predictions that are

consistent with, and take advantage of, the structure of GO.

There are several reasons for modeling protein function as a hierarchy of Bernoulli random variables. A standard multinomial representation would model the probability of  $K$  mutually exclusive outcomes, where  $K$  is the number of GO terms. Compared to this representation, Effusion directly predicts combinations of GO terms, without assuming that they are mutually exclusive. Second, we can model relationships between GO terms, so that annotations to GO terms related to a particular GO term of interest are considered. In particular, direct and indirect evidence for a general GO term is considered when making predictions about a more specific GO term of interest. Third, it enables the method to predict structured outputs, where the predictions for various GO terms are valid in relation to the structure of GO. It also allows the method to consume structured input with partial labels.

We implemented and evaluated two constructions for the protein template that generated models with different semantics. Our analysis revealed interesting tradeoffs between the two models. Although the bottom-up model has the advantage that it does not require negative evidence, it has other limitations that offset its value. Namely, it is sensitive to leaf probabilities, and since each factor typically depends on more variables than it does in the top-down model, it is also less amenable to inference.

We also evaluated several inference algorithms for use with our method. Although we could not get reasonable predictions for all queries, even with heuristics, we had much more success with a top-performing inference software identified by the UAI competition. We encourage further development of inference algorithms and software, and encourage participation in critical assessments on real world problems.

### 2.5.1 Comparisons to other methods

Many methods for automated function predication benefit from integrating many other types of data [73, 23], but it is very difficult to perform well controlled comparisons to each method. Therefore, to evaluate our methods against a large collection of diverse methods, we submitted predictions from a preliminary version of Effusion in the 2017 CAFA challenge (results and manuscript in preparation by the CAFA consortium).

Effusion’s main similarities to the method described in Carroll and Pavlovic [17] and Mitrofanova, Pavlovic, and Mishra [61] are the general use of protein networks, modeling of GO, and a probability model. However, there are fundamental differences with these works in each aspect of our methods. The previous methods used a model whose factors lack a global probabilistic interpretation, and since learning maximum likelihood parameters on a per network basis would have been infeasible, they used normalized similarity scores. Effusion, on the other hand, has a model whose factors are simply conditional probability functions, and therefore we can learn maximum likelihood parameters from all available experimental data. Significantly, this allows us to incorporate parameters that are specific to each GO term, thereby giving us reasonable results on problems that have a wide range of GO terms, rather than limiting predictions to only a few (i.e.,  $< 10$ ) GO terms. Additionally, our formulation allows the incorporation of data derived pseudocounts, inference with general purpose, rather than ad-hoc, inference algorithms, and ranked probabilities for all GO terms, rather than setting an arbitrary threshold for prediction. We’ve shown that the differences between these methods result in drastically different performance.

While it is common for labs to provide the algorithm for predicting protein function, they rarely provide the pipeline necessary for preparing the data and parameters, making it difficult to compare methods on the same training and test data. Fortunately, the high-

throughput pipeline for SIFTER was available. This method is highly relevant because it is based on sequence similarity, its aim is to discern functions among relatively close homologs rather than predicting general functions from remote homologs, and also uses a Bayesian network. There are substantial differences, however. rather than BLAST for homologous sequences, SIFTER uses Pfam [71] and uses its alignments to build phylogenetic trees. It uses a continuous time Markov chain in its model for protein function evolution, but does not use GO. Whereas Effusion currently requires edges to reflect alignment across the entire sequence, SIFTER combines results from all the Pfam domains in a query protein. While we have shown that Effusion outperforms SIFTER, we have not determined which of the differences in method result in the difference in performance.

## 2.5.2 Directions for future research

In this report, we describe the first version of Effusion, a simple, high performing method that suggests specific protein function with a model that uses protein networks and incorporates GO. Although Effusion only uses sequences that are highly similar across their entire lengths, and only models the molecular function aspect of GO, it is designed to be extendable to model additional aspects of protein function, and additional, more finely grained relationships between proteins and GO terms. By making use of resources such as domain-centric GO (dcGO)[33], we could include sequences that align only over a domain, and propagate function at higher granularity.

Because of the high availability of functional association data [83], and the strong relationship between a protein's biological process and its molecular function [12, 69], we are especially interested in extending Effusion by modeling each protein's biological process. This extension could scale manual genomic context analysis, which uses a protein's genomic

context and pathway information to infer a protein’s molecular function [74, 90, 91]. Calculating and incorporating potential ligands and substrate substructures could also be useful for discerning substrates from biological processes associated with enzymatic function and metabolism. We are also interested in modeling the cellular component of a protein, because of the availability of unbiased, high-throughput experimental data for this aspect of a protein, and the extent to which the location of a protein relates to its molecular role.

A probability distribution, including a PGM, can be used for other purposes besides computing marginal probabilities. Eventually, we envision the use of Effusion to identify targets for experimental characterization that would minimize the overall uncertainty in our belief of protein functions.

Since we only considered MST edges, Effusion could not perfectly capture network boundaries (see Figure 2.2). We think it would be valuable to pursue using more of the network edges, as long as we could still use all the experimental data to calculate parameters and priors. One approach would be to adopt a phylogenetic tree as the topology of our graphical model, which was shown to be beneficial for SIFTER [31].

Finally, there are additional ways that could be investigated to improve learning the parameters. For example, we could differentiate the functions and annotations of each protein, and model the functions as a latent random variable, resulting in a tree-structured HMM. We could also use a more inclusive neighborhood function when counting annotation changes over neighboring functions, or use a continuous Markov chain rather than the discrete one that we currently use.

## Chapter 3

# Effusion GCA: Prediction of Protein Function with a Probabilistic Model for Analysis of Genomic Context

### 3.1 Abstract

#### 3.1.1 Motivation

There are not enough function annotations to delineate functional boundaries using sequence similarity alone. Manual analysis of genomic context has been successfully used to infer functions in the absence of characterized close homologs or in the presence of conflicting evidence. Generalizing and scaling this approach may improve the performance of prediction algorithms and the quality of genome annotation. However, few methods to date have attempted to automate genomic context analysis for the prediction of protein function.

#### 3.1.2 Results

We present *Effusion GCA* a method that propagates – in a statistically rigorous way – information about a protein’s biological process among functionally associated proteins, and

uses that information in its simultaneous prediction of molecular function. We evaluate the performance of Effusion GCA, similar published methods, and various baselines using a critical method that reflects that real world task of function annotation. We also analyze the value of each component of the method. Finally, we discuss the results and propose future directions.

## 3.2 Introduction

Effusion, presented in the last chapter, establishes a framework for prediction of protein function via network analysis. The only feature of each protein that the method considered was pairwise sequence similarity. However, sequence similarity alone is insufficient for precisely differentiating protein function, and manual network analysis typically considers a broad array of distinctive features in the sample set, such as the residues in conserved positions in an alignment. This chapter attempts to extend Effusion in order to accommodate another analytical technique from computational biology. Analysis of genomic context has been useful in generating hypothesis for a protein’s molecular function in the absence of characterized homologs, and useful for pinpointing a protein’s function in the presence of conflicting evidence. We hypothesize that the incorporation of this analytical technique into Effusion will result in a method capable of achieving greater predictive performance.

While the goal is to automate genomic context, the specific approach is to proceed by asking: how can we use data about functionally associated proteins to predict molecular function? The method proposed here, *Effusion GCA*, represents one attempt.



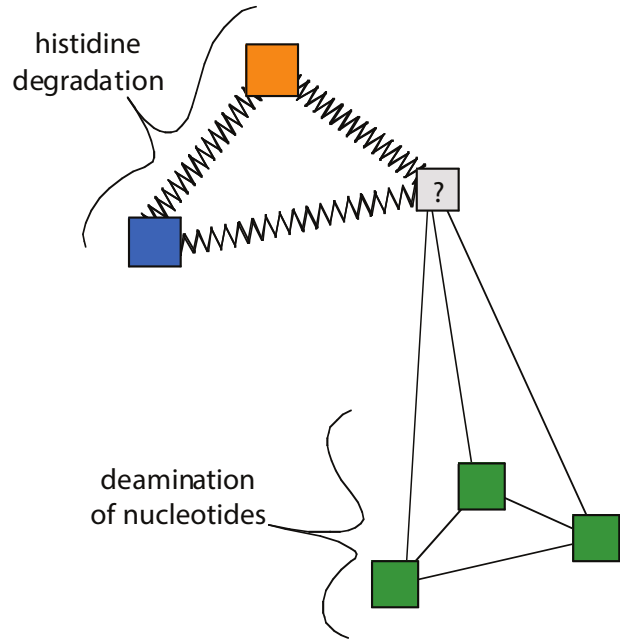
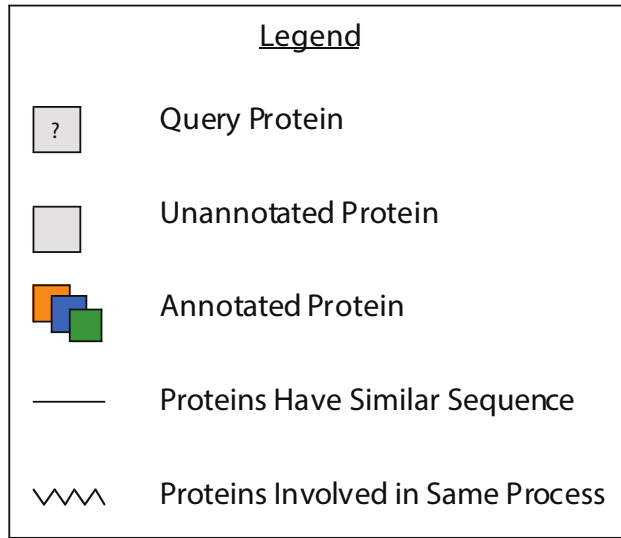
### 3.2.1 Genomic context analysis

A motivating example for the use of genomic context comes from Ricardo et al. [74], where a protein of unknown function was inferred from manual analysis of genomic context (see Figure 3.1b). The query protein was similar in sequence to nucleotide deaminases, and was therefore annotated as a putative nucleotide deaminase. However, the query protein also shared the genomic context of proteins in the histidine degradation pathway. Since histidine is not a nucleotide, the authors doubted the putative function. Instead, they hypothesized and validated that, rather than a nucleotide deaminase, the protein was actually a N-formimino-L-glutamate deiminase.

In the manual analysis just described, the query protein had characterized homologs and members of its operon were also characterized. However, a common scenario is where none of the close homologs to the query are characterized, and the either the query is not in an operon, or the members of the operon are not characterized (Figure 3.1c). The lack of information directly related to the query is unpromising, but a network view of this problem gives us hope that information can propagate to the query from more distantly related proteins. However, finding an assignment of functions to all the proteins that explains what's going on in the network is starting to get tough, and the actual scale of the problem cannot be done manually (Figure 3.1d).

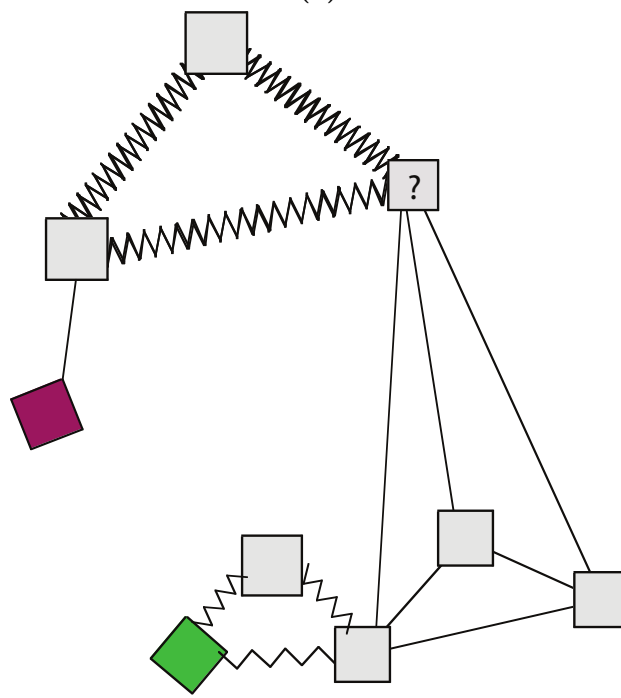
### 3.2.2 Functional associations

Functionally associated proteins generally describe two or more proteins that work together to carry out cellular functions. For example, functionally associated proteins may be involved in the same metabolic pathway, have an activator – enzyme relationship, or are components in a multiprotein complex. Functional associations can be detected through a combination

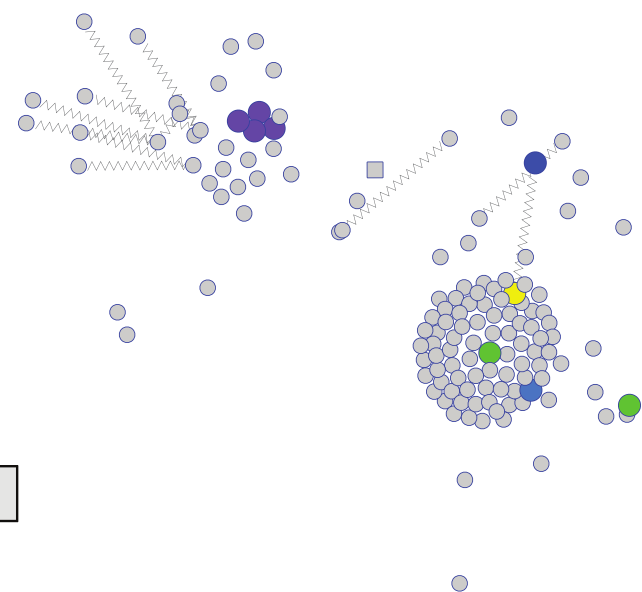


(a)

(b)



(c)



(d)

**Figure 3.2:** GCA case study. (a) Legend. (b) The approximate scale of the manual analysis in Ricardo et al. [74]. (c) The query is neither similar in sequence to any annotated protein, nor is it functionally associated with any annotated protein. The network view, however, suggests that it may still be possible to glean functional information about the query, but manual analysis would be very difficult. (d) A real-world SSFA network. It was generated using an early method that iteratively searched BLAST and STRING.

of experimental assay and computational inference, through techniques such as yeast two-hybrid screening, tests for synthetic lethality or genetic interaction, text-mining, phylogenetic profiling, genomic proximity, co-localization, and interolog inference.

STRING [58, 83] is a database of known and inferred functional associations. It also combines multiple data sources in order to give a combined score indicating the confidence of a functional association. For the dataset used in this work, STRING had 46,049,474 high confidence (combined score  $> 900/1000$ ) functional associations. STRING reports that the newer STRING v10.5 has 1.380 billion interactions, 2031 organisms, and 9.6 million proteins.

There are other sources that aggregate functional association data. MetaCyc provides well-curated pathway data [18]. BioGRID is a curated repository [19].

### 3.2.3 Functional associations networks

Pairwise functional association data are often represented as functional association networks. Analysis is often performed on these networks to identify topological features, such as hubs and motifs, that may be of biological significance. They have been used, for example, to identify potential drug-drug interactions.

### 3.2.4 Quantification of association between aspects of function

There have been several quantitative studies that measure the association between molecular function and biological process [7, 12, 40, 70, 50, 84, 11, 52, 68, 10, 53, 69]. These studies typically identify functional associations to enhance the structure of GO or to compare genes.

In addition to identifying novel pairs of related terms and new methods for identifying such relations, this research has yielded other insights, such as the ability to reconstitute manually derived associations, and the observation that different methods produce distinctive results.

### 3.2.5 Related work

#### Network-based methods

Many methods propose a novel method for inferring functional associations [55, 65, 67] or for combining sources of functional association data [54, 89], in order to predict protein function. These methods typically focus on predicting biological processes (BPs) [26, 87, 41, 63, 45].

Some methods predict MF via functional association, typically via protein-protein interaction networks that combine functional associations with sequence similarity. These methods propagate function without regard to the aspect the function: they assume that functionally associated proteins are likely to share the same BP *and* MF. The latter assumption, however seems unlikely; it seems more common that two proteins in the same pathway have different MFs. Expectedly, this assumption has two outcomes. Either methods that use functional association data perform better at predicting BP, or the method learns parameters that indicate functional association is relatively uninformative for predicting MF [60].

In a recent assessment of protein function prediction methods, only one of 17 MF predic-

tion methods self reported the use of genomic context. This method, ffPred [23], was one of the best performing methods. However, it integrated genomic context data while discarding biological knowledge about genomic context that may improve prediction of MF.

### 3.3 Methods

The method builds on Effusion, and is called *Effusion GCA*. Effusion models the MF of each protein, and has factors that include corresponding MF terms for each pair of proteins that are neighbors in the reduced network. Effusion GCA additionally models the BP of each protein. The SSFA network has edges that represent either sequence similarity or functional association. Effusion GCA has factors that include corresponding BP terms for each endpoint for each edge that represents functional association, in addition to the factors linking corresponding MF terms for sequence similarity edges.

#### 3.3.1 Preprocessing

In addition to the data used for the sequence similarity-only method, we also downloaded:

- functional associations: STRING, version 10

#### 3.3.2 Building the SSFA network

For a given query protein, we build a SSFA network, where nodes represent proteins and edges represent either pairwise sequence similarity or functional association. This network is quickly constructed by collecting homologs with BLAST, collecting functionally associated proteins from STRING, and then using DIAMOND to fill in the all-by-all sequence similarity edges of the network. BLAST is run with a permissive  $E$ -value threshold ( $1e-8$ ), but limited

to global alignments (`qcov_hsp_perc = 90`), and filtered to sequences with a bit score per residue of at least `.25`. STRING is queried for functionally associated proteins with a combined score of 90%. We use the resulting sequences to format a DIAMOND DB, and search each resulting sequence against this DB using DIAMOND (`evalue = 1e-8`, `query-cover = 90%`, `subject-cover = 90%`), and more restrictive parameters (`max-target-seqs = 1000`, minimum bit score per residue of  $1.4 \times .25$ ).

### 3.3.3 Constructing the probabilistic model

As with Effusion, the model network is made tractable by converting it to an MST, pruning it to nodes with evidence, and directing it outward from the query.

The graphical model then represents the following joint distribution:

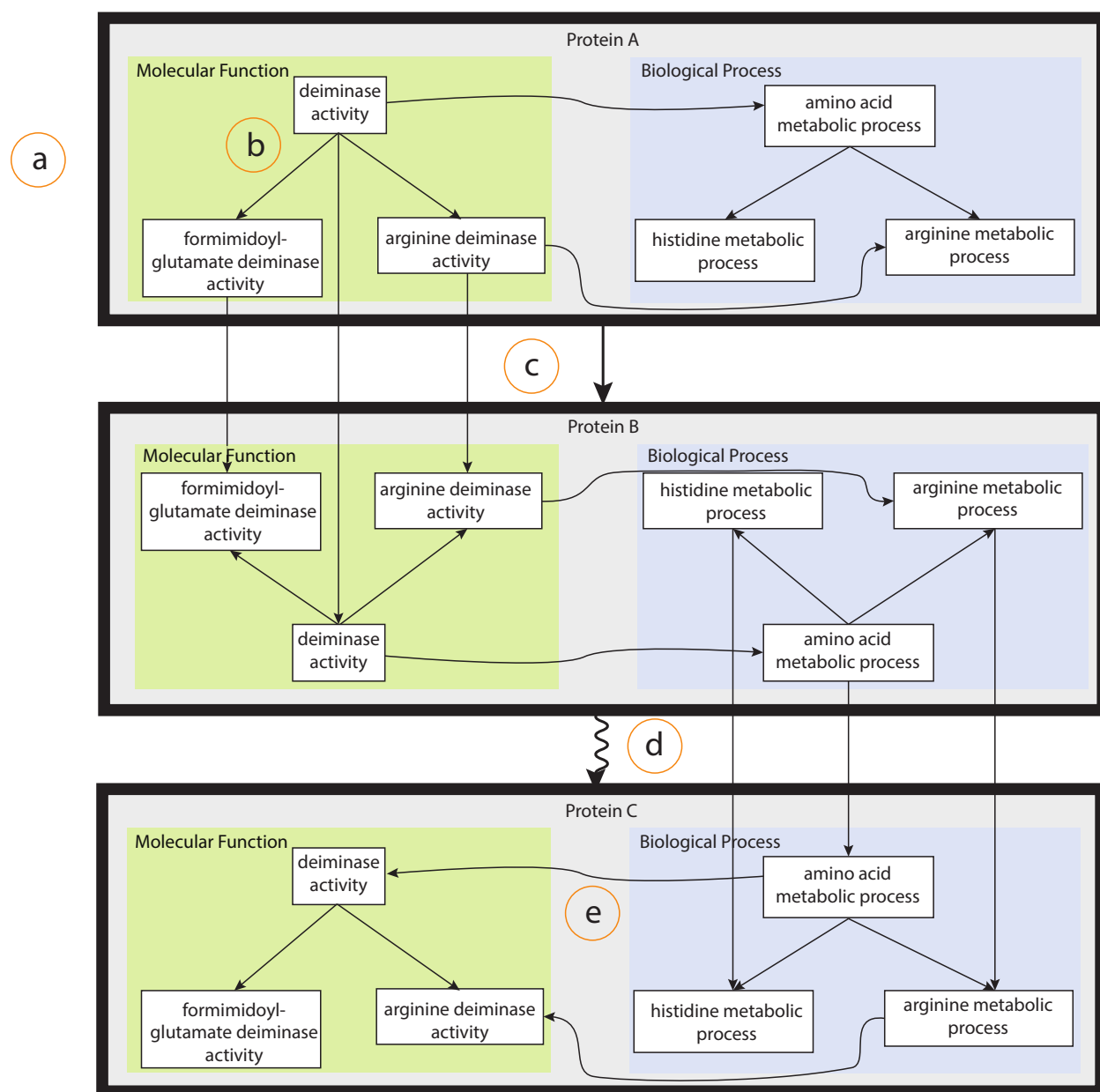
$$P(X = x) = \prod_{x_i \in x} P(x_i | \text{pa}(x_i))$$

where  $x := x_1, x_2, \dots, x_N$  and  $\text{pa}(x_i)$  represents the model parents of variable  $x_i$ .

