Assessing the Role of Clusters Derived from Large Sequence Similarity Networks for Gene Function Predictions

Parth H. Vora

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science and Applications

Shiv D. Kale, Co-chair
T. M. Murali, Co-chair
Lenwood S. Heath

May 08, 2020
Blacksburg, Virginia

Keywords: Bioinformatics, Gene Function Prediction
Copyright 2020, Parth H. Vora

Assessing the Role of Clusters Derived from Large Sequence Similarity Networks for Gene Function Predictions

Parth H. Vora

(ABSTRACT)

Large scale genomic sequencing efforts have resulted in a massive inflow of raw sequence data. This raw data, when appropriately processed and analyzed, can provide insight to a trained biologist and aid in hypothesis-driven research. Given the time and resource requirements necessary for biological experiments, computational predictions of gene functions can aid in reducing a large list of candidate genes to a few promising targets. Various computational solutions have been proposed and developed for gene function prediction. These solutions utilize various forms of data, such as DNA/RNA/protein sequences, protein structures, interaction networks, literature mining and a combination of these data sources. However, these methods do not always produce precise results as the underlying data sets used for training or modeling are quite sparse. We developed and used a massive sequence similarity network build over 108 million known protein sequences to aid in protein function prediction. Predictions are made through the alignment of query sequences to representative sequences for a given cluster derived from the massive sequence similarity network. Derived clusters aggregate information (particularly that from the Gene Ontology) from respective members, which we then consolidate through a novel weighted path method. We evaluate our method on four holdout datasets using CAFA evaluation metrics. Our results suggest that clustering significantly reduces the time and memory requirements, with a marginal impact on predictive power. At lower sequence similarity thresholds, our method outperforms other gold standard methods.

Assessing the Role of Clusters Derived from Large Sequence Similarity Networks for Gene Function Predictions

Parth H. Vora

(GENERAL AUDIENCE ABSTRACT)

We often think of a protein as a nutritional requirement. However, proteins are far more than just food, they play countless and unappreciated roles in facilitating life. From transporting nutrients in the body, synthesis of hormones, functioning as enzymes to expediting chemical reactions, serving as the scaffold for cells and tissues, to protecting the body against foreign pathogens. On a molecular level, each protein is made up of chains of 20 different amino acids, just like a chain of beads, that are then folded to create a 3-dimensional structure. The variations in the ordering of amino acids result in different types of proteins. There are millions of genes across known life, and they perform different functions when translated into proteins. Nature has given us many proteins with interesting properties, and the low cost of sequencing their precursors (DNA) has resulted in large amounts of sequence data that is not yet associated with a function. Biological experiments to determine the function of a protein can be time consuming and expensive. We built a massive network encompassing 108 million protein sequences based on sequence similarity. This ensures that we make use of as much data as possible to make better predictions. Specifically, our work focuses on utilizing this information of similar proteins to aid in predicting the functions of a protein given its sequences. It is based on the idea of guilt by association, such that if two proteins are similar in sequences, they perform similar functions. We show that using computationally efficient methods and large datasets, one can achieve fast and highly precise predictions.

Dedication

For Mom and Dad,

couldn't have done it without your genes

Acknowledgments

Foremost, I would like to express my most profound appreciation and gratitude for my advisor Dr. Kale who has continually conveyed a spirit of adventure and excitement in regards to research. The thesis would would not have been possible without his encouragement and patience. To my committee co-chair, Dr. Murali and committee member, Dr. Heath, I am extremely grateful for your valuable support, genius insight, feedback and much more. Lastly, a special thank you to Virginia Tech for providing an inspiring environment to learn and grow.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Proteins play a very important role across the domain of life. Knowledge about the functions of proteins is vital to understanding life at a molecular level. It can drive innovation and research in the medical field, like aiding, in the analysis of diseases and the development of targeted drugs [9, 37, 73, 74, 75]. Standard practices like wet-lab experiments and literature curation can be expensive, time-consuming, and even difficult in certain cases. Functional discovery is also limited by the ethics of experimentation and the interests of scientists. The steady logarithmic rate of increase in performance for DNA sequencing platforms over the past years and low-throughput methods to discover the functions of proteins has resulted in a growing gap between the accumulated genomic data and understanding the functions associated with the data [61]. Without a proper system to store, organize, and process the large amounts of data, we will miss out on the insight provided by the sequence data.

Dr. Kale built a network of over 108 million protein sequences to cluster proteins based on sequence similarity. Clustering similar proteins enable us to propagate functional information in a cluster. More specifically, it is based on the idea that if two proteins have similar sequences, then they must perform similar functions [52]. Every time there is a new protein, whose function is not known, we can associate it to a cluster based on sequence similarity and transfer functional information to the unannotated protein sequence.

In this thesis, we show that the incorporation of larger annotation database dramatically increases the performance of prediction using homology transfer. This increase in predictive

power comes at the cost of computing time and large output file sizes. Using sequence similarity networks, we can reduce the search space and consequently reduce the time and compute requirements.

Dr. Kale provided the Neighborly sequence similarity clusters, CD-HIT clusters, and representative sequences for analysis as well as the temporal holdout test sets. We devised a prediction pipeline for Neighborly networks based on rank and threshold, applying evaluation criteria used by the CAFA competition [33]. Additionally, to gain an appreciation for how close our predictions are from the true functions, we developed a weighted accuracy measure with guidance from the committee members. We also benchmarked our predictions against a smaller DIAMOND database and one of the current best performing sequence-based prediction method DeepGoPlus [39]. Additionally, we also evaluate the predictions made by CD-HIT networks to determine what is the impact of different networks on gene function prediction. To compare our prediction pipeline with current best-performing algorithms, we evaluate our cluster based annotation transfer method on the CAFA 3 datasets along with two temporal holdout datasets. We also devised a weighted-path method to consolidate aggregated functional information associated with Neighborly clusters. We compare the weighted-path annotations with enrichment analysis method using Jaccard's Index and Lin's Similarity measures. Further, we evaluate the effect of weighted-path annotations for gene function prediction.

1.1 Sequence Similarity

Determining the optimal alignment of nucleotide and protein sequences allows for a standardized and quantitative measure to determine the similarity or dissimilarity of two sequences. An alignment of two sequences is an arrangement of one sequence with respect to the other

such that most of their constituent elements match with each other. The similarity score is a quantitative value that is calculated based on several factors like the degree of overlap between two sequences, the number of positionally identical amino acid sequences, and the number of gaps between the two alignments. Given the explosion of sequencing data in the 21st century, methods to optimize the time and compute required to produce such alignments are in high demand. An algorithm for optimal global sequence alignment was initially put forth by Needleman and Wunsch [48]. Important variations of this algorithm include the Smith-Waterman algorithm [58], which has led to further optimizations, ultimately resulting in the BLAST/BLAT method [4, 5, 36]. Further, BLAST has provided a foundation for several high throughput methods such as DIAMOND [13], USEARCH [21], RAPSearch2 [76], and PAUDA [31]. These methods are designed to generate sensitive alignments for millions of proteins. While these current algorithms provide a needed scale up in terms of time and compute resources to process the size of ever-growing sequence datasets, they often significantly sacrifice sensitivity in regard to capturing all alignments.

1.2 Needleman-Wunsch Algorithm

The Needleman-Wunsch algorithm was designed to address if significant global homology or sequence similarity exists between two full length sequences through identification of their optimal alignment [48]. Optimal sequence alignment can be thought of as the arrangement of two linear sequences to maximize the number of positional matches between two sequences. Formally, this algorithm belongs to a class of dynamic programming problems, where the idea is to iteratively reach the best alignment by using optimal alignments of subsequences. A brute force approach to find the best global alingment, would consider all possible alignments, score each alignment and select the alignment which gets the best score. To align a sequence

of length n with a sequence of length m, we would have to do $\binom{m+n}{n}$ score calculations. Needleman-Wunsch algorithm reduces this time complexity to $\mathcal{O}(nm)$ which denotes the time needed to traverse the position scoring matrix of size n × m.

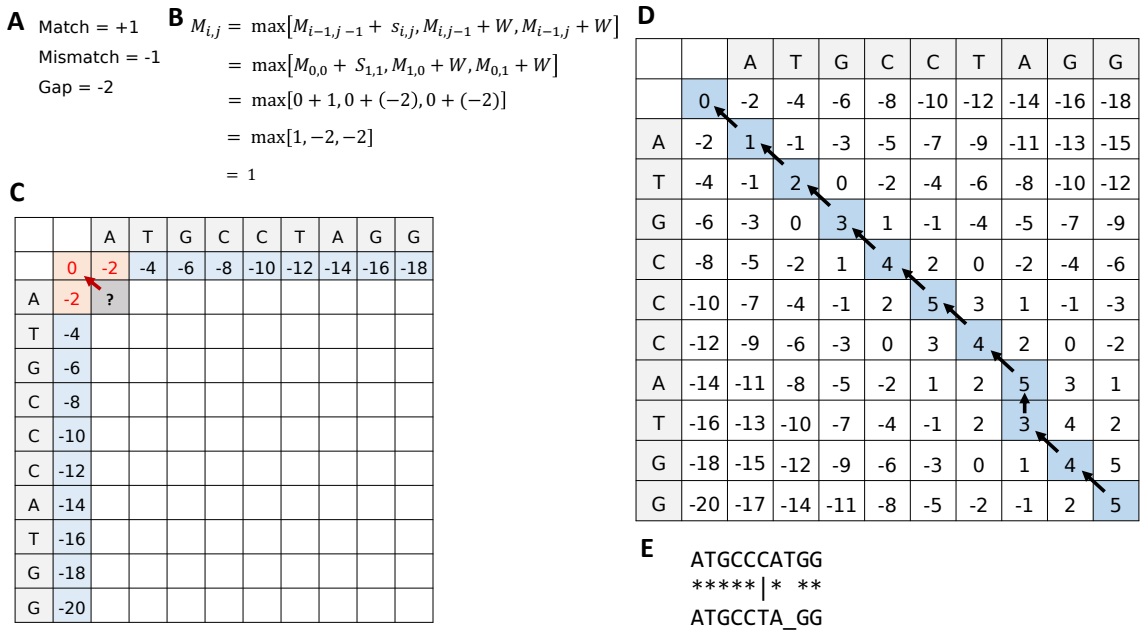

Figure 1.1: Alignment of sequences by Needleman-Wunsch algorithm. (A) Scoring criteria for positional matching. (B) Equation to determine value for a given cell in the 2 dimensional matrix. Values are determined in this instances for highlighted cell in (C). (C) 2 dimensional matrix of alignment scores based on scoring criteria devised by Needleman and Wunsch. The final alignment is shown below. (D) Completed matrix with traceback showing optimal alignment. (E) Inferred final alignment.

The algorithm achieves this end by first instantiating a position scoring matrix based upon the two sequences, then calculating the scores for this matrix, and finally tracing back through the matrix from the highest scoring path. For this matrix each sequential column (top to bottom) and row (left to right) position represents a sequential character from the two sequences being compared. A initial gap position is is put forth for the first position to cover cases where a gap occurs at the beginning of the alignment. The Needleman-Wunch algorithm then proceeds to fill in this matrix based on a formula that incorporates if that

cell is a match (score +1), mismatch (score -1), or gap (score -2) and the scores for the neighboring cells (Figure 1.1 A, Figure 1.1 B). It is also important to note that the first column and first row scores are calculated when the matrix is initialized. Upon completion of the matrix a weighted path can be deduced by tracing back to the origin from the last completed cell by identifying the nearest immediate neighbor with the highest value. (Figure 1.1 C). This trace back is guaranteed to produce an optimal alignment though other optimal alignments of equal weighted value may exist.

1.3 Smith–Waterman Algorithm

The Smith-Waterman algorithm is a variation of Needleman-Wunsch designed to identify not only global alignments, but also the optimal local alignments. Smith-Waterman algorithm does this through dynamic programming approach where small local alignments (solutions to the problem) are found then built upon to identfy the optimal solution. Additionally Smith-Waterman algorithm factors in a substitution matrix that appropriately weigh mismatches, and a gap-scoring scheme that remove harsh penalties associated with large gaps. Weights are determined by either PAM or BLOSUM matrices, which are built upon measured rates of residue frequency and substitution [29, 47]. In the Smith–Waterman Algorithm, negative scoring matrix cells are also set to zero (see Figure 1.2). Rather than start from the final column position, the Smith-Waterman algorithm looks for the highest cell value in the matrix and then traces back. This specific function facilitates local alignment rather than centering on global alignment. Local alignment helps find conserved domains and motifs between two sequences. In many cases, this algorithm can produce multiple alignments of a similar score, and therefore parameter weight for gaps, matches, and types of mismatches are critical for optimization (see Figure 1.2). In Figure 1.2, we show the multiple local alignments that can

be produced from a single sequence with identical scores, thus the ranking of alignments must incorporate other alignment metrics, such as length of overlap, to identify a a statistically significant or optimal alignment.

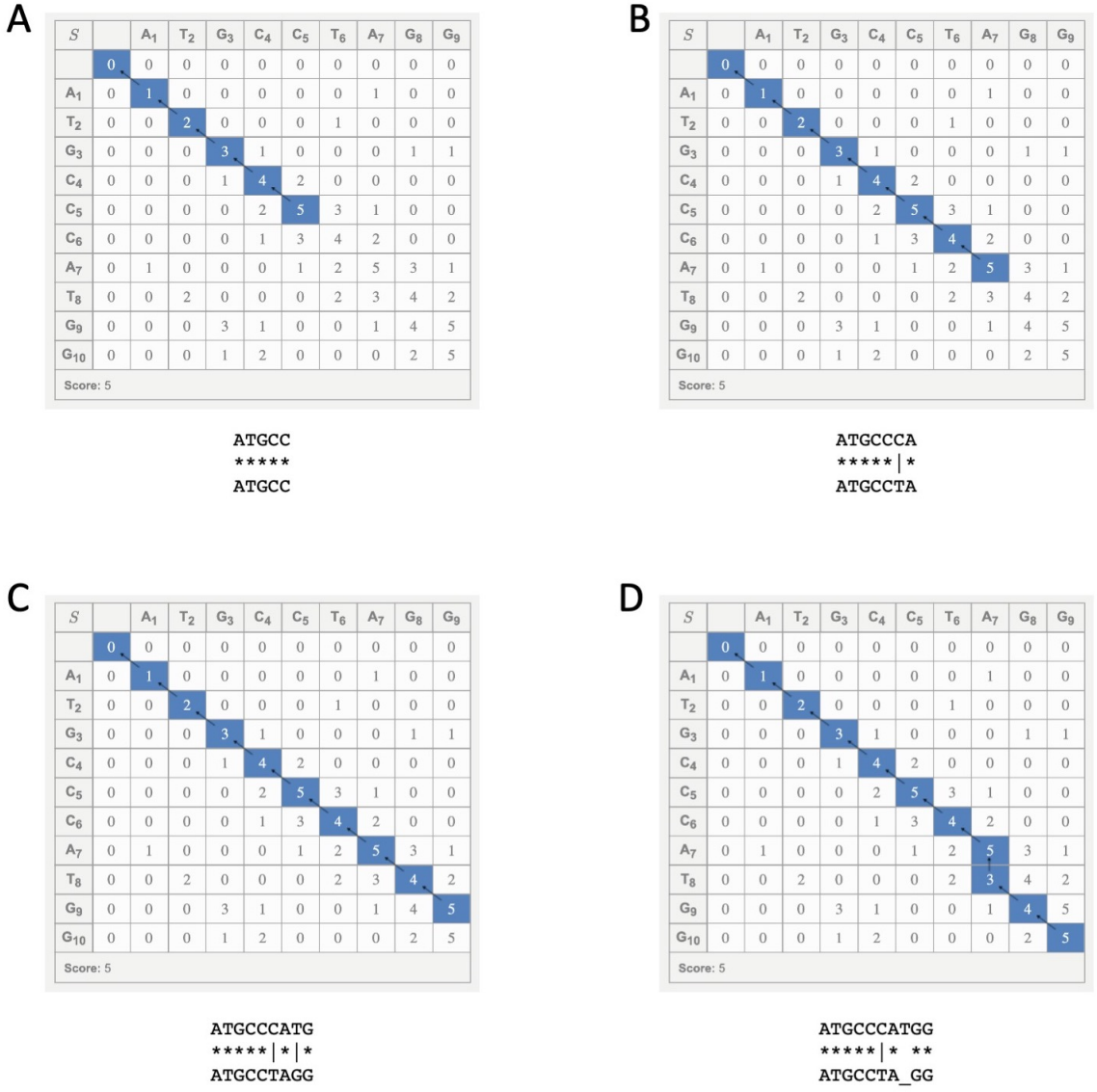

Figure 1.2: Alignment of sequences by Smith-Waterman algorithm. (A-D) Show the 4 optimal local alignments for the two given sequences based on the 2D matrix of alignment scores devised by Smith and Waterman. Final alignments of equal score are shown below for each of the 4 optimal alignments with equal weight. Matrix generation was created by University of Freiberg, Frieberg RNA tools (http://rna.informatik.uni-freiburg.de/Teaching/index.jsp?toolName=Needleman-Wunsch)

1.4 Alignment Metrics

A given alignment produces several metrics that provide insight into different aspects of the quality of the alignment, as well as the probability that the alignment could not be attributed to chance [4, 5]. Given the ever-growing size of sequence data, spurious alignments particular by short sequences can ill-advise researchers into believing instances of motif and domain homology. Critical metrics to assess the quality of alignment include the percentage sequence identity, alignment overlap, raw score, the bit-score, and E-value. Percentage sequence identity is indicative of the number of instances in an alignment where the positions match. The overlap is indicative of the length of the aligned region. In some instances of domain conservation, two proteins can share sequence similarity for a portion of their sequence, but not their whole sequence. This information is critical when determining domain level homology or complete sequence homology. The raw score is produced through the use of the aforementioned Smith-Waterman algorithm. The bit-score is a conversion of the raw score that removes the dependence/bias of the query sequence length, does not factor in database size (in contrast to E-value), and normalizes the score based on the conversion (BLOSUM, etc.) matrix. The E-value is an additional conversion of the raw score that is designed to determine the number of matches "expected" to be observed by chance for a database of a particular size.

1.5 BLAST/BLAT

Creating a two dimensional matrix for all possible alignments between a set of genes/proteins becomes a tedious and inefficient task when querying thousands of sequences against a database of thousand of sequences. The Basic Local Alignment Search Tool (BLAST)

algorithm was designed to reduce the time required to traverse large databases of sequences while maintaining sensitivity and specificity of alignments [4, 5]. BLAST attempts to reduce the number of full alignments by first conducting a heuristic search. It assumes that if two sequences are similar, they share common sub-sequences. Thus when a given query is broken into overlapping sub-strings, the subjects that contain these sub-strings are ideal candidates for subsequent alignment. Not all query sub-strings are utilized for this initial search as different sub-strings carry different weights. These weights are defined from a substitution matrix such as BLOSUM or PAM, which are built from a large number of previously computed sequence alignments [29]. The values in the substitution matrix are indicative of the relative frequency of that particular residue as well as the frequency of that residue being mutated into another residue. Infrequent residues carry higher weights than more frequently found residues. Thus a highly weighted sub-string would consist of a combination of infrequent amino acids [4, 5]. The determination of the highest weighted words is a critical step in the BLAST heuristic as it reduces the number candidate query sub-strings. These highly weighted words are then used to scan databases to identify all potential matches quickly and filter out candidates that do not contain these highly weighted words. A large number of matches between the weighted sub-string queries and a given subject indicate the probability for a larger alignment. These instances of local alignments are then expanded in an attempt to create contiguous sequences. Gaps are incorporated to enhance the alignments. A raw score is then determined for the final alignment by summing the weight matches, mismatches, and gaps of the alignment. This raw score can be converted to both bit-score or E-value. Methods such as BLAST provide a critical filtration setup that removes unnecessary alignments that are considered to be low scoring. However, despite the significant speedup BLAST does not scale well in regards to time when comparing millions of sequences.

1.6 DIAMOND

DIAMOND is one of several methods designed to minimize the time required to identify optimal alignments between millions of queries and subject sequences while retaining sensitivity and specificity [13]. DIAMOND, like BLAST, utilizes the seed and extend strategy; however, DIAMOND makes several additional optimizations to expedite this process significantly. First, DIAMOND utilizes double indexing to store a list of all seed sequences and their location in both the query and sequences in the database. These two lists are alphabetically sorted and then traversed together to determine all matching seeds and locations. Second, DIAMOND utilizes longer seed sequences, but only requires that a specific portion of the seed aligns. This partial matching has been shown to increase speed of matching without losing sensitivity [13]. Finally, DIAMOND utilizes a reduced alphabet space of 11 amino acids. This reduced alphabet is thought to take into account the functional redundancy of amino acids. Seed sequences that are conserved between the query and subject(s) are filtered through into the extension phase of alignment, via Smith-Waterman alignment. This optimization by DIAMOND has allowed for computations that have taken 800k CPU hours at a super-computing center to be completed on a single system in 2.3 hours. DIAMOND is thought to be approximately 2,000 times faster than BLAST based on the author's bench marking studies. DIAMOND, in its most sensitive mode, is capable of capturing 92% of all BLAST-based alignments in test sets. It is not known if the remaining 8% consist of high scoring alignments or low scoring alignments. Regardless, DIAMOND provides a rapid method to produce specific and sensitives alignments for a large number of sequences in a relatively short time frame. The ability to generate such a large data set then allows for the creation of networks based on sequence similarity.

1.7 Sequence Similarity Networks

The wealth of alignment data generated from algorithms such as BLAST and DIAMOND can be reduced to showcase the connections between genes/proteins [8]. These connections, or edges, represent instances of sequence similarity and can carry weights often defined by the bitscore or negative log of the E-value for the alignments between the two sequences [8, 56]. Arbitrarily manipulating the threshold for the edge weights can create stringent or lenient sequence similarity networks. The use of well established methods from graph theory can be applicable to these networks, thereby providing a method to appreciate aspects of these networks. Of particular interest by the life science community is the identification of clusters of sequence similar genes/proteins. The Markov clustering algorithm is amongst the most popular methods to identity clusters from sequence similarity networks due to its relative speed, ease of implementation, and incorporation into life science friendly software packages such as orthoMCL [23, 40].

1.8 Clustering proteins based on sequence similarity

The generation of sequence similar clusters from a list of sequences requires algorithmic solutions that minimize the number of alignments and comparisons. Two methods, CD-HIT [25, 41] and more recently linclust [63], have functioned as the gold-standards for cluster generation. Both methods have been at some point the default process utilized by UniProt (the Universal Protein knowledgebase) to generate the sequence similarity clusters for their global dataset at 90, 80, 70, and 50% sequence identity [6]. Criteria for clusters are based on the percentage of overall sequence identity as well as sequence overlap. Sequence overlap is a critical consideration as small regions of a protein sequence can share high sequence

similarity with each other, even if the overall alignment between the two proteins is poor.

The CD-HIT algorithms starts by sorting all proteins in the input in decreasing order of length. The list is then traversed with the longest sequence functioning as the representative sequence. Subsequent proteins are compared to the representative sequence and are included in the cluster until specific criteria such as percentage sequence identity and overlap are no longer satisfied. At that point, the current sequence becomes the new representative sequence for a new cluster, and the previous cluster is considered complete. According to the authors, this process reduced the number of comparisons from n^2 to nk, where k is the number of representative clusters.

Further, CD-HIT optimization occurs for the alignment as well. CD-HIT assumes that a given alignment at a certain threshold will result in a certain number of k-mers (specifically di- through pentapeptides). For each sequence, the program identifies the number and types of k-mers. The number of and type of k-mer is compared to the respective reference sequence. If the number of shared k-mers between two sequences is above a threshold then the sequences are considered to be within a cluster. CD-HIT forgoes the use of actually conducting pair-wise alignments and relies primarily on conserved k-mers. Linclust utilizes a similar process centered around k-mers, but through various optimizations and is said to function in linear time [62].

A recent unpublished method, Neighborly-GFP (gene function prediction), developed by Dr. Shiv Kale, utilizes clusters derived from a sequence similarity network of 108 million protein sequences. This network was constructed through bitscores generated from a all-versus-all comparison using DIAMOND. Nodes in this undirected network represent protein sequences, and every edge connects two proteins. The weight of an edge (a, b) is the lower of the bit-scores observed in the all-versus-all DIAMOND comparison. The threshold for edges in the high sequence similarity network require 90% sequence identity, 90% overlap between

the alignments, and a reciprocal match that meets the aforementioned criteria. Clusters are initially identified or generated from cliques, pairs, and cleavage of imperfect components via the Markov Clustering Algorithm (MCL) [23]. These clusters are then turned into "super-nodes", and all internal edges and nodes are collapsed. These super-nodes are then integrated into lower sequence similarity networks based upon percentage sequence identity at 80%, 70%, and 50%. Edges amongst super-nodes and nodes are collapsed into a single edge, where the weight is the lowest observed edge weight. Cliques, pairs, singletons, and clusters via MCL are identified once again for these networks. A representative sequence is chosen for each cluster based on being the longest sequence, similar to CD-HIT and linclust described below.

The generation of sequence similar clusters and to a broader extent networks provides an avenue to propagate biologically relevant information to other members of the clusters or nearby nodes in the network. This distribution of information facilities hypothesis for genes and protein that do not have an experimentally derived annotation, but share strong similarity to genes and proteins that do. It is important to note that most experimentally derived annotations for genes were diverse, varying in detail, and not standardized. The lack of a method to compare and contrast annotations between genes and proteins created an additional hurdle for gene function prediction.

1.9 Gene Ontology

The central aim of the Gene Ontology consortium is to create and maintain a "structured, precisely defined, common, controlled vocabulary for describing the roles of genes and gene products in any organism" [7, 19]. This vocabulary is manually curated by experts across the biological sciences. The Gene Ontology has had significant growth over the past 20

years and is contributed to by a large number of institutes and organizations [18]. The Gene Ontology can be broken into three sub-domains: biological processes, molecular function, and cellular location. A given gene product can be associated with none, one, or more than one term. These associations require evidence codes that range from experimental evidence, phylogenetic-inference, computational analysis, author/curator statements, and finally electronic annotations, which are not reviewed.

The Gene Ontology is a directed acyclical graph, where parent-child relationships provide the bulk of the hierarchical structure. In this graph, different parents may be connected to the same child using different relational vocabulary. The primary vocabulary includes "is a," "part of," "has part," and "regulates." Descent into the GO hierarchy generally results in more specific terms. It is vital to realize sections of the hierarchy in terms of depth and members within a branch at a given depth vary greatly. These graphs should be viewed as highly imbalanced. A significant portion of current genome annotation efforts, gene function prediction, and transcriptomic analysis rely heavily on the Gene Ontology. Though the Gene Ontology provides a controlled and relational vocabulary for describing the function and associated processes of genes, instances occur where a set of genes have multiple Gene Ontology annotations. In such cases a method to identify the most prevalent and/or meaningful terms is needed.

1.10 Functional Enrichment Analysis

There is a need to determine associated biological processes and pathways for a set of genes, such as those identified through clustering by sequence similarity. Proteins sharing a high degree of sequence similarity may have identical or similar molecular functions. There has been significant progress in creating such methods for transcriptomic studies, where researchers

wish to identify if a set of genes with similar expression patterns are associated with a given biological process and/or molecular function. The rationale for these studies is that the activation or deactivation of given biological processes will result in changes in gene expression for many genes associated with that pathway. While these two forms of enrichment analysis are used for different purposes, the core methods of enrichment analysis developed for transcriptomics are applicable to annotate clusters.

There are two broad methods for functional enrichment analysis that have been packaged for use by various software (reviewed by Geistlinger et al. [26]). Gene Ontology Enrichment Analysis (GOEA), also known as singular enrichment analysis (SEA), tests if a given pathway, process, or term contains a significant number of genes with significant expression change or shared property [26, 66]. Tools based upon this method include BinGO [45], DAVID [57], and BayGO [67]. Modular enrichment analysis (MEA) is a modification of the SEA strategy through the incorporation of the term-to-term relationships of the Gene Ontology [30, 66]. Tools, such as TopGO [3], GenGO [44], MGSA [11], Ontologizer [27], and GeneCodis [50], utilizing this strategy claim to improve discovery, sensitivity, and specificity in comparison to SEA. Gene set enrichment analysis (GSEA) tests if a given pathway process or term accumulates at the bottom or top of a ranked list of genes, e.g., sorted by a measure of differential gene expression [26, 64]. The use of GOEA based methods is directly applicable to annotate clusters such as those identified by CD-HIT, linclust, and Neighborly. Recently, the development of GOATools provides a python library for functional enrichment using Fisher's Exact test with over 10 methods for multiple test corrections [38].

For over-representation analysis, statistical significance has been determined through the application of the binomial, χ^2, and Fisher's exact/hypergeometric tests (reviewed in [55]). The Fisher's exact test has become the more commonly used method by the life science community. This test is applied to what is assumed to be a hypergeometric distribution for

null hypothesis testing. For the Fisher's exact test, a global set of genes, usually from an annotated genome, are divided into positive and negative examples for each GO term. The probability of the ratio of having a set of positive values based on the known distribution is then computed. A detailed example of Fisher's exact test is shown in Chapter 3. In many cases, researchers look for enrichment across all possible Gene Ontology terms, and the issue of multiple hypothesis testing arises. This is often corrected through the incorporation of the Bonferroni correction or the False Discovery Rate correction.

1.11 Gene Function Prediction

Methods like Gene Ontology Enrichment Analysis aid in deriving knowledge from a group of gene products whose functions are known. However, there are many protein who are yet to be associated with functional terms. Biological experiments to understand the functions of proteins can be slow, expensive and in some cases even difficult. The growing gap between the accumulation of genomic data and extracting functional information from it demands accurate computational methods to enhance our understanding of genomic data.

The field of gene function prediction is the culmination of several advances in sequence alignment, network propagation, enrichment analysis, aggregation of diverse data sources, and machine learning. For specific tasks, gene function prediction can be seen as a binary classification problem. The objective of binary classification methods is to predict if a protein is involved in a particular function or not. Zheng et al. [77] demonstrated the use of the nearest neighbor classifier to predict the association of bacterial virulence factors with proteins using interaction networks. Zeng et al. [72] predicted pathogenic-related secreted proteins using tools like Blast2GO [17] with PHI (protein-host interaction) database [69]. A range of prediction methods is reviewed by Sonah et al. [59] for the prediction of effector

proteins. However, most of the protein function prediction methods fall under the multi-class, multi-label classification problem. A given protein can be associated with multiple functions in the Gene Ontology. The multi-class classification umbrella includes methods that classify a given protein into multiple classes, while multi-label denotes the task of assigning a protein to more than one label [65].

The earliest predictors of gene function relied on sequence similarity using BLAST, PSI-BLAST, FASTA [51]. GO annotations from top matches are transferred to the query proteins. Older tools such as GeneWiz and relatively newer methods such as BLAST2GO provide a means for this basic task [17, 28]. One critical shortcoming of BLAST2GO like methods is the occurrence of diverse and contrasting GO terms amongst top matches. Likewise, the use of motif or domain conservation to propagate information to queries has been utilized extensively across the life and computational sciences. Notable examples include translocation motifs from plant pathogenic eukaryotes and lipid binding domains such as the FYVE, pH, and pX domains [20, 34, 60].

