GOING BEYOND HOMOLOGY FOR PREDICTING PROTEIN FUNCTION FOR NEWLY SEQUENCED ORGANISMS

A Dissertation

Presented to the Department of Computer Science

of Royal Holloway, University of London

in Partial Fulfilment of the Requirements for the Degree of

Doctor of Philosophy

by

Mateo Torres

Supervisor: Prof. Alberto Paccanaro

September 2019

Declaration

I Mateo Fernando Torres Bobadilla hereby declare that this thesis and the work presented in it is entirely my own. Where I have consulted the work of others, this is always clearly stated.

_____

Signature

_____

Date

To Nelly, Eduardo, Marco, Rocio, and Polo.

Acknowledgements

I am grateful to my supervisor, Prof. Alberto Paccanaro, for his guidance and continuous dedication to the research in the lab, in addition to his invaluable and ongoing contribution to Paraguay.

I would like to thank Prof. Haixuan Yang for his central contribution to the work presented in this thesis. I also thank Dr. Alfonso E. Romero for his contributions, the continuous support, friendship and guidance to understanding the vast field of protein function prediction. Also, to Prof. Lszl Bgre and his research group for their guidance in our collaboration.

I thank my fellow labmates at the PaccanaroLab. Dr. Juan Cceres, Diego Galeano, Dr. Horacio Caniza, Dr. Cheng Ye, Rubn Jimnez, Jessica Gliozzo, Michele Nacucchi, Vctor Yubero, Santiago Noto, and Phil Ovington. Innumerable discussions, conversations, and late nights contributed to our progress as scientists and friends. They are responsible for making the PhD process fun and the lab environment enjoyable.

I thank my parents, Nelly and Eduardo. Without their encouragement to my curiosity I would not have pursued a scientific career. Also my siblings, Marco and Rocio, growing up with them has been a great privilege.

Last, but not least, to my friends all over the world, with whom I shared so many laughs, worries, and a considerable number of pints! Thanks!

GOING BEYOND HOMOLOGY FOR PREDICTING PROTEIN FUNCTION FOR NEWLY SEQUENCED ORGANISMS

Mateo Torres

Royal Holloway, University of London 2019

The computational annotation of proteins has become a crucial step to the functional characterisation of genomes. Many computational methods predict protein function by exploiting experimental data such as protein-protein interactions and gene expression. For newly sequenced organisms these experiments are not available, limiting the feasible tools to sequence-based techniques.

In this thesis, I approach the problem of predicting protein function for newly sequenced organisms in three different ways. First, by exploiting the "guilt by association" principle in the context of protein-protein networks. Second, by elucidating the domain architecture of proteins and associating them with functions. Finally, by identifying protein complexes and the function enriched in every complex. Each approach considers different aspects of the problem and a wide variety of techniques are applied to address them. These techniques share the fundamental property of transferring information from well-studied organisms to those that are barely characterised, if at all.

Table of Contents

List of Tables

List of Figures

1

Introduction

1.1 Motivation

The study of protein function represents a major effort to explain the mechanisms behind the processes occurring in the living cell. A thorough understanding of such mechanisms will allow us not only to make better decision with regards to our own biology, but also regarding the biological environment that surrounds us. With applications possible in industries such as personalised medicine, food, and energy, there is a strong global motivation to determining protein function as accurately as possible. This is not an easy task.

The very concept of function is hard to define. Several attempts of characterising protein function were made but remained divergent until the creation of the Gene Ontology (GO) [1], a controlled vocabulary of terms that became the most widely adopted tool for the functional characterisation of genes and gene products (proteins). In this context, a functional annotation is the assignment of a GO term to a protein.

Experimentally, these associations are discovered by testing whether a particular protein is involved in the function. These experiments tend to be expensive, time-consuming, and often can handle only one protein at a time. This, coupled with next generation sequencing (NGS) techniques created a scenario in which experimental approaches fall short. Even the higher-throughput approaches for experimental elucidation are unable to cope with the exponential nature of NGS techniques [2]. In this scenario, computational approaches for elucidating protein function became a more important step of the process. Protein function prediction (PFP) is defined as the problem to predict such annotations.