A PGM enables us to build a probabilistic model from smaller components. The components of this model are shown in Figure 3.4.

#### Protein template

The protein template is based on the protein template in Effusion GCA, but is extended with a representation for the protein's BP. The representation for BP mirrors that for MF: the aspect of protein function is modeled as a hierarchy of Bernoulli random variables reflecting a subgraph of the biological process ontology of GO. Every GO term for which there is a positive or negative preprocessed annotation in the SSN is a candidate GO term, with two



**Figure 3.4:** GCA model. (a) A PGM for a network of 3 proteins. (b) The MF of each protein is represented by a hierarchy of Bernoulli random variables with the topology of GO. (c) Proteins that are similar in sequence are likely to have similar MFs. (d) Proteins that are functionally associated are likely to have similar BPs. (e) Correlations are modeled between ontologies so that the predicted functions for an individual protein are consistent.

exceptions. For reasons of computational practicability, candidate BP terms are filtered: (1) they must be a descendent of metabolic process (GO:0008152), or (2) be an ancestor of GO:0008152. For example, superoxide metabolic process (GO:0006801) might be a candidate GO term because it is descendent of GO:0008152, but an ancestor of GO:0006801, such as cellular process (GO:0009987) would be excluded, because it is not a metabolic process.

As with Effusion, we implemented both a top-down (parent GO terms are parents in the Bayesian network, referred to as *td*) and bottom-up model (parent GO terms are children in the Bayesian network, referred to as *bu*).

We add edges between MF terms and BP terms within a single protein, allowing belief to flow between the two aspects of protein function, and reducing the influence of assignments with unlikely combinations of terms. For reasons of computational tractability, rather than add edges between every combination of terms from each aspect, we aim to add the most informative edges, with the justification that each additional edge is one less independence assumption, and therefore should improve the results, as long as there is enough data to learn stable parameters. Including some inter-ontological edges and not others may introduce bias, but we have not determined the extent or effect of this bias.

The algorithm for determining the inter-ontological parents of a candidate term  $g$  in the top-down model is as follows. We collect up to 1000 proteins that are positively labeled with  $g$ , and up to 1000 proteins that are positively labeled with the parents of  $g$ , but not labeled with  $g$ . We calculate their labels, assuming a value of zero for GO terms without a positive preprocessed annotation. We then use the scikit-learn feature selection function `mutual_info_classif` to calculate the mutual information between the binary class  $g$  and the values for the candidate functions in the other ontology, and take the top  $iop$  candidate functions in the other ontology. Our analysis is based on  $iop = 1$ .



### Inter-protein edges

As with Effusion, when two proteins have similar sequences, we connect the corresponding terms in MF. This adds a factor that says that two proteins that are similar in sequence are likely to have similar values for each molecular function.

When the edge between two proteins represents a functional association, then we associate their corresponding BP terms.

Like the edges due to sequence similarity and the edges due to functional associations, inter-ontological edges are also directed. Every protein  $i$  in the reduced network except the root has a parent protein  $\pi(i)$ . If  $(\pi(i), i) \in E_{SS}$  in the reduced network, then there is a directed edge between each corresponding MF term from  $(\pi(i), i)$ , and a directed edge  $(IOP(t), t)$  for each BP term  $t$  that has an inter-ontological parent. On the other hand, if  $(\pi(i), i) \in E_{FA}$  in the reduced network, then there is a directed edge between each corresponding BP term from  $(\pi(i), i)$ , and a directed edge  $(IOP(t), t)$  for each MF term  $t$  that has an inter-ontological parent. We are free to choose which ontology is the parent of the other ontology for the root node; we chose BP  $\rightarrow$  MF.

### 3.3.4 Parameter learning

There are three sets of parameters derived from the structure of the graphical model described in the previous section: the probability of protein  $i$  having term  $t$ , given the parents of  $t$  for protein  $i$ , and either the corresponding  $go$  term for the blast parent, the corresponding term for the string parent, or the inter-ontological parents for the same protein. That is,

$$P(x_i | \text{pa}(x_i)) = \begin{cases} P(x_t^i | x_{\text{GO kin}(t)}^i, x_t^{\text{BLAST parent}(i)}) & \text{if } (x_t^{\text{BLAST parent}(i)}, x_t^i) \in E_{\text{BLAST}} \\ P(x_t^i | x_{\text{GO kin}(t)}^i, x_t^{\text{STRING parent}(i)}) & \text{if } (x_t^{\text{STRING parent}(i)}, x_t^i) \in E_{\text{STRING}} \\ P(x_t^i | x_{\text{GO kin}(t)}^i, x_{\text{interont. parent}(t)}^i) & \text{if } (x_{\text{interont. parent}(t)}^i, x_t^i) \in E_{\text{interont.}} \end{cases}$$

They are learned from weighted contingency tables with pseudocounts derived from contingency tables of similar terms, as described in the previous chapter.

## 3.4 Results

### 3.4.1 Comparative analysis

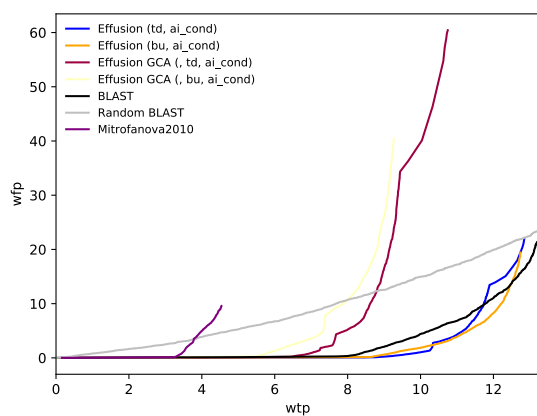
The dataset for the test phase contained 2757 proteins that had an IDA annotation to the MF ontology dated in 2016.

The BLAST-based method made non-root MF predictions for 74.3% (2048 / 2757) of the evaluation proteins. Effusion GCA (top down, adaptive inference with conditioning) made non-root MF predictions for 60.0% (1655 / 2757) of the total.

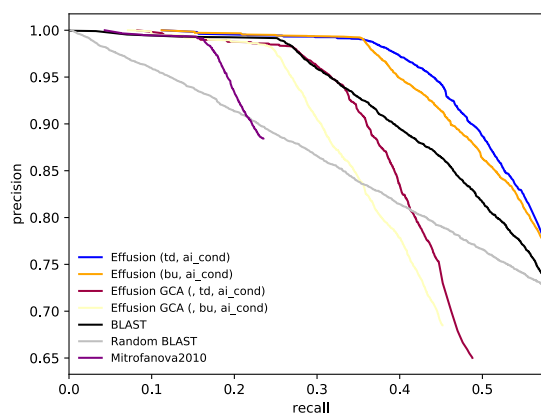
We compared Effusion GCA to Effusion, the most similar published method (Mitrofanova2010), and informative baseline methods.

We compared the performance over the proteins treated by Effusion GCA, Effusion, Mitrofanova2010, and the baselines. Mitrofanova2010 was excluded from evaluations over treated proteins because its low coverage would severely reduce the size of the evaluation set.

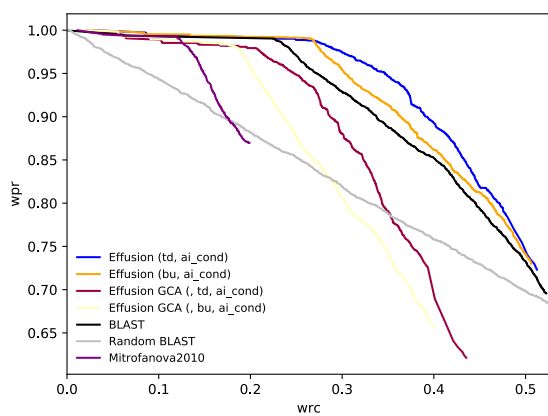
We compared the performance Effusion GCA, Effusion, Mitrofanova2010, and the baselines on the catalytic subset.



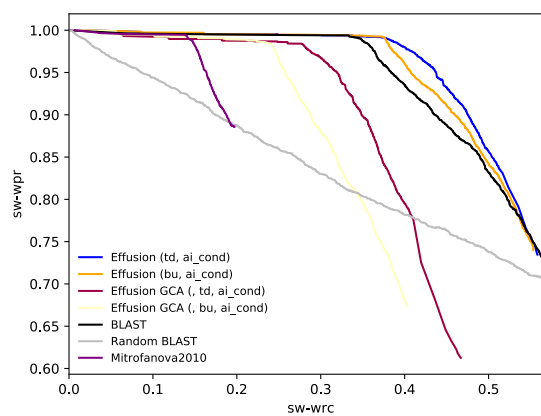
(a) WFP vs. WTP



(b) Precision vs. Recall

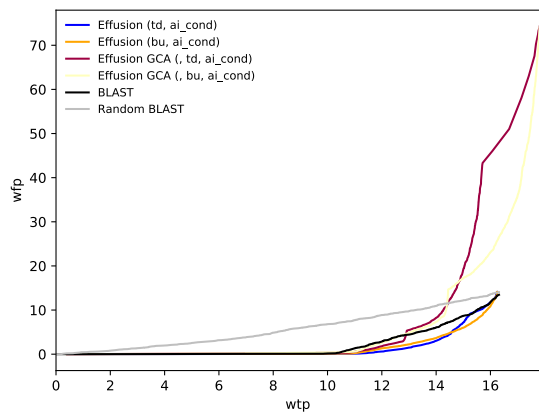


(c) Weighted Precision vs. Weighted Recall

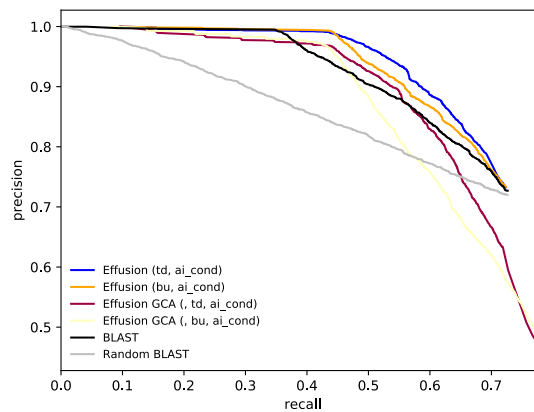


(d) Sample-Weighted Weighted Precision vs. Sample-Weighted Weighted Recall

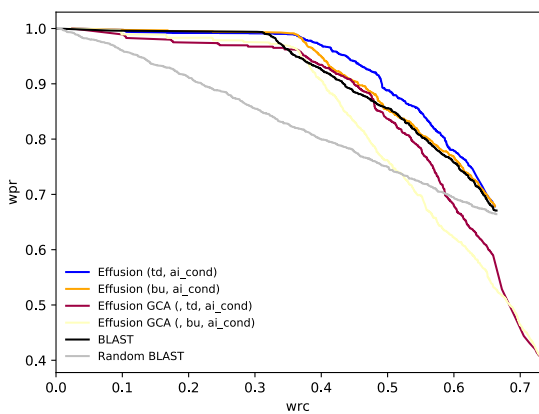
**Figure 3.5:** Performance plots over all proteins in the test set, regardless of whether any of the methods failed to make predictions.



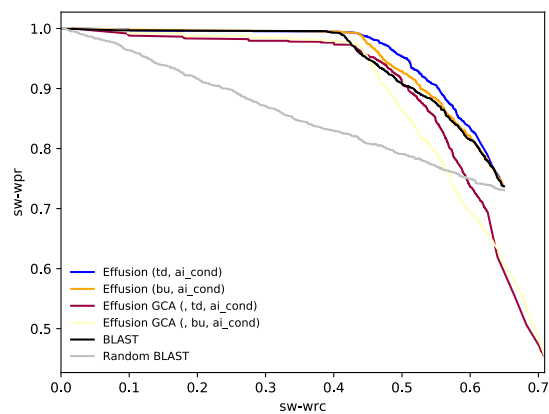
(a) WFP vs. WTP



(b) Precision vs. Recall

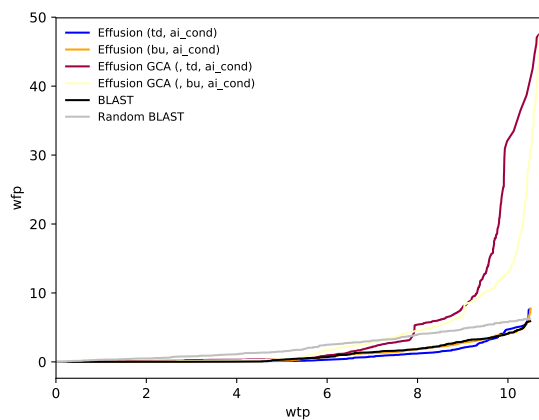


(c) Weighted Precision vs. Weighted Recall

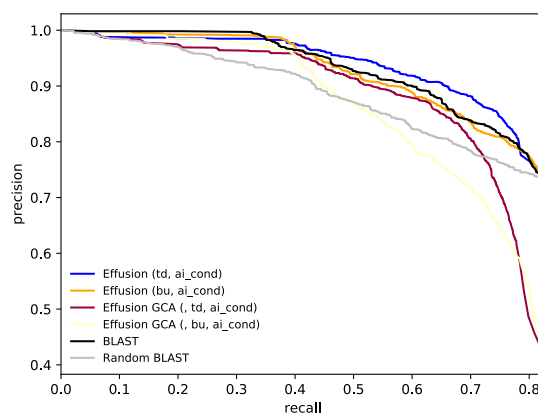


(d) Sample-Weighted Weighted Precision vs. Sample-Weighted Weighted Recall

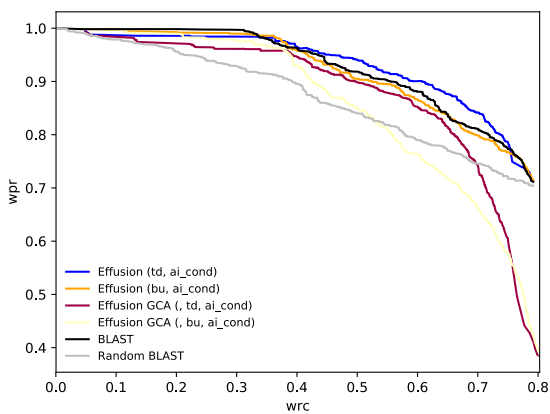
Figure 3.6: Performance plots over treated proteins.



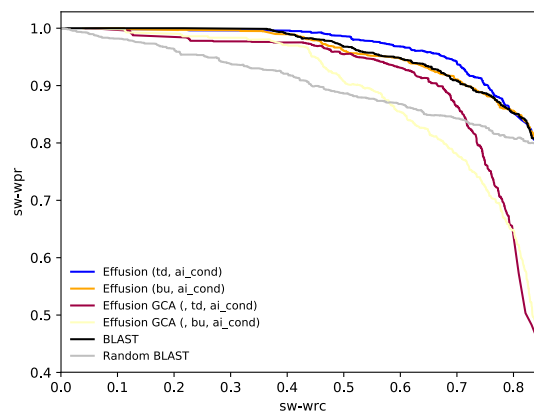
(a) WFP vs. WTP



(b) Precision vs. Recall



(c) Weighted Precision vs. Weighted Recall



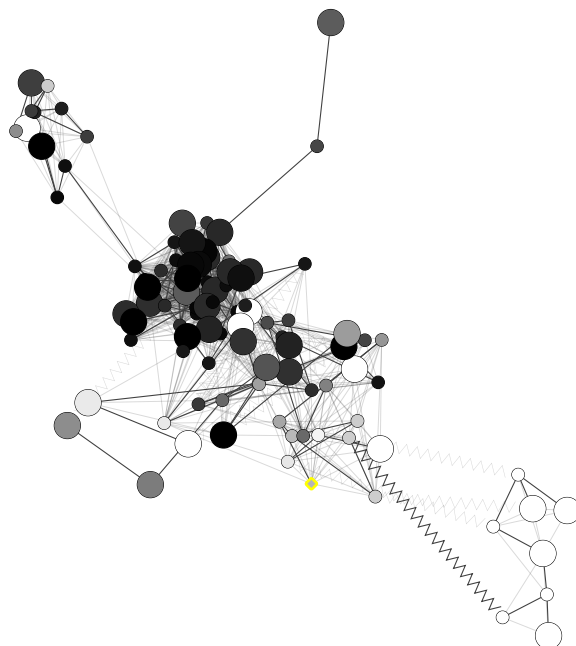
(d) Sample-Weighted Weighted Precision vs. Sample-Weighted Weighted Recall

**Figure 3.7:** Performance on catalytic terms. Performance plots over treated proteins. Annotations and predictions were filtered to catalytic terms

We compared Effusion GCA (top-down, ai\_cond) and Effusion (top-down, ai\_cond) on a per protein basis, by comparing the 1,496 classifiers of each that both had a non-root MF annotation, and measuring performance by area under the curve (AUC) under the WTP-WFP (x-y) curve, when the evaluation was limited to the GO terms predicted by both methods for the respective query. 1,229 of the classifiers had the same AUC. Effusion GCA performed better on 121 of the queries. Effusion performed better on 146 of the queries. When examining the predictions by the best and worst performers, we noticed that many of them had unreasonable probabilities. Effusion GCA was uncertain that 45 of the query proteins had the root MF term. In comparison, Effusion was certain of the root MF term for all except 8 of the query proteins.

An example where Effusion GCA did well compared to Effusion is gamma-aminobutyric acid receptor subunit epsilon (UniProt P78334). Effusion GCA predicted chloride transmembrane transporter activity (GO:0015108) at 29.1% before predicting excitatory extracellular ligand-gated ion channel activity (GO:0005231) at <1%. Meanwhile, Effusion predicted GO:0005231 at 89.2% and eventually predicted GO:0015108 at <1%. The performance plot is shown in Figure B.1, and the predictions are in Table B.1 and Table B.2.

Failed inference was responsible for all of the worst 6 queries Effusion GCA compared to Effusion, as measured by AUC under the WTP-WFP curve. As the 7th worst prediction by Effusion, Alpha-ketoglutarate-dependent dioxygenase alkB homolog 3 (UniProt Q96Q83) had only predictions with probabilities close to 0 or close to 1. Therefore, we suspect that inference for this query failed to converge, and the performance of Effusion GCA in general was limited by the performance of the inference algorithm.

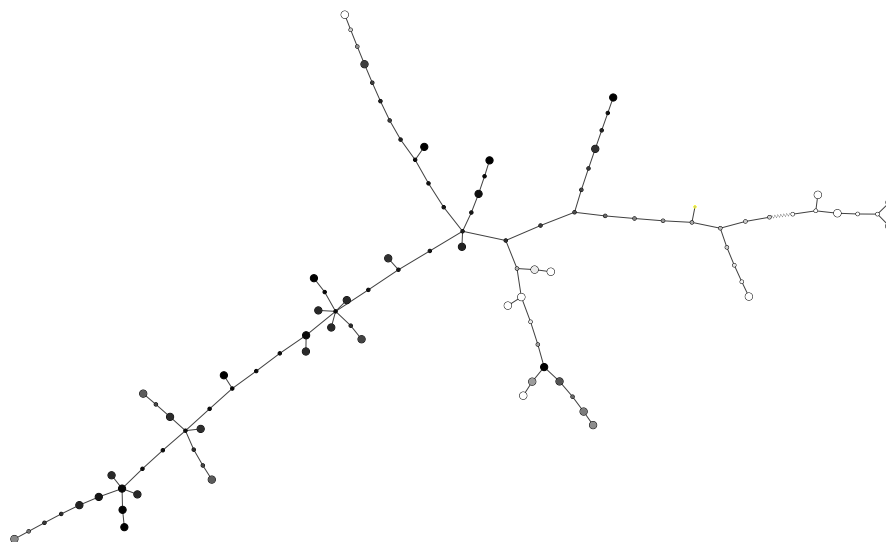


**Figure 3.8:** SSFA network for UniProt P78334. The nodes are pruned to the same nodes as in the reduced network, and colored by probability of GO:0015108.