Through advancements in both structural homology modeling and computational modeling, researchers can create putative structural models for protein sequences. These models facilitate molecular docking studies with other proteins as well as small molecules. Zheng et al. [78] created a service called I-TASSER that makes structure-based function annotation. Given a protein sequence, I-TASSER predicts the 3-D structure of the protein and then searches the database of known protein structures with their functional annotations to find the closest match. The annotations are then transfered from the closest match to the query protein. Recently, a method known as PANDA (Propagation of Affinity and Domain Architecture) predicts protein function through the use of domain identification and homology modeling against several databases [68]. PANDA then calculates the probability of having a specific GO term using Bayesian statistics.

Another set of methods rely on the occurrence of a specific string of amino acids (k-mers) that are associated with labels of the GO hierarchy. Machine learning models often require numerical representation of data. The raw protein sequence is converted into a feature matrix using the aforementioned k-mer frequencies. Various machine learning models like support vector machines [14], Bayesian classifiers [43], and neural networks [15] have been proposed with varying success. The large number of labels associated with the Gene Ontology as well as the inter-relation of terms has made it challenging to produce a single multi-classifier.

1.12 Critical Assessment of protein Function Annotation (CAFA)

Large number of methods to predict the function of proteins and a need to understand their performance and biological significance requires detailed evaluation of the methods. The Critical Assessment of protein Function Annotation algorithms (CAFA) is an ongoing competition that provides a platform for large-scale evaluation of protein function prediction methods on novel proteins whose functional information is not available to the research community. In short, CAFA provides a set of protein sequences that currently do not have published experimental annotations. Participants use their algorithms to associate these protein sequences with Gene Ontology terms. After the submission deadline, the organizers release the Gene Ontology annotations for the sequences that have been experimentally verified. These proteins along with their annotations serve as the benchmark set, against which the methods are evaluated.

The first CAFA experiment was organized in 2010–2011, and a target set of 866 proteins from the SwissProt dataset ranging across two domains (biological process and molecular function)

was released as the benchmark set [52]. A total of 54 algorithms submitted their predictions to be evaluated. For each target sequence, the algorithms predicted the functional annotation with a confidence score ranging between 0 and 1. The evaluations were done primarily using the F_{max} score.

Two algorithms, Naive and BLAST, served as the baseline tools. For BLAST, all GO terms of an experimentally annotated sequence (template) from SwissProt were transferred to the target sequence such that the scores equaled pairwise sequence identity between the template and the target. For the Naive method, each GO term for each target was scored with the relative frequency of this term in SwissProt over all annotated proteins. The best performing algorithm Jones-UCL had a F_{max} of around 0.6 for molecular functions, and a F_{max} score slightly below 0.4 for biological processes. Most methods used sequence alignment data based on an underlying hypothesis that sequence similarity is correlated with functional similarity.

Three years after the first experiment, CAFA 2 was organized [33]. They evaluated 126 prediction methods of a dataset of 3,681 proteins ranging across three domains (biological process, cellular components, and molecular function). Similar to CAFA, each method associated a GO term to a sequence along with a confidence score. Along with F_{max}, the evaluation was also done using the remaining uncertainty, misinformation, and the corresponding S_{min} (minimum semantic distance) score. Again, two baseline methods, Naive and BLAST, were used. The best performing method with respect to F_{max} for molecular functions was Ms-kNN, which achieved score of about 0.65 for molecular functions and 0.4 for biological functions. Based on S_{min} metric, Orengo-FunFams had a score of 6.2 for molecular functions and MS-kNN had a score of 17.6 for biological processes. On comparison, with CAFA 1 algorithms on the new test set, the top 5 CAFA 2 methods outperformed the top 5 CAFA 1 methods.

More recently, in 2016-2017, CAFA 3 experiment was conducted with a target set of over

66,000 protein sequences [79]. Similar to CAFA 1 and CAFA 2, each participating algorithm submitted a list of associated GO terms with a confidences core. Each method was evaluated using F_{max} and S_{min} scores. And Naive and BLAST methods served as the benchmark. The new methods had comparable performance to the previous CAFA 2 methods. More specifically, on comparing top 5 CAFA 3 methods with top 5 CAFA 2 methods, only one method, GOLabeler [71], outperformed all the CAFA 2 methods. Out of the top 12 ranked methods in CAFA 3 and CAFA 2, 7 belonged to CAFA 3 and 5 belonged to CAFA 2. Similarly, only the top 3 methods in CAFA 3 outperformed the top 5 methods in CAFA 2 and that too by a small margin.

There was very little performance difference between CAFA 2 and CAFA 3 methods. On computing the similarity of each pair of methods using the Euclidean distance of prediction scores, it was observed that CAFA 2 and CAFA 3 methods were very similar.

Chapter 2

Neighborly for Gene Function Prediction

2.1 Introduction

Gene function prediction is a rapidly growing field filled with diverse methodologies that rely on various data sources in addition to sequence information. Given the steep growth in the number of sequences and putative genes coupled with the considerable time and resources required for experimental validation, the need for accurate and efficient computational predictions is paramount to aiding research in understanding diseases, evolution, and drug development.

The majority of gene function prediction methods rely on some form of guilt by association. Query sequences are associated with known annotated sequences through total or partial sequence similarity, the consensus amongst k-mers, and more recently, comparison in protein structure. While some methods incorporate diverse data sources, such as protein-protein interaction data, phenotypic screens, and text mining to aid gene function prediction in addition to sequence similarity. The implementation of these methods varies greatly, with each possessing a respective set of strengths and weaknesses (reviewed in Chapter 1).

One recent unpublished method, Neighborly Gene Function Prediction (Neighborly-GFP),

leverages the massive amount of available protein sequences to create a network of similar gene products. For an edge to exist between two proteins (A, B), there must be an edge from protein A to B and an edge from B to A. The weight of an undirected edge is calculated as the minimum of two bit-scores calculated by aligning protein A with B and vice versa. Formally, the weight of an edge between two proteins A and B is calculated as

$$Edge\ Weight = \min\left(bit\ score(A,\ B),\ bit\ score(B,\ A)\right) \tag{2.1}$$

The network is built on 108 million non-redundant protein sequences and has approximately 6.8 trillion edges.

Filtrations of this network are extracted to create simpler networks through thresholds for:

- **Percent Identity**: Measures the proportion of amino acid characters that match exactly between two sequences.
- **Percentage Sequence Overlap**: Measures the overlap between two aligned sequences (refer Figure 2.1).

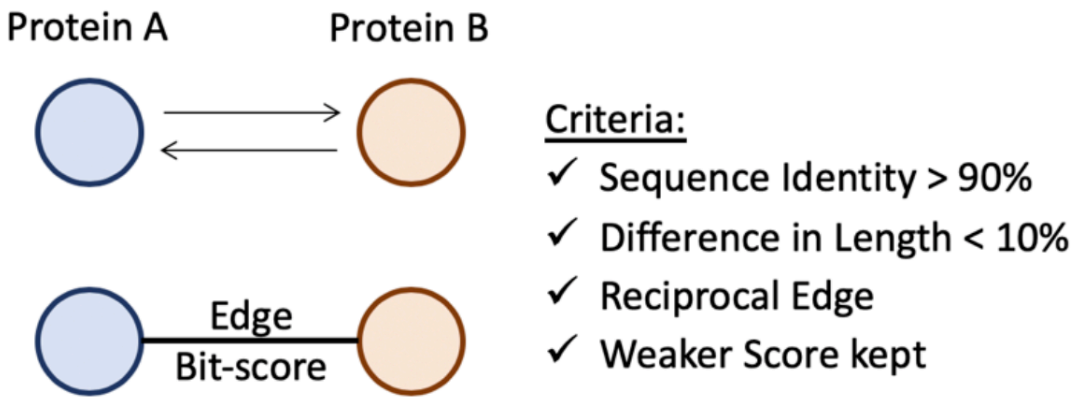

Figure 2.1: Edge Criteria for Neighborly Sequence Similarity Network

Neighborly Network Types

Varying the stringency of edge criteria in the initial network, we create five variations of the Neighborly networks.

Neighborly 90: The Neighborly 90 network connects proteins of high sequence similarity. For an edge to exist in this network, two proteins must have at least 90 % sequence identity and at least 90 % overlap between sequences. The resulting network consisted of $\sim$ 5.98 billion edges.

Neighborly 80: The Neighborly 80 network is created reducing the criteria of sequence identity to >80%. The criteria for percent sequence overlap is still > 90%.

Neighborly 70: Further reducing the sequence identity to > 70 % yeilds the Neighborly 70 network.

Neighborly 50L10: The Neighborly 50L10 network is created using the sequence identity threshold of >50%, and the difference in overlap between the two sequence can only be less than 10% (L10).

Neighborly 50L20: Increasing the threshold for difference in sequence overlap to less than 20% (L20) creates the Neighborly 50L20 network. Here, the criteria for percent sequence identity is still at 50%.

Not surprisingly, the number of edges significantly increases as the threshold for sequence similarity is reduced (Table 2.1). Global analysis of Neighborly 90 network suggests a highly disjointed network composed a large number of connected components, which include cliques

Table 2.1: Characteristics of Neighborly Networks

Network	Components	Connected Nodes	Edges	Perfect Components	Imperfect Components	Clusters from Imperfect Components	Pairs	Cluster Islands	Total Clusters (Excluding Singletons)	Singletons
S90L90	10,905,274	62,291,889	5,983,239,454	3,773,939	1,483,093	2,740,091	5,648,243	NA	12,162,273	45,892,114
S80L90	TLD	–	13,384,260,125	–	–	–	–	–	–	–
S80L90Red	4,425,332	72,291,663	316,029,360	637,057	201,557	1,022,855	2,665,671	5,267,924	9,593,507	35,892,340
S70L90	TLD	–	27,149,750,920	–	–	–	–	–	–	–
S70L90Red	4,753,698	79,072,573	388,993,692	955,599	1,272,809	3,132,864	2,525,290	3,307,265	9,921,018	29,111,430
S50L80	TLD	–	83,562,919,705	–	–	–	–	-	–	–
S50L80Red	3,359,350	86,130,192	3,917,496,316	718,993	381,376	3,513,104	1,629,382	1,225,164	7,086,643	22,053,811
S50L90	TLD	–	81,922,207,055	–	–	–	–	–	–	–
S50L90Red	3,213,745	82,949,170	3,054,239,073	649,925	367,700	4,120,159	1,565,075	1,640,339	7,975,498	25,234,833
CD-HIT 90L90	Clusters* (57,411,525)	64,288,236	NA	–	–	–	–	NA	12,655,109	43,895,767
CD-HIT80L10*	Clusters* (46569182)	67,177,695	NA	–	–	–	–	NA	5,562,874	41,006,308
CD-HIT70L10	Clusters* (38897788)	75,813,398	NA	–	–	–	–	NA	6,527,183	32,370,605

with three or more nodes, single edges, and disconnected nodes. Connected components that were not cliques were then further broken down into clusters through the use of the Markov clustering algorithm [22, 35, 70].

One issue that arose with the Neighborly 80, 70, and 50 networks is the massive amount of inter-connectivity. A large number of edges made it computationally infeasible to load the graphs in memory and identify the individual entities. Dr. Kale implemented the concept of a network reduction through super-nodes. Each super-node represented a cluster, clique, or pair identified in the Neighborly 90 network. All nodes and edges within the super-node were collapsed into this single representation. All edges connecting to this super-node were collapsed into a single representation. This method of reduction resulted in >95% reduction in edges as well as a 20–40% reduction in nodes (Table 2.1). These simplified networks were then used to further determine cliques, components, and disconnected nodes. Each cluster in the graph is represented by the longest sequence in that cluster.

In this chapter, we discuss the various methods we compare the Neighborly-GFP against, the datasets used to test gene function prediction, and the metrics used to evaluate our predictions. Specifically, we compare Neighborly-GFP networks with CD-HIT clusters, DeepGoPlus and a DIAMOND benchmark database. We use the CAFA 3 dataset, CAFA 3 test

set along with two temporal-holdout datasets to evaluate our predictions. These evaluations are done using the F_{max} score, information-theoretic measures and a new weighted accuracy method.

2.2 Methods

2.2.1 Function prediction methods

Function prediction by function transfer

The most basic information we know about proteins are their sequences, which are made of 20 amino acids. Variations in these sequences lead to generation of different types of proteins that perform different functions [49]. This sequence information is well organized in standard databases like the SwissProt and TrEMBL databases [12]. Databases like the Gene Ontology Annotations (GOA) provide a repository for known functional information of proteins from various sources [10]. Using sequence alignment and similarity methods like DIAMOND and BLAST [5, 13] we can transfer functional information between sequence similar proteins. The underlying principle being if two sequences are similar, they must perform similar functions [52].

However, alignment and consequent function transfer can be computationally expensive for larger databases. To find similar sequences for a given query sequence, the methods have to align and search through a large database. Contrastingly, searches in a smaller database are relatively quick but we are limited by the number of sequences in the database and the consequent functional knowledge we can derive from a smaller database.

Methods like CD-HIT and Neighborly-GFP provide a needed scale-up in performance by

grouping similar proteins into a cluster. Each cluster is then represented using the longest sequence in it. Clustering similar proteins has two main advantages. Firstly, clusters of proteins reduces the search space required to find similar sequences even if the original search space of known sequences was larger. Secondly, the functional information associated with a cluster is the aggregation of terms associated with each of its members. This ensures we can transfer more functional information to a query sequence. Formally, given a query sequence X whose function is not known, we first find the most similar clusters to that sequence using DIAMOND and then transfer the GO terms of all the associated members of the cluster to the query sequences. This set of transferred annotations forms the predicted set of terms.

DeepGoPlus

We wanted to compare our function prediction method with the current best sequence-based function prediction method. DeepGoPlus is a Convolutional Neural Network (CNN) based method, that combines scores from the CNN output with the DIAMOND bit-score [39]. DeepGoPlus starts by encoding a raw protein sequence into a 21×2000 vector, where 2000 represents the maximum length of the sequence and 21 represents the 20 amino acids along with an unknown character X. The first layer is a series of convolution filters of sizes $\{8, 16, 24...128\}$ (multiples of 8). For each filter size, 512 such filters are stacked at each layer. These filters identify specific domains and motifs in a sequence and map them to GO terms. The output of these layers is condensed using a Max-Pool layer. The final hidden layer is a classification layer of size 8192 (512×128). The output layer is the size of the number of GO tems in train set.

The final prediction scores for each GO term is defined as follows:

$$S_{final} = \alpha * S_{DiamondScore} + (1 - \alpha) * S_{DeepCNN} \tag{2.2}$$

where

- α is the weight parameter, decides which score gets more importance.
- $S_{DiamondScore}$ is the bit score from DIAMOND output.
- $S_{DeepCNN}$ is score from the CNN.

We used the code base from DeepGoPlus's Github repository (https://github.com/bio-ontology-research-group/deepgoplus) to train the model on the CAFA 3 dataset with the default value for $\alpha = 0.55$.

2.2.2 Datasets

One of our central issues was identifying an appropriate dataset that (i) contained sequences that were not used to build the Neighborly networks, (ii) was utilized by other research groups to test their specific methods, and (iii) contained an extensive number of examples. We were unable to identify a single dataset that met these criteria, but we did identify four datasets that satisfied at least one of these criteria (Table 2.2). We initially utilized the CAFA 3 testing set as it satisfied the criterion (ii) and allowed us to compare Neighborly-GFP with other methods. The CAFA 3 test set was produced before Neighborly sequences were downloaded in Feb 2018 and therefore, may contain sequences that were used to build our network. We also utilized the CAFA 3 training set (dataset) for predictions as this dataset provided over 35,000 sequences with annotated functions. To satisfy criterion (i) we identified

two temporal holdout datasets from TrEMBL and SwissProt databases respectively. This dataset consisted of experimentally annotated sequences that have been added to TrEMBL and SwissProt after we had created the Neighborly networks. Table 2.2 highlights the four datasets we use. Each dataset is broken into two entries in the Table 2.2 to highlight the number of proteins and functional terms present for each Gene Ontology domain.

Table 2.2: Biological Process and Molecular Function Test Sets

Dataset Name	Number of Proteins	Unique Terms
CAFA 3 Train BP	50,813	16,117
CAFA 3 Train MF	35,086	5,966
CAFA 3 Test BP	2,143	1,660
CAFA 3 Test MF	1,092	615
TrEMBL BP	3,498	2,451
TrEMBL MF	2,980	1,135
SwissProt BP	1,501	2,139
SwissProt MF	1,220	841

IEA and non-IEA annotations

We use the available functional information from the Gene Ontology Annotations (GOA) database to transfer terms from sequence similar proteins to a query sequence [10]. Each annotation in the GOA database is described by an evidence code. Evidence code describes how a protein was associated with a given protein [1]. For example, evidence code IEA represents the annotations are inferred from electronic annotations. Electronic annotations are not reviewed and curated experimentally. We divide the evidence codes into two simple categories with IEA and without IEA. With IEA annotations refer to all the annotations include the ones with the evidence code IEA, while without IEA refers to all the annotation evidence codes excluding the IEA.

2.2.3 Evaluation Methods

Threshold-based evaluation

Each protein can be associated with multiple terms from the GO. Predicted terms for a given protein, are associated with a confidence score. This score indicates how sure the model is about a prediction. For example, for some protein X if a predicted term t gets a score of 0.90 the model is 90% sure that the protein X is associated with term t. Confidence scores usually range from 0 to 1, where 1 denotes absolute certainty and 0 denotes zero certainty.

Threshold-based evaluation considers only those terms having a confidence score above a certain threshold value. This threshold value is varied from minimum possible confidence score to the maximum with a certain step size. At each threshold value, only the predicted terms that have a score greater than or equal to the defined threshold value are considered for evaluation. For our threshold-based evaluation we vary the threshold θ from 0 to 1, with a step size of 0.01. This is similar to how CAFA does its evaluation.

DIAMOND and BLAST-based methods do not explicitly output a set of predicted terms. They align and find similar proteins for a given query sequence from a database of known protein sequences. The functional terms of known sequences are then transferred to the query sequence. To associate a confidence score with each term transferred for a query sequence, we take the sequence identity score as the confidence value. Recall that sequence identity score measures the similarity between two sequences. The rationale is that if the two sequences are more similar, then the terms are transferred with a greater confidence. In the final list of predicted terms, if a single term, is present with multiple confidence scores because it was associated with two different sequence in the database, the higher of the two scores is kept.

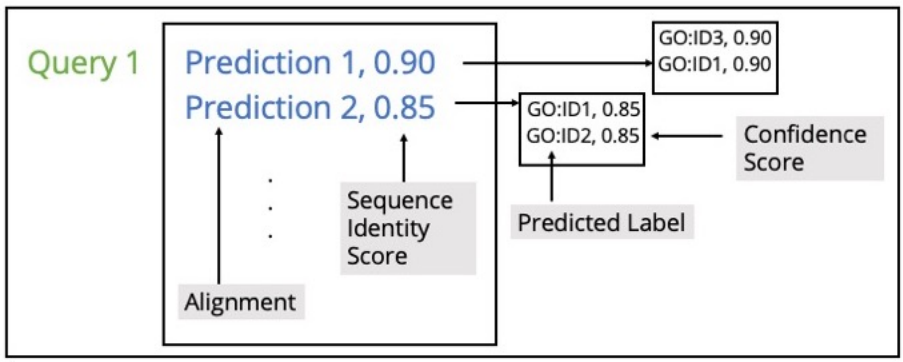

Figure 2.2: Threshold Based Method for Prediction. We have a query sequence *Query 1* (shown in green) and it is similar to two sequences *Prediction 1* and *Prediction 2* that have a sequence identity score of 0.90 and 0.85 with the query sequence respectively (shown in blue). *Prediction 1* is associated with two GO terms (GO:ID1, GO:ID3) and *Prediction 2* has two GO terms (GO:ID1, GO:ID2). Each of these terms is transferred to the query sequence *Query 1* with the confidence score equal to the sequence identity score. GO:ID1 gets a confidence score of 0.9, GO:ID2 gets a confidence score of 0.85 and GO:ID3 gets a score of 0.9. Note that GO:ID1 got a score of 0.90 and 0.85 from *Prediction 1* and *Prediction 2* respectively, but higher of the two scores is kept.

Rank-based evaluation

Given a query sequence, DIAMOND outputs a list of similar proteins that are ordered by the bit-score. The top protein in the list of similar proteins is the one that has the highest bit-score and as we go down the list, the bit-score generally decreases. The bit-score is a metric to quantify the similarity of two sequences, that is independent of the length of the two sequences, and the size of the database. Sequence identity score used for threshold-based evaluation quantify just the similar amino-acids at the same positions. Unlike the sequence identity score, the bit-score includes the statistical characteristics of the alignments as well. Based on the ordering of similar proteins for a query sequence, we can assign a rank to each similar protein. The protein sequence with the highest bit-score gets a rank of 1, the protein with the second highest bit-score gets a rank of 2 and so on.

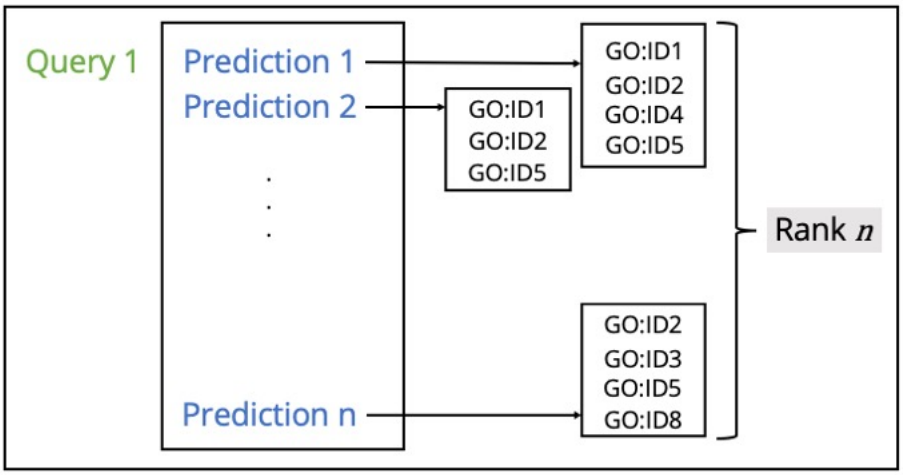

Figure 2.3: Rank Based Method for Prediction. We have a query sequence *Query 1* (shown in green) and it is similar to n proteins, which are ordered by their individual bit-scores. *Prediction 1* has the highest bit-score and gets a rank of 1, *Prediction 2* has the second highest bit-score and gets a rank of 2 and *Prediction n* has the lowest bit-score and gets a rank of n (shown in blue). At rank *1*, we consider only the GO terms associated with *Prediction 1* i.e., (GO:ID1, GO:ID2, GO:ID5, and GO:ID4). At rank 2, we consider all the GO terms associated with top 2 proteins (*Prediction 1* and *Prediction 1*). This will be the set of (GO:ID1, GO:ID2, GO:ID4, GO:ID5). Finally for rank n, all the GO terms associated with top n proteins are considered. This will make up the set of predicted terms at rank n as (GO:ID1, GO:ID2, GO:ID5, GO:ID4, GO:ID3, GO:ID8).

2.2.4 Evaluation Metrics

GO terms associated with a protein along with their ancestors form the subgraph of ground truth. The task of a predictor is to associate a subgraph of GO terms that is closest to the true subgraph for a protein whose functional information is being predicted. The large number of functional terms associated with a protein and the relation between them makes the evaluation task more complex. To assess the quality of our predictions we use the protein-centric evaluation criteria defined in CAFA 3 [79]. Protein-centric evaluation focuses on assessing the association of a GO terms for a specific protein. On the other hand, term-centric evaluation methods, focus on associating a protein with a GO term. Using this

published criteria allows us to compare our method against the methods published in CAFA 3 as well as ensure that our prediction pipeline is working as intended.

The CAFA 3 evaluation criteria are centered around a F_{max} score built from the precision-recall (PR) curve as well as a S_{min} score built from the misinformation and remaining-uncertainty (MI-RU) curve.

Precision, Recall and *F-measure*

Both the true and the predicted subgraphs can be represented using a set of GO terms representing the nodes in the subgrpah.

Precision: Precision of a method gives an idea of what proportion of predictions are correct.

Formally, it can be defined as:

$$precision(\theta) = \frac{1}{m(\theta)} \sum_{i=1}^{m(\theta)} \frac{\sum_f I(f \in Pred_i(\theta) \cap f \in True_i)}{\sum_f I(f \in Pred_i(\theta))} \tag{2.3}$$

where,

- θ represents the rank/threshold, f is a term in the ontology
- For rank-based evaluation
  - $m(\theta)$ represents the number of query proteins that have at least one prediction upto rank θ.
  - $Pred_i(\theta)$ represents all the GO terms associated with top θ alignments.
- For threshold-based evaluation

  - $m(\theta)$ represents the number of query proteins that have at least one prediction with a confidence score greater than equal to θ.
  - $Pred_i(\theta)$ represents set of predicted GO terms that have a confidence score greater than or equal to θ.

- $True_i$ represents the set of true GO terms.
- I represents the Indicator function.

Recall: The recall of a method gives an idea of what proportion of actual positives are captured by the method. More formally, it can be defined as:

$$recall(\theta) = \frac{1}{n} \sum_{i=1}^{n} \frac{\sum_f I(f \in Pred_i(\theta) \cap f \in True_i)}{\sum_f I(f \in True_i)} \tag{2.4}$$

where,

- θ represents the rank/threshold.
- n represents the total number of query proteins in the test set.
- For rank-based evaluation
  - $Pred_i(\theta)$ represents all the GO terms associated with top θ alignments.
- For threshold-based evaluation
  - $Pred_i(\theta)$ represents set of predicted GO terms that have a confidence score greater than or equal to θ.
- $True_i$ represents the set of true GO terms.
- I represents the Indicator function.

F-measure: *F-measure* provides a single metric to compare the performance of different predictors. It is defined as the harmonic mean of precision and recall. *F-measure* is calculated at each rank/threshold and the maximum across all calculated values gives the final F_{max}.

$$F_{max} = \max_{\theta} \left\{ \frac{2 \times precision(\theta) \times recall(\theta)}{precision(\theta) + recall(\theta)} \right\} \tag{2.5}$$

Information-theoretic evaluation

Rationale for information-theoretic evaluation: Measures like precision, recall and *F-measure* are interpretable but treat true and predicted subgraphs as sets of GO terms. They do not capture the relationship between GO terms and specificity of each GO term. The information-theoretic evaluation method is designed to calculate the similarity between the predicted and the true subgraphs while considering the dependency and information content of nodes in these two subgraphs.

This approach starts by modelling the GO as a Bayesian network [16]. Using this assumption, the probability of any node in the GO is dependent only on its immediate parents and independent of its ancestors. Consider we have a subgraph consisting of three nodes a, b and c. Relationship between them is as — a depends on b and b depends on c. Based on Bayesian rule, a is conditionally independent of c and dependent on b alone. This probability of a can be formulated as:

$$P(a) = \frac{P(a|b)P(b)}{P(b|a)} \tag{2.6}$$

Since, b is an ancestor of a, $P(b|a) = 1$ and the equation 2.6 becomes:

$$P(a) = P(a|b)P(b) \tag{2.7}$$

The probability of a subgraph can be calculated by taking the product of conditional prob-

abilities of each node in the subgraph. For some protein and its associated subgraph of GO terms (G_{sub}), the probability of the subgraph can be expressed as:

$$P(G_{sub}) = \prod_{v \in G_{sub}} P(v \mid Parent(v)) \tag{2.8}$$

where

- v denotes a node in the subgraph, and
- $Parent(v)$ represnts the parents nodes.

Consider the yellow highlighted subgraph in Figure 2.4 C. The probability of that subgraph can be factored into the individual conditional probabilities of each node in the subgraph as follows:

$$P(G_{sub}) = P(f|b)P(g|b)P(b|a)P(a) \tag{2.9}$$

These probabilties are calculated from some background database of annotations. For our test sets, these values will be calculated from the ground-truth terms. The information content of the subgraph (G_{sub}) is then calculated by taking the negative logarithm of the probability of the subgraph as follows:

$$ic(G_{sub}) = \log \frac{1}{P(G_{sub})} = -\log P(G_{sub}) \tag{2.10}$$

Figure 2.4 A highlights an oversimplified ontology with seven terms. Figure 2.4 B highlights the background information database to calculate the probabilities. There are six proteins in the database shown along six rows and columns represent the terms from the ontology. Each entry represents the association of a protein with a term. For example, the protein in row 1 is associated with terms a, b, f, and g.

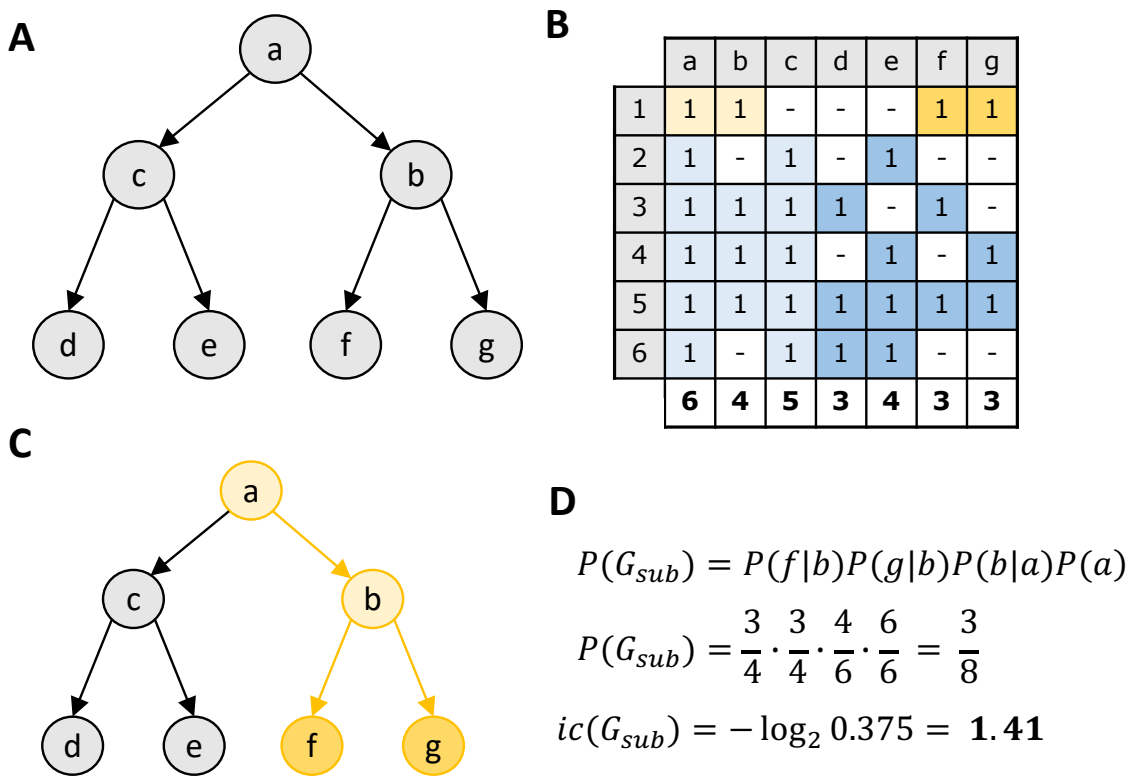

Figure 2.4: Example of information content calculation

The annotations for a protein 1, are highlighted in Figure 2.4 C. Probability of the subgraph in Figure 2.4 C is factored into individual conditional probabilities of each node in the subgraph as shown in Figure 2.4 D. The final values of the probability are calculated from the database in Figure 2.4 B. For example, $P(f|b) = 3/4$. The final information content of the subgraph is calculated by taking the negative binary log of the probability of the subgraph show in Figure 2.4 D.