Consider the current portion of annotated protein sequences. Less than 1% of all available proteins are annotated with reliable experimental annotations. Moreover, the existing annotations are focused on a relatively small subset of sequenced organisms. This subset of well-studied organisms are the model organisms, and the available information ranges from the sequence to the 3D structure, and even high-quality experimental annotations of GO terms. Conversely, the data available for newly sequenced organisms (NSO) is limited only to the sequence. This means that for the majority of the sequenced organisms, only sequenced-based methods for PFP are readily available.

When putting this in contrast with the current state of methods available for PFP, we find ourself facing a regrettable situation. The Critical Assessment for Functional Annotations (CAFA) [3] shows that PFP methods that exploit information such as gene expression, protein structure, and protein-protein interaction (PPI) networks outperform classical methods based on simple sequence similarity. This is, the best available methods are essentially unavailable for the vast majority of sequenced organism.

In this thesis, I delve deep into the current limitations for predicting protein function for NSOs, and present alternatives that go beyond establishing homology relations based on sequence similarity for the prediction of protein function. These share the core property of transferring useful experimental information from well-studied organsism to NSOs, and exploiting these transferred data with a wide variety of techniques. I show how this transferred knowledge can be used to predict protein function, but also infer protein complexes and protein domains.

1.2 Contributions

- **S2F:** A state-of-the-art framework for PFP. S2F focuses on the prediction of protein function for recently sequenced organisms. It is, to my knowledge, the first one to exploit available information in a way that allows going beyond sequence-based techniques for this task. Its three main parts consist of building a seed, building a network, and propagating the seed into this network. I report its performance for newly sequenced organisms, as well as its performance in several CAFA competitions.
- **ConSAT:** The Consensus Architecture Tool. This consists of both a tool and web applicaiton for protein function annotation for genome projects of any scale. It relies on protein domains to infer function. Particularly, it is focused in the "domain architecture" of proteins. Function is determined using GO terms as well as English keywords.
- **ICrep:** A comprehensive protein interaction and complexes repository. A database that contains protein-protein interactions, inferred interologs (computationally inferred interactions, derived by comparing the protein sequences [4]), protein complexes, and functional enrichments for the complexes. Built by performing a pairwise transference of information between non-redundant proteomes available in UniProtKB [5].

1.3 Structure of the Book

Chapter 2 consists of a literature review on the core concepts required for the proper understanding of this book. I attempt to introduce the reader to the state-of-the-art in PFP, and the relevant tools and methods relevant to this problem.

In chapter 3 I explain the S2F framework in depth. First, I provide an overview of the state-of-the-art for PFP for newly sequenced organisms. Then, I delve deep into the definition of the framework, evaluation, and experimental results. Finally, I discuss S2F for "regular" PFP, not focused only in newly sequenced organism. I present this in the context of CAFA.

In chapter 4 I introduce the ConSAT tool. First, I explain the methodology

used to obtain consensus domain architectures. Then, I explain the strategies to assign function to proteins, both GO terms as well as English keywords. Finally, I describe the features of the upcoming web tool.

In chapter 5 I showcase the ICrep database. I start with thorough explanation of the calculations of the interologs, protein complexes, and over-representation analysis. This is followed by several statistics and options available to the user of ICrep.

In chapter 6 I show my contributions to collaboration projects that do not have PFP as their main focus. After a quick glance at each project, I proceed to highlight the functional aspects of each one. These projects are:

- Studying the effects of Rapamycin on *Chlamydomonas reinhardtii*.
- prot2vec: A tool for the embedding of proteins in a vectorial space.

Finally, appendix A contains further S2F performance results.

2

Literature Review

2.1 The Gene Ontology

Protein function is predominantly described in the literature using terms from the Gene Ontology (GO) [1]. The GO is a controlled vocabulary of terms, each term describing a function in one of three domains: Molecular Function (MF), Cellular Component (CC), and Biological Process (BP). GO terms are related to each other with relations that operate between them. The structure can be described as a graph, in which nodes are GO terms, and edges the relations between terms. The main relations used in GO are:

1. **is a:** This is the relation that forms the basic structure of the ontology. If A *is a* B, it means that node A is a subtype of node B. For example, in the GO "mitotic cell cycle" *is a* "cell cycle". It does not mean, however, that A *is an instance of* B; e.g. a dog *is a* mammal, but Snoopy *is an instance of* a dog, rather than a subtype of dog.
2. **part of:** This relation is used to represent part-whole relationships. This

is, if A *is part of* B, it means that A is necessarily a part of B: wherever A exists, it is as part of B. The presence of A implies the presence of B, but not vice-versa. For example, in the GO "receptor ligand activity" is *part of* "signal transduction".