### 3.4.2 Creation and use of SSFA networks

Figure 3.8 shows an SSFA network. The subgraph induced by the dark edges is the reduced network, after taking the MST, directing, and pruning to evidence. The whole network is actually a subgraph induced by the nodes that exist in the reduced network. Figure 3.9 applies the Prefuse Force Directed layout to the reduced network to highlight the tree structure.

The method could create SSFA networks reliably and quickly (Table 3.1, Figure 3.10). Effusion GCA succeeded in making networks for 98.3% (2710 / 2757) of the queries. This is the same number as the Effusion (sequence only) method; Effusion GCA added a number of sequences that were, usually, non-homologous to the query, but these were only used as additional input sequences to DIAMOND run on a database of sequences homologous to the query, and therefore, did not prohibitively increase the number of edges.



**Figure 3.9:** Reduced SSFA network for UniProt P78334. The nodes are colored by probability of GO:0015108.

**Table 3.1:** Statistics for the time to build the SSFA networks. Times are rounded to the nearest second

Statistic	Time to Build Network (s)
count	2710
mean	2277
std	3748
min	121
25%	762
50%	1176
75%	1987
max	34219

Statistics for the number of proteins that were added to the protein network by querying STRING is shown in Table 3.2. The network reduction pruned many of these STRING proteins. The statistics for the number of functionally associated proteins remaining in the reduced network is also shown in Table 3.2. We report statistics based on a smaller number

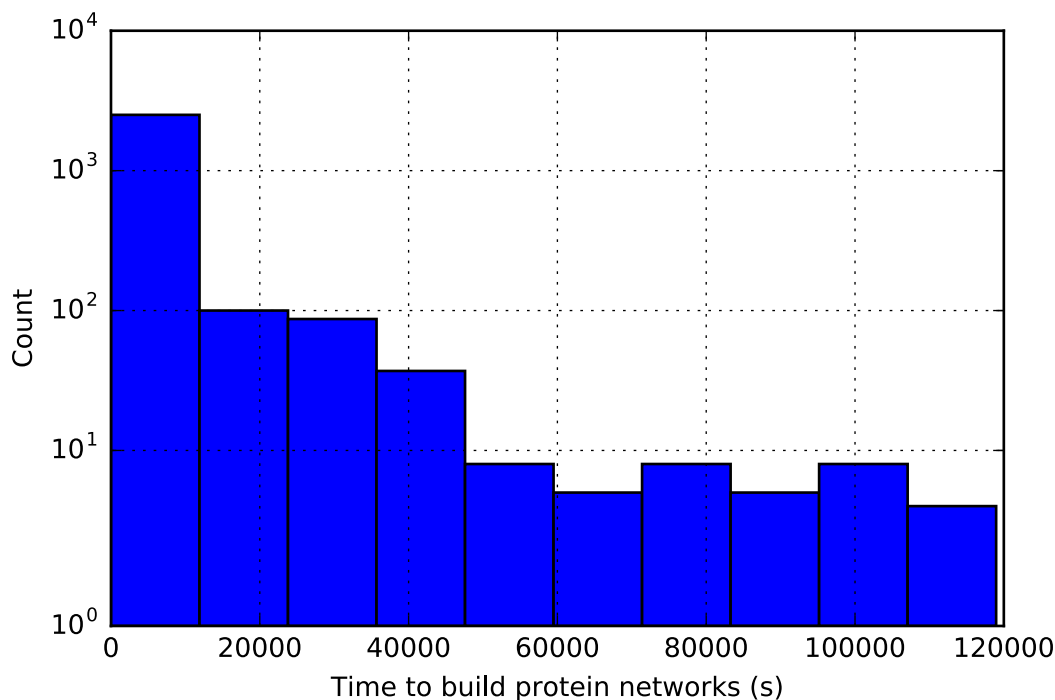


Statistic	FA proteins in reduced SSFA	
	FA proteins in full SSFA network	network
std	11264	30
min	0	0
25%	6	0
50%	137	2
75%	1315	16
max	118969	369

Importantly, whereas Effusion had 899 reduced SSN with only one node and 565 problem instances with no positive evidence, Effusion GCA had only 398 reduced SSFA networks with only one node and only 269 problem instances with no positive evidence.

### 3.4.3 Parameters for functional association edges

We investigated the multi-label similarity of two functionally associated proteins, excluding functionally associated pairs with either protein lacking any annotation for the aspect being considered. We only counted terms that were annotated by either of the two proteins, so that the counts would not be dominated by terms not annotated by either protein. We computed this for functional associations used to compute the parameters, and for the functional associations that occurred in the reduced networks. We report the results, based on 10,000 sampled functional associations, in Table 3.3. We were surprised that two functionally associated proteins had such similar MFs, even when we weighted the samples and the terms by information content. We noticed that there were many identical or highly similar pairs of MFs that were labeled with terms such as structural constituent of ribosome (GO:0003735)



**Figure 3.10:** Distribution of time to build the network

or GO:0005488. This may also have been due to differences in the structures and annotations between MF and BP. Either way, this suggests that it may be useful to differentially propagate terms within different sub-ontologies (e.g., binding, catalytic activity, structural molecule activity) for the same GO (e.g., MF). That is, some sub-ontologies of MF, such as GO:0003735, might be worth propagating over sequence similarity edges. However, we still assume that functionally associated proteins are less likely to have similar catalytic activities than proteins with highly similar sequences, and we are focused on a model that will best distinguish catalytic activities, rather than predicting whether a protein is capable of binding.

**Table 3.3:** Statistics for the number of functionally associated proteins in the full SSFA network and in the reduced SSFA network.

Statistic	in params	in reduced network
BP similarity by binary %	28.0%	28.4%
BP similarity by IC %	24.2%	22.3%
BP similarity by SW-IC %	42.3%	35.4%
MF similarity by binary %	29.4%	34.4%
MF similarity by IC %	22.9%	31.7%
MF similarity by SW-IC %	40.7%	50.3%
catalytic similarity by binary %	21.3%	8.82%
catalytic similarity by IC %	18.4%	7.54%
catalytic similarity by SW-IC %	22.7%	9.10%
MF+BP similarity by binary %	29.0%	31.6%
MF+BP similarity by IC %	23.8%	25.8%
MF+BP similarity by SW-IC %	39.8%	38.9%

We looked deeper into this by inspecting the raw contingency tables. As a baseline, we look at carbohydrate biosynthetic process (GO:0016051). The background probability for GO:0016051 is low (0.575%). However, given that a protein is functionally associated with a protein that is annotated to GO:0016051, the probability increases to 24.7%. Remarkably for non-catalytic molecular function GO:0003735, whereas the background probability is 3.13% the probability increases to 84.7% when the protein is functionally associated with another structural constituent of ribosome. However, if we look at an example of a catalytic activity, hydrolase activity, acting on carbon-nitrogen (but not peptide) bonds, in linear amidines

(GO:0016813), the background probability of the term is 0.0233%, and the probability of GO:0016813 given a functionally associated protein is only 8.33%. On the other hand, if the protein is similar in sequence to a GO:0016813, then our belief that the protein is also a GO:0016813 goes to 100%.

Mitrofanova, Pavlovic, and Mishra [61] claims that it is infeasible to learn a set of parameters for each GO term, and therefore uses a shared set of parameters for all GO terms, similar to those in Table 3.3. Their paper did not have a completely specified algorithm, but we calculated their parameters as follows. For each query, we determine the candidate ontology, sample 1000 functional associations where both proteins are annotated to all aspects within the candidate ontology. For each functional association, we construct the multi-label for each protein, such that unannotated terms are assumed negatively annotated, and count the instances. The final counts are normalized. The parameters calculated for query UniProt P78334 are shown in Table 3.4. As expected  $\psi(-, -)$  is the most likely configuration. Somewhat less expected is that  $\psi(+, +)$  is less likely than either of  $\psi(+, -)$  and  $\psi(-, +)$ . We did not expect these parameters to perform well, and we realize that assuming negative annotations for terms without annotations is pessimistic, but we were generous in limiting our counting to functional associations with annotated proteins and terms in the candidate ontology, and they did not suggest an alternative in text or in code.

**Table 3.4:** Mitrofanova parameters used for problem instance based on query UniProt P78334

Parameter	Value
$\psi(+, +)$	8.35%
$\psi(+, -)$	1.84%

Parameter	Value
$\psi(-, +)$	1.69%
$\psi(-, -)$	88.1%

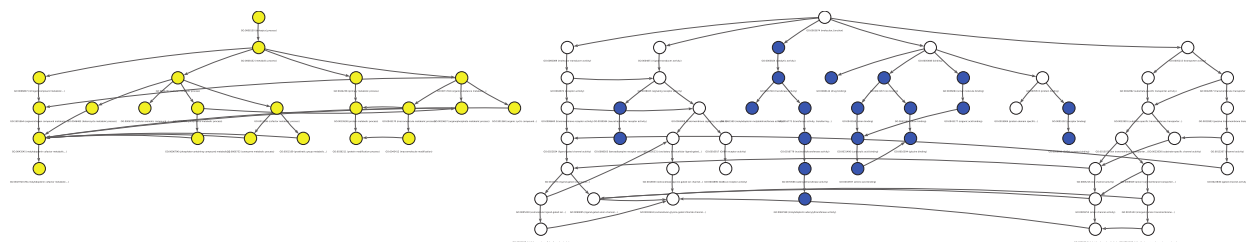
### 3.4.4 Protein template

An example protein template, including biological process, but excluding the inter-ontological edges, is shown in Figure 3.11.

The number of candidate GO terms in the model produced by Effusion GCA was typically much larger than the model produced by sequence-only Effusion. Effusion GCA collects candidate MFs from non-homologs in addition to homologs, so Effusion GCA greatly increases the number of candidate MF terms, and greatly increases the maximum false positives. Methods that predicted all of the candidate terms would have incurred, on average, 9.09 true positives (14.05 bits of information), but also have 31.64 false positives (74.84 bits of information) and 4.25 false negatives (8.94 bits of information). These additional MF terms are shaded blue in Figure 3.11.

There is a performance cost associated with the modeling additional MF terms, due to the added difficulty for inference, and due to limitations in the model discussed in the previous chapter. We implemented variants of our method to measure the costs of a more complex protein template. The *functionally-associated molecular functions (FAMF)* methods start with the sequence similarity-only Effusion, and add the additional MF terms. We show these results for the top-down model with and without supplementary negative evidence in Figure 3.12, and the results for the bottom-up model in Figure 3.13.

Effusion GCA also models is an additional aspect of protein function, BP, and proteins



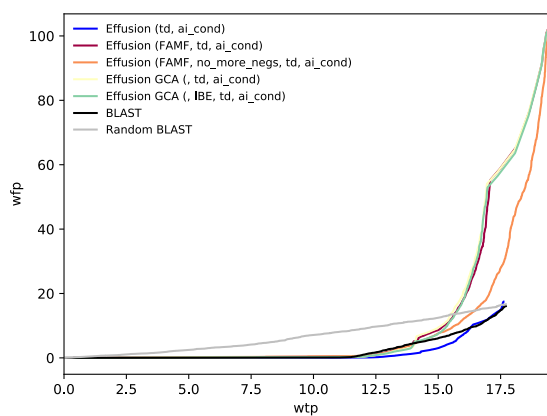
**Figure 3.11:** Protein template for Effusion GCA. Effusion’s model is shown in white. Additional candidate MF terms are shaded blue. New candidate BP terms are shaded yellow.

are often annotated to many different terms within this ontology. The new candidate BP terms are shaded in yellow in Figure 3.11.

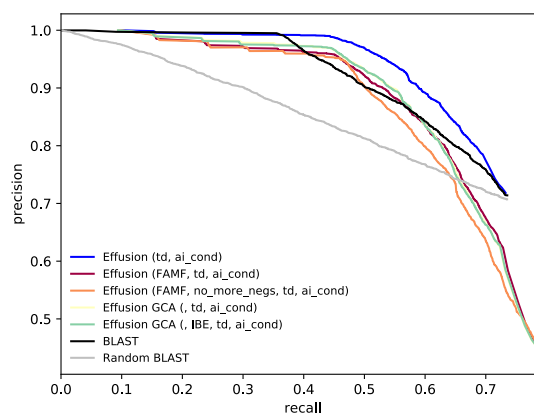
In order to measure the cost of modeling an additional ontology, we implemented variants of Effusion GCA that ignore evidence for BP, while keeping the model (template and network) the same, called the *ignore BP evidence (IBE)* methods. The performance plots are in Figure 3.12 for the top-down model and Figure 3.13 for the bottom-up model.

### Inter-ontological parameters

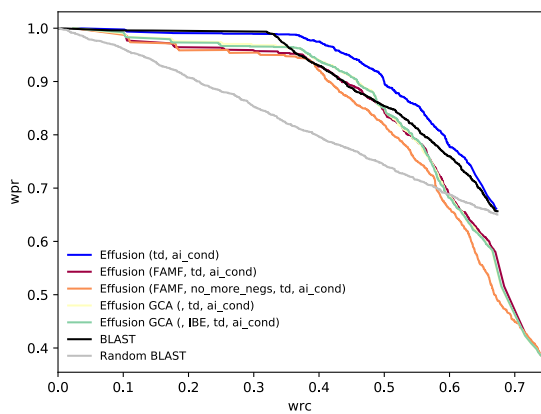
Each candidate catalytic activity term was linked to the candidate metabolic process term to which it shared the highest mutual information, and vice versa. The median mean-correlation-in-template was .236, and a histogram of the mean correlation in template is shown in Figure 3.14. Examples of such links include those in Table 3.5. An example is calculated from the raw contingency tables: whereas  $P(x_{GO:0070566}^i = 1 \mid x_{GO:0016779}^i = 1) = 30.1\%$ , and  $P(x_{GO:0070566}^i = 1 \mid x_{GO:0016779}^i = 1, x_{GO:1901564}^i = 1) = 85.3\%$ . In words, given we know a protein is a nucleotidyltransferase, it is 30.1% likely to be an adenylyltransferase. But if we also know the protein is involved in organonitrogen compound metabolism, then the probability that it is a nucleotidyltransferase increases to 85.3%.



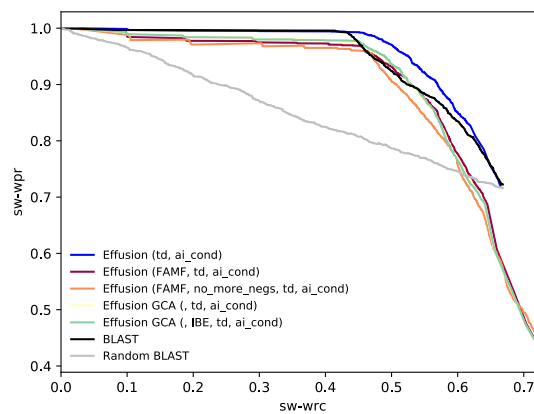
(a) WFP vs. WTP



(b) Precision vs. Recall

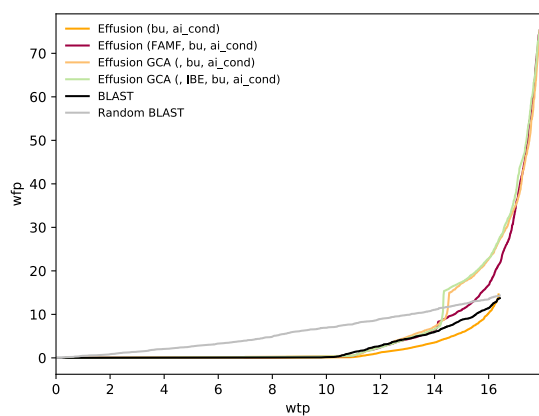


(c) Weighted Precision vs. Weighted Recall

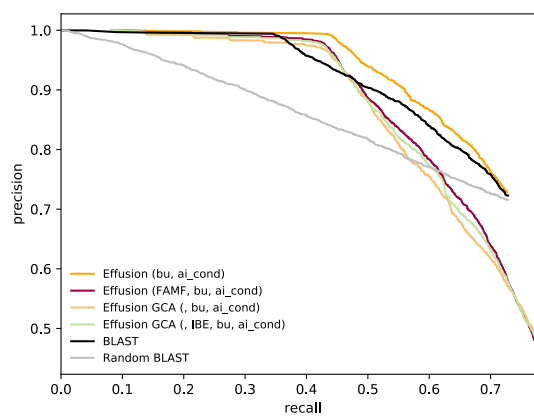


(d) Sample-Weighted Weighted Precision vs. Sample-Weighted Weighted Recall

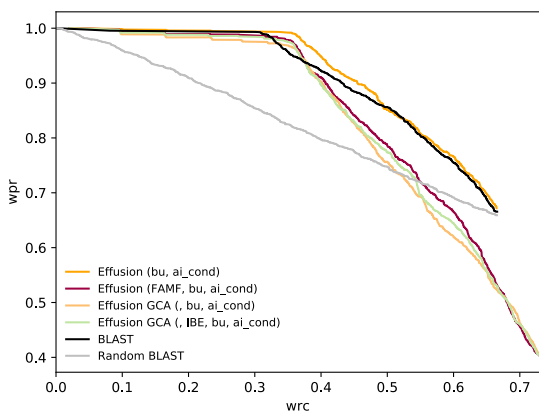
**Figure 3.12:** Performance plots over treated proteins showing the change in performance due to changes in the protein template for the top-down model.



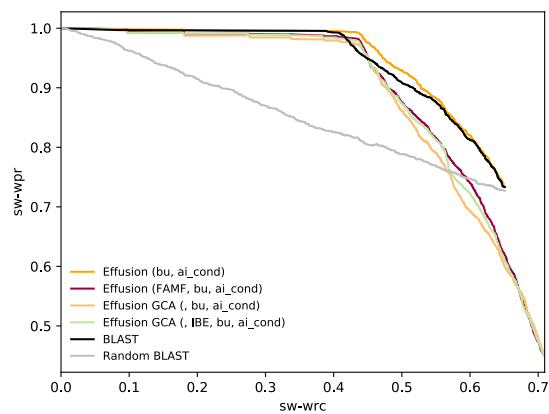
(a) WFP vs. WTP



(b) Precision vs. Recall



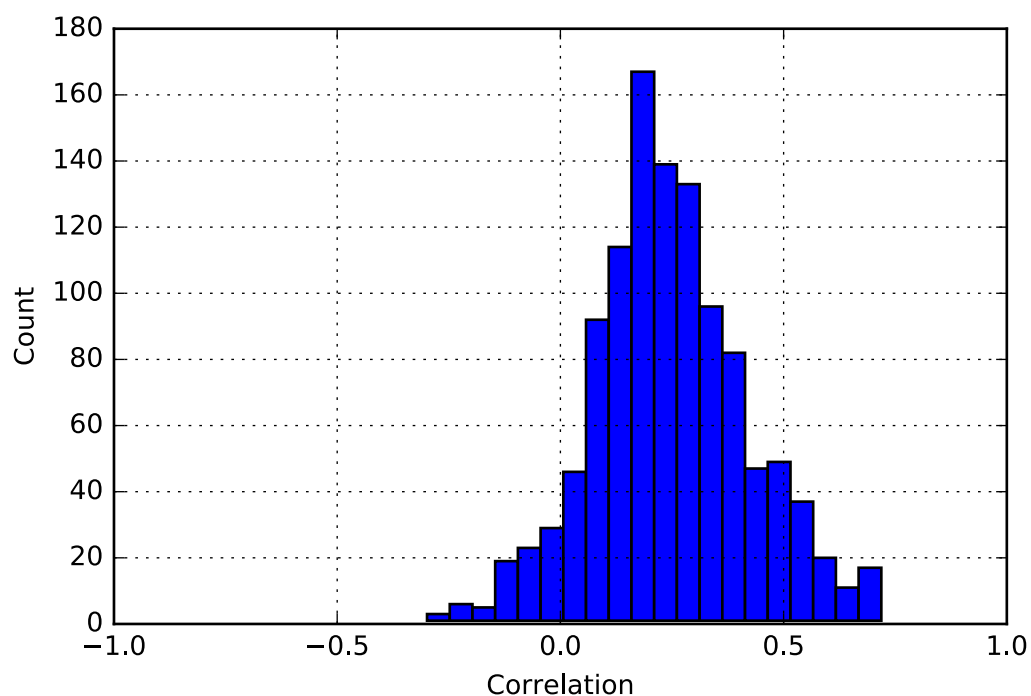
(c) Weighted Precision vs. Weighted Recall



(d) Sample-Weighted Weighted Precision vs. Sample-Weighted Weighted Recall

**Figure 3.13:** Performance plots over treated proteins showing the change in performance due to changes in the protein template for the bottom-up model.