This information content associated with the subgraphs and proteins is used to calculate two information-theoretic analogs of precision and recall: i.e misinformation and remaining uncertainty as proposed by Clark and Radivojac [16].

Misinformation: Consider we have two subgraphs, G_{True} and G_{Pred} that are formed by taking the terms in predicted GO terms and true GO terms, respectively. Misinformation is then defined as the total information content associated with incorrectly predicted nodes in the prediction subgraph G_{Pred}. More specifically, misinformation can be denoted as:

$$misinformation(\theta) = mi(\theta) = \frac{1}{n}\sum_{i=1}^{n} \sum_{v \in G_{Pred(\theta,i)} - G_{True(i)}} ic(v) \tag{2.11}$$

where

- θ represents the rank/threshold, v is a term in GO, n is the total number of query proteins and ic is the information content.
- $G_{True(i)}$ is subgraph representing true GO terms.
- $G_{Pred(\theta,i)}$ is the subgraph representing predicted GO terms at rank θ or having confidence score greater than equal to θ.

Remaining Uncertainty: Given two subgraphs, G_{True} and G_{Pred} formed using terms in the predicted set and true set. Remaining uncertainty can be defined as the total information content that is missed by the subgraph of predicted terms (G_{Pred}). More specifically, remaining uncertainty can be denoted as:

$$remaining\ uncertainty(\theta) = ru(\theta) = \frac{1}{n}\sum_{i=1}^{n} \sum_{v \in G_{True(i)} - G_{Pred(\theta,i)}} ic(v) \tag{2.12}$$

where the terms of the equation are as described previously.

Both the metrics can be visualized in Figure 2.5. Figure 2.5A shows the predicted subgraph G_{Pred}, and Figure 2.5B shows the true subgraph G_{True}. Both subgraphs G_{Pred} and G_{True} are

formed by using terms $p1$, $p2$, and $t1$, $t2$ which are the terms in the true and predicted set along with their ancestors (shown in light blue and light green respectively). In Figure 2.5C nodes highlighted in red contribute to the misinformation of the prediction, nodes highlighted purple add to the remaining uncertainty that the predicted subgraph failed to capture. Nodes colored in yellow represent the overlap between predicted and true subgraphs.

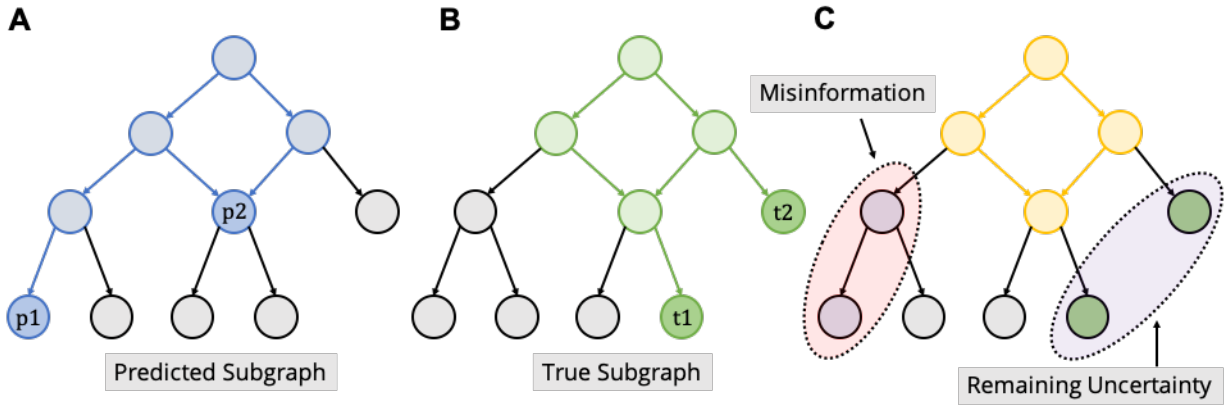

Figure 2.5: Visual Representation of Misinformation and Remaining Uncertainty

Semantic Distance: To provide a single value to evaluate the performance of the prediction task, we use the semantic distance. Semantic distance is defined as the distance from the origin to the *RU-MI* curve. *RU-MI* curve is obtained by plotting *ru* and *mi* values (*ru(θ), mi(θ)*) for each value of θ.

More specifically, it can be denoted as:

$$semantic\ distance(G_{True}, G_{Pred}) = s(G_{True}, G_{Pred}) = (mi^k + ru^k)^{\frac{1}{k}} \tag{2.13}$$

where k is greater than equal to one, we use $k = 2$, which denotes the Euclidean distance between the origin and the *RU-MI* curve. We take the minimum of semantic distance values achieved at each threshold as the final performance measure with $k = 2$ as shown:

$$s_{min}(G_{True}, G_{Pred}) = \min_{\theta} \left(mi^2(\theta) + ru^2(\theta)\right)^{\frac{1}{2}} \tag{2.14}$$

θ represents the threshold/rank. $mi(\theta)$ represents the misinformation of predicted terms at rank/threshold θ and $ru(\theta)$ represents the remaining uncertainty of predicted terms at rank/threshold θ.

2.2.5 Weighted Accuracy

We wanted to get an appreciation of how far/close our predictions are from the true GO terms. We designed a new metric weighted accuracy ($W_{accuracy}$) to measure this, while still being interpretable and considering the GO hierarchy. Weighted accuracy ($W_{accuracy}$) gives the relative distance between the GO terms in the GO and penalizes every less specific prediction than (ancestor of) the true GO term. Every perfect match (where the predicted term is an exact match with the true term) and a match where the predicted term is more specific than(descendant of) of the true GO term is given a $W_{accuracy}$ of 1. The *root* node in a particular domain is given a $W_{accuracy}$ of 0 because it provides us with no information about the protein under consideration. If a predicted GO Term is on the path from the true GO term to the root of the ontology (ancestor of the true GO term), the $W_{accuracy}$ is reduced based on the number of edges the predicted node is away from the true GO node. More formally, let v_p be the predicted node, and v_t be the true node.

If v_p and v_t are a perfect match or if v_p is the descendant of v_t

$$W_{accuracy} = 1 \tag{2.15}$$

If, v_p is an ancestor of v_t,

$$W_{accuracy} = 1 - \frac{n_e(v_p, v_t)}{n_e(v_t, root)} \tag{2.16}$$

where $n_e(v_p, v_t)$ denotes the number of edges between predicted node v_p and true node v_t and $n_e(v_t, root)$ denotes the number of edges between the true node v_t and root node $root$.

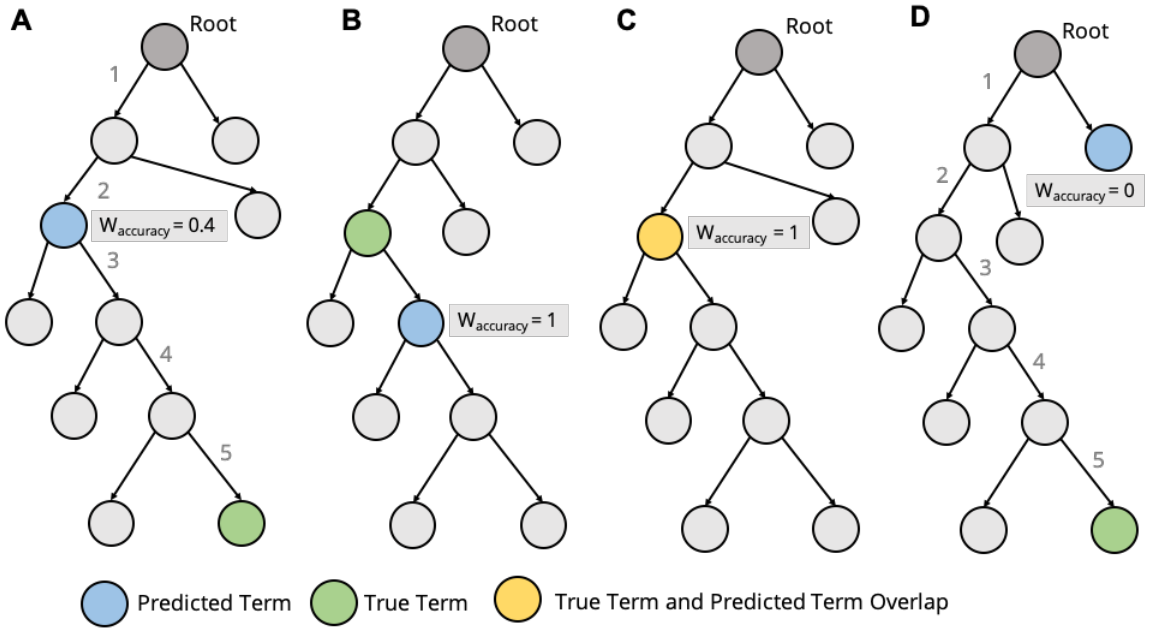

Figure 2.6: Example of output instances for Weighted Accuracy

For example, consider Figure 2.6A, where the predicted node (blue) is an ancestor of the true node (green). The true node v_t is five edges away from the root node ($root$) and the three edges away from the predicted node ((v_p)). The $W_{accuracy}$ for this pair will be $1-(3/5) = 0.4$. Figure 2.6B, the predicted node (blue) is a descendant of the true node (green) and the pair gets a $W_{accuracy}$ of 1. Similarly, in figure 2.6C, predicted node and true node are a perfect match (yellow) and the pair gets a $W_{accuracy}$ of 1. Figure 2.6D, the predicted node is in a different branch and get a score of 0. At each rank/threshold, the $W_{accuracy}$ is calculated as the average of maximum $W_{accuracy}$ for each true GO term.

2.3 Results

2.3.1 Alignment Metrics

We collected alignment metrics for the four test sets using the Neighborly Clusters as well as a baseline using the full 108 Million sequences. Table 2.3 highlights the results of the CAFA 3 test set. Use of Neighborly 90 clusters resulted in a 50% reduction in run time for alignments, and a 50% reduction number of overall pair-wise alignments. This was done without a loss in the number of queries in the CAFA 3 test set for which we could make predictions. Use of Neighborly 80, 70 50L10, 50L20 resulted in a 50-80% reduction in run time for alignments, and a 50-85% reduction for overall pair-wise alignments (Table 2.3). Similar results were observed using the two temporal hold-out test sets and the larger CAFA 3 dataset. It is important to note that reduction in file size (>50%) using the all sequences database (5.8GB) versus the Neighborly 90 (2.8 GB) for the CAFA 3 dataset. The use of Neighborly clusters significantly reduces the time for alignments and generates data without sacrificing the number of query sequences being predicted for.

Table 2.3: Alignment Metrics using Neighborly Clustes

	CAFA 3 Test Set, Biological Process					
Cluster	Time [s]	Percent Reduction	Queries Aligned	Pairwise Alignments	Percent Reduction	File Size
All	2,391.060	0.00%	2,143	67,573,977	0.00%	5.8G
Neighborly 90	1,178.610	50.71%	2,143	33,863,194	49.89%	2.8G
Neighborly 80	1,161.750	51.41%	2,143	30,747,011	54.50%	2.6G
Neighborly 70	707.578	70.41%	2,143	16,880,479	75.02%	1.4G
Neighborly 50L10	564.992	76.37%	2,143	12,571,995	81.40%	1.1G
Neighborly 50L20	473.267	80.21%	2,143	9,820,732	85.47%	831M
	CAFA 3 Test Set, Molecular Function					
Cluster	Time [s]	Percent Reduction	Queries Aligned	Pairwise Alignments	Percent Reduction	File Size
All	2,083.730	0.00%	1,092	31,908,198	0.00%	2.8G
Neighborly 90	1,011.690	51.45%	1,092	15,739,351	50.67%	1.4G
Neighborly 80	985.772	52.69%	1,092	15,605,615	51.09%	1.4G
Neighborly 70	626.392	69.94%	1,092	9,221,836	71.10%	776M
Neighborly 50L10	501.683	75.92%	1,092	7,105,376	77.73%	605M
Neighborly 50L20	430.739	79.33%	1,092	5,769,324	81.92%	490M

2.3.2 Temporal holdout datasets

We wanted to assess the quality of our predictions on unseen datasets using the well-established CAFA 3 evaluation criteria. We created two datasets, one from SwissProt and one from TrEMBL, that contained annotated genes not used to build our Neighborly sequence similarity network. These two datasets functioned as our temporal holdout. We utilized our Rank based approach (Top 1 to 100 alignments) to make predictions for both Gene Ontology domains of biological process and molecular function. We used DeepGOPlus as a baseline along with the all sequences database and databases created by CD-HIT 90, 80, and 70 Clusters. Clusters and sequences were annotated using terms that included (IEA) or excluded electronic annotations (non IEA).

PR curves indicated prediction for the TrEMBL dataset by Neighborly 90 clusters without IEA started at Rank 1 with a precision close to 0.95 and a recall close to 0.88 (Figure 2.7). As the ranks were traversed to 100, the precision decreased to 0.45 while the recall approached 1. The final calculated F_{max} was identified to be 0.92. Surprisingly, baseline results using the all sequences database was marginally better with a F_{max} of 0.94 (Figure 2.8). Neighborly 80 and 70 clusters had slight decreases in precision and recall, though the shape of their curves were consistent with Neighborly 90. The calculated Neighborly 50L10 and 50L20 had a greater decrease in precision down to 0.8 and recall to 0.6 at Rank 1. F_{max} scores decreased to 0.80 and 0.75, respectively. DeepGOPlus produced a F_{max} score of 0.49, similar to its previously published results on the CAFA 3 test set [39]. PR curve generated by predictions from CD-HIT 90 clusters overlapped near perfectly with the PR curve by Neighborly 90 clusters. CD-HIT 80 and 70 were significantly reduced in recall by 0.2 and a minor increase in precision in comparison to their Neighborly counterparts.

Analysis of biological process predictions for our TrEMBL dataset using misinformation-

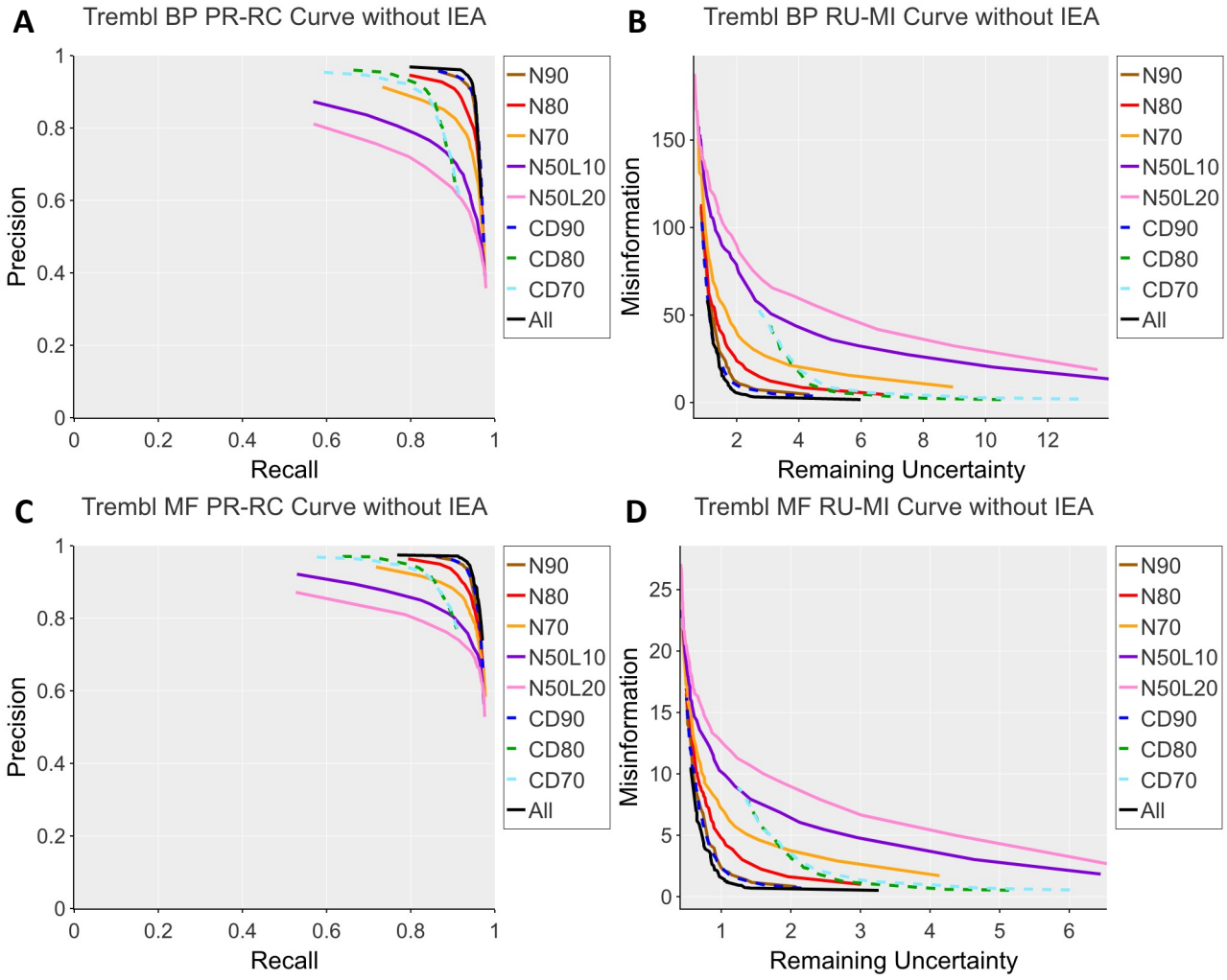

Figure 2.7: Precision-Recall Curves and Misinformation-Remaining Uncertainty Curves for Predictions on the temporal holdout TrEMBL dataset for non-IEA annotations.

remaining uncertainty curves resulted in a similar assessment for quality of predictions (Figure 2.7 B). Predictions via Neighborly 90, CD-HIT, and all sequences produced overlapping curves with a calculated S_{min} between 4.1 and 6.2 (Figure 2.9). Interestingly the S_{min} increased by approximately 4 units when the threshold for sequence similarity decreased by 10 percent for Neighborly clusters. Neighborly 80 and 70 clusters significantly out-performed CD-HIT 80 and 70 as CD-HIT clusters had a higher degree of remaining uncertainty even at lower ranks (Figure 2.7 B). However, the S_{min} for CD-HIT 80 and 70 were higher than

Neighborly 80 and 70 respectively as the misinformation values for predictions by CD-HIT 80 and 70 were significantly lower.

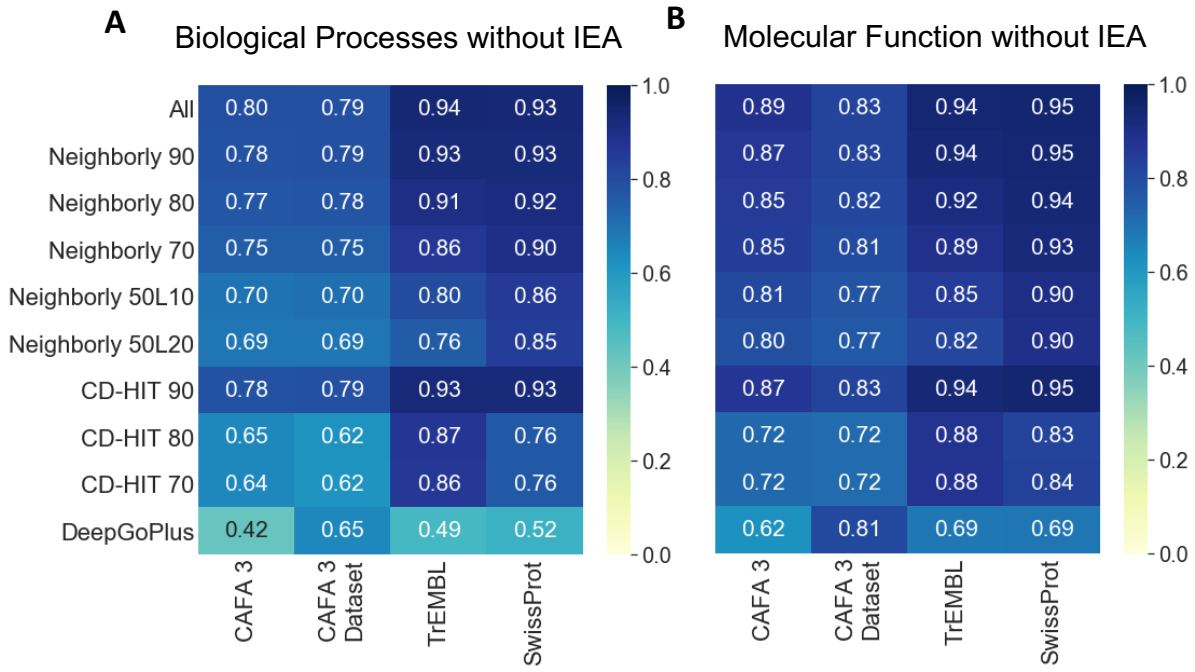

Figure 2.8: F_{max} values for Biological Process and Molecular Function terms without IEA annotations across four datasets

We then conducted a similar analysis for molecular function terms using Neighborly clusters, CD-HIT clusters, and all sequences database as well as DeepGoPlus. PR curves indicated prediction for the molecular function TrEMBL dataset by Neighborly 90 clusters without IEA started at Rank 1 with a precision close to 0.95 and a recall close to 0.88. As the ranks were traversed to 100, the precision decreased to 0.45 while the recall approached 1 (Figure 2.7). The final calculated F_{max} was 0.93. Surprisingly, baseline results using the all sequences database was marginally better with a F_{max} of 0.94 (Figure 2.8 B). Neighborly 80 and 70 clusters had slight decreases in precision and recall, though the shape of their curves were consistent with Neighborly 90. The calculated Neighborly 50L10 and 50L20 had a greater decrease in precision down to 0.8 and recall to 0.6 at Rank 1. F_{max} scores decreased to 0.85

and 0.81, respectively. DeepGOPlus produced a F_{max} score of 0.68, similar to its previously published results on the CAFA 3 test set. PR curve generated by predictions from CD-HIT 90 clusters overlapped near perfectly with the PR curve by Neighborly 90 clusters. Once again, CD-HIT 80 and 70 were significantly reduced in recall by 0.2 and a minor increase in precision in comparison to their Neighborly counterparts. Across all Neighborly and CD-HIT clusters predictions for molecular function terms, the inclusion of terms inferred by electronic annotations resulting in significantly decreased F_{max} due to a drop in precision (0.1–0.2) and a minor increase in recall.

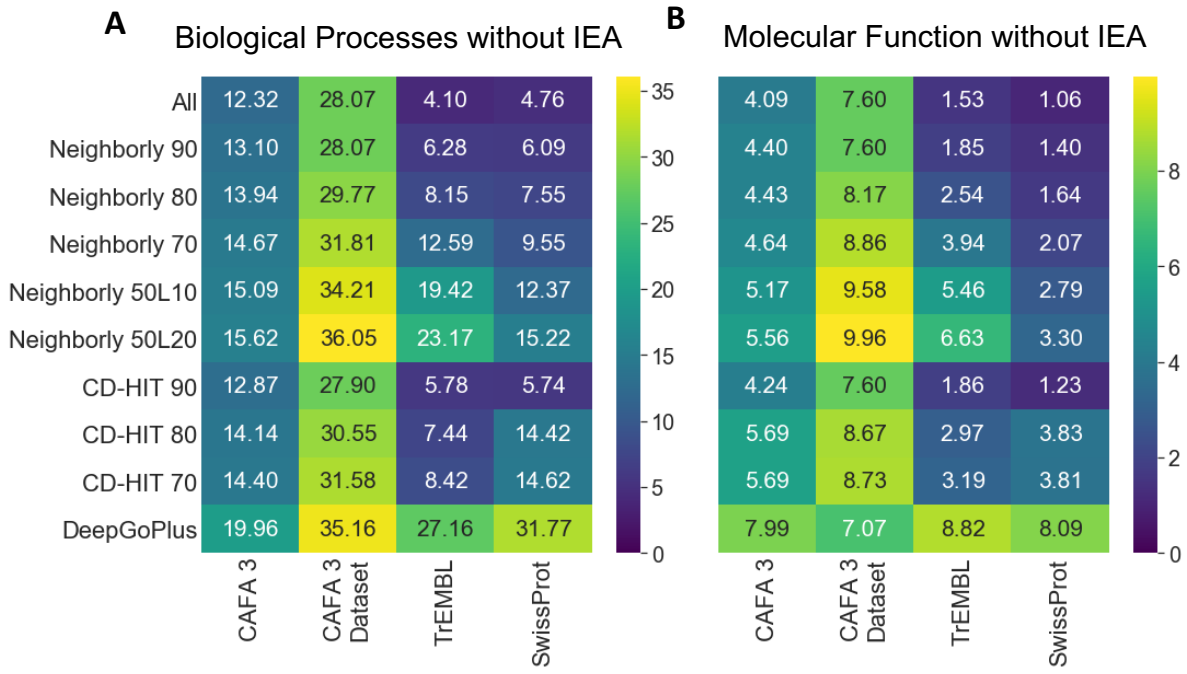

Figure 2.9: S_{min} values for Biological Process and Molecular Function terms without IEA annotations across four datasets

Analysis of misinformation-remaining uncertainty curves for molecular function predictions for the TrEMBL dataset resulted in a very low S_{min} scores for the majority of the Neighborly, CD-HIT, and all methods (Figure 2.9 B). Predictions via Neighborly 90, CD-HIT, and all sequences produced overlapping curves with a calculated S_{min} between 1.0 and 1.4 (Figure

2.7 D). The S_{min} increased for Neighborly 80, 70 and 50L10/L20 clusters, but only to 3.3 (Figure 2.9 B). Again, Neighborly 80 and 70 clusters significantly out-performed CD-HIT 80 and 70 as CD-HIT clusters had a higher degree of remaining uncertainty even at lower ranks. The S_{min} for CD-HIT 80 and 70 were higher than Neighborly 80 and 70 respectively, as the misinformation values for predictions by CD-HIT 80 and 70 were not much lower than Neighborly yet the remaining uncertainty was again more substantial.

We repeated this analysis using the SwissProt dataset, with similar results and consistent trends (Figure A.1, Figure 2.8, Figure 2.9).

2.3.3 CAFA 3 test set and dataset

The quality of our predictions by Neighborly as well as by CD-HIT and all sequences database was quite surprising. Alignment methods, such as BLAST/DIAMOND, function as "low" benchmarks for new methods seeking to show improvement. Recall that the CAFA 3 dataset is the training set of proteins (proteins with functional annotations) and test set is the testing set of proteins that were released as a part of the CAFA 3 competition. We decided to use Neighborly to predict for the CAFA 3 test set (Figure 2.10, Figure 2.8, Figure 2.9). It is important to reiterate that since CAFA 3 was released before the generation of Neighborly, sequences in this dataset may be included in our sequence similarity network. Against the CAFA 3 test set Neighborly, CD-HIT and all sequence database scored with a slightly decreased values of F_{max} and slightly increased values of S_{min} in comparison to the results from the TrEMBL and SwissProt test sets. General trends observed for or between the Neighborly Cluster and CD-HIT prediction methods were once again conserved. Our evaluation of DeepGoPlus was in accord with previously published results, suggesting our evaluation pipeline was appropriately implemented.

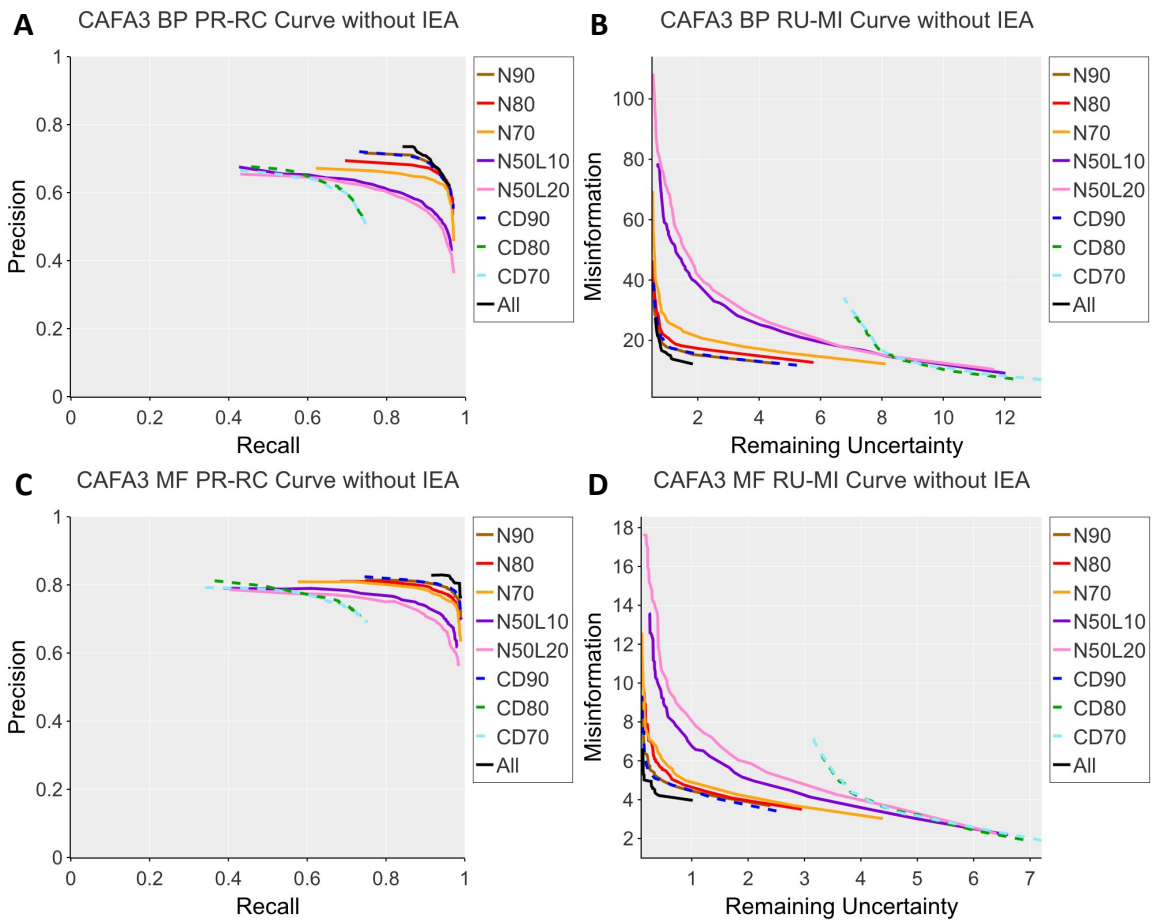

Figure 2.10: Precision-Recall Curves and Misinformation-Remaining Uncertainty Curves for Predictions on the temporal holdout CAFA 3 test set for non-IEA annotations.

We then thought to use the CAFA 3 training set as an additional test set for Neighborly as this set of annotated protein was comprehensive in size and scope of GO terms (Figure A.2, Figure 2.8, Figure 2.9). Once again similar trends observed for the temporal hold out and CAFA 3 test set were observed, though the F_{max} were generally lower by 0.1–0.2 and the S_{min} was significantly increased.

2.3.4 Threshold-based evaluation results

One major concern of ours was the significantly increased F_{max} score of Neighborly, CD-HIT, and all sequences in comparison to the BLAST alignment method utilized by CAFA 3 and others benchmarking their tools. We, therefore, used the CAFA 3 training set to re-create the BLAST based database used for predictions (Figure 2.11). Use of the CAFA3 training set for BLAST/Diamond based predictions resulted in a poor F_{max} (0.47 for molecular function, 0.35 for biological process) and S_{min} (9.911 for molecular function, 18.95 for biological process) as previously published. Thus the inclusion of a more extensive sequence library or derivatives of that library significantly increases the predictive power of the alignment-based method, as noted through a near doubling of the F_{max}. It is essential to additionally note that the F_{max} scores for all methods were comparable using either our rank-based assessment or the percentage sequence identity-based assessment (Figure A.9, Figure A.10, Figure A.11).

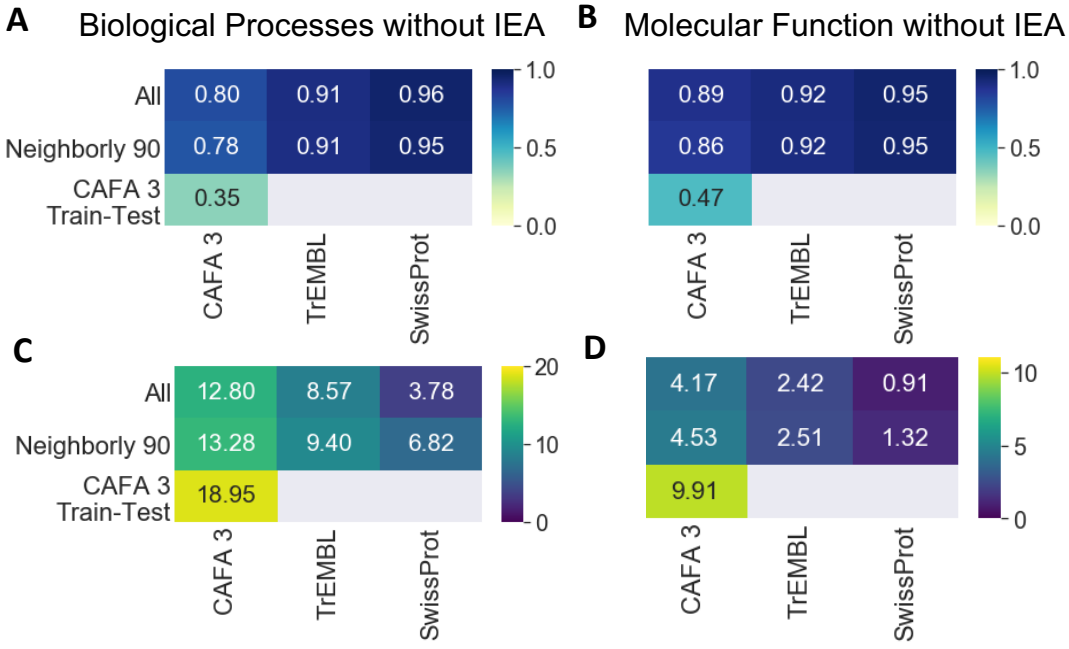

Figure 2.11: Summary of F_{max} and S_{min} Scores for Molecular Function and Biological Process without IEA labels across three datasets using threshold-based evaluation.

It is also important to note that across all datasets and both biological process and molecular function terms inclusion of IEA annotations resulted in decrease in performance (Figure A.7, Figure A.8, Figure A.3, Figure A.4, Figure A.5, Figure A.6).

2.3.5 Weighted Accuracy

Our analysis of PR and MI-RU indicated Neighborly 90, 80, and 70 clusters performed close to the all sequence database predictions. We then thought to gain an appreciation for the proximity of our Neighborly predictions to the known annotations. Using our weighted accuracy formulation, we found a convergence to a plateau at a score of 0.95 and 1 by rank 60 for all Neighborly, CD-HIT, and all clusters except for CD-HIT 80 and 70 for biological process and molecular function annotations respectively (Figure 2.12, Figure A.12). From rank 1–20 Neighborly 90,80, 70 and CD-HIT 90 clusters overlapped significantly with the all sequences based predictions for both sub-ontologies. Both Neighborly 50L10 and 50L20 were notably lower; however, both outscored CD-HIT 80 and 70. Interestingly, the inclusion of IEA terms increased the overall accuracy for predictions. This was surprising given its detrimental effects on both F_{max} and S_{min}. We suspect the method to calculate weighted accuracy introduces bias as the highest-ranking score is kept if there are multiple annotations associated with a given term.

2.4 Discussion

We were quite surprised at the high quality of our alignment-based methods. The approximate F_{max} published by CAFA using local alignment-based methods were 0.26 for biological process and 0.42 for molecular functions. Even the best reported methods had a F_{max} of

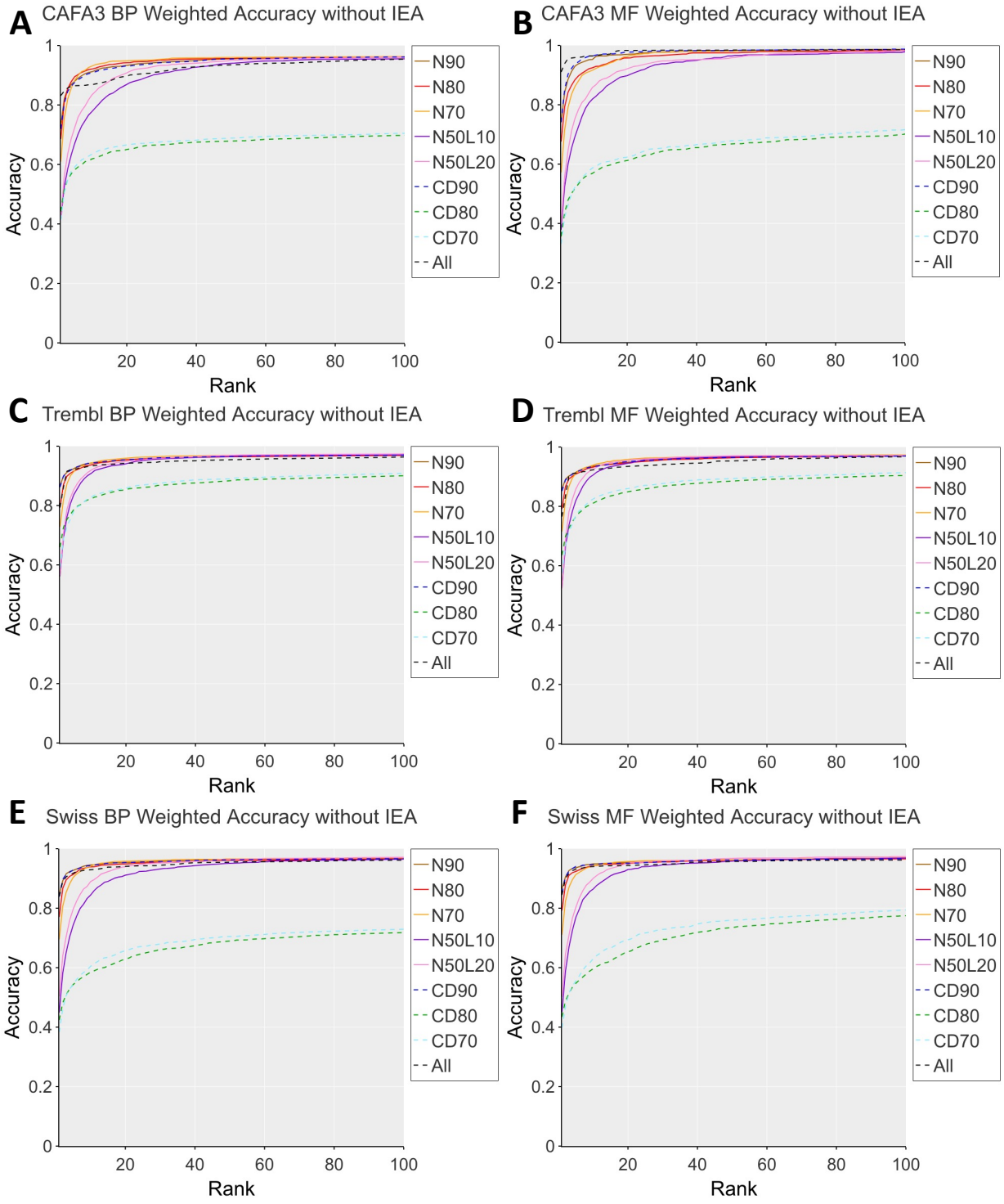

Figure 2.12: Weighted Accuracy Plots for biological process and molecular function terms without IEA annotations

0.40 for biological process and F_{max} of 0.62 for molecular functions. The use of Neighborly, CD-HIT, and all sequences produced a significantly higher score than other methods for all four of our datasets. The use of the CAFA 3 test set, training set, and evaluation criteria, as well as DeepGOPlus, showed our results were in accord with previously published results and that our analysis pipeline was functioning as expected. This critical piece of information, coupled with our F_{max} results using the more extensive database of all sequences, justified that the dramatic increase in F_{max} was due to the increase in available data. The use of our temporal holdout dataset showed that we produced a similar F_{max} scores on data that Neighborly had not previously been built upon. Thus the improvement cannot be attributed to the redundancy of sequences. Finally, the use of the CAFA 3 training set as a test set gave a sense of how Neighborly scores in regards to a large group of diverse sequences, similar to those one would observe in a genome.

Our resultant F_{max} for the all sequences database predictions indicates that the sheer size of our original dataset of sequences provides the respective increase and decrease in F_{max} and S_{min}. This quality of prediction comes at a cost associated with compute time as well as the size of the output file size. Neighborly 90, 80, and 70 clusters provide a significant decrease in compute time and output file size for all four datasets without a significant sacrifice in F_{max} or S_{min}. Interestingly, CD-HIT 80 and 70 had significantly weaker performance than their Neighborly counterparts, while CD-HIT 90 and Neighborly 90 produced similar scores. Unpublished and forthcoming comparisons of CD-HIT 90 and Neighborly 90 clusters indicate a very low Jaccard Index between clusters and low bias between the size of clusters for a given method. This finding makes the similar PR curves and F_{max} values produced by both methods remarkable at the 90 percent threshold. We suspect the greedy nature of how CD-HIT clusters are generated does not transfer well when using the CD-HIT 90 representative sequences as a basis to generate of CD-HIT 80 and 70 clusters. This may manifest itself in

the observed decrease in the F_{max} of CD-HIT 80 and 70 in comparison to their Neighborly counterparts. It will be interesting to determine if other greedy algorithms such as linclust also have a drop in predictive quality as observed for CD-HIT or if they function as well or better than Neighborly at the 80 and 70 percent sequence identity.

The use of GO terms inferred from electronic annotations to aid in gene function prediction has been widely discussed for the potential benefits and detriments. We show that the incorporation of IEA reduced the overall F_{max} and S_{min}. The decrease in F_{max} can be attributed to a more substantial drop in precision and a marginal increase in recall. Similarly, the increase S_{min} can be attributed to an increase is misinformation that is not negated by a decrease in remaining uncertainty. Thus the use of IEA for BLAST/DIAMOND based prediction methods is more detrimental than beneficial.

Chapter 3

Weighted Path Method to Reduce GO annotations

3.1 Introduction

Prediction of protein function is a field full of diverse methods relying primarily on a form of sequence similarity to facilitate annotation. When a given protein shares high sequence similarity with several well-annotated genes, the transfer of appropriate annotations becomes an issue. Similarly, deciding the appropriate annotations for a cluster of sequence similar proteins is not a trivial problem. There are many potential solutions to appropriately and effectively annotate clusters. Several of these methods derive from tools utilized in transcriptomics analysis to determine pathway enrichment via Gene Ontology Enrichment Analysis (summarized in Chapter 1).

One simple method for cluster annotation would be to keep all unique annotations associated with proteins in the cluster. This approach could increase the number of false positives as inaccurate annotations would be of equal value to well-annotated and potentially more abundant terms. A second method, looking at only the most abundant terms, could result in a selection of shallow terms in the GO, as most genes would not have deep descriptive annotations. This method would eliminate many spurious terms with low abundance; however meaningful results with rigorous experimental annotation could be removed as well. A third

method, enrichment of terms via Gene Ontology Enrichment Analysis (GOEA), provides a way to capture terms within the GO using statistical methods to remove abundant terms. These statistical methods are dictated by the number of terms annotated within the GO and can factor in the abundance of parent terms. Incorporation of post-hoc methods to then identify child terms identified in the set from enriched parent terms have been incorporated by methods such as TOPGO to aid in providing the most descriptive term [2]. While GOEA methods work well for small datasets such as a given transcriptome, they do not necessarily scale well for 10s -100s of millions of clusters, where the statistical tests need to be repeated for each GO term associated with the cluster.

We propose a weighted path-based method to identify the most descriptive terms for gene annotation and protein function prediction. In principle, this method identifies all leaf terms in a graph and creates a subgraph of possible paths to the root node. Each path starts from the leaf terms and consists of all the terms along shortest path from the leaf/child term to the root node. Each term in the path is weighted by the number of genes annotated to that term, the final weight for the path is the sum of individual terms present in the path. The leaf term from that path with highest weight is selected as the final annotation. The compute time and resource requirement is dramatically reduced because we are looking at annotations for each cluster locally, contrary to enrichment methods that conduct a large number of statistical tests and require the background annotation information.

3.2 Methods

3.2.1 Path Based Method for Cluster Reduction

Let C be a cluster of genes. For every GO term t, let C_t be the set of genes in C that are annotated by t. We define the weight $w_{t,C}$ of t in C as $|C_t|$. We define t to be a leaf with respect to C if $w_{t,C} > 0$ and $w_{s,C} = 0$ for every descendant of t in the GO DAG. Consider the path π_t in the GO DAG from a term t that is a leaf with respect to C to the root of the (corresponding domain in the) GO DAG. We define the weight of this path as the sum of the weights of the terms in it. We select the path with the highest weight and assign the leaf term in it as the annotation for C. If multiple paths are tied for the highest weight, we include all the corresponding leaf terms as annotations for C.

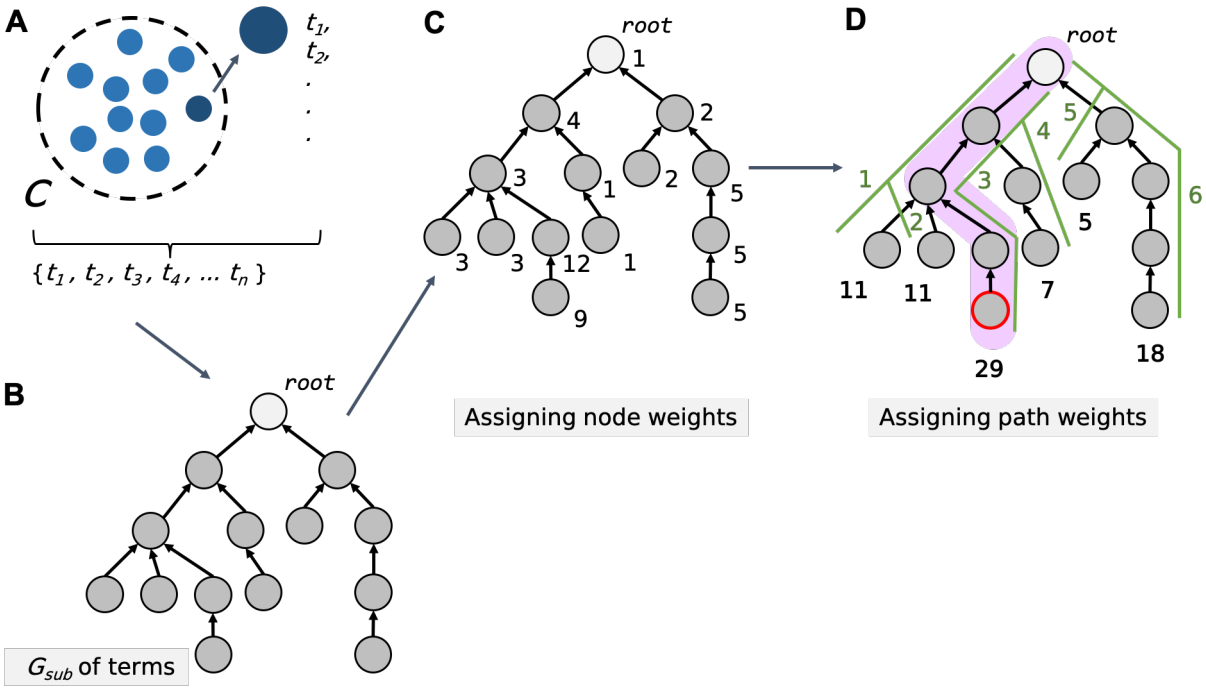

Figure 3.1: Example of calculating Weighted Path Annotations

For example, consider a cluster C, shown in Figure 3.1A. To create a set of GO terms $\{t_1, t_2, ...t_n\}$ associated with the cluster C, we aggregate all the annotations of genes in the cluster. Assume a subgraph G_{sub}, that is formed by the set of GO terms $\{t_1, t_2, ...t_n\}$ (shown in Figure 3.1B). The light grey node denotes the root of the domain in the GO DAG, while dark grey nodes represent the terms associated with the cluster. For each term $t \in G_{sub}$, the weight $w_{t,C}$ of the term t equal to the $|C_t|$, where $|C_t|$ denotes the number of genes in cluster C annotated to the term t. The weights of the terms are shown in Figure 3.1C. The weights of each path π_t in G_{sub} is the sum of all weights of terms t in the path and is shown in Figure 3.1D (where green line shows the path and the numbers in green denote the path weight). For instance, the weight of path 3 is equal to 29. Finally, the leaf term t from the path with the highest weight is selected as the final annotation of the cluster C (path with the highest weight is highlighted in purple in Figure 3.1D and the leaf term of that path is highlighted with a red boundary).

3.2.2 Gene Ontology Enrichment Analysis

Gene ontology enrichment analysis (GOEA) encompasses a set of statistical methods that characterize the composition of gene/protein sets. More specifically, given a set of genes and their associated GO terms, GOEA methods aim to identify a set of GO terms that are underrepresented or enriched with statistical confidence. GOEA methods are often utilized in transcriptomics to determine if a set of genes are associated with a given pathway, process, or function in a statistically significant manner. Different statistical tests like the hypergeometric/Fisher's exact test, chi-squared test, Z-score, and Kolmogorov-Smirnov test have been proposed for enrichment analysis [30]. Rivals et al. [55] discusses the advantages and disadvantages in detail for various statistical tests for testing enrichment of GO terms.

To perform GOEA, we used the GOATOOLs package, which uses the Fisher's Exact test to compute an enriched set of gene products for a given cluster [38] with minor modifications. Fisher's Exact test requires the following set of inputs:

1. Background population of gene products.
2. GO terms associated with these gene products.
3. Gene Ontology DAG.
4. Members of a given cluster.

Fisher's exact test compares the proportions of one sample with respect to the other and checks if the difference in proportion is by chance or there is indeed a difference. More formally, for Gene Ontology Enrichment Analysis, given a cluster and the GO terms associated with the genes in that clusters, Fisher's exact test tries to find "enriched" set of GO terms. Enriched set refers to the GO terms that characterize the functional information associated with the cluster. This is done by comparing the frequency of annotations of one GO term in the set of genes (clusters of genes) with the frequency of that term in the entire annotation database (Gene Ontology Annotations database). The comparison leads to the calculation of a p-value associated with every term in the GO term set.

Consider the contingency table in Table 3.1, where the total of the rows represents the distribution of genes in the cluster and the annotation file, while the columns represent the distribution of a particular GO term T.

The *null hypothesis* assumes that there is no association between the values in the contingency table. Fisher's test assumes that the marginal values are fixed, i.e., the row and column totals are constant. Based on this assumption, Fisher's Test provides a statistical basis to measure how likely our particular configuration of values in the table is, compared

Table 3.1: Example of calculating p-values for Fisher's Exact Test.

	Associated with GO term T	NOT associated with GO term T	
Number of Genes in the Cluster	12	3	**15**
Number of Genes in the Annotation File	3	12	**15**
	15	**15**	**30**

to all possible arrangements we can get by having the same marginal row and column totals. Figure 3.2 highlights all the 16 possible configurations we can get by keeping the row and column sums fixed.

	C1		C2		C3		C4	
R1	0	15	1	14	2	13	3	12
	15	0	14	1	13	2	12	3
R2	4	11	5	10	6	9	7	8
	11	4	10	5	9	6	8	7
R3	8	7	9	6	10	5	11	4
	7	8	6	9	5	10	4	11
R4	12	3	13	2	14	1	15	0
	3	12	2	13	1	14	0	15

Figure 3.2: All possible configurations possible given the row and column sums are fixed.

Consider the arrangement in the first row(R1) and second column(C2) from Figure 3.2 shown in Table 3.2. Fisher's Exact Test uses rules from combinatorics to calculate the probability of this unseen configuration under the null hypothesis. This is done by calculating three values (one for each row). For the configuration in Table 3.2, these would be as follows:

Table 3.2: Table from Row 1(R1) and Column 2(C2)

	Associated with GO term T	NOT associated with GO term T	
Number of Genes in the Cluster	1	14	**15**
Number of Genes in the Annotation File	14	1	**15**
	15	**15**	**30**

1. Ways in which 1 gene product in a cluster is associated with the GO Term T out of a total of 15 cluster members.

2. Ways in which 14 gene products are associated with the GO term T out of total 15 gene products in the annotation file.

3. Ways in which 15 gene products are associated with the GO term T from a total of 30 gene products.

This would be put forth as 15 Choose 1 $\binom{15}{1}$, 15 choose 14 $\binom{15}{14}$ and 30 choose 15 $\binom{30}{15}$ respectively. The probability can then be determined as:

$$p = \frac{\binom{15}{1} * \binom{15}{14}}{\binom{30}{15}} = \frac{15 * 15}{155117520} = 1.45E-6 \tag{3.1}$$

We follow the same steps to calculate the probability of each configuration, as shown in Figure 3.3. To get the final p-value, we sum all the probabilities that are lesser than equal to our input configuration in Table 3.1. These would be the sum of all the probabilities highlighted in red in Figure 3.3.

$$
\begin{aligned}
p &= \sum_{p \leq 1.33\times10^{-3}} 2 * \left(1.33 \times 10^{-3} + 7.11 \times 10^{-6} + 1.45 \times 10^{-6} + 6.45 \times 10^{-9}\right) \\
&\approxeq 0.0028
\end{aligned} \tag{3.2}
$$

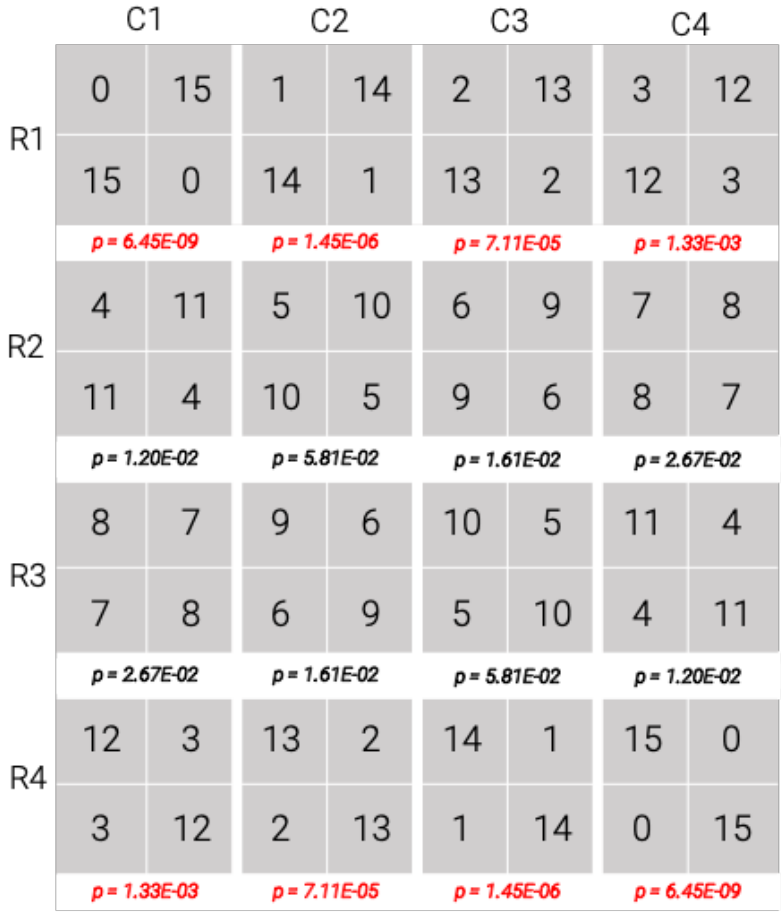

Figure 3.3: Probabilities calculated for each configuration

In this example, the final value p-value for a given GO term T is 0.0028. The same procedure is repeated for all GO terms, resulting in a final list of GO terms, each associated with an uncorrected p-value. We refer to them as uncorrected p-values because when we perform a large number of statistical tests, there will be some p-values that are less than 0.05, purely by chance, even if all null hypotheses were true. When we perform Fisher's exact test for a

broad set of GO terms for a particular cluster, we must account for this phenomenon and adjust the p-values accordingly. Unadjusted p-values result in some false positive GO terms being associated with a cluster just by chance. The simplest way one can correct the p-values is by using the Bonferroni Correction [46]. The cut-off alpha is divided by the total number of tests performed, and all the values lesser than this new alpha value are considered to be statistically significant. For example, if we perform a hundred tests, the new alpha value becomes 0.005 (0.05/100). However, for a large number of tests, this method fails and may result in a large number of false negatives.

An alternate approach to correct the false discovery is to use the Benjamini-Hochberg correction [46]. To apply Benjamini-Hochberg correction, we start by arranging all the p-values for a given cluster, into ascending order, from smallest to largest and ranking them. Correspondingly, the smallest p-value gets a rank of 1; the next smallest gets a rank of 2, and so on. The Benjamini-Hochberg cut-off value for a given value is defined as:

$$\textit{cut-off value} = \left(\frac{i}{m}\right) Q \tag{3.3}$$

where i is the rank of the p-value, m is the total number of tests performed and Q is the user-defined alpha value. The largest p-value that is smaller than equal to its cut-off value is significant, and all the p-values less than it are also considered significant. This also includes the ones that are not less than their cut-off value. We use $Q = 0.05$ for our experiments. Hence, for a given set of proteins belonging to a cluster, GOATOOLS returns a set of GO terms along with their corrected and uncorrected p-values. We choose to perform detailed evaluation only on corrected samples as the uncorrected results provided minimal reduction (data not shown).

3.2.3 Set Comparison for Assessment of Reduction (Jaccard Index)

We initially compared weighted path annotations, GOEA annotations, and original annotations to each other using the Jaccard Index. Jaccard Index is a measure used to compare two sets by looking at the elements that are common and elements that are different [32]. More formally, it is defined as the ratio of the number of elements in the intersection of two sets to the number of elements in the union of two sets.

If X and Y are two sets, then Jaccard Index is calculated as:

$$Jaccard\ Index(X,\ Y)\ =\ \frac{|X\ \cap\ Y|}{|X\ \cup\ Y|} \tag{3.4}$$

Given that both the weighted path and GOEA annotations are a subset of the original annotations, the ratio between the union and intersect can be additionally defined as the terms unique to the original annotations. This does not apply to comparisons between weighted path annotations and GOEA annotations.

3.2.4 Semantic Similarity

Resnik's similarity measure uses information content to define the similarity between two GO terms. Information content for any term t is defined as the negative log to the base 10 of the probability of that term [53, 54]. The probability is calculated from the frequency of that term in the annotations database. More formally, the frequency of any term t is defined as:

$$frequency(t) \;=\; count(t) \;+\; \sum_{c \,\in\, child(t)} frequency(c) \tag{3.5}$$

where $count(t)$ represents the number of proteins that are associated with term t in the database, $child(t)$ represents the set of child nodes of term t.

The probability of the term is t then is defined as the ratio of its frequency in the database of annotations to the frequency of the $root$.

$$p(t) \;=\; \frac{frequency\,(t)}{frequency\,(root)} \tag{3.6}$$

The probabilities are calculated independently for each domain. Intuitively, the probability keeps on increasing as we move from a leaf node to the root of the ontology.

Resnik's similarity is based on the assumption that two terms have strong similarity when they share greater amounts of information. Moreover, this shared information is derived from the set of common ancestors. More formally, Resnik's similarity between two terms t_1 and t_2 is defined as:

$$similarity_{Resnik}(t_1,\ t_2) \;=\; \max_{c \,\in\, A(t_1,\ t_2)} (-\log_{10}(p(c))) \tag{3.7}$$

where $A(t_1,\ t_2)$ represents the set of common ancestors of the terms t_1 and t_2. One important consideration of Resnik's similarity score is that it is not bounded by a set range from above.

Lin's semantic similarity is based on the information-theoretic definition of similarity between two sets of GO terms that assume a probabilistic model [42]. It assumes that the similarity of two GO terms is based on their commonality. More formally, the similarity of the two terms is defined as the ratio of their commonality and the information needed to describe

them adequately. The commonality for two terms is measured by their common ancestor. The information needed to describe them is the sum of their information content. Lin's Semantic Similarity is then mathematically defined as:

$$similarity_{Lin}(t_1,\ t_2) \ = \ \max_{c \ \in \ A(t_1,\ t_2)} \left\{ \ \frac{2 \ log(p(c))}{log(p(t_1)) \ + \ log(p(t_2))} \ \right\} \tag{3.8}$$

where $A(t_1,\ t_2)$ represents the set of common ancestors of the terms t_1 and t_2.

Unlike Resnik's similarity measure, which has a minimum value of 0 and no upper bound for maximum value, Lin's similarity ranges from 0 to 1. We chose to use Lin's Similarity as it provides a normalized value with easy interpretation. A score of 1 represents the two terms are identical, while a score of 0 means there is no similarity between two GO terms. To compute the Lin's Similarity Score between two annotation types, we create two sets, each representing the annotations from that type. Then for each pair of annotations between the two sets, we compute Lin's similarity score for that pair. After computing the score for each pair, we take the mean of that score as the final score associated with the cluster.