**Figure 3.14:** Histogram of the mean correlation in the protein template

**Table 3.5:** inter-ontological associations modeled for problem instance based on UniProt P78334

catalytic activity	metabolic process	$\rho$
catalytic activity (GO:0003824)	metabolic process (GO:0008152)	0.661003
transferase activity (GO:0016740)	macromolecule modification (GO:0043412)	0.472582
transferase activity, transferring phosphorus-containing groups (GO:0016772)	phosphate-containing compound metabolic process (GO:0006796)	0.618583

catalytic activity	metabolic process	$\rho$
molybdopterin molybdotransferase activity (GO:0061599)	molybdenum incorporation into molybdenum-molybdopterin complex (GO:0018315)	1.000000
nucleotidyltransferase activity (GO:0016779)	heterocycle metabolic process (GO:0046483)	0.630559
adenylyltransferase activity (GO:0070566)	organonitrogen compound metabolic process (GO:1901564)	0.553178
molybdopterin adenylyltransferase activity (GO:0061598)	molybdenum incorporation into molybdenum-molybdopterin complex (GO:0018315)	0.993850

## 3.5 Discussion

### 3.5.1 Comparative analysis

We created and analyzed a method for the prediction of protein function that has the essential components to automate basic analysis of genomic context analysis. Our model is based on proteins that are similar in sequence having similar MFs, proteins that are functionally associated having similar BPs, and a protein's MF being dependent of its BP.

Effusion GCA was able to make predictions for more query proteins than was Mitrofnova2010. Our predictions were also more accurate than Mitrofnova2010.

However, Effusion GCA performed worse than Effusion. Our results suggest that the main reason for the decrease in performance was due to failed inference. Both Effusion GCA

and Effusion were limited to only 40 minutes for inference. However, Effusion GCA's model had many more variables, and could have benefited from having more time. This is especially supported by our implementation of a variant of Effusion GCA that modeled the additional MF terms and BP, but it then ignored the evidence during inference.

There are alternative hypothesis for the difference in performance between Effusion GCA and Effusion. Effusion GCA had more candidate functions to choose from. While we violated an assumption of the top-down model, our data showed that we effectively accounted for this by adding supplementary negative evidence. Furthermore, the bottom-up model did not violate this assumption.

We violate other assumptions of our model, but we are not aware of any violation that would have a major effect on performance. For example, when calculating the parameters, we assume that the multi-label for each protein is complete and also assume that the multi-label represents the protein's function. Rather than formally model the difference between function and annotation and the missing data, we account for missing data by weighting the samples by their information content. Also, for each annotated protein, we only consider the the annotated protein that is most similar in sequence for the BLAST parameters, and the annotated protein that is most functionally associated for the STRING parameters. However, the model then applies these parameters to edges in the reduced network, and these edges are not random samples from the population of edges used to compute the parameters.

### 3.5.2 Creation and use of SSFA networks

We created a method for the fast creation of SSFA networks, which can be visualized with Cytoscape and used for manual network analysis for function annotation. The SSFA networks not only show proteins that are homologous to the query that have function annotations, but

also proteins that are functionally associated to homologs of the query that have function annotations.

### 3.5.3 Functional associations

The parameters we learned from the data indicated that our belief in certain BPs strongly depends on the BPs of functionally associated proteins. To our surprise, we also found that functionally associated proteins also have similar MFs, although we expect this is largely attributable to subontologies of MF for which we are less interested.

Each functional association edge in the reduced network is represented by several edges between corresponding BP terms in the graphical model. As far as we know, this is the first time that the parameters for those functional association edges are GO-term specific.

### 3.5.4 Inter-ontological parameters

Finally, in the protein template, we only added at most one inter-ontological parent for each catalytic term or metabolic term, and for no other GO terms. Our results show that the parameters based on these associations were information rich. While adding any edge should have corresponded with fewer independence assumptions, it is not clear that we did not bias our results in any way. Alternatively, it is not clear that only considering one inter-ontological parent for each catalytic term or metabolic term added enough information to offset the additional candidate GO terms and proteins.

### 3.5.5 Alternative representations

We only modeled functional associations within a single species, and our model realized that functionally associated proteins were more likely to be involved in similar BPs. Alternatively,

we could have represented *contextual similarity* that could be measured across species. This logic would have allowed us to realize that homologous proteins with the similar contexts were more likely to be orthologous and have the same function.

We used BP to model a protein's context, and propagated BP over functional associations. Let us consider an alternative representation for the functional context of a protein: the *Pfams in genome neighborhood (PGN)*. The PGN for a protein is calculated as the set of Pfam domains [71] in the proteins upstream and downstream the given protein. Now, let us compare BP with PGN.

PGN can be calculated precisely for every protein which belongs to an organism with a fully sequenced genome. BP, on the other hand, may be available for every protein, but depends on annotations that are always incomplete.

In both cases, the micro-labels are not mutually exclusive, and the size of the multi-label space is the size of the powerset of the micro-labels. Both BP and PGN use a controlled vocabulary of micro-labels, but they can be combined in ways to represent novel functional contexts. Pfam domains are defined to prevent overlaps, whereas a single protein can be annotated with a large number of BP terms. BP, on the other hand, is structured as a directed acyclic graph (DAG), and the number of valid multi-labels is limited to those that satisfy the true path rule. The hierarchical structure of GO is useful, but relying on it as canon may have disadvantages.

In both BP with PGN, it is possible to calculate the similarity between the functional context of two proteins. Interestingly, this feature would allow us to calculate a hierarchical clustering of PGNs. Also, both BP and PGN can be correlated with other aspects of function.

Ultimately, we chose BP because it is a curated, human readable, extensively used and extensively studied.

### 3.5.6 Other contributions

Although our focus is on predicting MFs, and catalytic activities in particular, research in other labs focus instead on prediction of BP or pathway. Although we did not evaluate it yet, our method simultaneously predicts BP terms.

Additionally, our model can be used for ensuring the quality of function annotations, by flagging proteins with an unlikely combination of annotations.

### 3.5.7 Directions for future research

Our method was motivated by genomic context analysis, but more research is needed to fully automate that technique. First of all, the analysis in Ricardo et al. [74] used knowledge of the likely substrate substructures in a pathway. Therefore, we would need to extend Effusion GCA to model substrate and substrate similarity. Second, as previously mentioned, it would also be useful to compare the context of a protein across organisms. Third, it may be useful to add a factor including the MF terms of functionally associated proteins. The reason for this would be to induce a particular composition of MF terms between cliques of functionally associated proteins. It is, of course, unlikely that all proteins in an operation perform the same function.

# Chapter 4

## Discussion and conclusions

### 4.1 Contributions

This dissertation represents a significant advance in prediction of protein function. It presents two methods, Effusion and Effusion GCA.

Effusion is a framework for prediction of protein function on a sequence similarity network. It is a modern probabilistic approach that can tolerate the sparsity of function annotations. It also takes advantage of the structure of GO. Using a critical method for evaluation, we have shown that while previously published similar works were unable to make predictions that were more accurate than a BLAST-based method, Effusion performs much better than BLAST.

Effusion GCA extended Effusion to add the essential components for automation of basic analysis of genomic context. Although it performed better than the most similar published method, its performance in aggregate did not perform better than Effusion. After detailed analysis of instances where Effusion GCA performed particularly poorly, we suspect that this degradation in performance was largely due to limited time given to a more complex model. Whatever the reason, there are many avenues for enhancing the performance of the

method.

Each component of our model was interesting. For example, our parameters can tell us how likely a methylase is in a nitrogen compound metabolic process, or how likely that two proteins that are similar in sequence have the same catalytic activity. These components can be used in ways besides prediction of protein function. For example, the protein template can flag a protein as having an unlikely combination of protein functions.

We were rigorous in our evaluation and analysis. We withheld all annotations from a point in time onward for testing. The methods that were compared used the same training and test data. We evaluated variants of our method to see the value or cost associated with different components of our model.

As a probabilistic model, our method can do more than compute marginal probabilities. It can also be used for computing the variance, or uncertainty, of the model given the evidence. This would make Effusion useful for target selection, i.e., determining which proteins to subject to further study.

Effusion and Effusion GCA provide networks for manual analysis of protein networks. These networks can be visualized with Cytoscape to see protein similarity, functional associations, functional annotation data, and predictions.

## 4.2 Failed attempts

We experimented with variations on network building, probabilistic model semantics, priors, and inference algorithms.

Conventionally, a sequence similarity network, with no natural directionality and tight cliques, would be an undirected PGM, known as a Markov random field (MRF). Indeed, this is the representation used by previously published methods. However, as we can tell from the



literature and from our own implementation, that representation required concessions that precluded practical prediction of protein function. Specifically, these methods used shared parameters among all GO terms, prediction on very few GO terms, excessive, unfounded independence assumptions, ad-hoc inference algorithms and post-processing steps, and custom evaluation methods.

Due to transitivity of sequence similarity, sequence similarity networks have tight loops. Particularly in MRFs with this topology, belief propagation may fail to converge, or converge to the wrong answer. While sampling-based algorithms like Gibbs sampling are guaranteed to converge to the right answer, we observed that the tight cliques in the network prevented convergence.

We also experimented with inference using GraphLab for distributed inference, Alchemy for lifted inference, OpenGM for fast inference with C++ templates, and the various algorithms implemented in libDAI.

One approach we tried was to use a Metropolis-Hastings algorithm that clustered the network and used a proposal that jumped over unlikely states by flipping all of the values of all random variables in a cluster. While we found that this was effective with a network of proteins using a single GO term, it was not clear how we would flip a protein's value when that value had the structure of our protein template.

We then tried to get rid of the loops in our protein network via a MST. That approach remained in our final incarnation of Effusion. However, we could not apply a MST to remove loops within each instance of the protein template, since they were necessary for our hypothesis, which required that we model the relationship between MF and BP.

Therefore, based on our observation that only a small subset of possible assignments to GO are valid, we tried to consider each protein as a single variable with a scope equal to the size of the number of functions that are valid with respect to GO. However, we soon

found that we could not model enough candidate GO terms for high performance. A high performing method for protein function prediction must not limit inclusion of experimental or manually curated function annotation data. However, there have been recent developments in the combinatorics and dimensionality reduction of the function space defined by GO.

Significantly, we could not avoid learning the parameters on a per network basis. Although we came up with a formulation of our potentials with very few parameters, this model did not benefit from learning the parameters from all data in GOA, and it suffered in the common case of extremely sparsely annotated networks.

### 4.3 Limitations

Unfortunately, our model adds new problems that do not affect other methods.

In our evaluation, we excluded IEA annotations. This is the standard in the field, because of the leakage that results from including IEA annotations when doing standard cross validation, and because the inclusion of IEA evidence obfuscates the contributions of a method. In addition, IEA annotations older than one year are deleted, so they would only appear in the evaluation set. However, there are methods that would contribute information that would be complementary to Effusion. For example, one method may generate electronic annotations that are produced through docking experiments. Unfortunately, these electronic annotations are so numerous that their inclusion would result in explosive growth in the number of candidate GO terms, rendering our method infeasible.

For computational reasons, our model reduced the protein network into a MST. We observed that the edges of the MST typically connected nodes within the same cluster of the reduced network. However, a MST cannot perfectly capture network boundaries. We are very interested in including more edges in our model, as long as it does not compromise

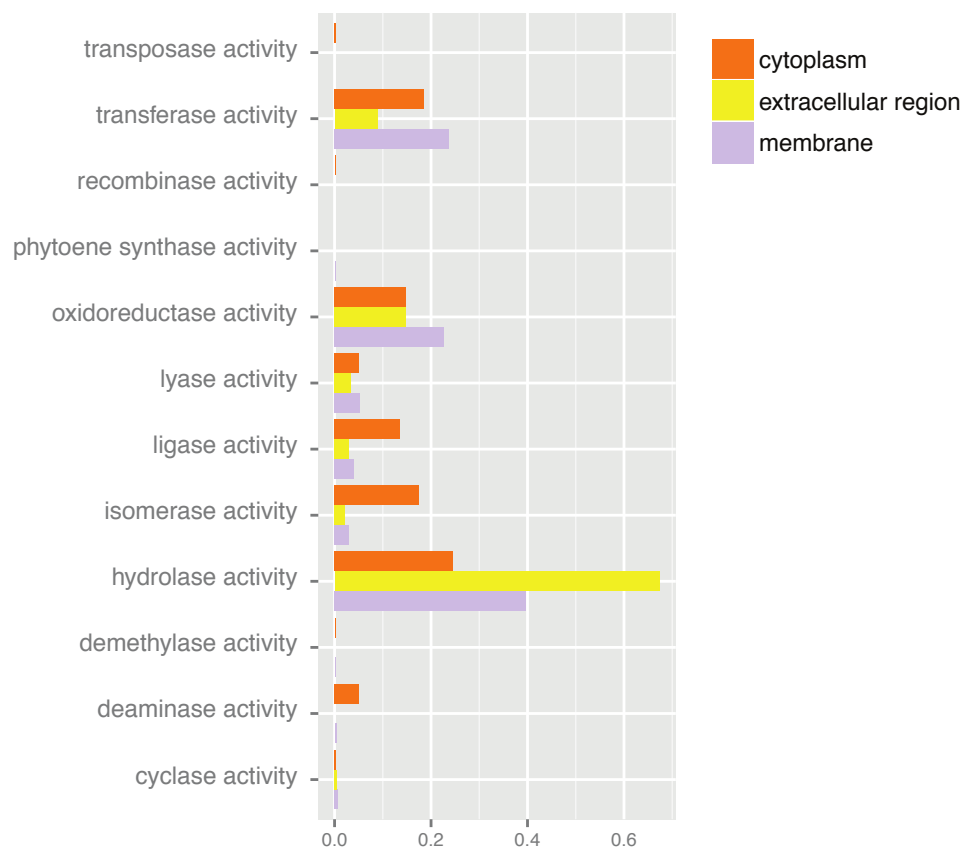
the integrity and performance of our model.

Effusion blurs the lines between annotation and function. During training, labels represent annotations. Any protein which missing evidence for a GO term is considered to be negatively annotated to that GO term. However, during inference, our labels represent functions, and any protein which does not have evidence for a GO term is considered to be unknown. Using parameters learned for annotations may not be the maximum likelihood parameters for functions. We mitigate this by weighting the samples we use for calculating the parameters by the information content of the samples. That way, an idealized protein with a complete annotation would carry the most weight for calculating parameters, but proteins with incomplete annotations are still useful for computing parameters of rare terms. This could be addressed with more rigor by using a model during learning that represent functions as latent variables, if such a model were tractable.

Our model utilizes the dependencies between various aspects of protein function. However, as more aspects are modeled, this may become very complicated. The modeling semantics of conditional random fields suggest one avenue for addressing this problem.

## 4.4 Directions for future research

We started with modeling molecular function and sequence similarity, and demonstrated how to add biological process and functional associations. However, our framework is designed to extend to additional aspects of protein function and protein similarity. One aspect we are interested in modeling is the cellular component of each protein. Exploratory analysis indicates that the probability of certain classes of enzymes depend on the cellular location of a protein Figure 4.1. Also, unlike MF and BP, there is high throughput, relatively unbiased experimental data indicating the cellular component for proteins for some organisms. Finally,

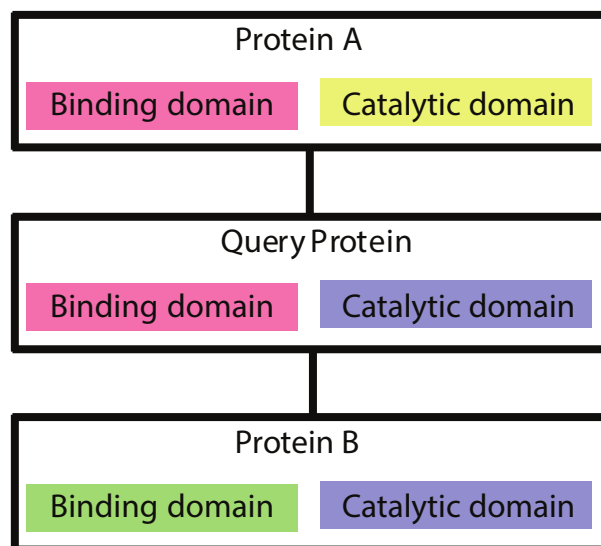


**Figure 4.1:** Probability of certain enzyme classes given cellular location

it would be straightforward to add to our framework, since, like MF and BP, it is another aspect for which GO provides an ontology.

As a proof of concept, Effusion only collected homologous proteins that were aligned over 90% with the query protein. While this avoided predicting functions that were due to a domain not shared by the query, it could result in a significant decrease in recall. However, we could also include proteins that only align partially, and use a resource like dcGO to only propagate the GO terms relevant to that domain, as in Figure 4.2.

There could be several benefits to modeling substrate and propagating it via substrate binding site similarity. The results of docking experiments could provide evidence where it



**Figure 4.2:** A finer grained representation of sequence similarity to propagate function more carefully. Only functions corresponding to shared domains should be propagated. Consider two highly similar proteins with domain architecture A-B-C and A-B-Z. We should not transfer functional information caused by domain C from the former protein to the latter protein.

is available. Homologous proteins may be more likely to act on the same substrate if they have similar substrate binding sites. Similarly, functionally associated enzymes may be more likely to act on substrates of with similar substructures.

As we discussed in Directions for future research, factors that include GO terms from at most two proteins may be insufficient for representing all of our knowledge regarding analysis of genomic context. A factor including the entire set of functionally associated proteins would be necessary to enforce a particular distribution of functions among the proteins. This may be feasible, since operons generally only contain a few proteins, but it would still be a computational burden.

# Bibliography

- [1] UA Acar, AT Ihler, and R Mettu. “Adaptive inference on general graphical models”. In: *arXiv preprint arXiv:1206.3234* (2012).
- [2] Daniel E Almonacid and Patricia C Babbitt. “Toward mechanistic classification of enzyme functions”. In: *Curr Opin Chem Biol* 15.3 (2011), pp. 435–442. ISSN: 1367-5931. DOI: 10.1016/j.cbpa.2011.03.008.
- [3] SF Altschul et al. “Gapped BLAST and PSI-BLAST: a new generation of protein database search programs.” In: *Nucleic Acids Res.* 25.17 (1997), pp. 3389–402. ISSN: 0305-1048.
- [4] B Andres and T and Beier. “OpenGM: A C++ library for discrete graphical models”. In: *arXiv preprint arXiv:1206.0111* (2012).
- [5] Michael Ashburner et al. “Gene Ontology: tool for the unification of biology”. In: *Nat Genet* 25.1 (2000), pp. 25–29. ISSN: 1061-4036. DOI: 10.1038/75556.
- [6] Holly J Atkinson et al. “Using sequence similarity networks for visualization of relationships across diverse protein superfamilies.” In: *PLoS ONE* 4.2 (2009), e4345. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0004345.
- [7] M Bada et al. “Using reasoning to guide annotation with gene ontology terms in GOAT”. In: *ACM SIGMOD Record* (2004).

- [8] Alan E Barber and Patricia C Babbitt. “Pythoscape: a framework for generation of large protein similarity networks.” In: *Bioinformatics* 28.21 (2012), pp. 2845–6. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bts532.
- [9] Zafer Barutcuoglu, Robert E Schapire, and Olga G Troyanskaya. “Hierarchical multi-label prediction of gene function”. In: *Bioinformatics* 22.7 (2006), pp. 830–836. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btk048.
- [10] F Benites, S Simon, and Sapozhnikova - E one. “Mining rare associations between biological ontologies”. In: *PloS one* (2014). DOI: 10.1371/journal.pone.0084475.
- [11] Fernando Benites and Elena P Sapozhnikova. “Generalized Association Rules for Connecting Biological Ontologies.” In: (2013), pp. 229–236.
- [12] Olivier Bodenreider, Marc Aubry, and Anita Burgun. “Non-lexical approaches to identifying associative relations in the gene ontology.” In: *Pac Symp Biocomput* (2005), pp. 91–102. ISSN: 2335-6936.
- [13] Shoshana D Brown and Patricia C Babbitt. “Inference of functional properties from large-scale analysis of enzyme superfamilies.” In: *J. Biol. Chem.* 287.1 (2012), pp. 35–42. ISSN: 0021-9258. DOI: 10.1074/jbc.R111.283408.
- [14] Shoshana D Brown and Patricia C Babbitt. “New insights about enzyme evolution from large scale studies of sequence and structure relationships.” In: *J. Biol. Chem.* 289.44 (2014), pp. 30221–8. ISSN: 0021-9258. DOI: 10.1074/jbc.R114.569350.
- [15] Benjamin Buchfink, Chao Xie, and Daniel H Huson. “Fast and sensitive protein alignment using DIAMOND.” In: *Nat. Methods* 12.1 (2015), pp. 59–60. ISSN: 1548-7091. DOI: 10.1038/nmeth.3176.