3.3 Results

3.3.1 Effect of path-based reduction on number of unique annotations per cluster

We implemented our weighted path-based method on all the biological process and molecular function annotations without IEA annotations for Neighborly clusters to determine the extent of the reduction in terms (Figure 3.4). 125,339 Neighborly 90 Clusters originally had exactly one annotation for biological process, while 90,189 had between 2-5, and 21,035 had

greater than 5 annotations. In general, the majority of clusters 235,585 were reduced to a small number of labels (1-5) from a range of initial values (1-250) (Figure 3.4A). 19 Clusters out of the 21,035 clusters with greater than five annotations did not have a reduction. This was not too surprising since a given gene may be involved in several biological processes. Similar results were observed for Neighborly 80-50L20 clusters for biological processes annotations (Figure A.13).

We then thought to plot the calculated Jaccard index (ratio of intersection over union) for each cluster. This plot indicated two distinct features: a reduction in original annotations to 1-10 terms, and a small subset of clusters were the original annotations (10+) were reduced by less than half (Figure 3.4B). Similar results were observed for Neighborly 80-50L20 clusters for biological process annotations (Figure A.13). We then utilized Lin's similarity score (LSS) to assess the semantic similarity between the retained biological process annotations and the original annotated terms (Figure 3.4E). Our results indicate the median LSS for all Neighborly clusters was between 0.75 and 0.8, and a range of 1.0 to 0.5 from the 100^{th} percentile to the 25^{th} percentile. Given the massive reduction in GO terms by weighted path methods, the preservation of over half of the semantic similarity was notable. We repeated our weighted path based reduction for the molecular function annotations without IEA for Neighborly clusters and evaluated the extent of reduction (Figure 3.4C). 152,141 Neighborly 90 Clusters originally had exactly one annotation for molecular functions, while 82,747 had between 2-5, and 6,509 had greater than 5, but no more than 40. We again noted that the majority of 240,905 clusters had a significant reduction in GO terms down to 1-5 final annotations, while a small subset (50) with greater than five terms had a marginal reduction ($\sim 13\%$) (Figure 3.4C, Figure 3.4D). Similar results were observed for Neighborly 80-50L20 clusters (Figure A.14). We then utilized Lin's similarity score (LSS) to assess the semantic similarity between the retained molecular function annotations and the original annotated

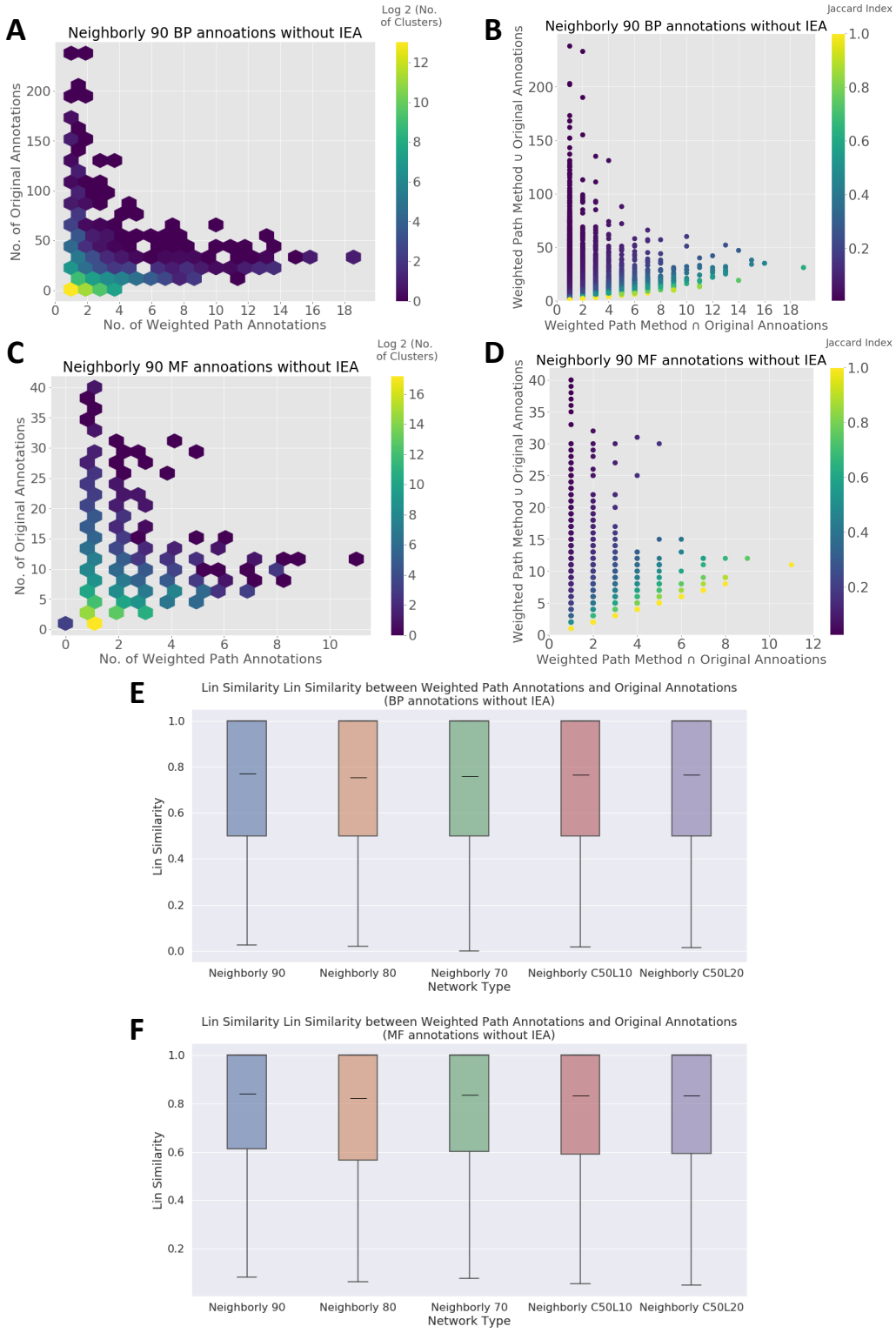

Figure 3.4: Comparison of Weighted Path Annotations and Original Annotations

terms. Our results indicate the median LSS for all Neighborly clusters was between 0.8 and 0.85, and a range of 1.0 to 0.55 from the 100th percentile to the 25th percentile.

3.3.2 Effect of GOEA reduction on number of unique annotations per cluster

We implemented the GOEA for biological process annotations without IEA on a subset of clusters from the Neighborly 90-50L20 networks. It is important to note that the time consumption for GOEA analysis per cluster was significant as it would have taken 42 days to perform the enrichment on Neighborly 90 clusters alone. Therefore, we performed our analysis on a subset of clusters from each network. This subset was created by randomly sampling 100 clusters that had either 10, 20, 30, 40, and 50 number of original annotations for a total of 500 samples. In contrast to the weighted path based method, we observed more of an even gradient and overall decrease in term reduction for GOEA. Median reduction was 73.07 for GOEA methods versus 95.0 by weighted path based reduction (Figure 3.5A, Figure 3.5B). Similar results were found for GOEA reduction for molecular functions, with a median reduction of 66.67 versus 90.00 by weighted path based reduction (Figure 3.5C, Figure 3.5D). Similar decreases in reduction were observed for Neighborly 80–50L20 clusters for both biological process and molecular function annotations (Figure A.15, Figure A.16). We then utilized Lin's similarity score (LSS) to assess the semantic similarity between the retained biological process annotations and the original annotated terms. Neighborly 90 clusters had a mean LSS of 0.72, but the scores from the 100th percentile to the 25th percentile ranged from 1 to 0. Oddly, this was not the case for Neighborly 80-50L20 clusters as the distribution of LSS indicated the enrichment of terms correctly represented (score of 1) the original biological process annotations (Figure 3.5E, Figure 3.5F). Similarly, nearly all molecular function annotations for Neighborly 90–50L20 clusters had near-perfect LSS (Figure 3.5D,

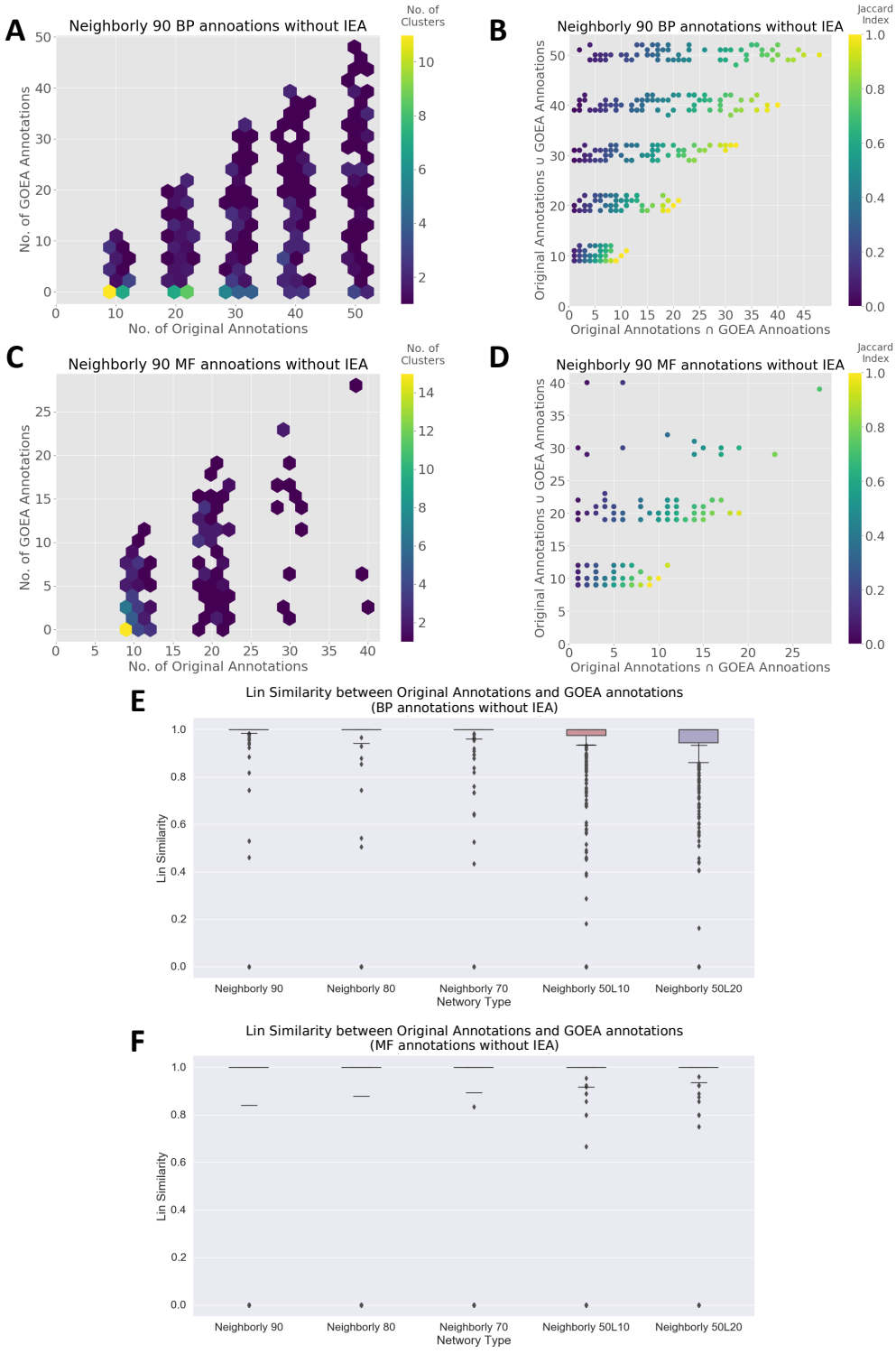

Figure 3.5: Comparison of Gene Ontology Enrichment Annotations and Original Annotations.

Figure 3.5E)

3.3.3 Comparison of Annotations by Weighted Path and GOEA

We thought of comparing the enriched terms by GOEA versus our weighted path based terms for biological processes annotations (Figure 3.6). In general, we noted a decreased number of terms for weighted path based methods in comparison to GOEA for both biological process and molecular function (Figure 3.6A, Figure 3.6C, Figure A.17, Figure A.18). A set-based comparison of annotations for Neighborly clusters indicated a modest overlap (intersection) by the two methods (Figure 3.6B, Figure 3.6D, Figure A.17, Figure A.18). LSS between weighted path and GOEA methods for Neighborly clusters reiterated this finding as the median score ranged from 0.18–0.24 (Figure 3.6E). Similar findings were observed for molecular function terms with a median score ranging from 0.27–0.29 (Figure 3.6F).

In summary, the sets of enriched terms by both methods shared marginal similarity. GOEA recapitulated the original semantic similarity, with a modest reduction in the number of annotated terms. Conversely, our weighted path reduction resulted in a significant reduction of terms, while sacrificing semantic similarity against the original annotations, but a large amount of similarity against the GOEA for the tested terms.

3.3.4 Weighted Path Annotations Reductions for Function Transfer

We thought to then test the quality of weighted path annotations in comparison to original annotations for predicting biological process and molecular function annotations using the CAFA 3, TrEMBL, and SwissProt test sets (See Chapter 2 for a description of test sets).

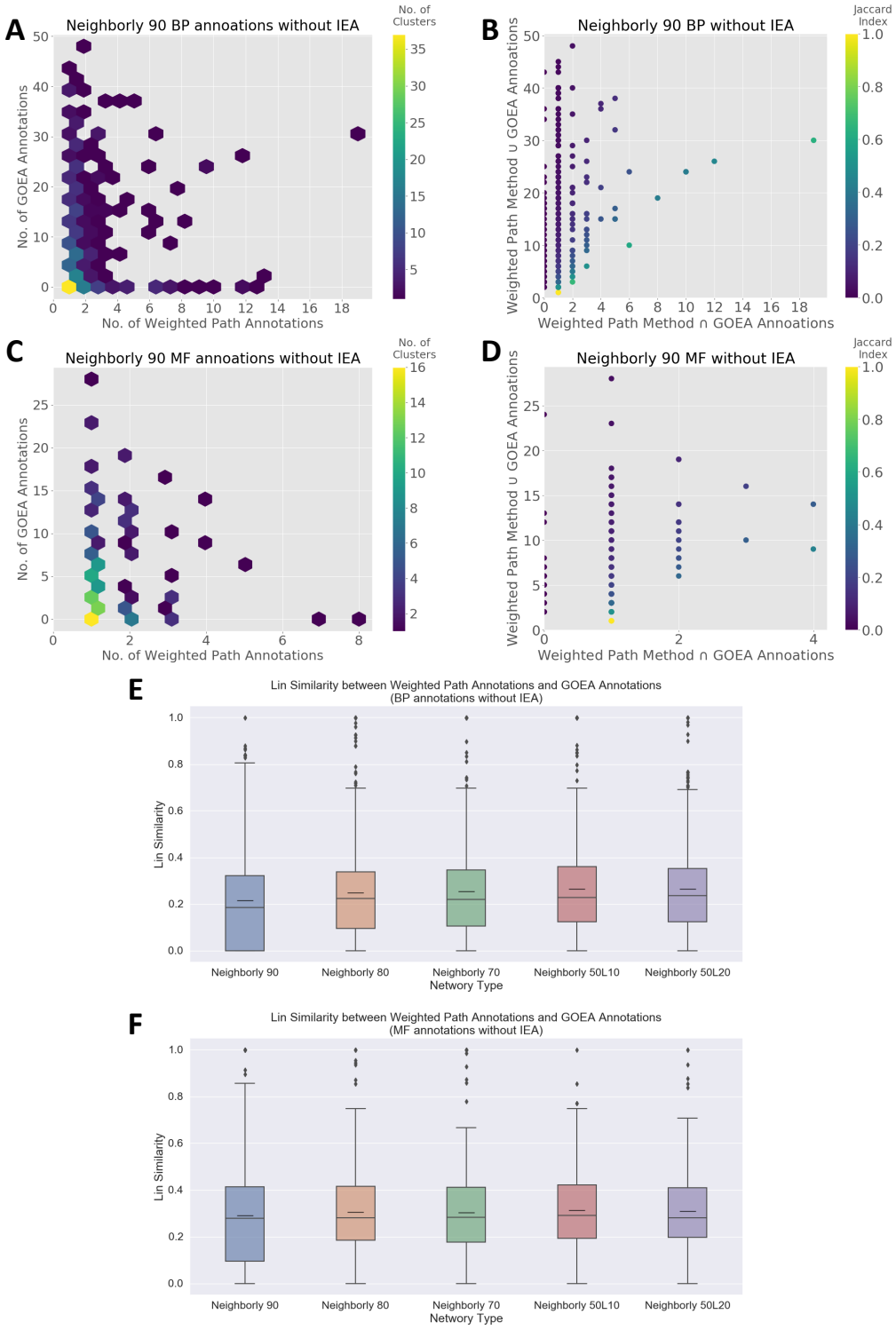

Figure 3.6: Comparison of Weighted Path Annotations and Gene Ontology Enrichment Annotations

Table 3.3: Summary F_{max} scores for weighted path based annotations in comparison to original annotations for CAFA 3 test set

	CAFA3 Biological Process w/o IEA			CAFA3 Molecular Function w/o IEA		
	F_{max}			F_{max}		
Method	Original	Weighted Path	% reduction	Original	Weighted Path	% reduction
All	0.80	0.68	14.21	0.89	0.84	5.33
Neighborly 90	0.78	0.66	15.88	0.87	0.82	5.42
Neighborly 80	0.77	0.64	17.42	0.85	0.81	5.12
Neighborly 70	0.75	0.60	20.48	0.85	0.80	4.98
Neighborly 50L10	0.70	0.59	15.05	0.81	0.78	3.57
Neighborly 50L20	0.69	0.58	15.36	0.80	0.76	3.83
CD-HIT 90	0.78	0.66	15.75	0.87	0.83	5.02
CD-HIT 80	0.65	0.63	2.12	0.72	0.81	-12.45
CD-HIT 70	0.64	0.62	3.38	0.72	0.80	-10.94

Across nearly all query sets, the F_{max} decreased by approximately 0.12 to 0.08 for biological predictions and 0.05 for molecular function predictions for cluster and non-cluster based predictions (Table 3.3, Table B.1, Table B.2). Closer analysis indicates the drop is due primarily to an initial decrease in recall and not precision. Conversely, the difference in S_{min} scores for biological processes varied between test set used. The weighted path method aided in decreasing S_{min} for CAFA 3 based predictions, but resulted in large increases for TrEMBL and SwissProt datasets despite the cluster or non-clustering method used. More specifically, we noted a decrease in misinformation, but a more significant increase in remaining-uncertainty is causing an increase in S_{min} (Figure A.19, Figure A.20).

3.4 Discussion

Rapid and scalable annotation of genes and clusters remains an important endeavor as the number of unique sequences continues to increase past hundreds of millions. The application of GOEA provides a solution that does not scale well for the tens of millions of clusters

generated as it requires a significantly massive compute, in the time frame of ∼42 days, to find enriched annotations for the Neighborly 90 clusters. Conversely, the development of our weighted path-based method allowed calculations to be completed in less than two days for a given cluster family. The trade-offs between the two methods include enhanced reduction by the weighted path-based method with a decreased Lin's Similarity Score with the original annotations. Annotations achieved by Gene Ontology Enrichment Analysis had a lower reduction in terms, but a near-perfect Lin's Similarity Score in reference to the original annotations. The low semantic similarity between the two methods suggests they are selecting different terms from the original annotations set.

The weighted path annotations resulted in a 5–20% decrease in F_{max}, suggesting that the reduced terms were having a dramatic impact on overall prediction for three of the test sets. It would be interesting to determine how GOEA effects the F_{max}, but due to the extensive resources required to compute, this is not possible.

Throughout this study, we identified two significant yet addressable shortcomings of our weighted-path method. The first is the lack of incorporation of additional weighted paths that are determined by our method. The second is the lack of a statistical analysis to address the likelihood that the identified annotation has not occurred by chance as well as discriminate between significant and insignificant annotations.

Incorporation of additional criteria to assess the significance of our paths is a challenging endeavor. We propose two solutions to aid this process. The first is the incorporation of a threshold. After assigning the weight to each path, we rank these paths based on the weights. The path with the highest weight gets a rank of 1, the path with the second-highest weight receives a rank of 2, and so on. We also use Lin's Similarity Score between the original annotations and reduced annotations as the threshold. We then include the minimum number of ranked paths incrementally to achieve Lin's score greater than equal to a

given threshold. The second method is a modification of the weighted path that incorporates propagation. This modified method would be to propagate the counts to the root node. If a node n_i is present five times in the list of annotations, all of its parents' counts will be incremented by 5. This will alleviate the apparent bias created by having paths that have a weight greater than the other path but only by a small margin. It is not yet known how computationally expensive these methods are.

One additional thought is that nearly all the terms annotated to a cluster or gene is done through a form of experimentation. In some instances, detailed experimental assessments are painstakingly conducted for an annotated term, which may hold for one member in a cluster due to the technical nature of the experiment. Our concern with GOEA and our current path method is that it negates this effort only because other research groups have not conducted the same experiment, which is often frowned upon due to an emphasis on novelty in science. This unfortunate bias potentially removes vital information for a given cluster. Thus, the rationale for enrichment analysis, that the selection of a term may be attributed to chance, and utilized for transcriptomics, may not be as directly applicable to annotations for sequence similarity clusters. Community evaluation methods such as CAFA focus and reward capturing the totality of terms, which creates a bias to identify more possible associations over depth of description. Ultimately, we propose emphasis should be put forth for a middle ground method that captures both detail and breadth of associations. Proposed modification to our weighted path method may hopefully facilitate this in a computationally efficient manner.

Chapter 4

Conclusion and Future Work

4.1 Summary

In this thesis, we introduced Neighborly networks—a novel way of organizing and deriving knowledge by clustering sequence similar proteins. In Chapter 2, we discussed how we could use larger databases of annotations to aid in protein function prediction. The inclusion of clustering information helped to significantly reduce the computational time and resource requirements of dealing with more extensive databases while sacrificing a modest amount of performance. We also discussed different metrics to evaluate the performance of predictions and how each metric provides a different view of predictor performance. Finally, we observed that the inclusion of IEA (electronically inferred) annotations are more detrimental than useful for function prediction using function transfer.

In Chapter 3, we described different ways to consolidate functional information associated with the genes in a cluster. We also introduced a new weighted-path based method to find the appropriate GO terms for a cluster of genes. Further, we presented the advantages and disadvantages of our approach in contrast to existing enrichment methods.

The results show that the knowledge derived from sequence similarity networks holds promise for aiding in gene function prediction. At the same time, we also discuss some shortcomings with our approach that will require future work to leverage the knowledge to its full potential.

4.2 Impact on Life Sciences

Advances in other fields have primarily shaped biology. While Physics has shaped our understanding of the molecular world through the advent of the advanced microscopy, and Chemists have made fundamental discoveries in the nature and structure of DNA. In the 21st century, Biology has been shaped through the explosion of information, often through -omics-based technology. This massive explosive of information has required the development of computational tools to analyze and simplify a large amount of data. Neighborly clusters and methods for annotation have three broad applications to the life science community:

- Neighborly can be used as a tool for gene function prediction and genome annotation. We have shown how Neighborly has been beneficial for function prediction, but this technique is readily applicable for genome annotation. We can also envision this technique applied to meta-genomics studies as our data suggests our predictions outperform those of CD-HIT at lower sequence similarity thresholds. CD-HIT based clusters are often used in large scale meta-genomics software [24]. It will be exciting to determine if Neighborly clusters outperform those by CD-HIT in this context.

- Neighborly can provide insight into how genes were inter-connected in regards to sequence similarity across the tree of life. One topic not covered in this thesis is the use of neighborly to gain novel taxonomic insight into sequence similar proteins. Neighborly clusters can also potentially provide us unique insight into how specific "genes," or Dawkin's "memes," have changes throughout time as well as their spread throughout the tree of life. One crucial question we hope to explore is how taxonomically diverse "house-keeping" genes are in comparison to virulence factors and could certain virulence factors be conserved across distinct branches of the tree of life.

- Neighborly can be used to transfer information to previously un-annotated genes within

a cluster. Another area not discussed in our thesis, but relevant to the life sciences, is the propagation of annotations amongst sequence similar cluster members. Though a sizable number of clusters contain many annotations, there is an even higher number of sparsely annotated or not annotated clusters. In the instances of sparse annotation, GO annotations can be propagated to a given cluster member. It will be necessary to determine the ideal sequence similarity threshold to allow for this process, as there is considerable variation at 50% sequence identity. Regardless, this backpropagation could allow for millions of genes in our data set to now have a form of annotation.

4.3 Future Work

The task of protein function prediction remains a multi-faceted open problem despite the various methods and their successes. We showed that function transfer using BLAST/DIAMOND based alignments, which serve as benchmarks for many research works, can provide excellent results if the reference database is extensive. However, we suspect that this could be an artifact of identical sequences between the test set and the reference database. Protein databases are usually a result of information curation from various sources and could contain redundant sequences for different identifiers. Analysis of redundant sequences and pre-processing steps to remove any such instances can help validate the accuracy of our results.

We compared our method with a more sophisticated deep learning method—DeepGoPlus, but it was trained on a smaller CAFA 3 dataset. It would be interesting to see if the results for methods with excellent performance on smaller datasets, translate well to more massive datasets. However, we are not sure if these methods can be trained on such an extensive dataset in a reasonable amount of time. We repeatedly came across this notion in our

studies, as many common methods do not scale well for millions of clusters of genes. Given the explosion of sequence information, a targeted approach to handle such large amounts of data is warranted.

The journey to characterize and annotate Neighborly Clusters has provided us novel insight into the challenges in building and scaling computational methods. Our future efforts are not limited to one particular aspect of Neighborly as we see opportunities to improve all aspects of the technique. One particularly important area of improvement is the ability to generate the sequence similarity networks used for this study in a significantly shorter period while utilizing fewer resources.

Further, our current methods for cluster generation could be expanded as we saw substantial variability between our approach and CD-HIT. Another critical aspect of future work is the inclusion of alternate information sources outside the gene ontology for annotation. These sources include PantherDB, KEGG pathways, but could be expanded to transcriptomics and phenotypic data. We have extensively discussed the need to improve upon a method for cluster annotation in Chapter 3. Applying methods to scale for an alternate form of annotations will be needed as well. Given the desire to make Neighborly broadly applicable to the life sciences community, we would like to develop aspects of a graphical user interface for general use. Our current public offering of Neighborly provides a simple skeleton for cluster analysis via Diamond based querying as well as looking up specific genes and clusters, but this method could be expanded upon considerably into the future. Facilitating ease of use could help propel use and has been the case for several bioinformatics packages.

Bibliography

[1] Guide to go evidence codes. URL http://geneontology.org/docs/guide-go-evidence-codes/.

[2] Adrian Alexa and Jörg Rahnenführer. Gene set enrichment analysis with topgo. *Bioconductor Improv*, 27, 2009.

[3] Adrian Alexa, Jörg Rahnenführer, and Thomas Lengauer. Improved scoring of functional groups from gene expression data by decorrelating go graph structure. *Bioinformatics*, 22(13):1600–1607, 2006.

[4] Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, 1990.

[5] Stephen F Altschul, Thomas L Madden, Alejandro A Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic acids research*, 25(17):3389–3402, 1997.

[6] Rolf Apweiler, Amos Bairoch, Cathy H Wu, Winona C Barker, Brigitte Boeckmann, Serenella Ferro, Elisabeth Gasteiger, Hongzhan Huang, Rodrigo Lopez, Michele Magrane, et al. Uniprot: the universal protein knowledgebase. *Nucleic acids research*, 32 (suppl_1):D115–D119, 2004.

[7] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

[8] Holly J Atkinson, John H Morris, Thomas E Ferrin, and Patricia C Babbitt. Using sequence similarity networks for visualization of relationships across diverse protein superfamilies. *PloS one*, 4(2), 2009.

[9] Albert-László Barabási, Natali Gulbahce, and Joseph Loscalzo. Network medicine: a network-based approach to human disease. *Nature reviews genetics*, 12(1):56–68, 2011.

[10] Daniel Barrell, Emily Dimmer, Rachael P Huntley, David Binns, Claire O'Donovan, and Rolf Apweiler. The goa database in 2009—an integrated gene ontology annotation resource. *Nucleic acids research*, 37(suppl_1):D396–D403, 2009.

[11] Sebastian Bauer, Julien Gagneur, and Peter N Robinson. Going bayesian: model-based gene set analysis of genome-scale data. *Nucleic acids research*, 38(11):3523–3532, 2010.

[12] Brigitte Boeckmann, Amos Bairoch, Rolf Apweiler, Marie-Claude Blatter, Anne Estreicher, Elisabeth Gasteiger, Maria J Martin, Karine Michoud, Claire O'Donovan, Isabelle Phan, et al. The swiss-prot protein knowledgebase and its supplement trembl in 2003. *Nucleic acids research*, 31(1):365–370, 2003.

[13] Benjamin Buchfink, Chao Xie, and Daniel H Huson. Fast and sensitive protein alignment using diamond. *Nature methods*, 12(1):59, 2015.

[14] CZ Cai, WL Wang, LZ Sun, and YZ Chen. Protein function classification via support vector machine approach. *Mathematical biosciences*, 185(2):111–122, 2003.

[15] Wyatt T Clark and Predrag Radivojac. Analysis of protein function and its prediction from amino acid sequence. *Proteins: Structure, Function, and Bioinformatics*, 79(7): 2086–2096, 2011.

[16] Wyatt T Clark and Predrag Radivojac. Information-theoretic evaluation of predicted ontological annotations. *Bioinformatics*, 29(13):i53–i61, 2013.

[17] Ana Conesa, Stefan Götz, Juan Miguel García-Gómez, Javier Terol, Manuel Talón, and Montserrat Robles. Blast2go: a universal tool for annotation, visualization and analysis in functional genomics research. *Bioinformatics*, 21(18):3674–3676, 2005.

[18] Gene Ontology Consortium. Expansion of the gene ontology knowledgebase and resources. *Nucleic acids research*, 45(D1):D331–D338, 2017.

[19] Gene Ontology Consortium. The gene ontology resource: 20 years and still going strong. *Nucleic acids research*, 47(D1):D330–D338, 2019.

[20] Daolong Dou, Shiv D Kale, Xia Wang, Rays HY Jiang, Nathan A Bruce, Felipe D Arredondo, Xuemin Zhang, and Brett M Tyler. Rxlr-mediated entry of phytophthora sojae effector avr1b into soybean cells does not require pathogen-encoded machinery. *The Plant Cell*, 20(7):1930–1947, 2008.

[21] Robert C Edgar. Search and clustering orders of magnitude faster than blast. *Bioinformatics*, 26(19):2460–2461, 2010.

[22] A. J. Enright, S. Van Dongen, and C. A. Ouzounis. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Research*, 30(7):1575–1584, 04 2002. ISSN 0305-1048. doi: 10.1093/nar/30.7.1575. URL https://doi.org/10.1093/nar/30.7.1575.

[23] Anton J Enright, Stijn Van Dongen, and Christos A Ouzounis. An efficient algorithm for large-scale detection of protein families. *Nucleic acids research*, 30(7):1575–1584, 2002.

[24] Eric A Franzosa, Lauren J McIver, Gholamali Rahnavard, Luke R Thompson, Melanie Schirmer, George Weingart, Karen Schwarzberg Lipson, Rob Knight, J Gregory Ca-

poraso, Nicola Segata, et al. Species-level functional profiling of metagenomes and metatranscriptomes. *Nature methods*, 15(11):962–968, 2018.

[25] Limin Fu, Beifang Niu, Zhengwei Zhu, Sitao Wu, and Weizhong Li. Cd-hit: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, 28(23):3150–3152, 2012.

[26] Ludwig Geistlinger, Gergely Csaba, Mara Santarelli, Marcel Ramos, Lucas Schiffer, Charity Law, Nitesh Turaga, Sean Davis, Vincent Carey, Martin Morgan, et al. Towards a gold standard for benchmarking gene set enrichment analysis. *BioRxiv*, page 674267, 2019.

[27] Steffen Grossmann, Sebastian Bauer, Peter N Robinson, and Martin Vingron. Improved detection of overrepresentation of gene-ontology annotations with parent–child analysis. *Bioinformatics*, 23(22):3024–3031, 2007.

[28] Peter F Hallin, Tim T Binnewies, and David W Ussery. The genome blastatlas—a genewiz extension for visualization of whole-genome homology. *Molecular BioSystems*, 4(5):363–371, 2008.

[29] Steven Henikoff and Jorja G Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22):10915–10919, 1992.

[30] Da Wei Huang, Brad T Sherman, and Richard A Lempicki. Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic acids research*, 37(1):1–13, 2009.

[31] Daniel H Huson and Chao Xie. A poor man's blastx—high-throughput metagenomic protein database search using pauda. *Bioinformatics*, 30(1):38–39, 2014.

[32] Paul Jaccard. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bull Soc Vaudoise Sci Nat*, 37:547–579, 1901.

[33] Yuxiang Jiang, Tal Ronnen Oron, Wyatt T Clark, Asma R Bankapur, Daniel D'Andrea, Rosalba Lepore, Christopher S Funk, Indika Kahanda, Karin M Verspoor, Asa Ben-Hur, et al. An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome biology*, 17(1):184, 2016.

[34] Shiv D Kale, Biao Gu, Daniel GS Capelluto, Daolong Dou, Emily Feldman, Amanda Rumore, Felipe D Arredondo, Regina Hanlon, Isabelle Fudal, Thierry Rouxel, et al. External lipid pi3p mediates entry of eukaryotic pathogen effectors into plant and animal host cells. *Cell*, 142(2):284–295, 2010.

[35] Shiv D Kale, Tariq Ayubi, Dawoon Chung, Nuria Tubau-Juni, Andrew Leber, Ha X Dang, Saikumar Karyala, Raquel Hontecillas, Christopher B Lawrence, Robert A Cramer, and Josep Bassaganya-Riera. Modulation of Immune Signaling and Metabolism Highlights Host and Fungal Transcriptional Responses in Mouse Models of Invasive Pulmonary Aspergillosis. *Scientific Reports*, 7(1):17096, 2017. ISSN 2045-2322. doi: 10.1038/s41598-017-17000-1. URL https://doi.org/10.1038/s41598-017-17000-1.

[36] W James Kent. Blat—the blast-like alignment tool. *Genome research*, 12(4):656–664, 2002.

[37] Maria Kissa, George Tsatsaronis, and Michael Schroeder. Prediction of drug gene associations via ontological profile similarity with application to drug repositioning. *Methods*, 74:71–82, 2015.

[38] DV Klopfenstein, Liangsheng Zhang, Brent S Pedersen, Fidel Ramírez, Alex Warwick Vesztrocy, Aurélien Naldi, Christopher J Mungall, Jeffrey M Yunes, Olga Botvinnik,

Mark Weigel, et al. Goatools: A python library for gene ontology analyses. *Scientific reports*, 8(1):1–17, 2018.

[39] Maxat Kulmanov and Robert Hoehndorf. Deepgoplus: improved protein function prediction from sequence. *Bioinformatics*, 36(2):422–429, 2020.

[40] Li Li, Christian J Stoeckert, and David S Roos. Orthomcl: identification of ortholog groups for eukaryotic genomes. *Genome research*, 13(9):2178–2189, 2003.

[41] Weizhong Li, Lukasz Jaroszewski, and Adam Godzik. Clustering of highly homologous sequences to reduce the size of large protein databases. *Bioinformatics*, 17(3):282–283, 2001.

[42] Dekang Lin et al. An information-theoretic definition of similarity. In *Icml*, volume 98, pages 296–304, 1998.

[43] Jun S Liu, Andrew F Neuwald, and Charles E Lawrence. Bayesian models for multiple local sequence alignment and gibbs sampling strategies. *Journal of the American Statistical Association*, 90(432):1156–1170, 1995.

[44] Yong Lu, Roni Rosenfeld, Itamar Simon, Gerard J Nau, and Ziv Bar-Joseph. A probabilistic generative model for go enrichment analysis. *Nucleic Acids Research*, 36(17): e109–e109, 2008.

[45] Steven Maere, Karel Heymans, and Martin Kuiper. Bingo: a cytoscape plugin to assess overrepresentation of gene ontology categories in biological networks. *Bioinformatics*, 21(16):3448–3449, 2005.

[46] John H McDonald. *Handbook of biological statistics*, volume 2. sparky house publishing Baltimore, MD, 2009.

[47] Tobias Müller, Rainer Spang, and Martin Vingron. Estimating amino acid substitution models: a comparison of dayhoff's estimator, the resolvent approach and a maximum likelihood method. *Molecular biology and evolution*, 19(1):8–13, 2002.

[48] Saul B Needleman and Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.

[49] Pauline C Ng and Steven Henikoff. Predicting the effects of amino acid substitutions on protein function. *Annu. Rev. Genomics Hum. Genet.*, 7:61–80, 2006.

[50] Ruben Nogales-Cadenas, Pedro Carmona-Saez, Miguel Vazquez, Cesar Vicente, Xiaoyuan Yang, Francisco Tirado, Jose María Carazo, and Alberto Pascual-Montano. Genecodis: interpreting gene lists through enrichment analysis and integration of diverse biological information. *Nucleic acids research*, 37(suppl_2):W317–W322, 2009.

[51] William R Pearson and David J Lipman. Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences*, 85(8):2444–2448, 1988.

[52] Predrag Radivojac, Wyatt T Clark, Tal Ronnen Oron, Alexandra M Schnoes, Tobias Wittkop, Artem Sokolov, Kiley Graim, Christopher Funk, Karin Verspoor, Asa Ben-Hur, et al. A large-scale evaluation of computational protein function prediction. *Nature methods*, 10(3):221–227, 2013.

[53] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*, 1995.

[54] Philip Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of artificial intelligence research*, 11:95–130, 1999.

[55] Isabelle Rivals, Léon Personnaz, Lieng Taing, and Marie-Claude Potier. Enrichment or depletion of a go category within a class of genes: which test? *Bioinformatics*, 23(4): 401–407, 2007.

[56] Roded Sharan, Igor Ulitsky, and Ron Shamir. Network-based prediction of protein function. *Molecular systems biology*, 3(1), 2007.

[57] Brad T Sherman, Qina Tan, Jack R Collins, W Gregory Alvord, Jean Roayaei, Robert Stephens, Michael W Baseler, H Clifford Lane, Richard A Lempicki, et al. The david gene functional classification tool: a novel biological module-centric algorithm to functionally analyze large gene lists. *Genome biology*, 8(9):R183, 2007.

[58] Temple F Smith, Michael S Waterman, et al. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, 1981.

[59] Humira Sonah, Rupesh K Deshmukh, and Richard R Bélanger. Computational prediction of effector proteins in fungi: opportunities and challenges. *Frontiers in plant science*, 7:126, 2016.

[60] Robert V Stahelin. Lipid binding domains: more than simple lipid effectors. *Journal of lipid research*, 50(Supplement):S299–S304, 2009.

[61] Patrik L Ståhl and Joakim Lundeberg. Toward the single-hour high-quality genome. *Annual review of biochemistry*, 81:359–378, 2012.

[62] Martin Steinegger and Johannes Söding. Linclust: clustering billions of protein sequences per day on a single server. *bioRxiv*, page 104034, 2017.

[63] Martin Steinegger and Johannes Söding. Clustering huge protein sequence sets in linear time. *Nature communications*, 9(1):1–8, 2018.

[64] Aravind Subramanian, Pablo Tamayo, Vamsi K Mootha, Sayan Mukherjee, Benjamin L Ebert, Michael A Gillette, Amanda Paulovich, Scott L Pomeroy, Todd R Golub, Eric S Lander, et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, 102(43):15545–15550, 2005.

[65] Fadi A Thabtah, Peter Cowling, and Yonghong Peng. Mmac: A new multi-class, multi-label associative classification approach. In *Fourth IEEE International Conference on Data Mining (ICDM'04)*, pages 217–224. IEEE, 2004.

[66] Hannah Tipney and Lawrence Hunter. An introduction to effective use of enrichment analysis software. *Human genomics*, 4(3):202, 2010.

[67] Ricardo ZN Vêncio, Tie Koide, Suely L Gomes, and Carlos A de B Pereira. Baygo: Bayesian analysis of ontology term enrichment in microarray data. *BMC bioinformatics*, 7(1):86, 2006.

[68] Zheng Wang, Chenguang Zhao, Yiheng Wang, Zheng Sun, and Nan Wang. Panda: Protein function prediction using domain architecture and affinity propagation. *Scientific reports*, 8(1):1–10, 2018.

[69] Rainer Winnenburg, Thomas K Baldwin, Martin Urban, Chris Rawlings, Jacob Köhler, and Kim E Hammond-Kosack. Phi-base: a new database for pathogen host interactions. *Nucleic acids research*, 34(suppl_1):D459–D464, 2006.

[70] Ying Wu, Xianfeng Ma, Zhiyong Pan, Shiv D Kale, Yi Song, Harlan King, Qiong Zhang, Christian Presley, Xiuxin Deng, Cheng-I Wei, and Shunyuan Xiao. Comparative genome analyses reveal sequence features reflecting distinct modes of host-adaptation between dicot and monocot powdery mildew. *BMC Genomics*, 19(1):705,

2018. ISSN 1471-2164. doi: 10.1186/s12864-018-5069-z. URL https://doi.org/10.1186/s12864-018-5069-z.

[71] Ronghui You, Zihan Zhang, Yi Xiong, Fengzhu Sun, Hiroshi Mamitsuka, and Shanfeng Zhu. Golabeler: improving sequence-based large-scale protein function prediction by learning to rank. *Bioinformatics*, 34(14):2465–2473, 2018.

[72] Rong Zeng, Shigang Gao, Lihui Xu, Xin Liu, and Fuming Dai. Prediction of pathogenesis-related secreted proteins from stemphylium lycopersici. *BMC microbiology*, 18(1):191, 2018.

[73] Xiangxiang Zeng, Xuan Zhang, and Quan Zou. Integrative approaches for predicting microrna function and prioritizing disease-related microrna using biological interaction networks. *Briefings in bioinformatics*, 17(2):193–203, 2016.

[74] Jingpu Zhang, Zuping Zhang, Zhigang Chen, and Lei Deng. Integrating multiple heterogeneous networks for novel lncrna-disease association inference. *IEEE/ACM transactions on computational biology and bioinformatics*, 16(2):396–406, 2017.

[75] Zuping Zhang, Jingpu Zhang, Chao Fan, Yongjun Tang, and Lei Deng. Katzlgo: large-scale prediction of lncrna functions by using the katz measure based on multiple networks. *IEEE/ACM transactions on computational biology and bioinformatics*, 16(2): 407–416, 2017.

[76] Yongan Zhao, Haixu Tang, and Yuzhen Ye. Rapsearch2: a fast and memory-efficient protein similarity search tool for next-generation sequencing data. *Bioinformatics*, 28 (1):125–126, 2012.

[77] Lu-Lu Zheng, Yi-Xue Li, Juan Ding, Xiao-Kui Guo, Kai-Yan Feng, Ya-Jun Wang, Le-

Le Hu, Yu-Dong Cai, Pei Hao, and Kuo-Chen Chou. A comparison of computational methods for identifying virulence factors. *PLoS One*, 7(8), 2012.

[78] Wei Zheng, Chengxin Zhang, Eric W Bell, and Yang Zhang. I-tasser gateway: A protein structure and function prediction server powered by xsede. *Future Generation Computer Systems*, 99:73–85, 2019.

[79] Naihui Zhou, Yuxiang Jiang, Timothy R Bergquist, Alexandra J Lee, Balint Z Kacsoh, Alex W Crocker, Kimberley A Lewis, George Georghiou, Huy N Nguyen, Md Nafiz Hamid, et al. The cafa challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome biology*, 20(1):1–23, 2019.

Appendices

Appendix A

Supplementary Figures

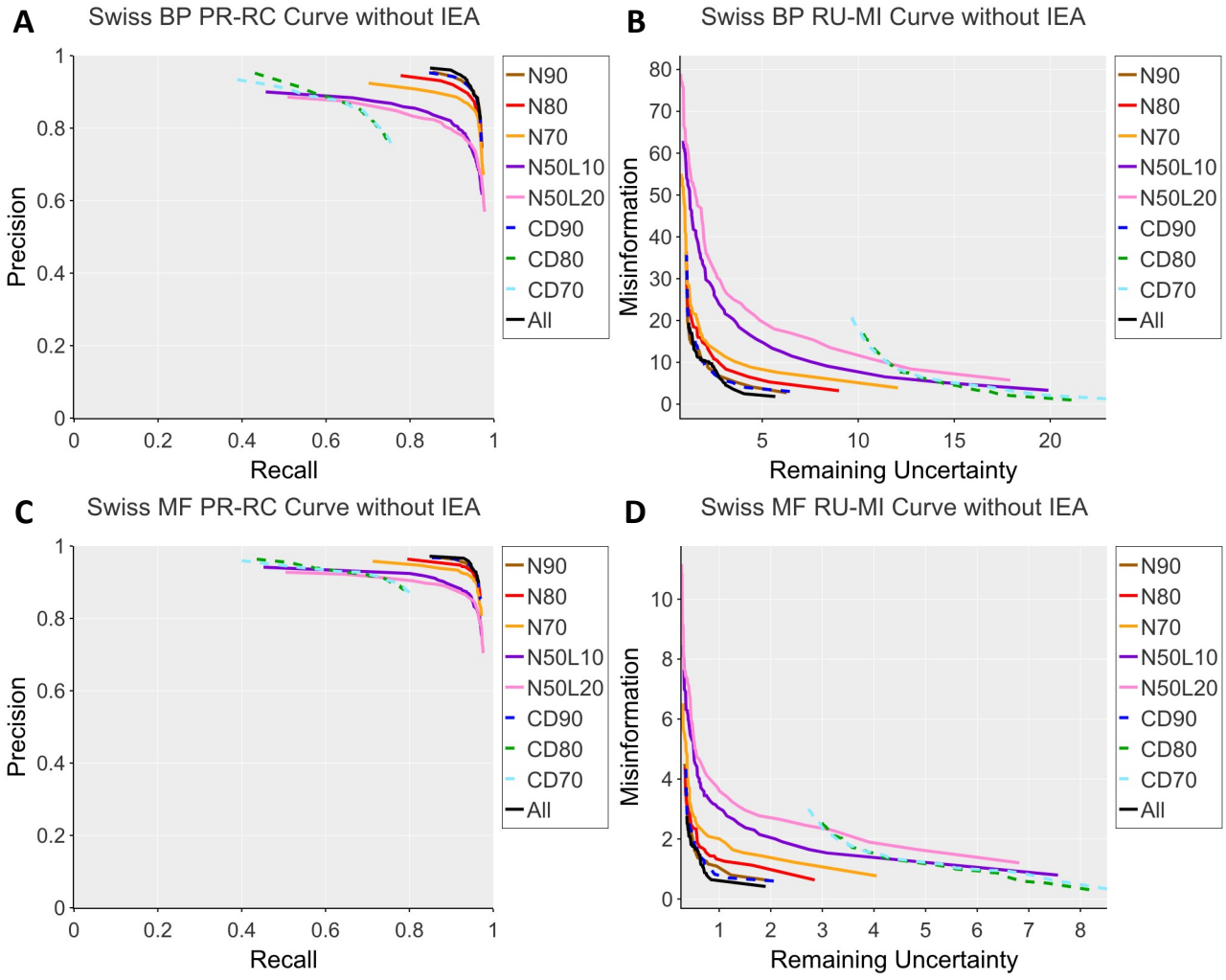

Figure A.1: Precision-Recall Curves and Misinformation-Remaining Uncertainty Curves for Predictions on the temporal holdout SwissProt test set without IEA annotations.

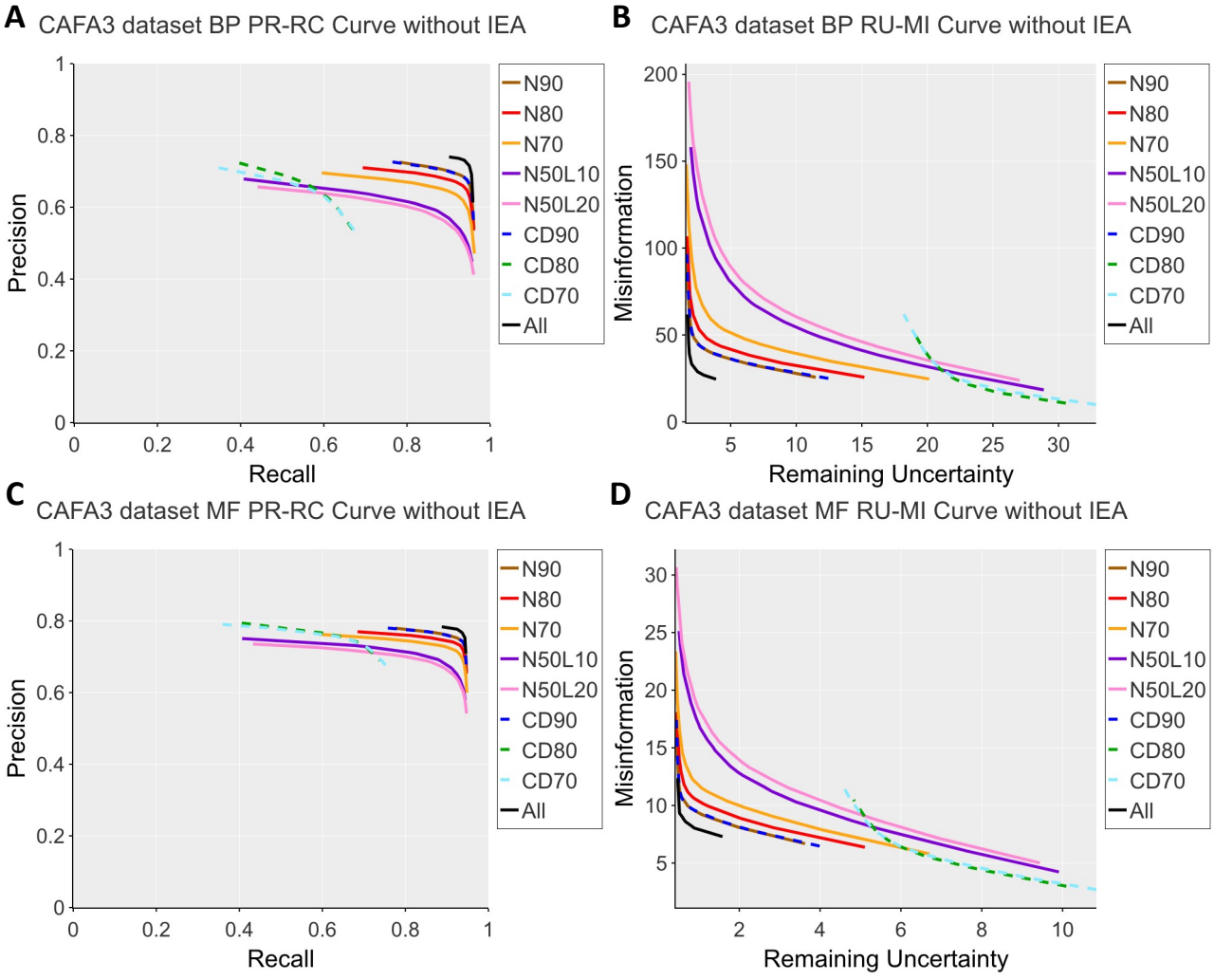

Figure A.2: Precision-Recall Curves and Misinformation-Remaining Uncertainty Curves for Predictions on the CAFA 3 data set without IEA annotations.

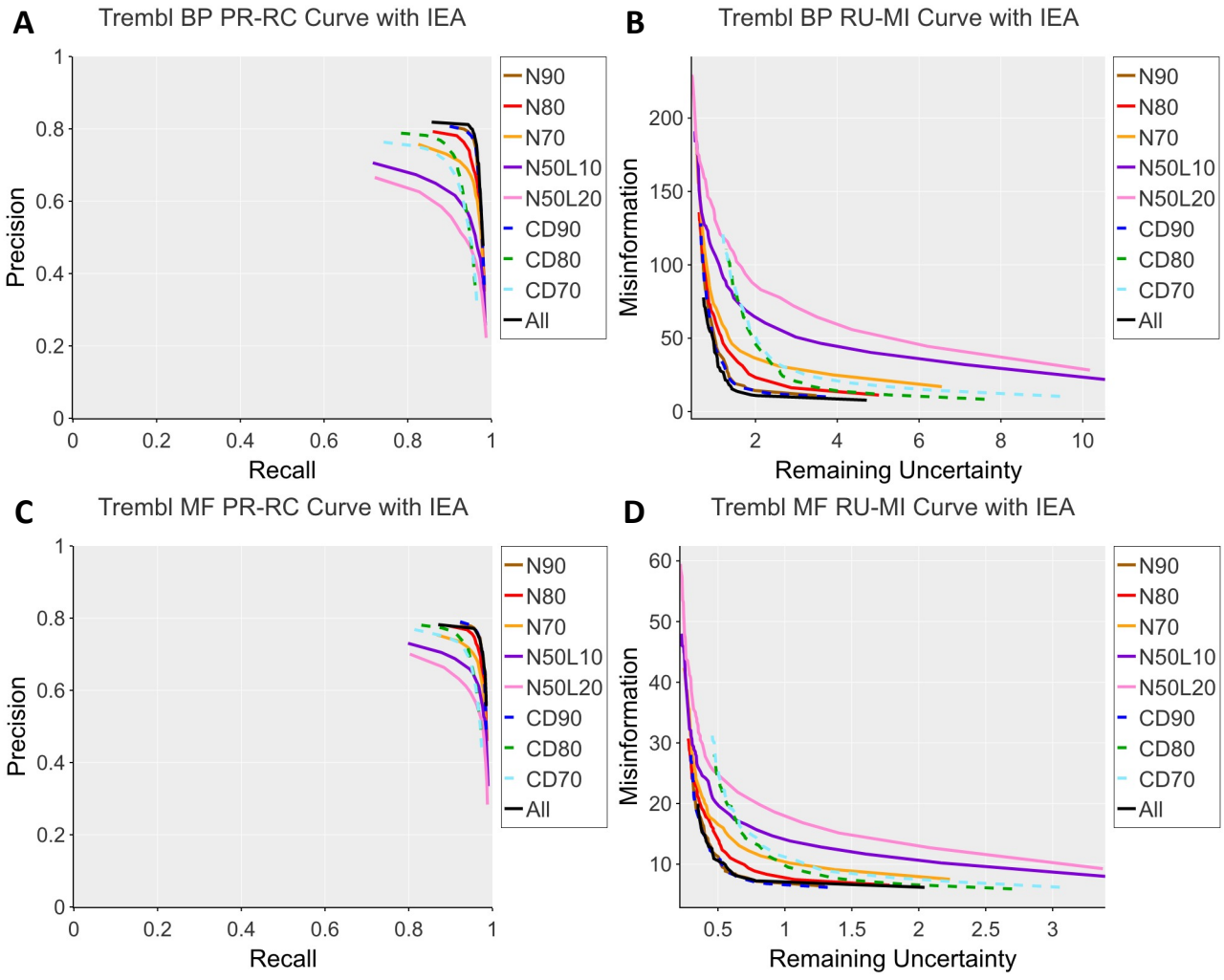

Figure A.3: Precision-Recall Curves and Misinformation-Remaining Uncertainty Curves for Predictions on the temporal holdout TrEMBL test set with IEA annotations.

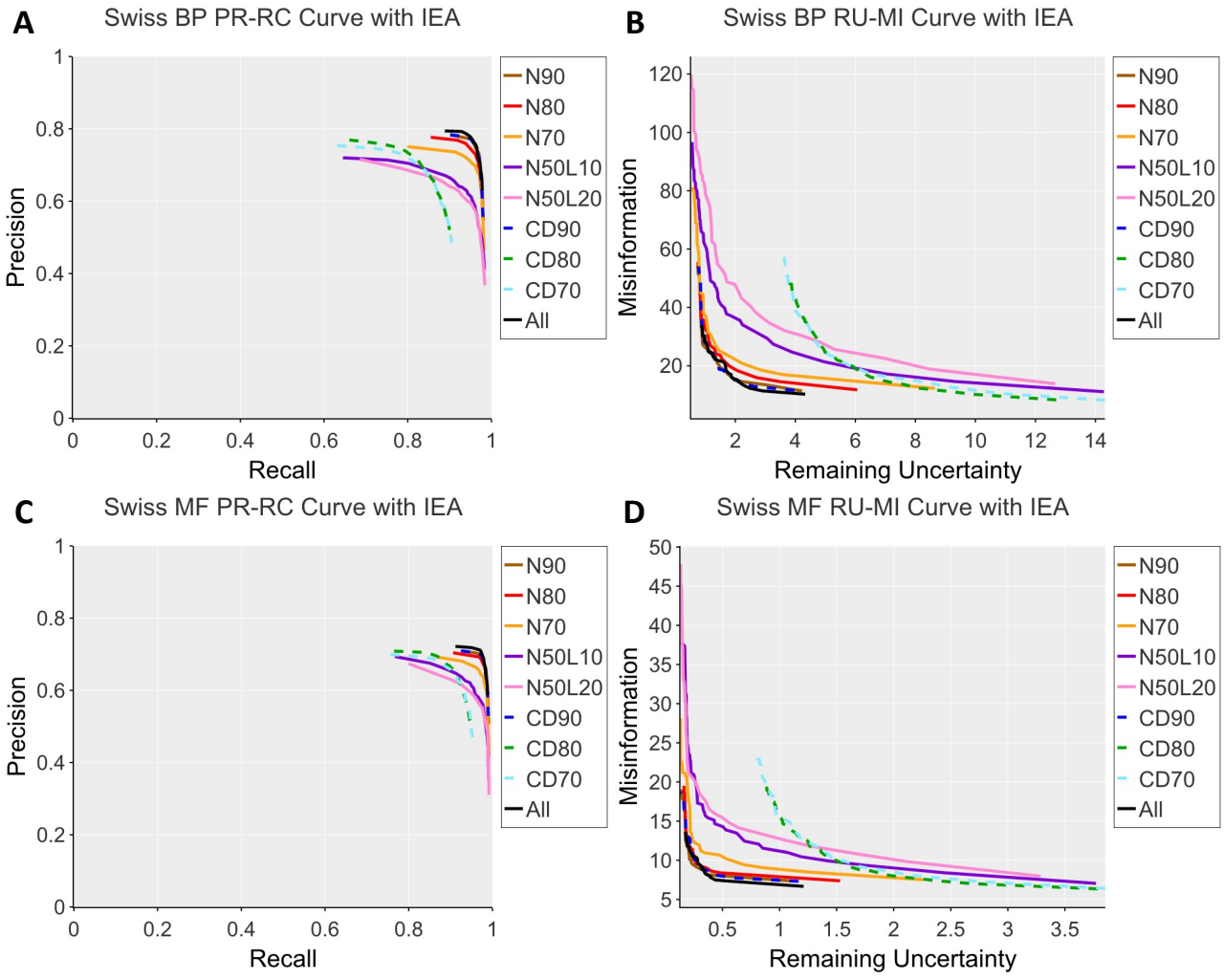

Figure A.4: Precision-Recall Curves and Misinformation-Remaining Uncertainty Curves for Predictions on the temporal holdout SwissProt test set with IEA annotations.

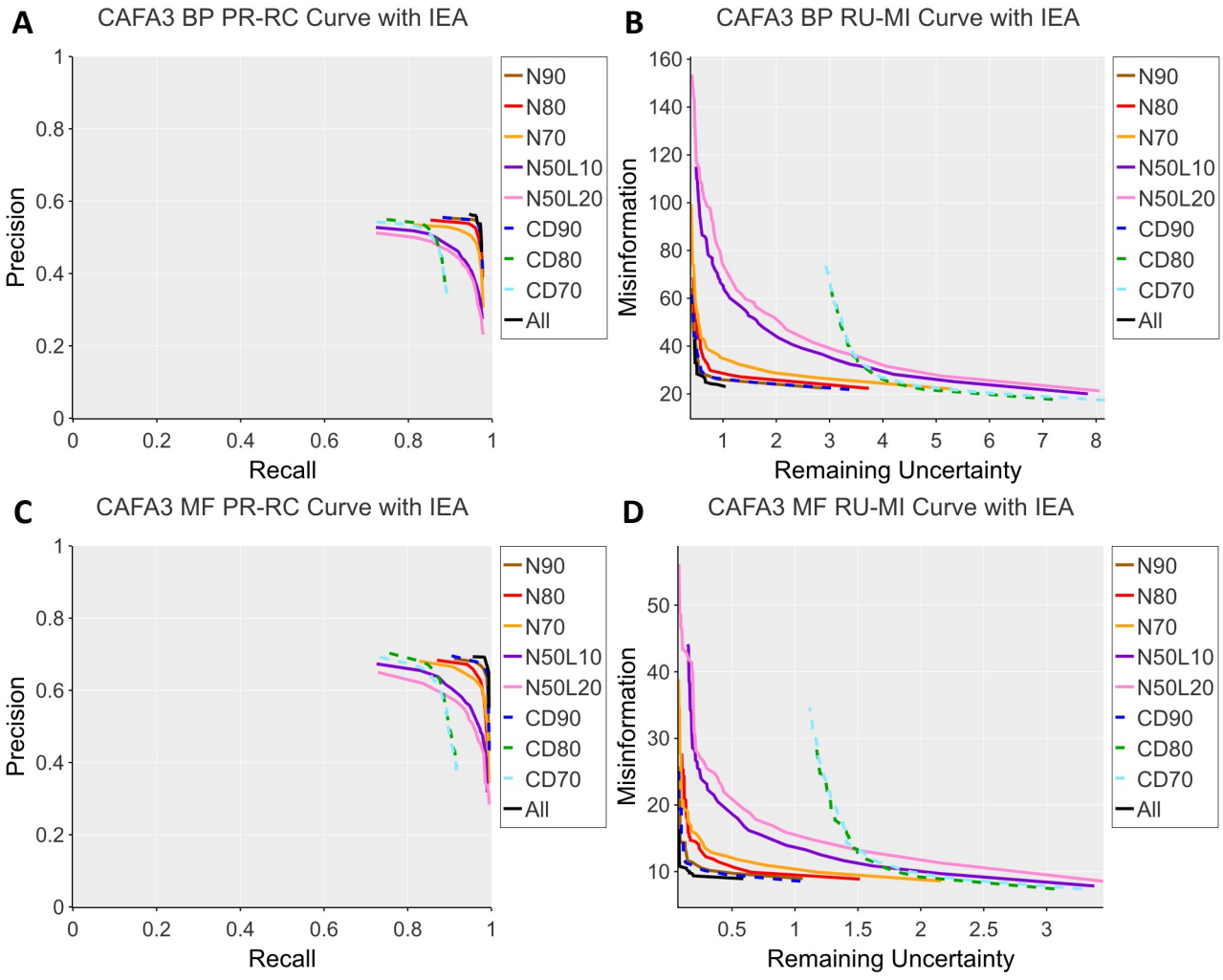

Figure A.5: Precision-Recall Curves and Misinformation-Remaining Uncertainty Curves for Predictions on the CAFA 3 test set with IEA annotations.