- [16] Evelyn Camon et al. “The Gene Ontology Annotation (GOA) Database: sharing knowledge in Uniprot with Gene Ontology”. In: *Nucleic Acids Res* 32.suppl\_1 (2004), pp. D262–D266. ISSN: 0305-1048. DOI: 10.1093/nar/gkh021.
- [17] Steven Carroll and Vladimir Pavlovic. “Protein classification using probabilistic chain graphs and the Gene Ontology structure”. In: *Bioinformatics* 22.15 (2006), pp. 1871–1878. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btl187.
- [18] Ron Caspi et al. “The MetaCyc database of metabolic pathways and enzymes.” In: (2017), gkx935–. ISSN: 1362-4962. DOI: 10.1093/nar/gkx935.
- [19] Andrew Chatr-aryamontri et al. “The BioGRID interaction database: 2017 update”. In: (2017), pp. D369–D379. ISSN: 0305-1048. DOI: 10.1093/nar/gkw1102.
- [20] Wyatt T Clark and Predrag Radivojac. “Information-theoretic evaluation of predicted ontological annotations”. In: *Bioinformatics* 29.13 (2013), pp. i53–i61. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btt228.
- [21] Ana Conesa et al. “Blast2GO: a universal tool for annotation, visualization and analysis in functional genomics research”. In: *Bioinformatics* 21.18 (2005), pp. 3674–3676. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bti610.
- [22] Domenico Cozzetto et al. “FFPred 3: feature-based function prediction for all Gene Ontology domains”. In: *Sci Reports* 6.1 (2016), p. 31865. ISSN: 2045-2322. DOI: 10.1038/srep31865.
- [23] Domenico Cozzetto et al. “Protein function prediction by massive integration of evolutionary analyses and multiple data sources.” In: *BMC Bioinformatics* 14 Suppl 3.S3 (2013), S1. ISSN: 1471-2105. DOI: 10.1186/1471-2105-14-S3-S1.



- [24] Rebecca Davidson et al. “A global view of structure-function relationships in the tautomerase superfamily.” In: *J. Biol. Chem.* (2017), jbc.M117.815340. ISSN: 0021-9258. DOI: 10.1074/jbc.M117.815340.
- [25] Minghua Deng, Ting Chen, and Fengzhu Sun. “An integrated probabilistic model for functional prediction of proteins.” In: *J. Comput. Biol.* 11.2-3 (2004), pp. 463–75. ISSN: 1066-5277. DOI: 10.1089/1066527041410346.
- [26] Minghua Deng et al. “Prediction of protein function using protein-protein interaction data.” In: *Proc IEEE Comput Soc Bioinform Conf 1* (2002), pp. 197–206. ISSN: 1555-3930.
- [27] Jonathan A Eisen. “Phylogenomics: Improving Functional Predictions for Uncharacterized Genes by Evolutionary Analysis”. In: *Biotechfor* 8.3 (1998), pp. 163–167. ISSN: 1088-9051. DOI: 10.1101/gr.8.3.163.
- [28] Roman Eisner et al. “Improving Protein Function Prediction using the Hierarchical Structure of the Gene Ontology”. In: (2005), pp. 1–10. DOI: 10.1109/CIBCB.2005.1594940.
- [29] Gal Elidan and Amir Globerson. *Summary of the 2010 UAI approximate inference challenge*. 0.
- [30] Barbara E Engelhardt et al. “Phylogenetic molecular function annotation”. In: *J Phys Conf Ser* 180.1 (2009), p. 012024. ISSN: 1742-6596. DOI: 10.1088/1742-6596/180/1/012024.
- [31] Barbara E Engelhardt et al. “Protein Molecular Function Prediction by Bayesian Phylogenomics”. In: *PLOS Comput. Biol.* 1.5 (2005), e45. ISSN: 1553-734X. DOI: 10.1371/journal.pcbi.0010045.

- [32] BE Engelhardt, MI Jordan, and Srouji - JR Genome “Genome-scale phylogenetic function annotation of large and diverse protein families”. In: *Genome* (2011). DOI: 10.1101/gr.104687.109.
- [33] Hai Fang and Julian Gough. “DcGO: database of domain-centric ontologies on functions, phenotypes, diseases and more.” In: *Nucleic Acids Res.* 41.Database issue (2013), pp. D536–44. ISSN: 0305-1048. DOI: 10.1093/nar/gks1080.
- [34] JA Gerlt and PC Babbitt. “Divergent evolution of enzymatic function: mechanistically diverse superfamilies and functionally distinct suprafamilies.” In: *Annu. Rev. Biochem.* 70.1 (2001), pp. 209–46. ISSN: 0066-4154. DOI: 10.1146/annurev.biochem.70.1.209.
- [35] John A Gerlt et al. “Enzyme Function Initiative-Enzyme Similarity Tool (EFI-EST): A web tool for generating protein sequence similarity networks.” In: *Biochim. Biophys. Acta* 1854.8 (2015), pp. 1019–37. ISSN: 0006-3002. DOI: 10.1016/j.bbapap.2015.04.015.
- [36] Gogate and Rina Dechter. “Samplesearch: A scheme that searches for consistent samples”. In: 2007, pp. 147–154.
- [37] Casey S Greene and Olga G Troyanskaya. “Accurate evaluation and analysis of functional genomics data and methods”. In: 1260.1 (2012), pp. 95–100. ISSN: 1749-6632. DOI: 10.1111/j.1749-6632.2011.06383.x.
- [38] Xiaoyu Jiang et al. “Integration of relational and hierarchical network information for protein function prediction”. In: *BMC Bioinform.* 9.1 (2008), pp. 1–15. ISSN: 1471-2105. DOI: 10.1186/1471-2105-9-350.
- [39] Yuxiang Jiang et al. “An expanded evaluation of protein function prediction methods shows an improvement in accuracy.” In: *Genome Biol.* 17.1 (2016), p. 184. ISSN: 1474-7596. DOI: 10.1186/s13059-016-1037-6.

- [40] HL Johnson and Cohen - KB on “Evaluation of lexical methods for detecting relationships between concepts from multiple ontologies”. In: *Pacific Symposium on* (2006).
- [41] Ulas Karaoz et al. “Whole-genome annotation by using evidence integration in functional-linkage networks”. In: *P Natl Acad Sci Usa* 101.9 (2004), pp. 2888–2893. ISSN: 0027-8424. DOI: 10.1073/pnas.0307326101.
- [42] Yiannis AI Kourmpetis et al. “Bayesian Markov Random Field Analysis for Protein Function Prediction Based on Network Data”. In: *Plos One* 5.2 (2010), e9293. DOI: 10.1371/journal.pone.0009293.
- [43] GR Lanckriet et al. “Kernel-based data fusion and its application to protein function prediction in yeast.” In: *Pac Symp Biocomput* (2004), pp. 300–11.
- [44] H Lee et al. “Diffusion kernel-based logistic regression models for protein function prediction”. In: *OmicS: a journal of* (2006). DOI: 10.1089/omi.2006.10.40.
- [45] S Lehtinen et al. “Gene function prediction from functional association networks using kernel partial least squares regression”. In: *PloS one* (2015). DOI: 10.1371/journal.pone.0134668.
- [46] Stanley Letovsky and Simon Kasif. “Predicting protein function from protein/protein interaction data: a probabilistic approach”. In: *Bioinformatics* 19.suppl\_1 (2003), pp. i197–i204. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btg1026.
- [47] Wenlin Li, Lisa N Kinch, and Nick V Grishin. “Pclust: protein network visualization highlighting experimental data”. In: *Bioinformatics* 29.20 (2013), pp. 2647–2648. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btt451.

- [48] Yongjin Li and Jagdish C Patra. “Genome-wide inferring gene-phenotype relationship by walking on the heterogeneous network.” In: *Bioinformatics* 26.9 (2010), pp. 1219–24. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btq108.
- [49] Anna Lobley et al. “Inferring Function Using Patterns of Native Disorder in Proteins”. In: *Plos Comput Biol* 3.8 (2007), e162. ISSN: 1553-734X. DOI: 10.1371/journal.pcbi.0030162.
- [50] Javier F Lopez et al. “Extracting biological knowledge by fuzzy association rule mining”. In: (2007), pp. 1–6.
- [51] Peña-Castillo Lourdes et al. “A critical assessment of *Mus musculus* gene function prediction using integrated genomic evidence.” In: *Genome Biol.* 9 Suppl 1 (2008), S2. ISSN: 1474-7596. DOI: 10.1186/gb-2008-9-s1-s2.
- [52] P Manda, McCarthy F, and Bridges - SM of biomedical informatics. “Interestingness measures and strategies for mining multi-ontology multi-level association rules from gene ontology annotations for the discovery of new GO ” In: *Journal of biomedical informatics* (2013).
- [53] P Manda et al. “Information-theoretic Interestingness Measures for Cross-Ontology Data Mining”. In: *arXiv preprint arXiv* (2015).
- [54] Edward M Marcotte et al. “A combined algorithm for genome-wide prediction of protein function”. In: *Nature* 402.6757 (1999), pp. 83–86. ISSN: 0028-0836. DOI: 10.1038/47048.
- [55] Edward Marcotte et al. “Detecting Protein Function and Protein-Protein Interactions from Genome Sequences”. In: (1999). ISSN: 0036-8075. DOI: 10.1126/science.285.5428.751.

- [56] Alberto JM Martin et al. “PANADA: Protein Association Network Annotation, Determination and Analysis”. In: *Plos One* 8.11 (2013), e78383. DOI: 10.1371/journal.pone.0078383.
- [57] David MA Martin, Matthew Berriman, and Geoffrey J Barton. “GOTcha: a new method for prediction of protein function assessed by the annotation of seven genomes”. In: *Bmc Bioinformatics* 5.1 (2004), pp. 1–17. ISSN: 1471-2105. DOI: 10.1186/1471-2105-5-178.
- [58] Christian von Mering et al. “STRING: a database of predicted functional associations between proteins”. In: *Nucleic Acids Res* 31.1 (2003), pp. 258–261. ISSN: 0305-1048. DOI: 10.1093/nar/gkg034.
- [59] Huaiyu Mi, Anushya Muruganujan, and Paul D Thomas. “PANTHER in 2013: modeling the evolution of gene function, and other gene attributes, in the context of phylogenetic trees”. In: *Nucleic Acids Res* 41.D1 (2013), pp. D377–D386. ISSN: 0305-1048. DOI: 10.1093/nar/gks1118.
- [60] Antonina Mitrofanova, Vladimir Pavlovic, and Bud Mishra. “Integrative Protein Function Transfer using Factor Graphs and Heterogeneous Data Sources”. In: *2008 Ieee Int Conf Bioinform Biomed* (2008), pp. 314–318. DOI: 10.1109/bibm.2008.65.
- [61] Antonina Mitrofanova, Vladimir Pavlovic, and Bud Mishra. “Prediction of Protein Functions with Gene Ontology and Interspecies Protein Homology Data”. In: *Ieee Acm Transactions Comput Biology Bioinform* 8.3 (2010), pp. 775–784. ISSN: 1545-5963. DOI: 10.1109/TCBB.2010.15.
- [62] Joris Mooij. “libDAI: A free and open source C++ library for discrete approximate inference in graphical models”. In: *J. Mach. Learn. Res.* 11.Aug (2010), pp. 2169–2173.

- [63] Sara Mostafavi and Quaid Morris. “Fast integration of heterogeneous data sources for predicting gene function with limited annotation.” In: *Bioinformatics* 26.14 (2010), pp. 1759–65. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btq262.
- [64] Guillaume Obozinski et al. “Consistent probabilistic outputs for protein function prediction.” In: *Genome Biol.* 9 Suppl 1.S1 (2008), S6. ISSN: 1474-7596. DOI: 10.1186/gb-2008-9-s1-s6.
- [65] R Overbeek, M Fonstein, and D’souza - M of the “The use of gene clusters to infer functional coupling”. In: *Proceedings of the* (1999).
- [66] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier, 2014.
- [67] Matteo Pellegrini et al. “Assigning protein functions by comparative genome analysis: Protein phylogenetic profiles”. In: *Proc National Acad Sci* 96.8 (1999), pp. 4285–4288. ISSN: 0027-8424. DOI: 10.1073/pnas.96.8.4285.
- [68] J Peng, J Chen, and Wang - Y bioinformatics. “Identifying cross-category relations in gene ontology and constructing genome-specific term association networks”. In: *BMC bioinformatics* (2013).
- [69] Jiajie Peng et al. “Identifying term relations cross different gene ontology categories”. In: *Bmc Bioinformatics* 18.Suppl 16 (2017), p. 573. DOI: 10.1186/s12859-017-1959-3.
- [70] Christian Posse et al. “Cross-Ontological Analytics: Combining associative and hierarchical relations in the Gene Ontologies to assess gene product similarity”. In: (2006), pp. 871–878.
- [71] Marco Punta et al. “The Pfam protein families database”. In: *Nucleic Acids Res* 40.D1 (2012), pp. D290–D301. ISSN: 0305-1048. DOI: 10.1093/nar/gkr1065.

- [72] Predrag Radivojac et al. “A large-scale evaluation of computational protein function prediction.” In: *Nat. Methods* 10.3 (2013), pp. 221–7. ISSN: 1548-7091. DOI: 10.1038/nmeth.2340.
- [73] Robert Rentzsch and Christine A Orengo. “Protein function prediction the power of multiplicity”. In: *Trends Biotechnol* 27.4 (2009), pp. 210–219. ISSN: 0167-7799. DOI: 10.1016/j.tibtech.2009.01.002.
- [74] Martí-Arbona Ricardo et al. “Annotating Enzymes of Unknown Function: N-Formimino-l-glutamate Deiminase Is a Member of the Amidohydrolase Superfamily”. In: *Biochemistry-us* 45.7 (2006), pp. 1997–2005. ISSN: 0006-2960. DOI: 10.1021/bi0525425.
- [75] Burkhard Rost. “Enzyme function less conserved than anticipated”. In: *J Mol Biol* 318.2 (2002), pp. 595–608.
- [76] Sayed M Sahraeian, Kevin R Luo, and Steven E Brenner. “SIFTER search: a web server for accurate phylogeny-based protein function prediction.” In: *Nucleic Acids Res.* 43.W1 (2015), W141–7. ISSN: 0305-1048. DOI: 10.1093/nar/gkv461.
- [77] Alexandra M Schnoes et al. “Annotation Error in Public Databases: Misannotation of Molecular Function in Enzyme Superfamilies”. In: *Plos Comput Biol* 5.12 (2009), e1000605. ISSN: 1553-734X. DOI: 10.1371/journal.pcbi.1000605.
- [78] Paul Shannon et al. “Cytoscape: a software environment for integrated models of biomolecular interaction networks.” In: *Genome Res.* 13.11 (2003), pp. 2498–504. ISSN: 1088-9051. DOI: 10.1101/gr.1239303.

- [79] Roded Sharan, Igor Ulitsky, and Ron Shamir. “Networkbased prediction of protein function”. In: *Mol Syst Biol* 3.1 (2007), p. 88. ISSN: 1744-4292. DOI: 10.1038/msb4100129.
- [80] A Sokolov and Ben-Hur A. “GOstruct: PREDICTION OF GENE ONTOLOGY TERMS USING METHODS FOR STRUCTURED OUTPUT SPACES”. In: (2009).
- [81] Artem Sokolov et al. “Combining heterogeneous data sources for accurate functional annotation of proteins”. In: *Bmc Bioinformatics* 14.S3 (2013), pp. 1–13. ISSN: 1471-2105. DOI: 10.1186/1471-2105-14-s3-s10.
- [82] B Szalkai and preprint arXiv:1703.10663, Grolmusz - V. “Near Perfect Protein Multi-Label Classification with Deep Neural Networks”. In: *arXiv preprint arXiv:1703.10663* (2017).
- [83] Damian Szklarczyk et al. “STRING v10: proteinprotein interaction networks, integrated over the tree of life”. In: *Nucleic Acids Res* 43.D1 (2015), pp. D447–D452. ISSN: 0305-1048. DOI: 10.1093/nar/gku1003.
- [84] Ying Tao et al. “Information theory applied to the sparse gene ontology annotation network to predict novel gene function”. In: *Bioinformatics* 23.13 (2007), pp. i529–i538. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btm195.
- [85] Weidong Tian and Jeffrey Skolnick. “How well is enzyme function conserved as a function of pairwise sequence identity?” In: *J. Mol. Biol.* 333.4 (2003), pp. 863–82. ISSN: 0022-2836.
- [86] AE Todd, CA Orengo, and JM Thornton. “Evolution of function in protein superfamilies, from a structural perspective.” In: *J. Mol. Biol.* 307.4 (2001), pp. 1113–43. ISSN: 0022-2836. DOI: 10.1006/jmbi.2001.4513.



- [87] Olga G Troyanskaya et al. “A Bayesian framework for combining heterogeneous data sources for gene function prediction (in *Saccharomyces cerevisiae*)”. In: 100.14 (2003), pp. 8348–8353. ISSN: 0027-8424. DOI: 10.1073/pnas.0832373100.
- [88] The UniProt Consortium. “UniProt: the universal protein knowledgebase”. In: *Nucleic Acids Res* 45.D1 (2017), pp. D158–D169. ISSN: 0305-1048. DOI: 10.1093/nar/gkw1099.
- [89] Alexei Vazquez et al. “Global protein function prediction from protein-protein interaction networks”. In: *Nat Biotechnol* 21.6 (2003), nbt825. ISSN: 1546-1696. DOI: 10.1038/nbt825.
- [90] Suwen Zhao et al. “Discovery of new enzymes and metabolic pathways by using structure and genome context.” In: *Nature* 502.7473 (2013), pp. 698–702. ISSN: 0028-0836. DOI: 10.1038/nature12576.
- [91] Suwen Zhao et al. “Prediction and characterization of enzymatic activities guided by sequence similarity and genome neighborhood networks.” In: *Elife* 3.0 (2014), e03275. ISSN: 2050-084X. DOI: 10.7554/eLife.03275.

# Appendix A

## Supplementary material for Chapter 2

**Table A.1:** Predictions for Q4ZIL6 made by Effusion. For comparison, predictions made by BLAST are in Table A.2. The GO terms are filtered to those that are predicted by both methods.

GO Term	Posterior		True Positives	False Positives
	Probability	In GOA	(Bits)	(Bits)
GO:0003674	1.000000	✓	0.414455	0.000000
GO:0003824	0.998374	✓	1.645753	0.000000
GO:0016491	0.985996	✓	4.580627	0.000000
GO:0004497	0.917051	✓	6.774468	0.000000
<b>GO:0008395</b>	<b>0.363443</b>	✓	<b>11.116429</b>	<b>0.000000</b>
GO:0005488	0.235120	✗	11.116429	1.475205
GO:0016705	0.211351	✓	13.463748	1.475205
GO:0016709	0.166540	✗	13.463748	1.940758
GO:0005515	0.154005	✗	13.463748	2.957393
GO:0008391	0.118320	✗	13.463748	6.180406
GO:0016712	0.105213	✗	13.463748	12.223869
GO:0071614	0.102991	✗	13.463748	21.014567
GO:0008389	0.101734	✓	21.739872	21.014567
GO:0097159	0.091303	✗	21.739872	22.277779
GO:1901363	0.088618	✗	21.739872	23.563451
GO:0004508	0.062760	✗	21.739872	26.181203
GO:0004509	0.059321	✗	21.739872	31.335009
GO:0019899	0.043275	✗	21.739872	33.466246
GO:0035302	0.024479	✗	21.739872	41.348889

GO Term	Posterior		True Positives	False Positives
	Probability	In GOA	(Bits)	(Bits)
GO:0008392	0.023916	<b>X</b>	21.739872	41.375201
GO:0008144	0.022433	<b>X</b>	21.739872	48.423917
GO:0019825	0.020329	<b>X</b>	21.739872	58.033484
GO:0046996	0.016909	<b>X</b>	21.739872	65.150414
GO:0018631	0.015712	<b>X</b>	21.739872	76.282914
GO:0072532	0.015700	<b>X</b>	21.739872	87.415414
GO:0097007	0.015683	<b>X</b>	21.739872	98.547914
GO:0097008	0.015683	<b>X</b>	21.739872	109.680413
GO:0047084	0.015680	<b>X</b>	21.739872	120.812913
GO:0072533	0.015641	<b>X</b>	21.739872	130.945413
GO:0052722	0.010906	<b>X</b>	21.739872	139.263127
GO:0046906	0.009599	<b>X</b>	21.739872	144.652214
GO:0016829	0.008688	<b>X</b>	21.739872	149.390134
GO:0016725	0.006943	<b>X</b>	21.739872	155.292879
GO:0032451	0.006759	<b>X</b>	21.739872	163.751381
GO:0008405	0.002005	<b>X</b>	21.739872	170.515146
<b>GO:0008404</b>	<b>0.001913</b>	<b>X</b>	<b>21.739872</b>	177.863874
GO:0047055	0.001685	<b>X</b>	21.739872	182.818071
GO:0072547	0.001561	<b>X</b>	21.739872	182.818071
GO:0072549	0.001558	<b>X</b>	21.739872	182.818071
GO:0072548	0.001553	<b>X</b>	21.739872	182.818071
GO:0072552	0.001526	<b>X</b>	21.739872	182.818071

GO Term	Posterior		True Positives (Bits)	False Positives (Bits)
	Probability	In GOA		
GO:0072551	0.001525	<b>X</b>	21.739872	182.818071
GO:0072550	0.001524	<b>X</b>	21.739872	182.818071
GO:0009055	0.001438	<b>X</b>	21.739872	191.343344
GO:0020037	0.001426	<b>X</b>	21.739872	191.763952
GO:0016830	0.001401	<b>X</b>	21.739872	193.490608
GO:0033695	0.000685	<b>X</b>	21.739872	198.198953
GO:0016832	0.000304	<b>X</b>	21.739872	200.394209
GO:0034875	0.000108	<b>X</b>	21.739872	200.394209
GO:0047442	0.000045	<b>X</b>	21.739872	204.564134

**Table A.2:** Predictions for Q4ZIL6 made by BLAST. The GO terms are filtered to those that are predicted by both methods.