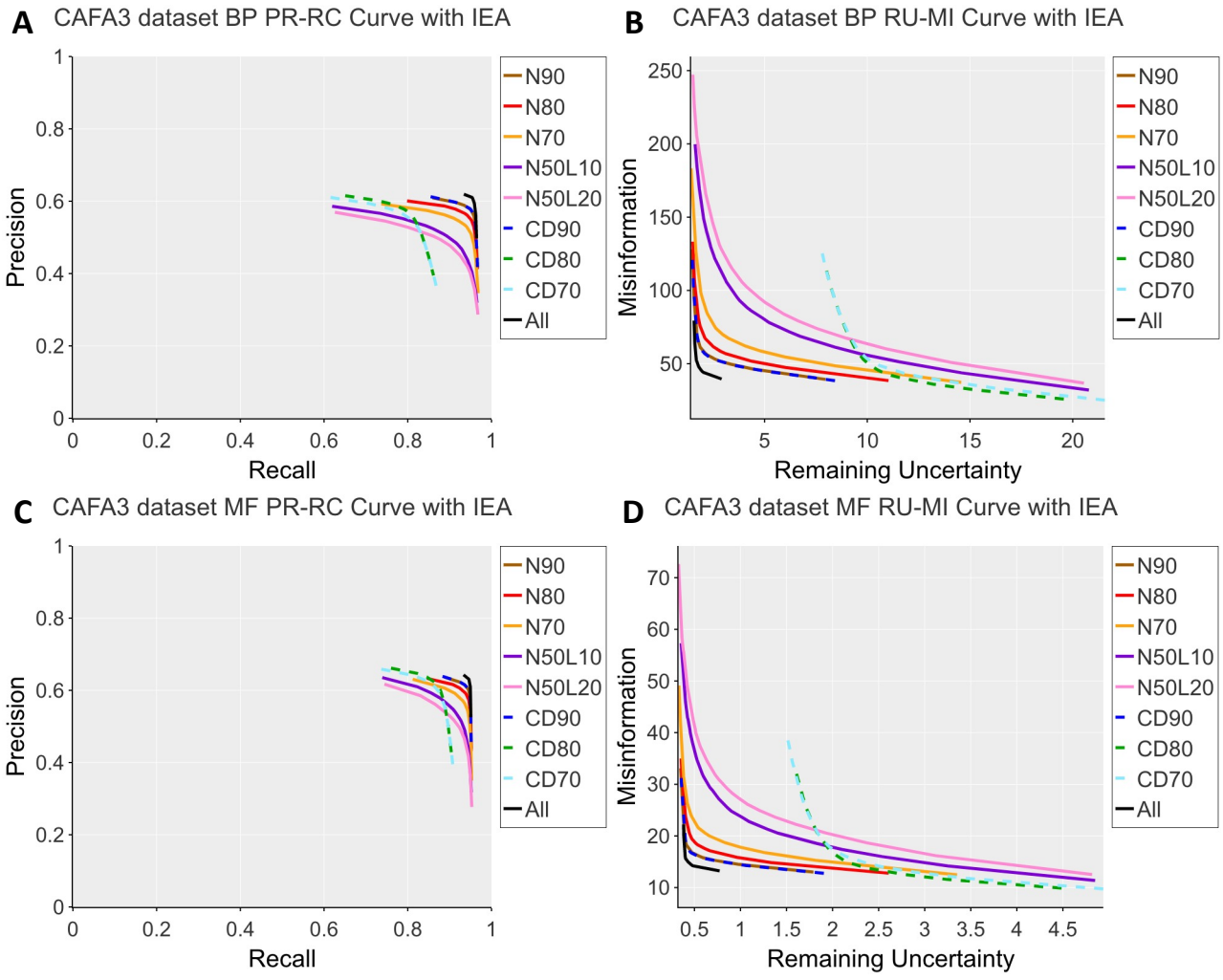

Figure A.6: Precision-Recall Curves and Misinformation-Remaining Uncertainty Curves for Predictions on the CAFA 3 data set with IEA annotations.

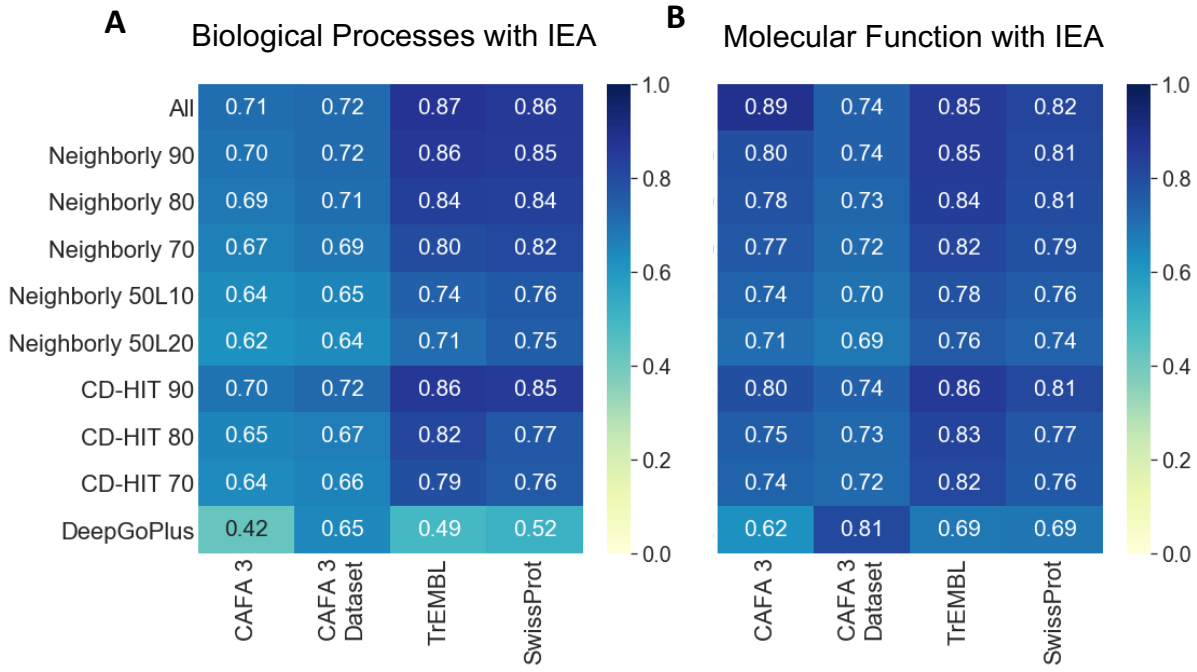

Figure A.7: F_{max} values for Biological Process and Molecular Function terms with IEA annotations across four datasets

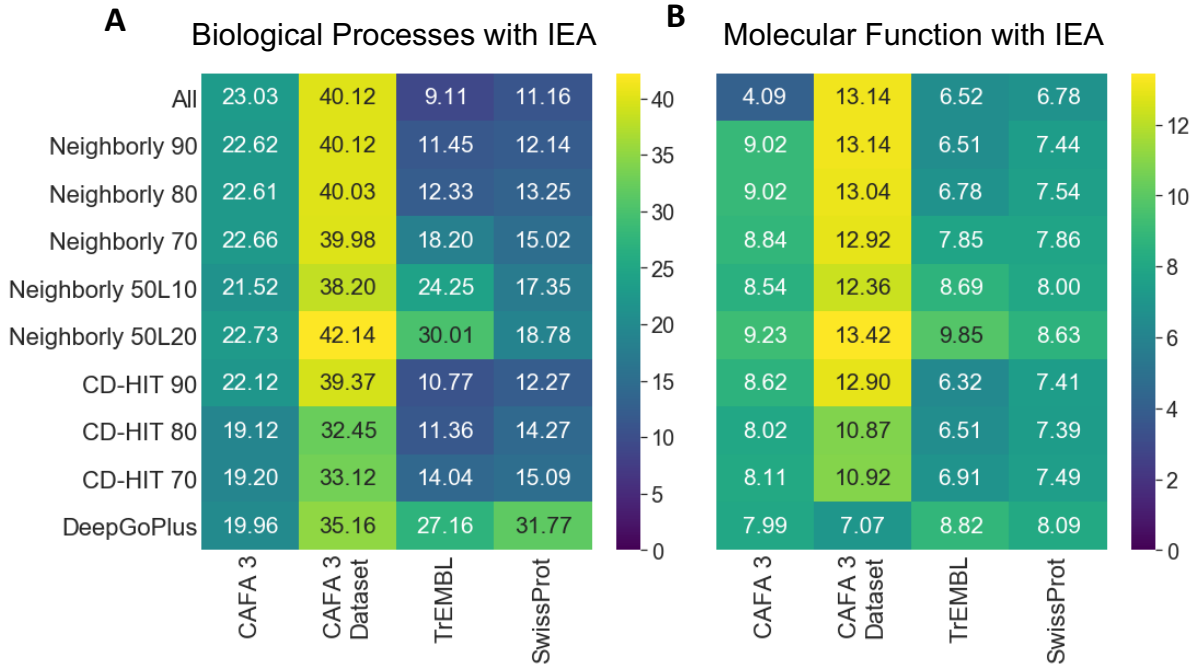

Figure A.8: S_{min} values for Biological Process and Molecular Function terms with IEA annotations across four datasets

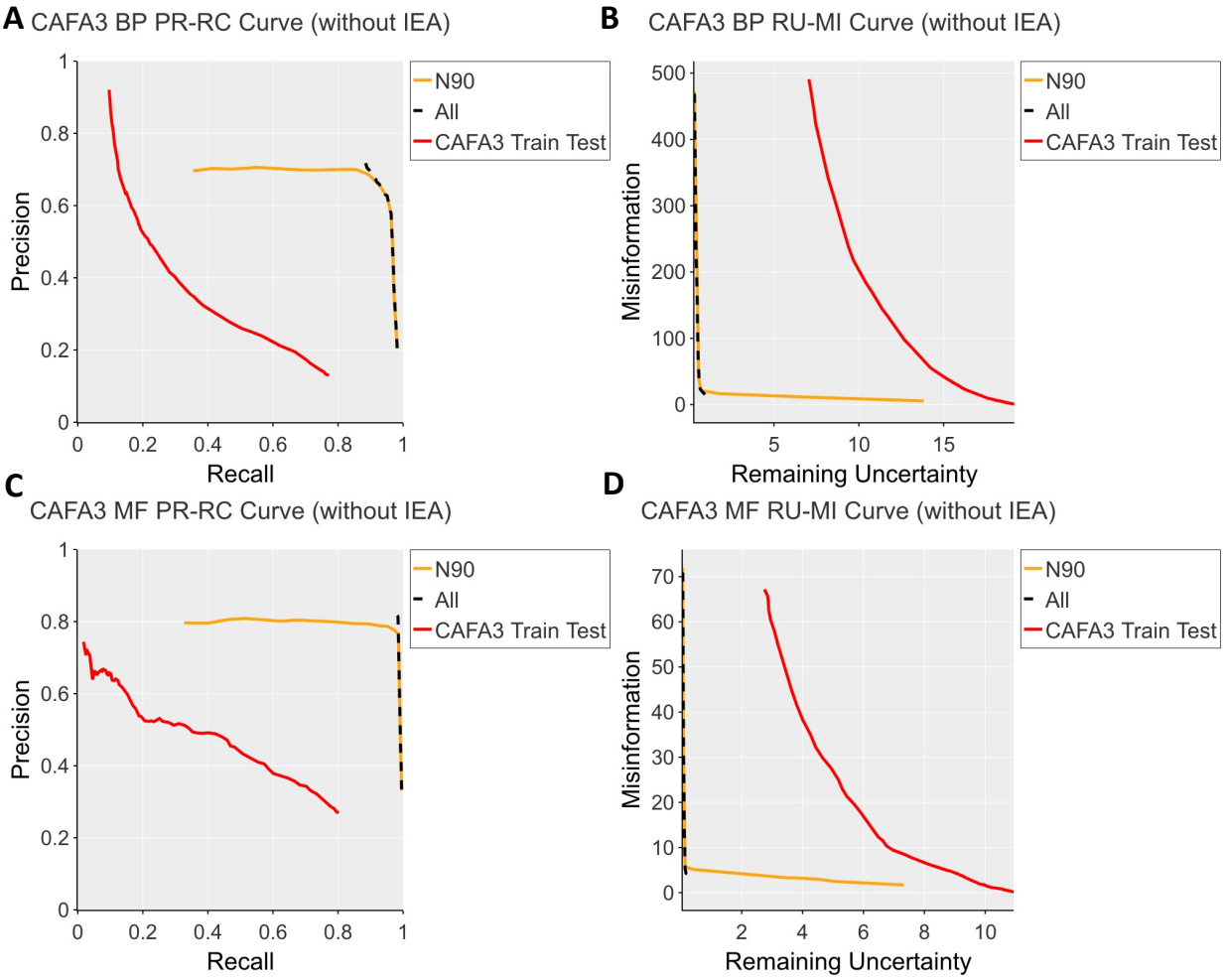

Figure A.9: Precision-Recall Curves and Misinformation-Remaining Uncertainty Curves for threshold-based evaluation on the CAFA 3 test set without IEA annotations.

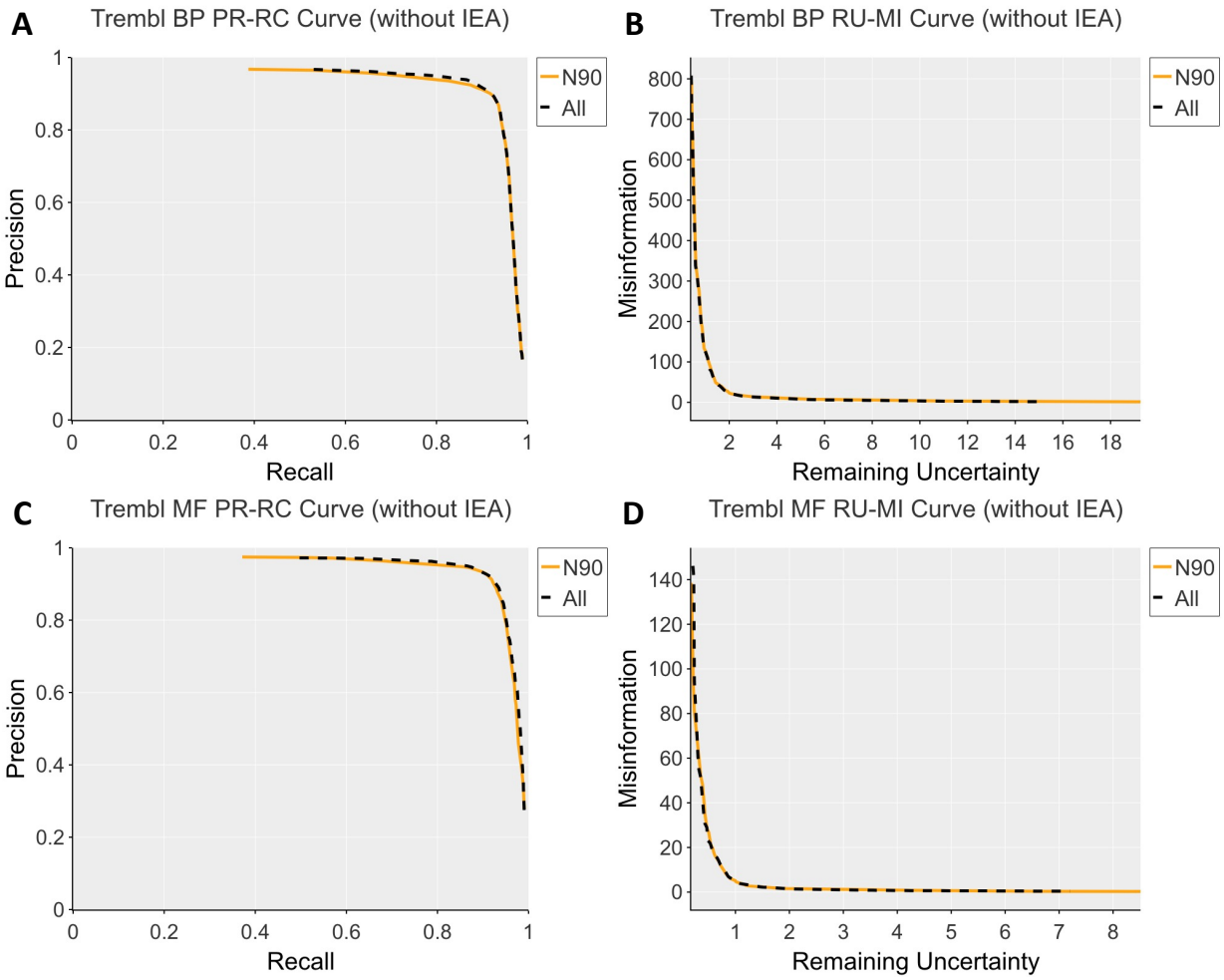

Figure A.10: Precision-Recall Curves and Misinformation-Remaining Uncertainty Curves for threshold-based evaluation on the TrEMBL test set without IEA annotations.

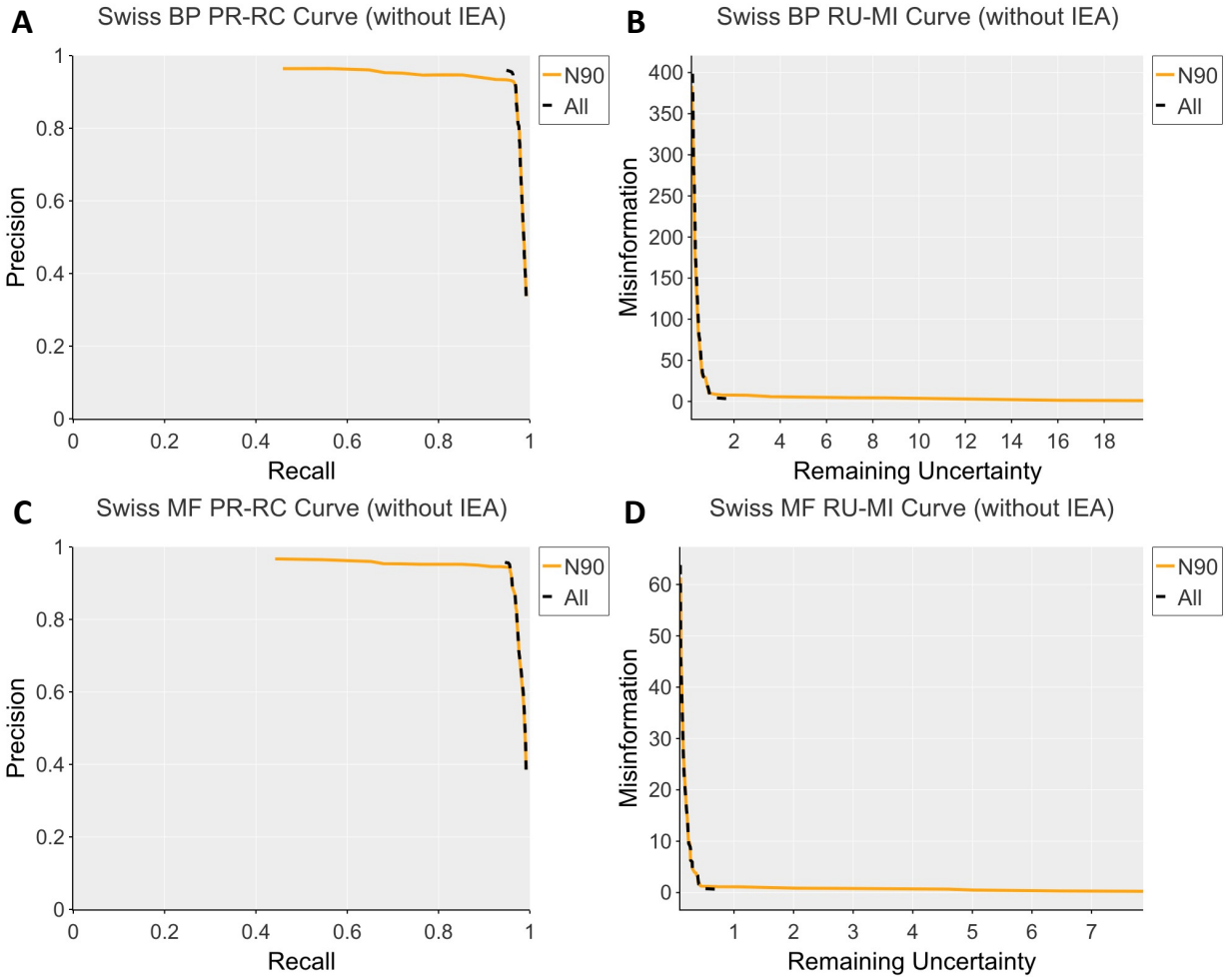

Figure A.11: Precision-Recall Curves and Misinformation-Remaining Uncertainty Curves for threshold-based evaluation on the SwissProt test set without IEA annotations.

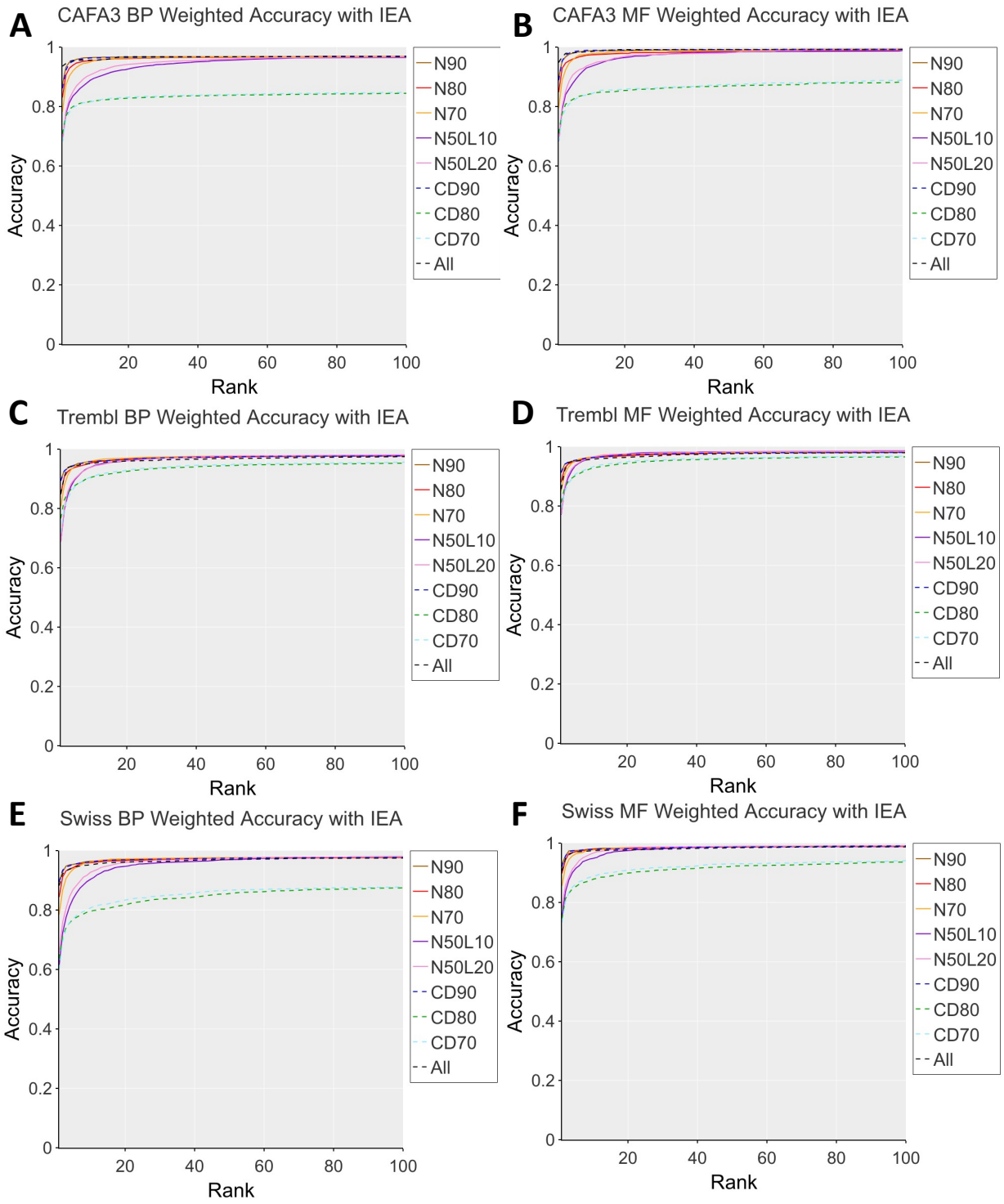

Figure A.12: Weighted Accuracy Plots for biological process and molecular function terms with IEA annotations

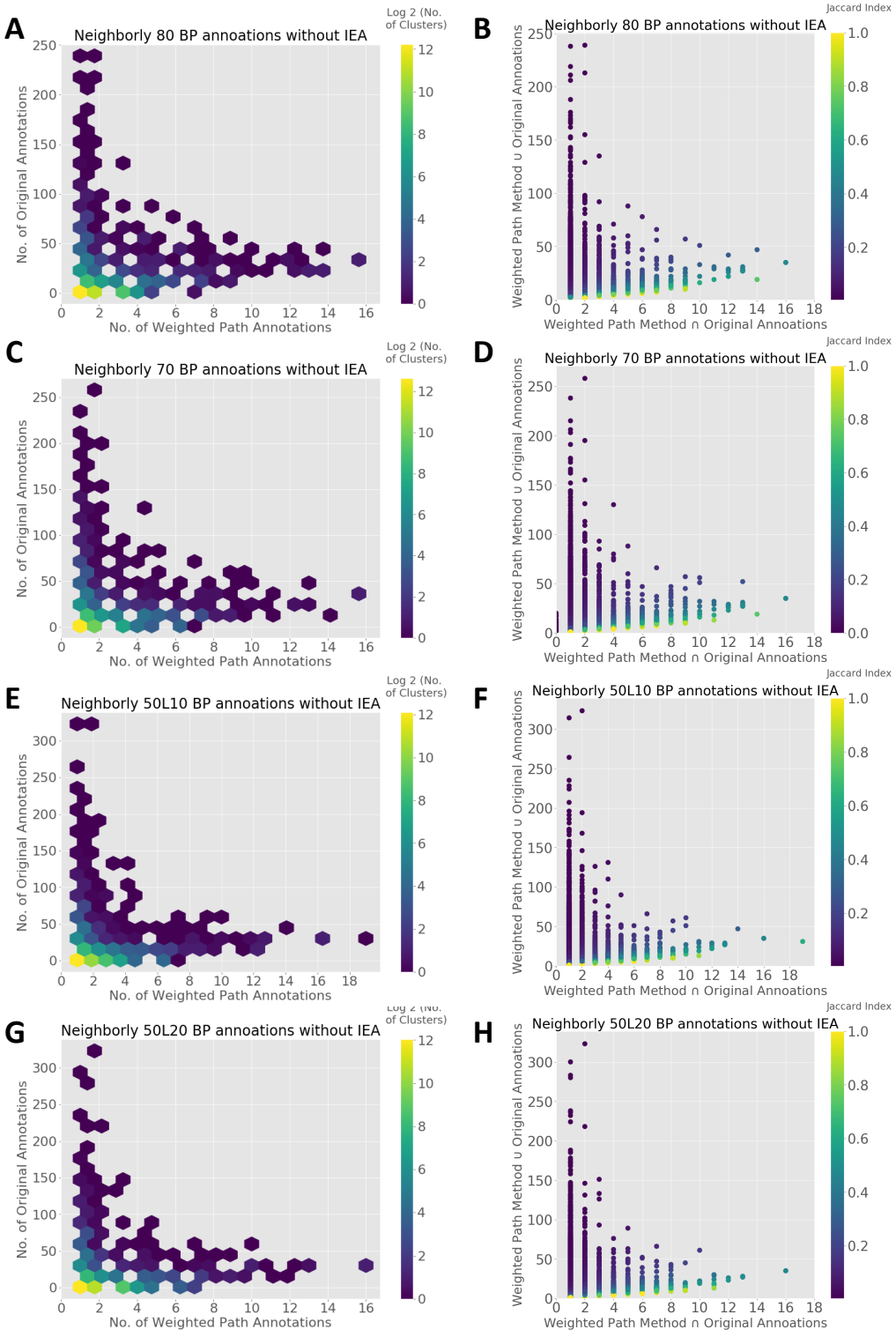

Figure A.13: Comparison of Weighted Path Annotations and Original Annotations for Biological Process annotations on Neighborly 80, 70, 50L10 and 50L20 networks