GO Term	Posterior		True Positives	False Positives
	Probability	In GOA	(Bits)	(Bits)
GO:0003674	0.315330	✓	0.414455	0.000000
GO:0005488	0.315299	✗	0.414455	1.475205
GO:1901363	0.315267	✗	0.414455	2.760877
GO:0097159	0.315267	✗	0.414455	4.024089
GO:0046906	0.315236	✗	0.414455	9.413176
GO:0020037	0.315204	✗	0.414455	9.833784
GO:0009055	0.305691	✗	0.414455	18.359057
GO:0003824	0.305691	✓	1.645753	18.359057
GO:0032451	0.305661	✗	1.645753	26.817559
GO:0019825	0.305661	✗	1.645753	36.427125
GO:0016491	0.305661	✓	4.580627	36.427125
GO:0005515	0.305661	✗	4.580627	37.443760
GO:0019899	0.305630	✗	4.580627	39.574997
GO:0016725	0.305630	✗	4.580627	45.477743
GO:0016705	0.305630	✓	6.927945	45.477743
GO:0004497	0.305630	✓	9.121786	45.477743
GO:0033695	0.305599	✗	9.121786	50.186088
GO:0016712	0.305599	✗	9.121786	56.229552
GO:0034875	0.305569	✗	9.121786	56.229552
GO:0071614	0.197727	✗	9.121786	65.020249

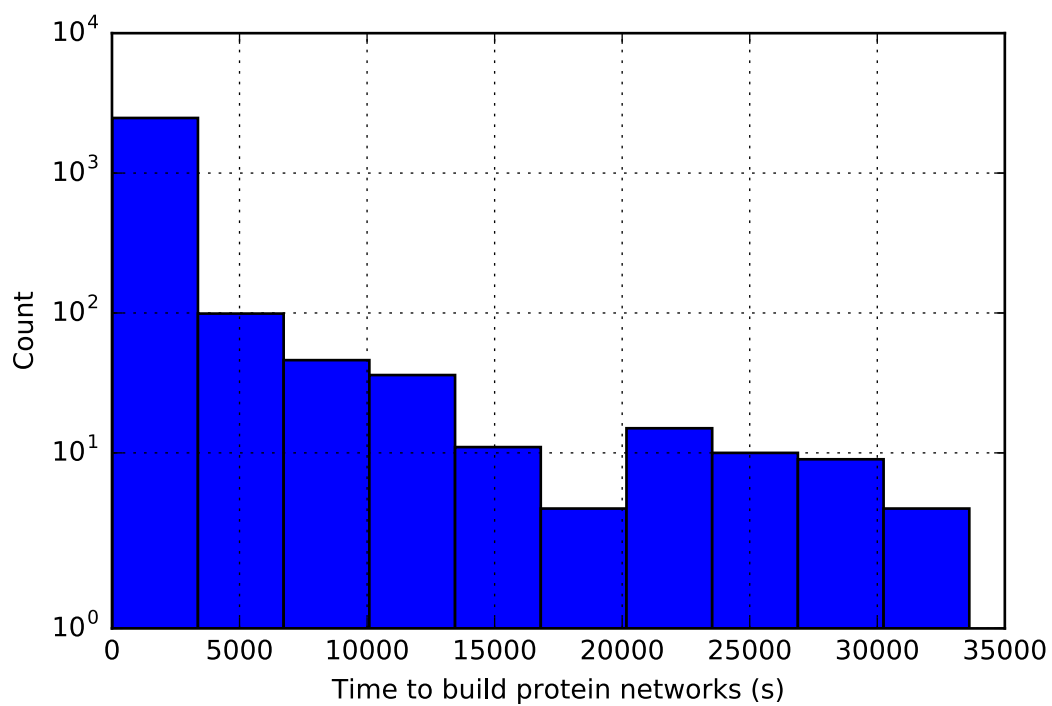
GO Term	Posterior Probability	In GOA	True Positives (Bits)	False Positives (Bits)
GO:0008391	0.197727	<b>X</b>	9.121786	68.243262
GO:0008392	0.197708	<b>X</b>	9.121786	68.269574
GO:0008405	0.197688	<b>X</b>	9.121786	75.033339
<b>GO:0008404</b>	<b>0.197688</b>	<b>X</b>	<b>9.121786</b>	<b>82.382068</b>
<b>GO:0008395</b>	<b>0.176671</b>	<b>✓</b>	<b>13.463748</b>	<b>82.382068</b>
GO:0004509	0.176653	<b>X</b>	13.463748	87.535873
GO:0008144	0.166730	<b>X</b>	13.463748	94.584589
GO:0008389	0.163372	<b>✓</b>	21.739872	94.584589
GO:0016709	0.133449	<b>X</b>	21.739872	95.050142
GO:0016829	0.132736	<b>X</b>	21.739872	99.788061
GO:0016830	0.132722	<b>X</b>	21.739872	101.514718
GO:0016832	0.132709	<b>X</b>	21.739872	103.709974
GO:0047442	0.132696	<b>X</b>	21.739872	107.879899
GO:0004508	0.132696	<b>X</b>	21.739872	110.497652
GO:0072533	0.076919	<b>X</b>	21.739872	120.630151
GO:0072532	0.076919	<b>X</b>	21.739872	131.762651
GO:0072552	0.076911	<b>X</b>	21.739872	131.762651
GO:0072551	0.076911	<b>X</b>	21.739872	131.762651
GO:0072550	0.076911	<b>X</b>	21.739872	131.762651
GO:0072549	0.076911	<b>X</b>	21.739872	131.762651
GO:0072548	0.076911	<b>X</b>	21.739872	131.762651
GO:0072547	0.076911	<b>X</b>	21.739872	131.762651

---

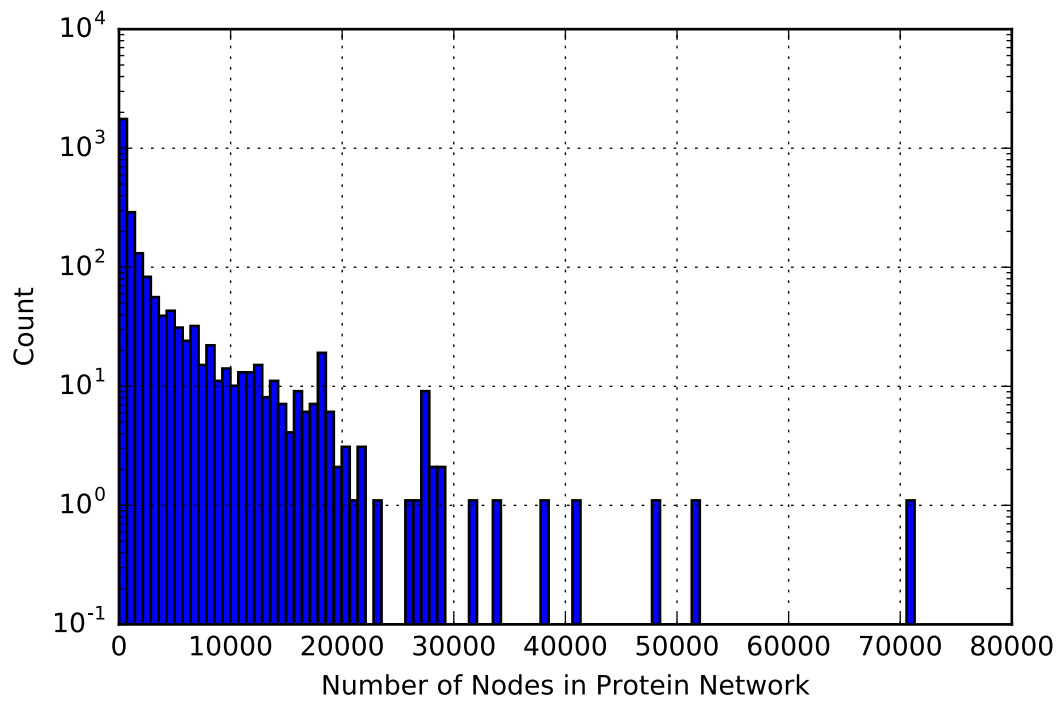
GO Term	Posterior		True Positives	False Positives
	Probability	In GOA	(Bits)	(Bits)
GO:0035302	0.076867	<b>X</b>	21.739872	139.645294
GO:0018631	0.061405	<b>X</b>	21.739872	150.777794
GO:0052722	0.058086	<b>X</b>	21.739872	159.095508
GO:0097008	0.057711	<b>X</b>	21.739872	170.228008
GO:0097007	0.057711	<b>X</b>	21.739872	181.360508
GO:0047084	0.057711	<b>X</b>	21.739872	192.493007
GO:0046996	0.056978	<b>X</b>	21.739872	199.609938
GO:0047055	0.056972	<b>X</b>	21.739872	204.564134

---

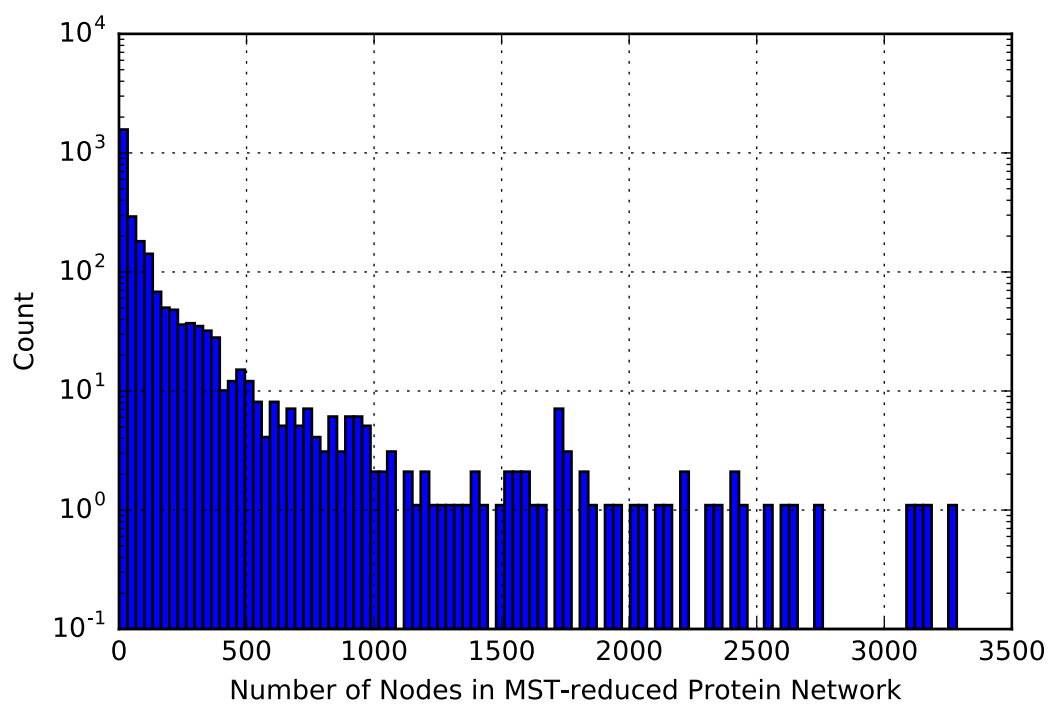




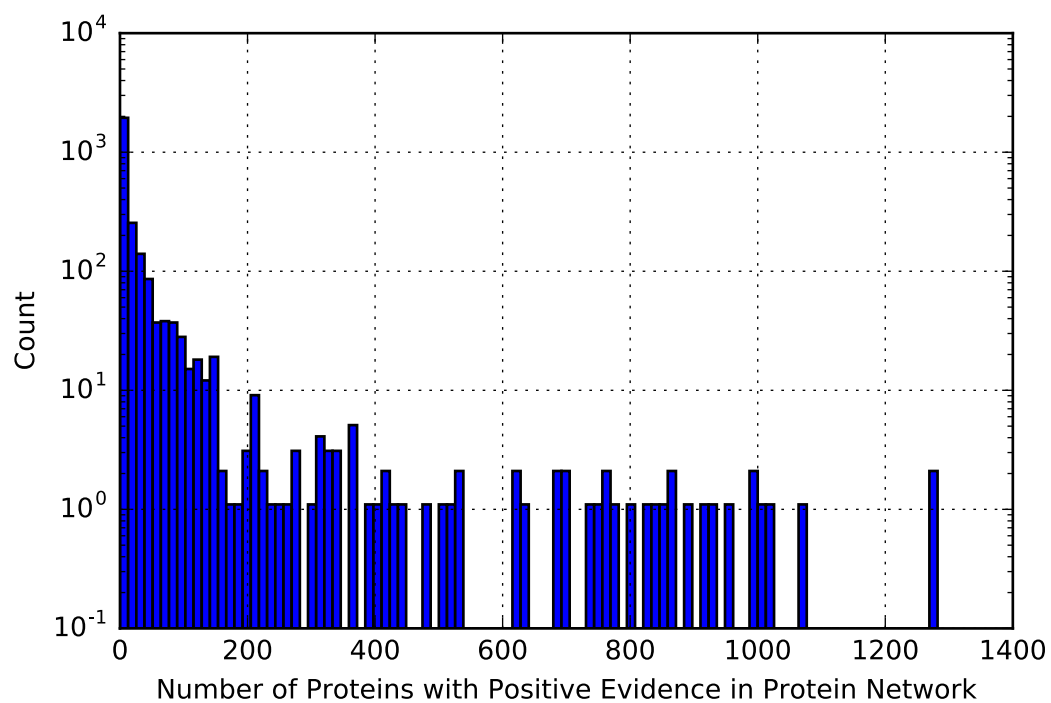
**Figure A.1:** Distribution of time to build the network.



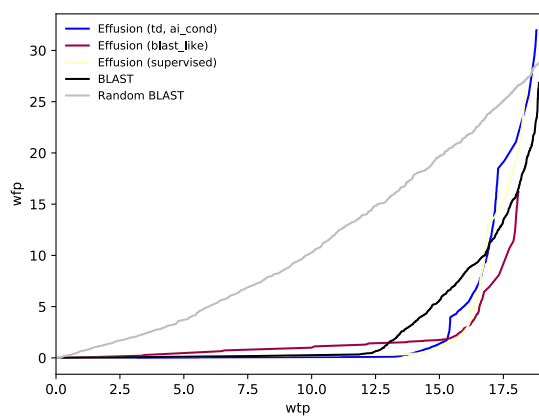
**Figure A.2:** Distribution of number of nodes in protein network.



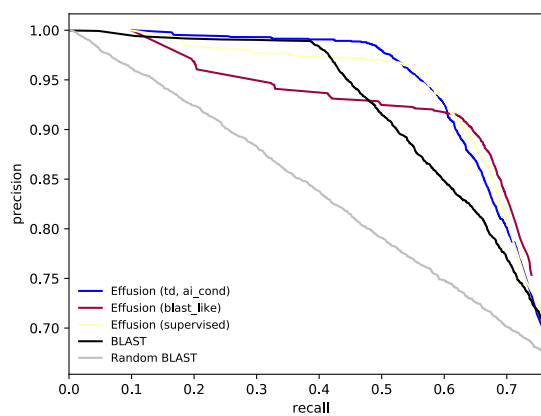
**Figure A.3:** Distribution of number of nodes in protein network after directing it, rooting it, and pruning it.



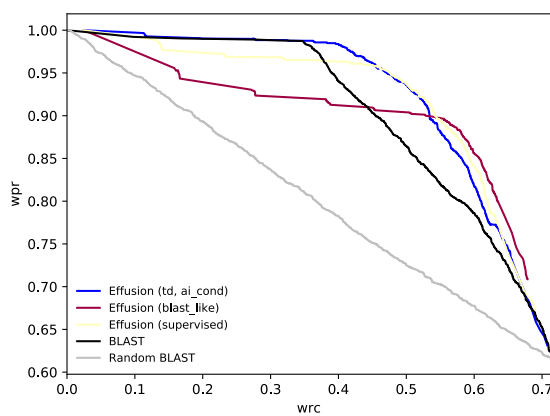
**Figure A.4:** Distribution of number of nodes in protein network with positive evidence.



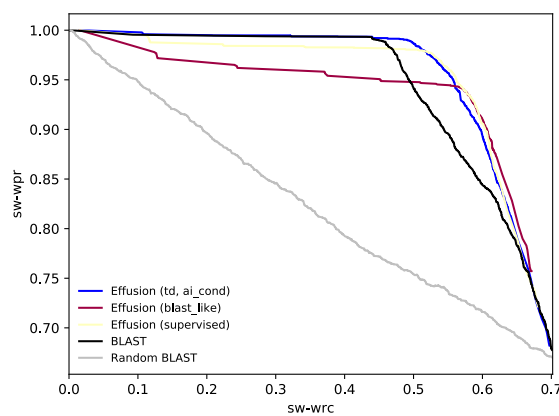
(a) WFP vs. WTP



(b) Precision vs. Recall

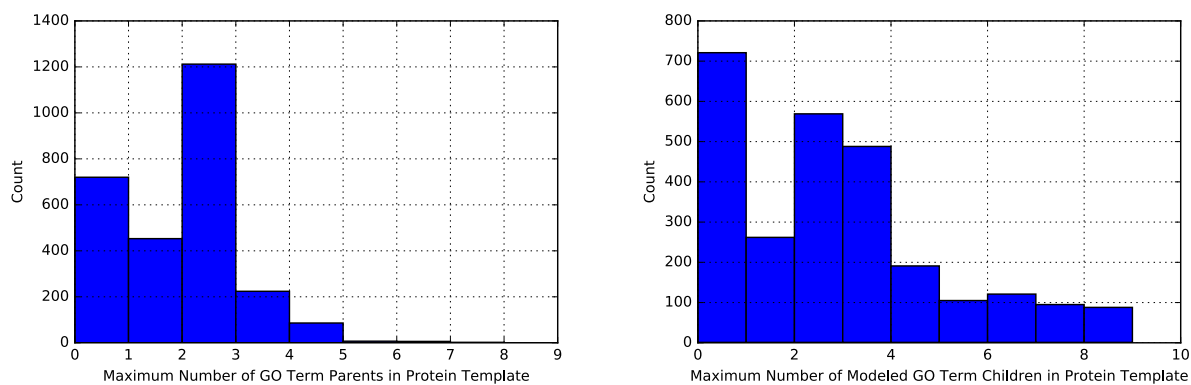


(c) Weighted Precision vs. Weighted Recall



(d) Sample-Weighted Weighted Precision vs. Sample-Weighted Weighted Recall

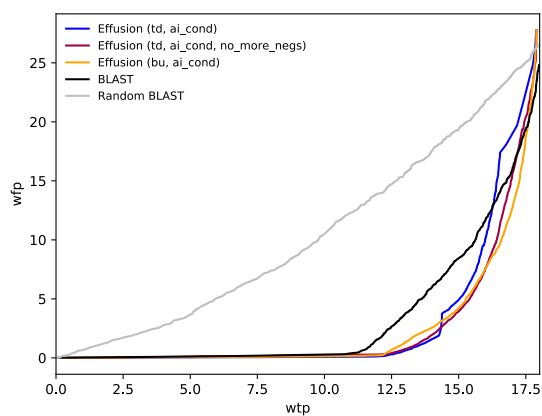
**Figure A.5:** Performance curves showing value of adding unannotated proteins. Evaluated over treated proteins, using the metrics indicated



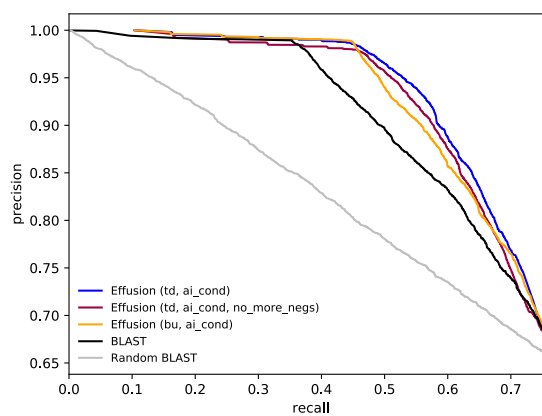
(a) Distribution of maximum number of GO parents in protein template

(b) Distribution of maximum number of modeled GO children in protein template

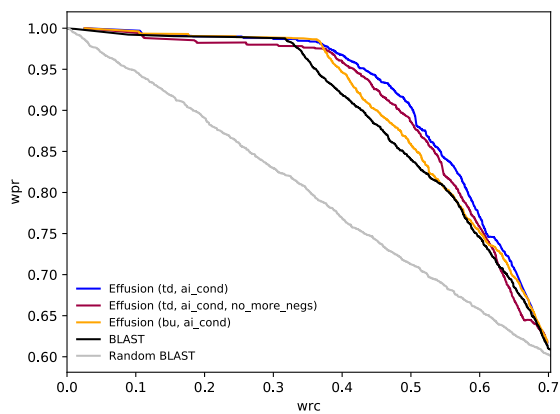
**Figure A.6:** Distributions for the maximum number of parents or modeled children in the protein template. The difficulty of the inference problem strongly depends on the number of GO parents in the top-down model, or GO children in the bottom-up model



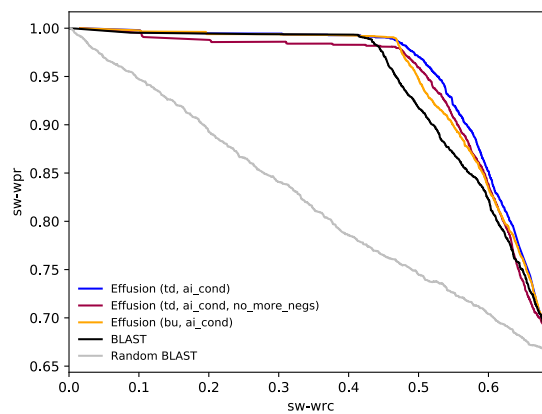
(a) WFP vs. WTP



(b) Precision vs. Recall

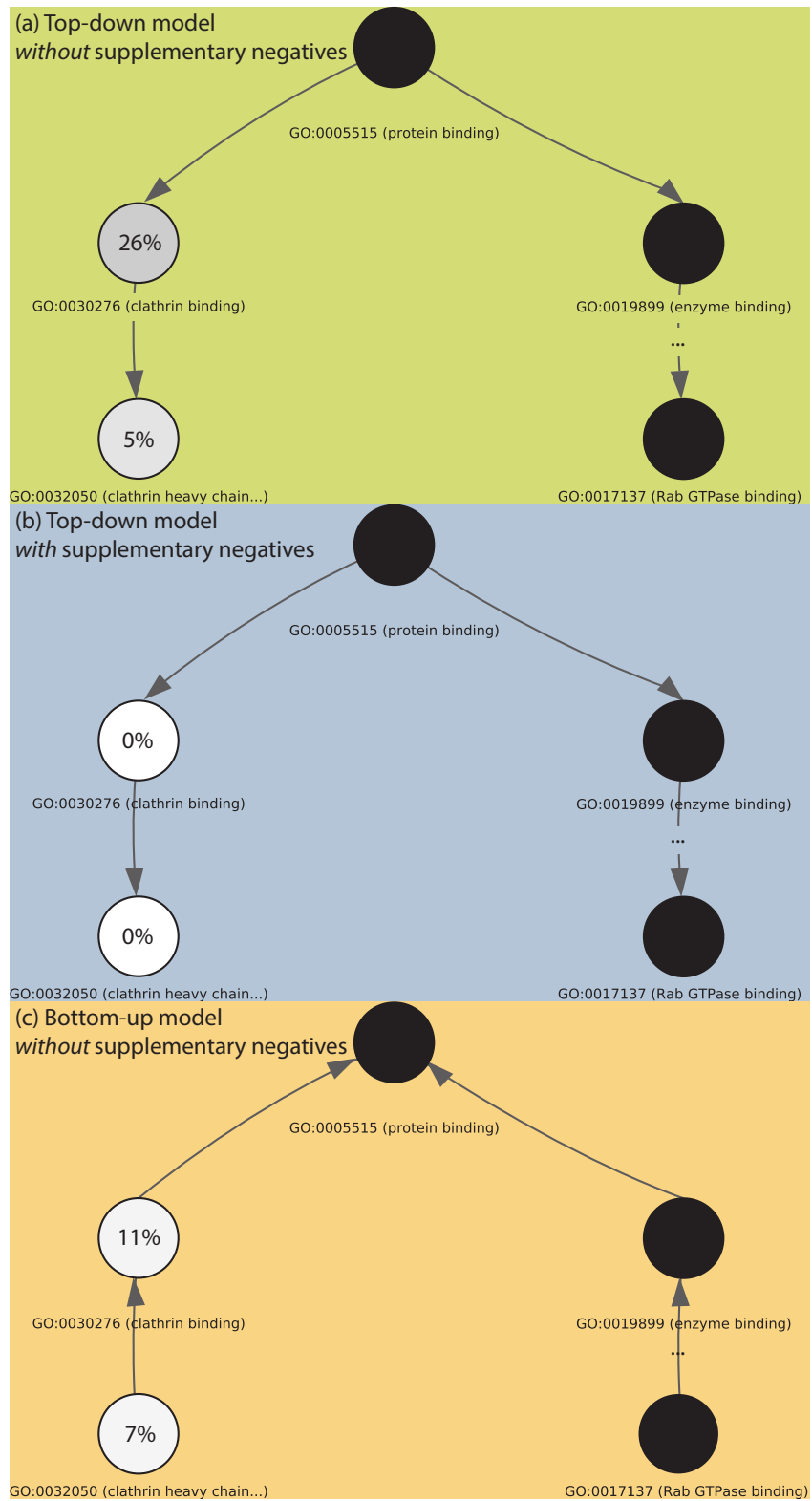


(c) Weighted Precision vs. Weighted Recall



(d) Sample-Weighted Weighted Precision vs. Sample-Weighted Weighted Recall

**Figure A.7:** Performance curves showing value of adding negative evidence. Evaluated over treated proteins, using the metrics indicated





**Figure A.9:** Example comparing top-down model, without and with supplementary negative evidence, and bottom-up model. The query protein is Q921C5. (a) With the top-down model, but without supplementary negative evidence, evidence for protein binding increases the posterior probability for clathrin binding, even though the evidence for protein binding is explained by Rab GTPase binding. (b) In the top down model, we compensate for this by adding negative evidence for clathrin binding, because a protein is rarely positively annotated with clathrin binding when it is positively annotated to sibling enzyme binding. (c) The bottom up model has factors over each GO term and their child GO terms, so the probability for clathrin binding is lowered upon observing enzyme binding.

**Table A.3:** Predictions for Q921C5 made by Effusion with the top-down model, but without supplementary negative evidence. The GO terms are filtered to those that are predicted by all methods.

GO Term	Posterior		True Positives	False Positives
	Probability	In GOA	(Bits)	(Bits)
GO:0003674	1.000000	✓	0.414455	0.000000
GO:0005488	0.999900	✓	1.889660	0.000000
GO:0005515	0.999800	✓	2.906295	0.000000
<b>GO:0019899</b>	<b>0.999700</b>	✓	<b>5.037532</b>	<b>0.000000</b>
GO:0051020	0.999600	✓	6.984703	0.000000
GO:0031267	0.999500	✓	7.071294	0.000000
GO:0017016	0.999400	✓	7.109188	0.000000
GO:0017137	0.999300	✓	7.642070	0.000000
GO:0005102	0.504263	✗	7.642070	2.625682
GO:0097159	0.388429	✗	7.642070	3.888894
GO:1901363	0.376987	✗	7.642070	5.174566
GO:0008092	0.349818	✓	10.888182	5.174566
GO:0005198	0.311069	✗	10.888182	9.722838
GO:0060090	0.280266	✗	10.888182	16.635034
GO:0030674	0.275064	✗	10.888182	16.796222
<b>GO:0030276</b>	<b>0.264024</b>	✗	<b>10.888182</b>	<b>23.120815</b>
GO:0044877	0.236979	✓	14.996187	23.120815
GO:0032403	0.229623	✓	15.118474	23.120815
GO:0001664	0.209859	✗	15.118474	25.202049

GO Term	Posterior Probability	In GOA	True Positives (Bits)	False Positives (Bits)
GO:0003676	0.125381	<b>X</b>	15.118474	25.425997
GO:0008093	0.096530	<b>X</b>	15.118474	26.387523
GO:0034452	0.064105	<b>✓</b>	23.107159	26.387523
GO:0003723	0.050614	<b>X</b>	23.107159	27.359985
GO:0005200	0.049353	<b>X</b>	23.107159	31.392109
GO:0032050	0.049330	<b>X</b>	23.107159	34.289929
GO:0031871	0.045123	<b>X</b>	23.107159	42.868437
GO:0070840	0.029559	<b>✓</b>	29.211342	42.868437
GO:0003729	0.012855	<b>X</b>	29.211342	45.374533
GO:0000339	0.005552	<b>X</b>	29.211342	51.639895
GO:0017091	0.005322	<b>X</b>	29.211342	59.301416
GO:0003730	0.002745	<b>X</b>	29.211342	61.946893
GO:0048027	0.002030	<b>X</b>	29.211342	66.173285
GO:0098808	0.001400	<b>X</b>	29.211342	69.260747
GO:0030350	0.001339	<b>X</b>	29.211342	77.681517
GO:0035368	0.001335	<b>X</b>	29.211342	83.871990
GO:1990715	0.001331	<b>X</b>	29.211342	93.408237
GO:1990825	0.001302	<b>X</b>	29.211342	101.281519
GO:1903231	0.001296	<b>X</b>	29.211342	101.281521
GO:0035925	0.000014	<b>X</b>	29.211342	101.419025

**Table A.4:** Predictions for Q921C5 made by Effusion with the top-down model, including supplementary negative evidence. The GO terms are filtered to those that are predicted by all methods.

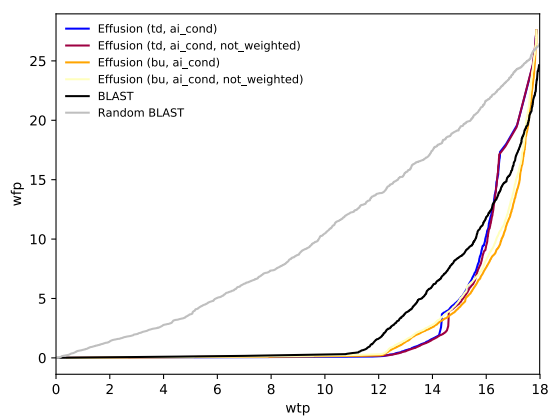
GO Term	Posterior		True Positives	False Positives
	Probability	In GOA	(Bits)	(Bits)
GO:0003674	1.000000	✓	0.414455	0.000000
GO:0005488	0.999900	✓	1.889660	0.000000
GO:0005515	0.999800	✓	2.906295	0.000000
<b>GO:0019899</b>	<b>0.999700</b>	✓	<b>5.037532</b>	<b>0.000000</b>
GO:0051020	0.999600	✓	6.984703	0.000000
GO:0031267	0.999500	✓	7.071294	0.000000
GO:0017016	0.999400	✓	7.109188	0.000000
GO:0017137	0.999300	✓	7.642070	0.000000
GO:0005198	0.000000	✗	7.642070	4.548272
GO:1901363	0.000000	✗	7.642070	5.833944
GO:0097159	0.000000	✗	7.642070	7.097156
GO:0060090	0.000000	✗	7.642070	14.009352
GO:0044877	0.000000	✓	11.750074	14.009352
GO:0005200	0.000000	✗	11.750074	18.041477
GO:0032403	0.000000	✓	11.872362	18.041477
GO:0030674	0.000000	✗	11.872362	18.202665
<b>GO:0030276</b>	<b>0.000000</b>	✗	<b>11.872362</b>	<b>24.527259</b>
GO:0008092	0.000000	✓	15.118474	24.527259
GO:0005102	0.000000	✗	15.118474	27.152940

GO Term	Posterior		True Positives	False Positives
	Probability	In GOA	(Bits)	(Bits)
GO:0003676	0.000000	<b>X</b>	15.118474	27.376888
GO:0070840	0.000000	<b>✓</b>	21.222658	27.376888
GO:0034452	0.000000	<b>✓</b>	29.211342	27.376888
GO:0032050	0.000000	<b>X</b>	29.211342	30.274708
GO:0008093	0.000000	<b>X</b>	29.211342	31.236234
GO:0003723	0.000000	<b>X</b>	29.211342	32.208695
GO:0001664	0.000000	<b>X</b>	29.211342	34.289929
GO:0031871	0.000000	<b>X</b>	29.211342	42.868437
GO:0017091	0.000000	<b>X</b>	29.211342	50.529958
GO:0003729	0.000000	<b>X</b>	29.211342	53.036054
GO:0000339	0.000000	<b>X</b>	29.211342	59.301416
GO:1990825	0.000000	<b>X</b>	29.211342	67.174699
GO:1990715	0.000000	<b>X</b>	29.211342	76.710946
GO:1903231	0.000000	<b>X</b>	29.211342	76.710948
GO:0098808	0.000000	<b>X</b>	29.211342	79.798411
GO:0048027	0.000000	<b>X</b>	29.211342	84.024802
GO:0035368	0.000000	<b>X</b>	29.211342	90.215275
GO:0030350	0.000000	<b>X</b>	29.211342	98.636045
GO:0003730	0.000000	<b>X</b>	29.211342	101.281521
GO:0035925	0.000000	<b>X</b>	29.211342	101.419025

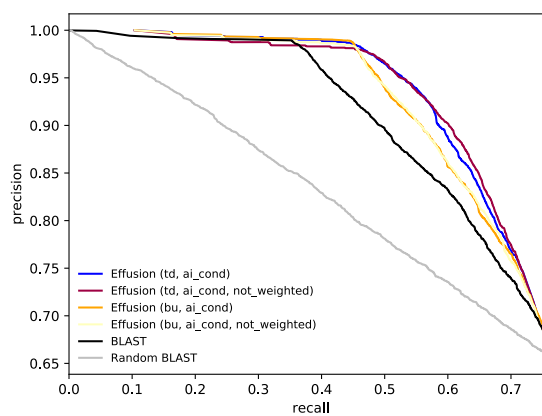
**Table A.5:** Predictions for Q921C5 made by Effusion with the bottom-up model. The bottom-up model does not require supplementary negative evidence. The GO terms are filtered to those that are predicted by all methods.

GO Term	Posterior		True Positives	False Positives
	Probability	In GOA	(Bits)	(Bits)
GO:0003674	1.000000	✓	0.414455	0.000000
GO:0005488	0.999900	✓	1.889660	0.000000
GO:0005515	0.999800	✓	2.906295	0.000000
<b>GO:0019899</b>	<b>0.999700</b>	✓	<b>5.037532</b>	<b>0.000000</b>
GO:0051020	0.999600	✓	6.984703	0.000000
GO:0031267	0.999500	✓	7.071294	0.000000
GO:0017016	0.999400	✓	7.109188	0.000000
GO:0017137	0.999300	✓	7.642070	0.000000
GO:0005198	0.386561	✗	7.642070	4.548272
GO:0008092	0.361044	✓	10.888182	4.548272
GO:0060090	0.327731	✗	10.888182	11.460468
GO:0030674	0.327699	✗	10.888182	11.621656
GO:0008093	0.309126	✗	10.888182	12.583182
GO:0005200	0.298606	✗	10.888182	16.615307
GO:0044877	0.250371	✓	14.996187	16.615307
GO:0032403	0.244758	✓	15.118474	16.615307
GO:0005102	0.188700	✗	15.118474	19.240989
GO:0070840	0.172669	✓	21.222658	19.240989
GO:0097159	0.107491	✗	21.222658	20.504201

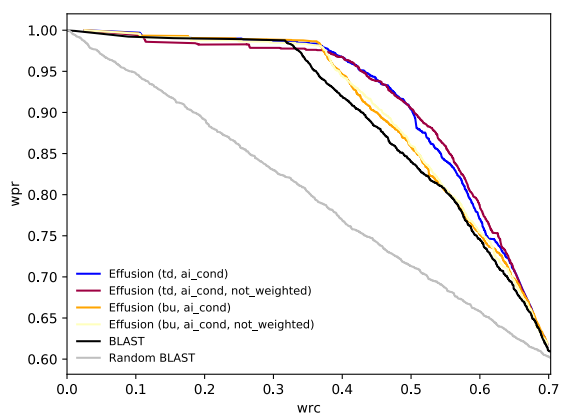
GO Term	Posterior Probability	In GOA	True Positives (Bits)	False Positives (Bits)
<b>GO:0030276</b>	<b>0.106014</b>	<b>X</b>	<b>21.222658</b>	<b>26.828794</b>
GO:1901363	0.099213	X	21.222658	28.114466
GO:0001664	0.098010	X	21.222658	30.195700
GO:0031871	0.072663	X	21.222658	38.774207
GO:0032050	0.067249	X	21.222658	41.672027
GO:0003676	0.058159	X	21.222658	41.895975
GO:0034452	0.053829	✓	29.211342	41.895975
GO:0003723	0.045016	X	29.211342	42.868437
GO:0003729	0.045011	X	29.211342	45.374533
GO:0035368	0.021988	X	29.211342	51.565005
GO:0000339	0.007958	X	29.211342	57.830367
GO:0098808	0.007957	X	29.211342	60.917830
GO:1903231	0.002738	X	29.211342	60.917832
GO:1990825	0.002601	X	29.211342	68.791114
GO:1990715	0.002322	X	29.211342	78.327361
GO:0048027	0.001620	X	29.211342	82.553753
GO:0003730	0.001333	X	29.211342	85.199230
GO:0030350	0.001226	X	29.211342	93.620000
GO:0017091	0.000342	X	29.211342	101.281521
GO:0035925	0.000200	X	29.211342	101.419025



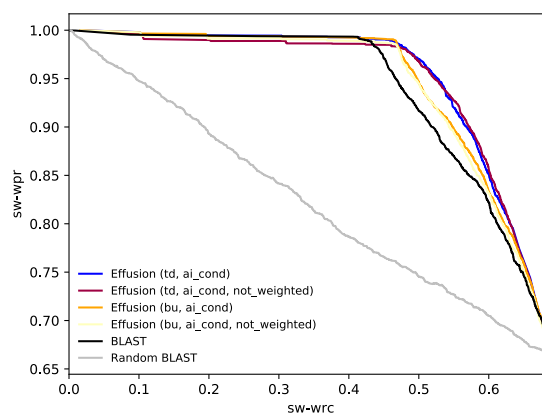
(a) WFP vs. WTP



(b) Precision vs. Recall



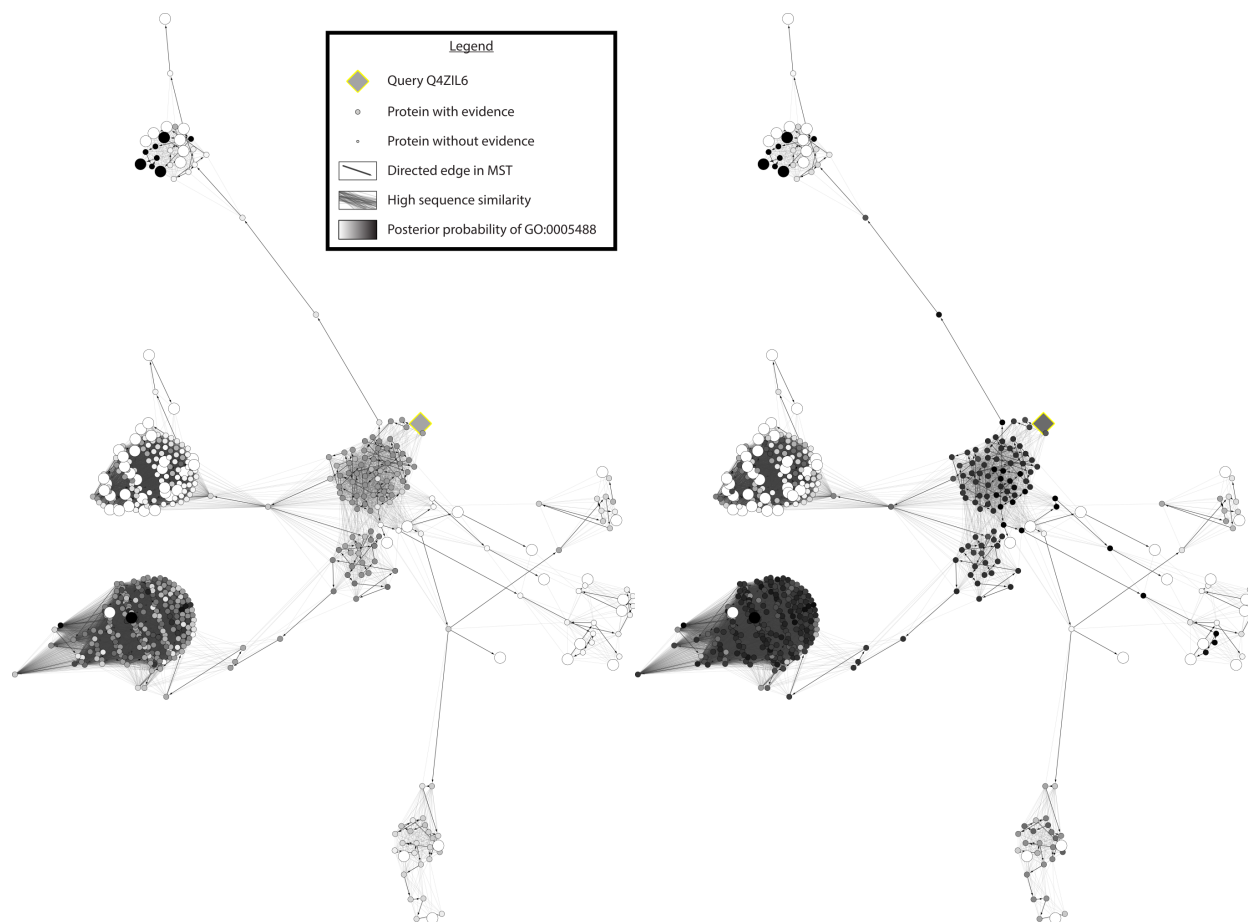
(c) Weighted Precision vs. Weighted Recall