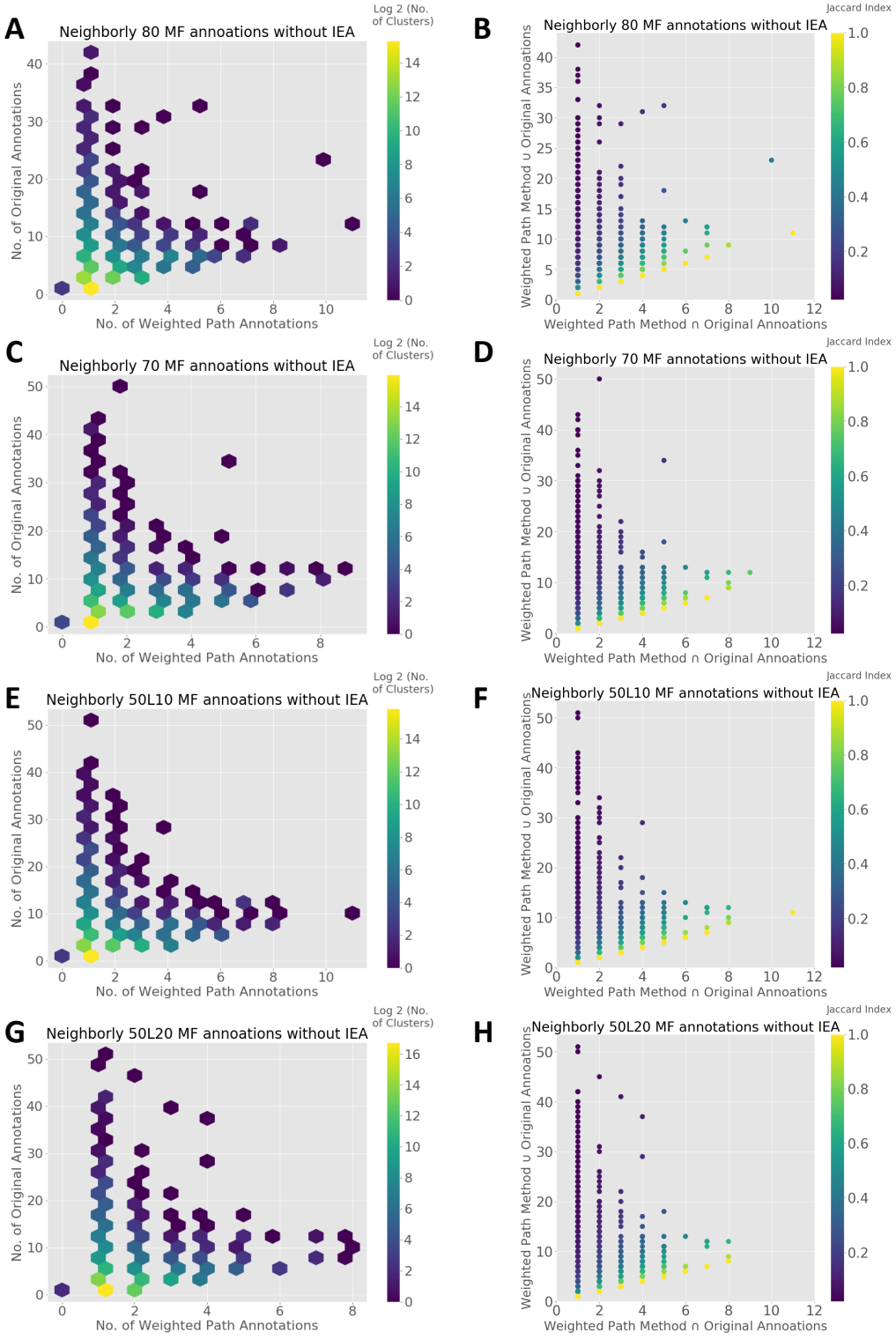

Figure A.14: Comparison of Weighted Path Annotations and Original Annotations for Molecular Function annotations on Neighborly 80, 70, 50L10 and 50L20 networks

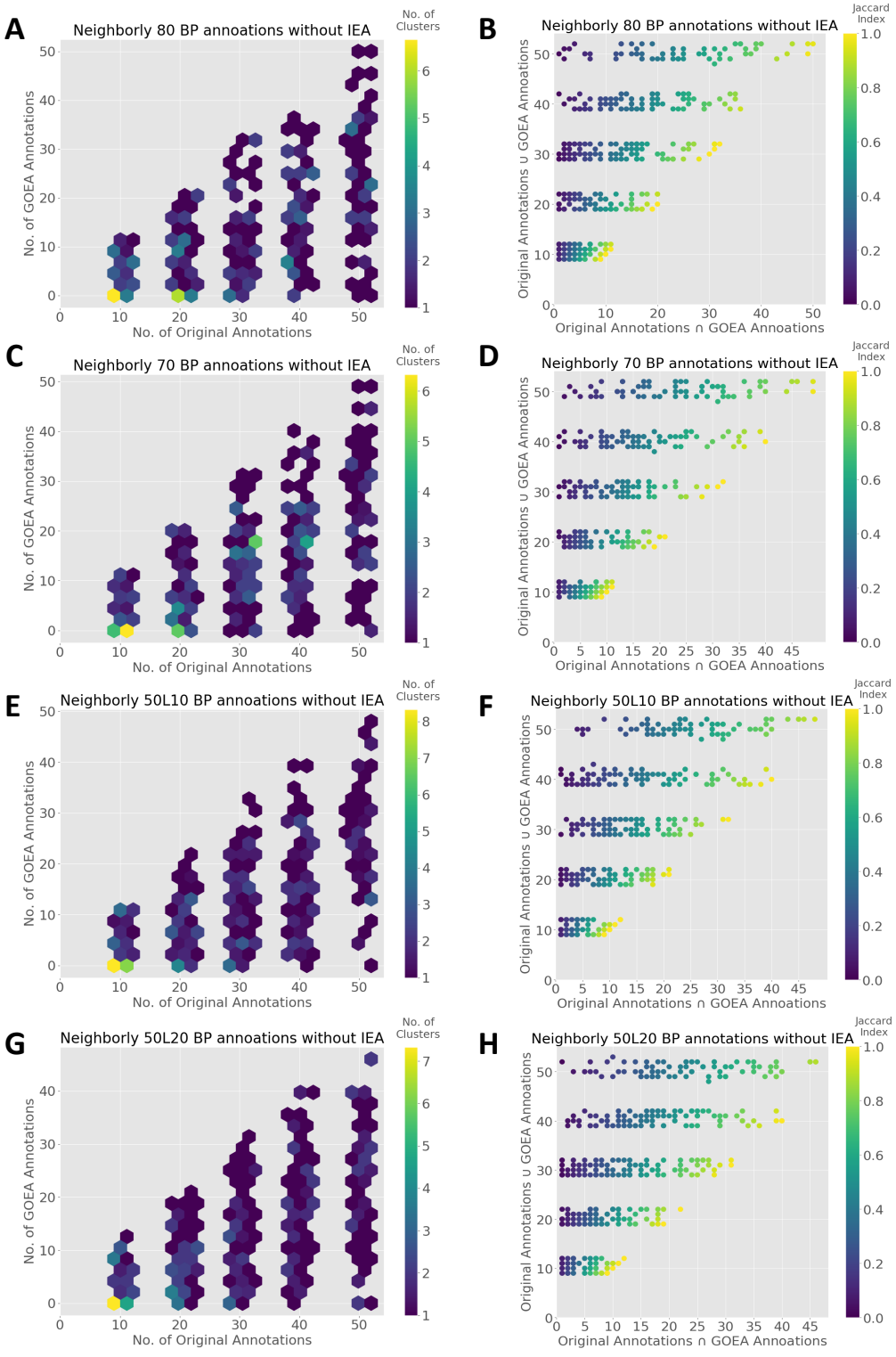

Figure A.15: Comparison of Gene Ontology Enrichment Annotations and Original Annotations for Biological Process annotations on Neighborly 80, 70, 50L10 and 50L20 networks

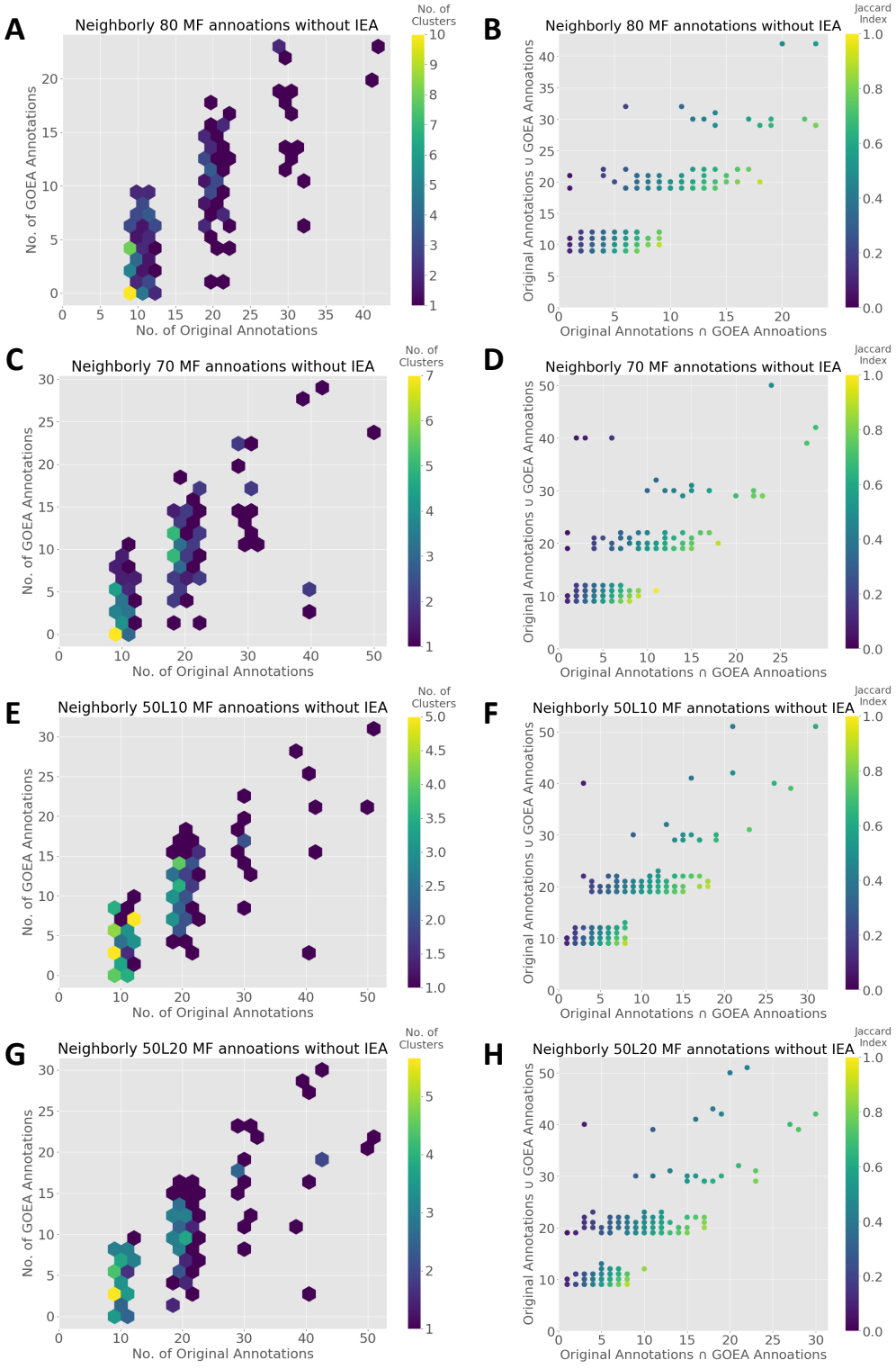

Figure A.16: Comparison of Gene Ontology Enrichment Annotations and Original Annotations for Molecular Function annotations on Neighborly 80, 70, 50L10 and 50L20 networks

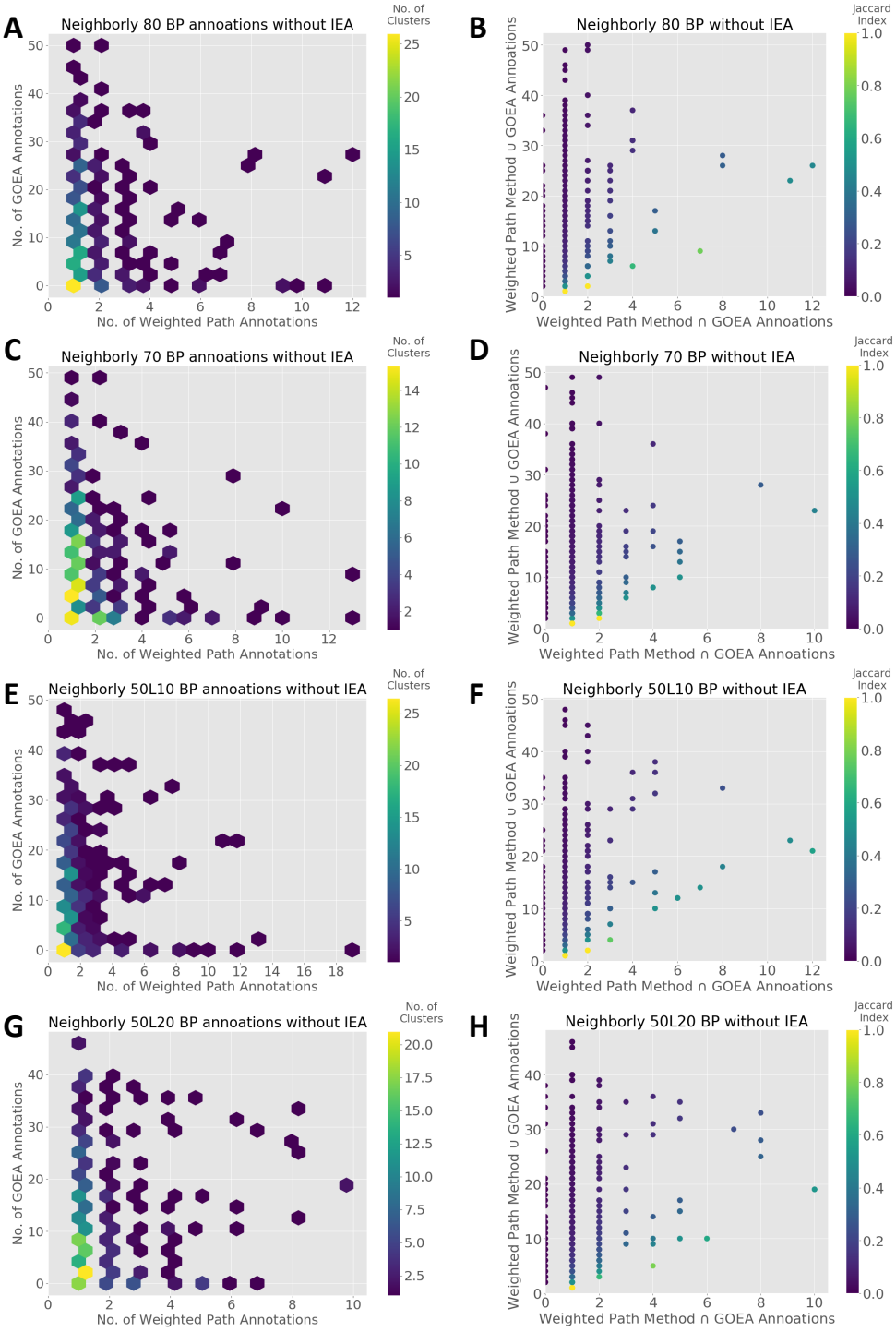

Figure A.17: Comparison of Weighted Path Annotations and Gene Ontology Enrichment Annotations for Biological Process annotations on Neighborly 80, 70, 50L10 and 50L20 networks

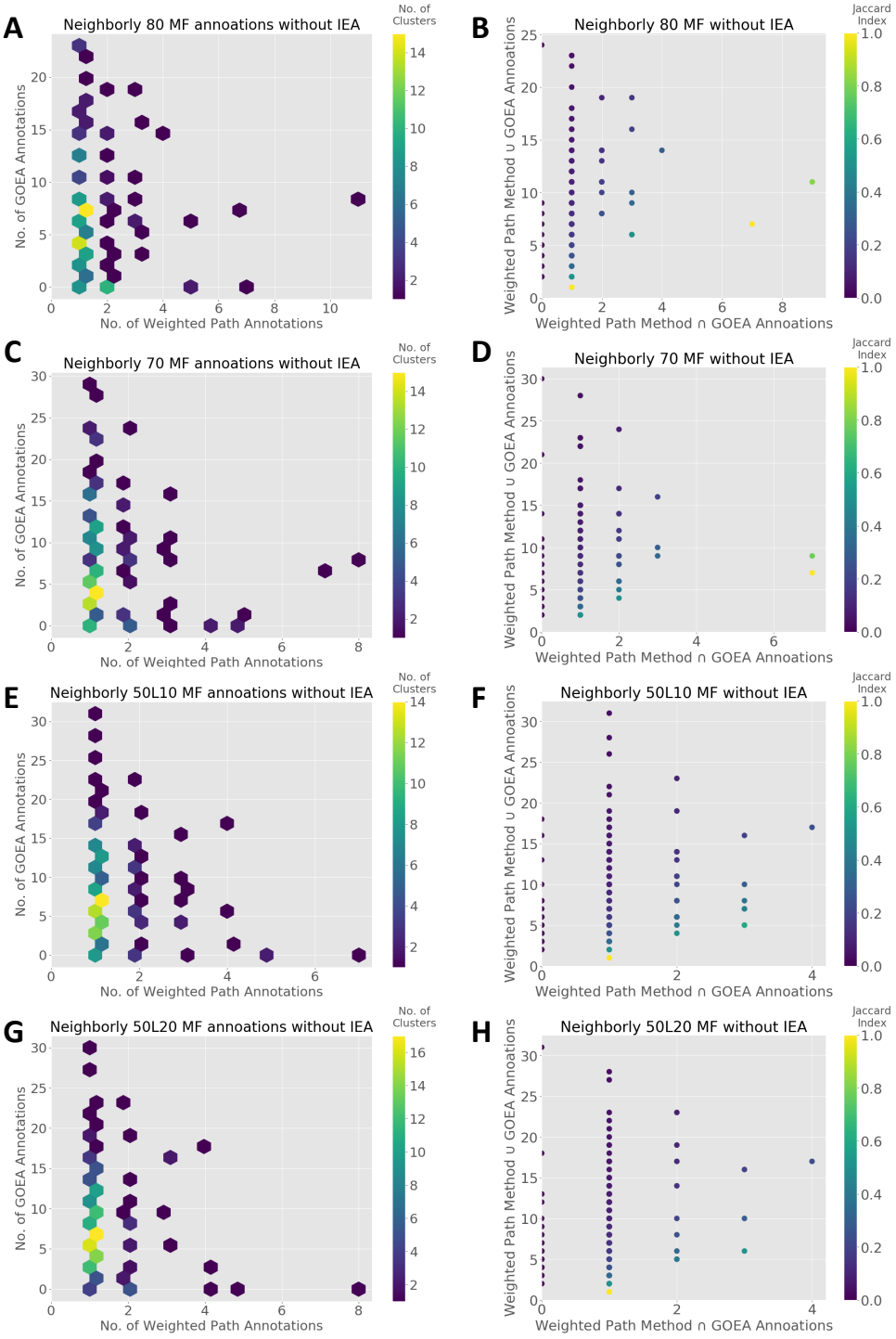

Figure A.18: Comparison of Weighted Path Annotations and Gene Ontology Enrichment Annotations for Molecular Function annotations on Neighborly 80, 70, 50L10 and 50L20 networks

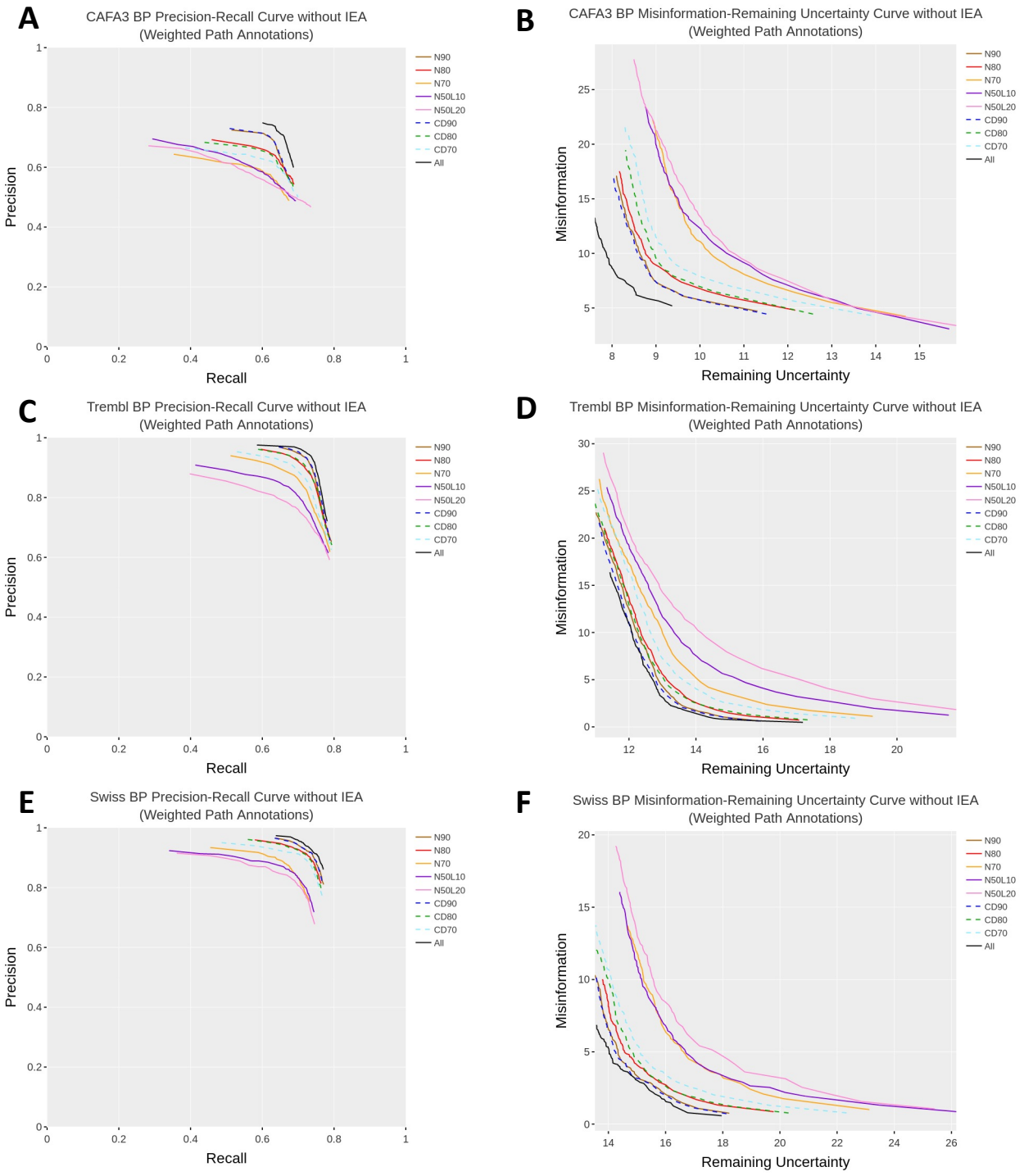

Figure A.19: Precision-Recall Curves and Misinformation-Remaining Uncertainty Curve for Cluster based predictions for biological processes using weighted path annotations

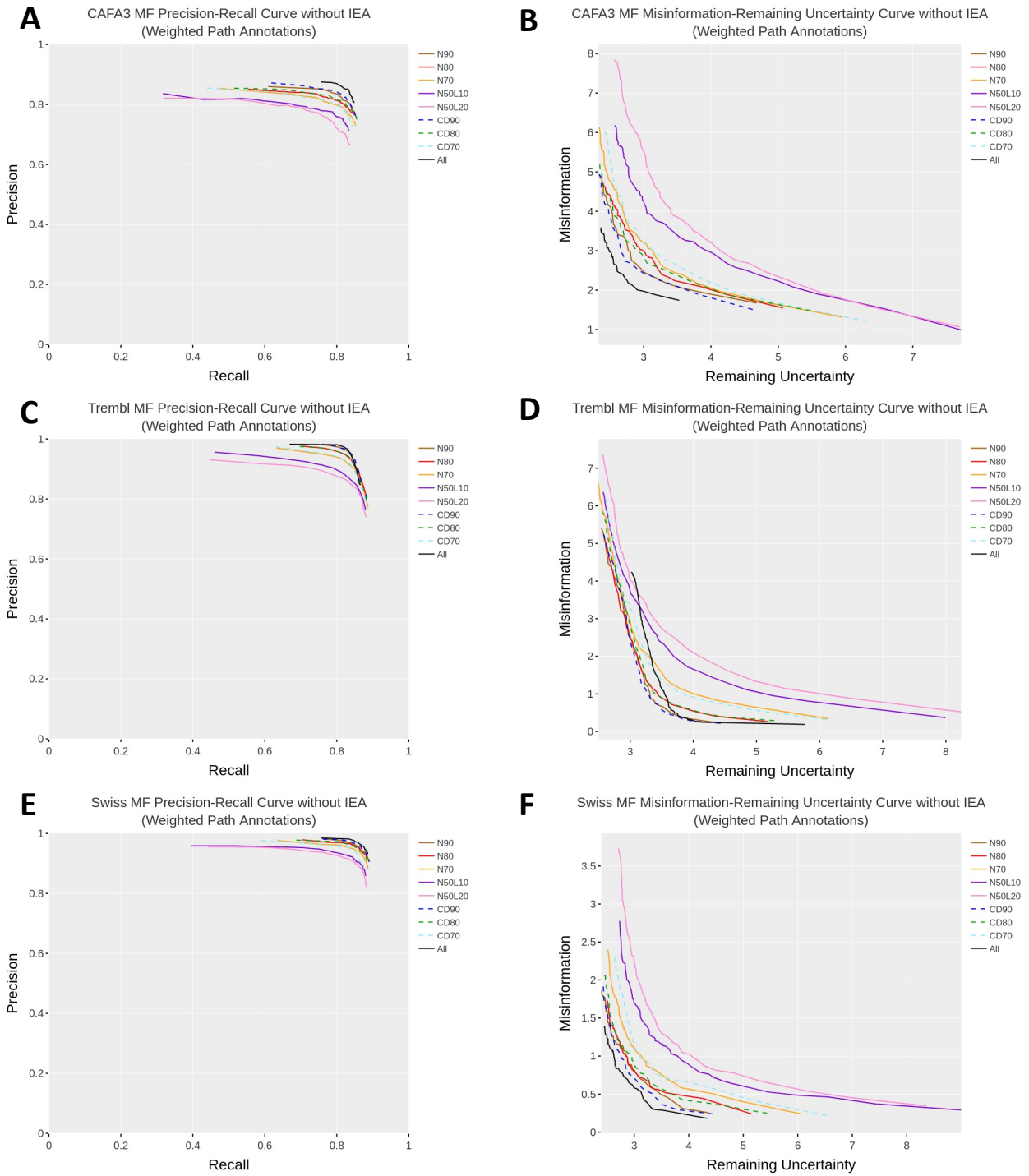

Figure A.20: Precision-Recall Curves and Misinformation-Remaining Uncertainty Curve for Cluster based predictions for molecular function using weighted path annotations

Appendix B

Supplementary Tables

Table B.1: Summary F_{max} and S_{min} scores for weighted path based annotations in comparison to original annotations for biological process labels

	CAFA3 Biological Process					
	F_{max}			S_{min}		
Method	Original	Weighted Path	% reduction	Original	Weighted Path	Original - WP
All	0.80	0.68	14.21	12.32	10.55	14.40
Neighborly 90	0.78	0.66	15.88	13.10	11.37	13.18
Neighborly 80	0.77	0.64	17.42	13.94	12.08	13.36
Neighborly 70	0.75	0.60	20.48	14.67	13.61	7.25
Neighborly 50L10	0.70	0.59	15.05	15.09	13.91	7.81
Neighborly 50L20	0.69	0.58	15.36	15.62	14.13	9.54
CD-HIT 90	0.78	0.66	15.75	12.87	11.41	11.39
CD-HIT 80	0.65	0.63	2.12	14.14	12.18	13.86
CD-HIT 70	0.64	0.62	3.38	14.40	12.73	11.57

	TrEMBL Biological Process					
	F_{max}			S_{min}		
Method	Original	Weighted Path	% reduction	Original	Weighted Path	Original - WP
All	0.94	0.82	12.29	4.10	13.34	-9.24
Neighborly 90	0.93	0.81	12.14	6.28	13.65	-7.37
Neighborly 80	0.91	0.80	11.77	8.15	14.00	-5.85
Neighborly 70	0.86	0.77	10.67	12.59	14.92	-2.33
Neighborly 50L10	0.80	0.75	6.77	19.42	15.80	3.61
Neighborly 50L20	0.76	0.73	3.71	23.17	16.90	6.26
CD-HIT 90	0.93	0.81	12.41	5.78	13.53	-7.76
CD-HIT 80	0.87	0.80	7.73	7.44	13.87	-6.43
CD-HIT 70	0.86	0.79	8.81	8.42	14.50	-6.08

	SwissProt Biological Process					
	F_{max}			S_{min}		
Method	Original	Weighted Path	% reduction	Original	Weighted Path	Original - WP
All	0.93	0.82	11.83	4.76	14.78	-10.02
Neighborly 90	0.93	0.82	11.94	6.09	15.10	-9.01
Neighborly 80	0.92	0.81	11.77	7.55	15.39	-7.84
Neighborly 70	0.90	0.76	15.53	9.55	17.22	-7.67
Neighborly 50L10	0.86	0.76	11.12	12.37	17.30	-4.92
Neighborly 50L20	0.85	0.75	11.28	15.22	17.86	-2.64
CD-HIT 90	0.93	0.82	11.97	5.74	15.04	-9.30
CD-HIT 80	0.76	0.81	-5.65	14.42	15.61	-1.19
CD-HIT 70	0.76	0.80	-4.74	14.62	15.88	-1.26

Table B.2: Summary F_{max} and S_{min} scores for weighted path based annotations in comparison to original annotations for molecular function labels

	CAFA3 Molecular Function					
	F_{max}			S_{min}		
Method	Original	Weighted Path	% reduction	Original	Weighted Path	Original - WP
All	0.89	0.84	5.33	4.09	3.53	0.56
Neighborly 90	0.87	0.82	5.42	4.40	3.88	0.51
Neighborly 80	0.85	0.81	5.12	4.43	4.05	0.38
Neighborly 70	0.85	0.80	4.98	4.64	4.19	0.45
Neighborly 50L10	0.81	0.78	3.57	5.17	4.86	0.31
Neighborly 50L20	0.80	0.76	3.83	5.56	5.11	0.45
CD-HIT 90	0.87	0.83	5.02	4.24	3.85	0.38
CD-HIT 80	0.72	0.81	-12.45	5.69	4.06	1.63
CD-HIT 70	0.72	0.80	-10.94	5.69	4.32	1.37

	TrEMBL Molecular Function					
	F_{max}			S_{min}		
Method	Original	Weighted Path	% reduction	Original	Weighted Path	Original - WP
All	0.94	0.89	5.47	1.53	3.67	-2.14
Neighborly 90	0.94	0.89	4.76	1.85	3.44	-1.59
Neighborly 80	0.92	0.88	3.52	2.54	3.53	-0.98
Neighborly 70	0.89	0.87	1.84	3.94	3.83	0.12
Neighborly 50L10	0.85	0.85	-0.18	5.46	4.20	1.27
Neighborly 50L20	0.82	0.85	-3.27	6.63	4.44	2.18
CD-HIT 90	0.94	0.89	4.66	1.86	3.42	-1.56
CD-HIT 80	0.88	0.88	-0.43	2.97	3.52	-0.55
CD-HIT 70	0.88	0.87	0.31	3.19	3.72	-0.53

	SwissProt Molecular Function					
	F_{max}			S_{min}		
Method	Original	Weighted Path	% reduction	Original	Weighted Path	Original - WP
All	0.95	0.91	3.70	1.06	2.77	-1.71
Neighborly 90	0.95	0.91	4.03	1.40	2.88	-1.48
Neighborly 80	0.94	0.90	3.80	1.64	2.92	-1.29
Neighborly 70	0.93	0.90	3.39	2.07	3.11	-1.05
Neighborly 50L10	0.90	0.88	2.43	2.79	3.45	-0.66
Neighborly 50L20	0.90	0.88	2.46	3.30	3.63	-0.34
CD-HIT 90	0.95	0.91	3.87	1.23	2.87	-1.64
CD-HIT 80	0.83	0.90	-8.73	3.83	2.95	0.88
CD-HIT 70	0.84	0.89	-7.04	3.81	3.21	0.60