(d) Sample-Weighted Weighted Precision vs. Sample-Weighted Weighted Recall

**Figure A.10:** Performance curves showing value of weighting counts when computing the parameters. Evaluated over treated proteins, using the metrics indicated

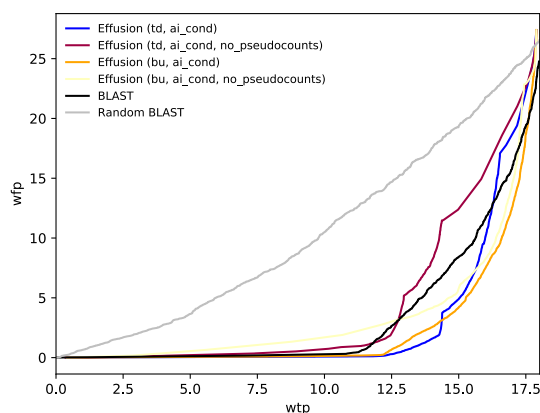




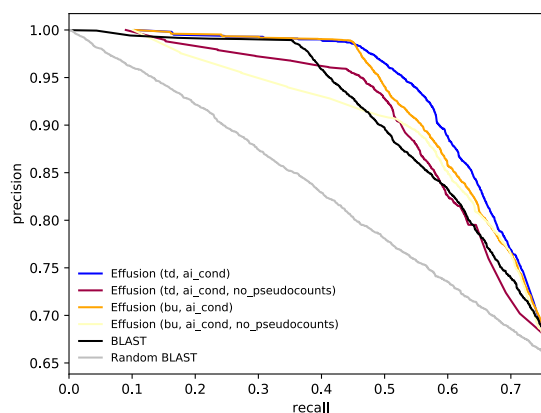
(a) Network view of predictions for UniProt P34945 without weighting counts in the contingency tables by the information content of the sample, colored by GO:0005488

(b) Network view of predictions for UniProt P34945 with weighting counts in the contingency tables by the information content of the sample, colored by GO:0005488

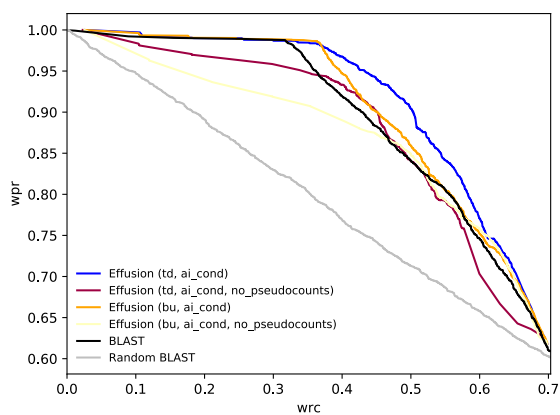
**Figure A.11:** A network view of an example, with and without weighting.



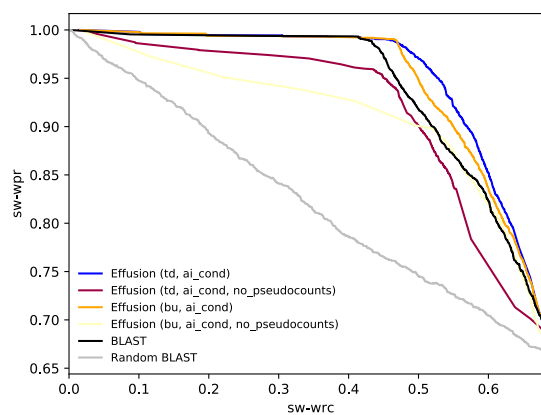
(a) WFP vs. WTP



(b) Precision vs. Recall

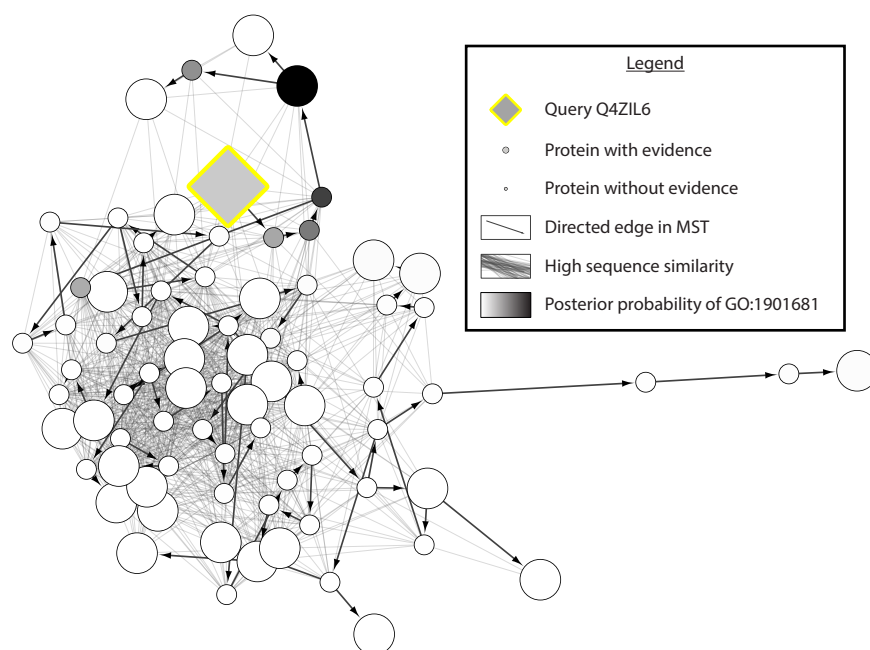


(c) Weighted Precision vs. Weighted Recall



(d) Sample-Weighted Weighted Precision vs. Sample-Weighted Weighted Recall

**Figure A.12:** Performance curves showing value of adding pseudocounts to the contingency tables. Evaluated over proteins that had predictions by all methods represented, using the metrics indicated

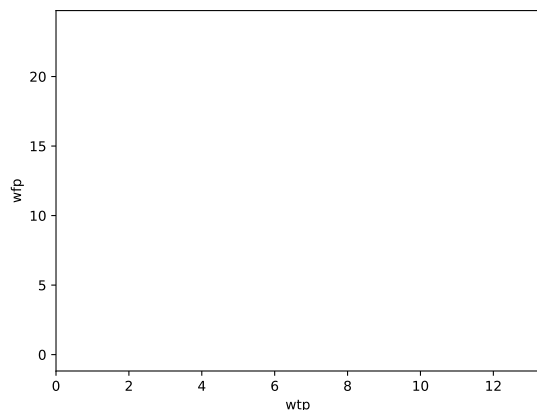


**Figure A.13:** Network view of predictions for UniProt Q6UWY2 without using pseudocounts, colored by GO:1901681. GO:1901681 has not often been experimentally observed, so without pseudocounts, its probability decays more quickly than desired.

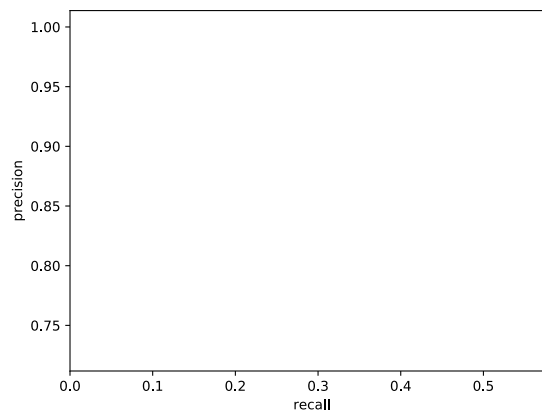
Effusion (bu, ai)  
 Effusion (bu, ai\_cond)  
 Effusion (bu, ai\_ijgp)  
 Effusion (bu, ai\_ijgp\_cond)  
 Effusion (bu, bp)  
 Effusion (bu, glc)  
 Effusion (bu, ijgp)  
 Effusion (bu, samplesearch)  
 Effusion (bu, trwbp)  
 Effusion (td, ai)  
 Effusion (td, ai\_cond)  
 Effusion (td, ai\_ijgp)  
 Effusion (td, ai\_ijgp\_cond)  
 Effusion (td, bp)  
 Effusion (td, glc)  
 Effusion (td, ijgp)  
 Effusion (td, samplesearch)  
 Effusion (td, trwbp)  
 BLAST  
 Random BLAST

**Figure A.14:** Legend for Figure A.15.**Table A.6:** Table of inference algorithms

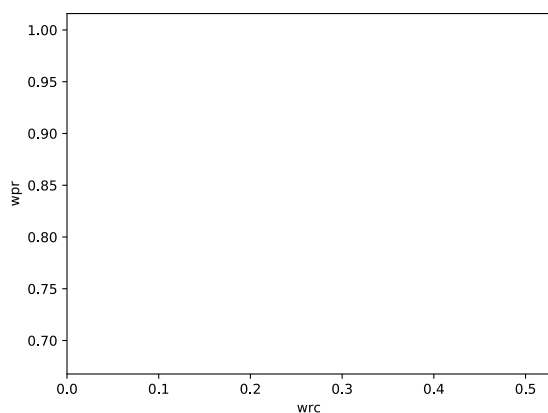
Short Name	Full Name
ai	adaptive inference
ai_cond	adaptive inference with conditioning
ai_ijgp_cond	adaptive inference with iterative-join-graph propagation and conditioning
bp	belief propagation
glc	generalized loop correction
ijgp	iterative-join-graph propagation
samplesearch	SampleSearch
trwbp	tree-reweighted belief propagation



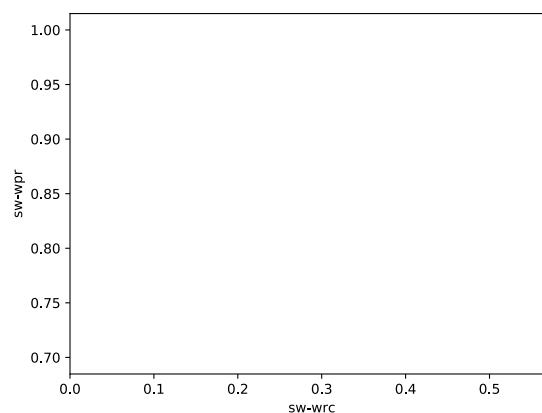
(a) WFP vs. WTP



(b) Precision vs. Recall



(c) Weighted Precision vs. Weighted Recall

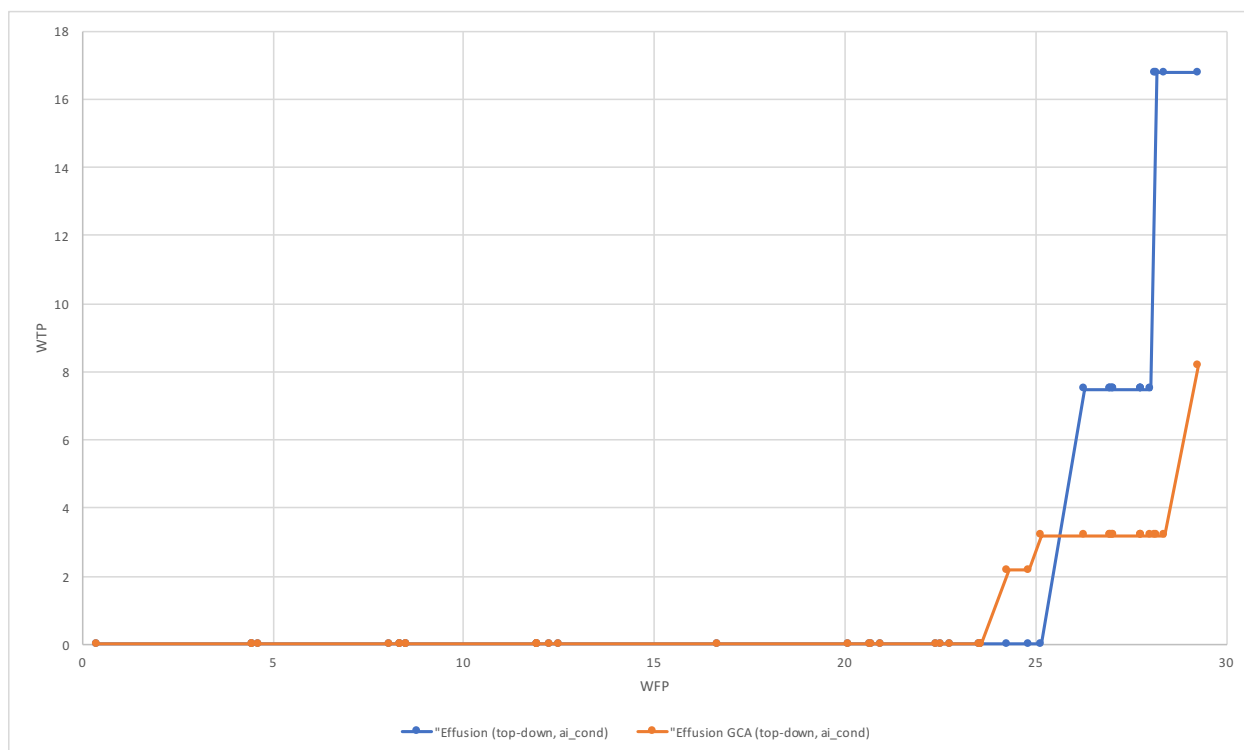


(d) Sample-Weighted Weighted Precision vs. Sample-Weighted Weighted Recall

**Figure A.15:** Plots showing the performance of different inference engines on the top-down model and the bottom-up model. Evaluated over all proteins, including those for which some methods failed to make predictions

## Appendix B

### Supplementary material for Chapter 3



**Figure B.1:** Performance plot comparing Effusion GCA (top-down, ai\_cond) and Effusion (top-down, ai\_cond) on UniProt P78334

**Table B.1:** Predictions for P78334 made by Effusion GCA (top\_down, ai\_cond). For comparison, predictions made by Effusion (top\_down, ai\_cond) Table B.2. The GO terms are filtered to those that are predicted by both methods.

GO Term	Posterior Probability	In GOA	True Positives (Bits)	False Positives (Bits)
GO:0003674	1.000000	✓	0.414455	0.000000
GO:0060089	0.999510	✓	4.512552	0.000000
GO:0004872	0.998071	✓	4.512740	0.000000
GO:0099600	0.995562	✓	4.672222	0.000000
GO:0005215	0.994323	✓	8.086758	0.000000
GO:0004871	0.993672	✓	11.944359	0.000000
GO:0038023	0.993112	✓	11.950580	0.000000
GO:0004888	0.989607	✓	11.950725	0.000000
GO:0022892	0.973083	✓	12.271890	0.000000
GO:0022857	0.971282	✓	12.526606	0.000000
GO:0016917	0.968074	✓	20.682580	0.000000
GO:0022891	0.953591	✓	20.701747	0.000000
GO:0015075	0.937574	✓	20.944265	0.000000
GO:0008509	0.854694	✓	22.408299	0.000000
GO:0004890	0.774346	✗	22.408299	0.703018
GO:0005488	0.622386	✗	22.408299	2.178223
GO:0015103	0.524925	✓	24.286373	2.178223
GO:0005515	0.438734	✗	24.286373	3.194858
GO:0015108	0.290903	✓	25.155245	3.194858



GO Term	Posterior Probability	In GOA	True Positives (Bits)	False Positives (Bits)
GO:0022803	0.150374	✓	26.970277	3.194858
GO:0015267	0.150359	✓	26.970442	3.194858
GO:0022838	0.150344	✓	27.033732	3.194858
GO:0005216	0.146677	✓	27.038145	3.194858
GO:0005253	0.135799	✓	27.042008	3.194858
GO:0022836	0.094703	✓	27.778316	3.194858
GO:0022834	0.094694	✓	27.795363	3.194858
GO:0015276	0.094684	✓	27.796507	3.194858
GO:0099095	0.094675	✓	28.015148	3.194858
GO:0005254	0.072974	✓	28.015148	3.194858
GO:0019904	0.062237	✗	28.015148	8.173894
GO:0005230	0.042338	✓	29.256364	8.173894
GO:0005237	0.042333	✗	29.256364	12.476998
GO:0005231	0.042333	✗	29.256364	13.145229
GO:0016933	0.042329	✗	29.256364	16.761029
GO:0016934	0.042325	✗	29.256364	16.761029

**Table B.2:** Predictions for P78334 made by Effusion (top-down, ai\_cond). For comparison, predictions made by Effusion GCA (top-down, ai\_cond) are in Table B.1. The GO terms are filtered to those that are predicted by both methods.

GO Term	Posterior Probability	In GOA	True Positives (Bits)	False Positives (Bits)
GO:0003674	1.000000	✓	0.414455	0.000000
GO:0060089	0.999351	✓	4.512552	0.000000
GO:0004871	0.998586	✓	8.370153	0.000000
GO:0004872	0.998486	✓	8.370341	0.000000
GO:0099600	0.998386	✓	8.529823	0.000000
GO:0038023	0.998386	✓	8.536044	0.000000
GO:0004888	0.998287	✓	8.536189	0.000000
GO:0016917	0.972512	✓	16.692164	0.000000
GO:0005215	0.904364	✓	20.106700	0.000000
GO:0022892	0.904273	✓	20.427865	0.000000
GO:0022857	0.904273	✓	20.682580	0.000000
GO:0022891	0.904183	✓	20.701747	0.000000
GO:0022803	0.904183	✓	22.516779	0.000000
GO:0015267	0.904092	✓	22.516943	0.000000
GO:0015075	0.904092	✓	22.759461	0.000000
GO:0022838	0.904002	✓	22.822751	0.000000
GO:0022836	0.904002	✓	23.559060	0.000000
GO:0022834	0.903911	✓	23.576107	0.000000
GO:0005216	0.903911	✓	23.580520	0.000000

GO Term	Posterior Probability	In GOA	True Positives (Bits)	False Positives (Bits)
GO:0015276	0.903821	✓	23.581664	0.000000
GO:0005230	0.903730	✓	24.822881	0.000000
GO:0005231	0.891673	✗	24.822881	0.668232
GO:0016933	0.789369	✗	24.822881	4.284031
GO:0004890	0.743026	✗	24.822881	4.987050
GO:0005488	0.580448	✗	24.822881	6.462255
GO:0005515	0.380296	✗	24.822881	7.478890
GO:0008509	0.058065	✓	26.286915	7.478890
GO:0019904	0.056282	✗	26.286915	12.457925
GO:0005237	0.036488	✗	26.286915	16.761029
GO:0015103	0.019908	✓	28.164989	16.761029
GO:0005253	0.019906	✓	28.168851	16.761029
GO:0099095	0.019904	✓	28.387492	16.761029
GO:0015108	0.000953	✓	29.256364	16.761029
GO:0005254	0.000953	✓	29.256364	16.761029
GO:0016934	0.000953	✗	29.256364	16.761029

# Appendix C

## Code

Source code is available at <https://github.com/babbittlab/effusion>.

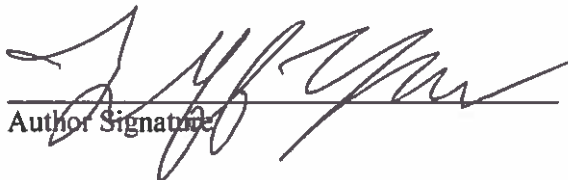
**Publishing Agreement**

*It is the policy of the University to encourage the distribution of all theses, dissertations, and manuscripts. Copies of all UCSF theses, dissertations, and manuscripts will be routed to the library via the Graduate Division. The library will make all theses, dissertations, and manuscripts accessible to the public and will preserve these to the best of their abilities, in perpetuity.*

---

***Please sign the following statement:***

*I hereby grant permission to the Graduate Division of the University of California, San Francisco to release copies of my thesis, dissertation, or manuscript to the Campus Library to provide access and preservation, in whole or in part, in perpetuity.*

  
\_\_\_\_\_  
Author Signature

3/29/2018  
Date