3. **has part:** This is the logical complement of *part of*, representing the part-whole relationship, but from the point of view of the parent. In similar fashion if A *has part* B, then A necessarily has part B, meaning that if A exists, then always B exists as part of A, but not vice-versa. i.e. all A *have part* B; some B *part of* A. For example, in the GO "cytokinesis" *has part* "membrane fission".

4. **regulates:** This relation is present when one process directly affects the manifestation of another process or characteristic. A *regulates* B means that necessarily, when both A and B are present, B is regulated by A. But B may not always be regulated by A, i.e. all A *regulate* B; some B are regulated by A. In many occasions, the nature of the regulation (positive or negative) is added to the relation, e.g. A *positively regulates* B. For example, in the GO "regulation of mRNA cleavage" *regulates* "mRNA cleavage". However, annotations to regulation terms modify the relation between the annotated protein and the GO term. In this case, a protein *X* annotated with "regulation of mRNA cleavage" is considered to be involved in that process (the one of regulation). It is not correct, however, to assume that *X* is involved in "mRNA cleavage".

These relations, and in particular the *is a* and *part of* relations establish a structure in which terms are naturally organised with the most general terms at the "top" and more specific terms at the "bottom". The former being isolated without being related with *is a* or *part of* to any other GO term. Moreover, these relations allow for logical transitivity, i.e. if A *is a* B, and B is *part of* C, then we can infer that A is *part of* C, see Figure 2.1. Importantly, these relations allow for the annotations to be propagated through the ontology. This is, if GO Term A is associated with protein P, because A *is a* B, then the association between B and P also exists. The process of "up-propagating" GO annotations follows the *true path rule*: if a protein is annotated with a GO term, it is also annotated with all the ancestors of the GO term. The advantage of this rule is that quite a big portion of the associations can be transferred by the most specific annotations,

knowing that the association will be up-propagated to the root of the ontology.

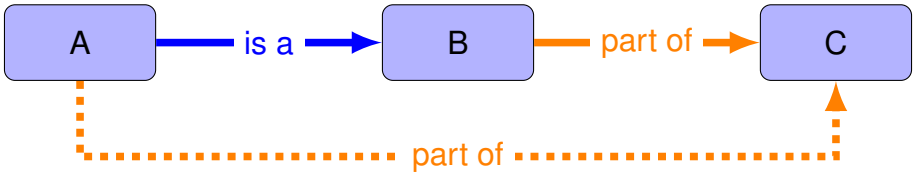


Figure 2.1 – *Transitivity of relations in the GO. Given that A **is a** B, and B **part of** C (represented by solid arrows), we can infer a new relation A **part of** C (dashed arrow).*

The GO describes and organises the knowledge of the biological in three sub-domains:

1. **Molecular Function:** This domain describes molecular-level activities performed by proteins. Examples are "transporter activity" (GO:0005215), and "catalytic activity" (GO:0003824). MF GO terms describe activities rather than the entities that perform the actions (molecules or complexes), and do not specify where, when, or the context in which these actions take place.
2. **Cellular Component:** This domain is concerned with the locations relative to cellular structures in which the actions are performed by proteins. Examples of are "mitochondrion" (GO:0005739), and "ribosome" (GO:0005840).
3. **Biological Process:** This domain describes the larger and more complex processes accomplished by multiple molecular activities. Examples are "DNA repair" (GO:0006281), or very specific ones such as "pyrimidine nucleobase biosynthetic process" (GO:0019856). It is important to note that a biological process is not the same as a pathway, as these terms do not represent the dynamics and dependencies required to fully describe a pathway.

The GO is constantly being revised and updated to reflect the latest research on functional genomics. At the time of writing, there are 44,733 valid terms, that are distributed in the sub-domains: 29,457 BP terms, 11,093 MF terms, and 4183 CC terms.

2.2 The evolution of available sequences and GO annotations

The GO provides an invaluable tool for the functional characterisation of proteins. These come in the form of GO annotations, which are statements about the function of a particular protein. A very important characteristic of these annotations is their evidence code, which indicates how the annotation is supported. Evidence codes are organised as follows:

- **Experimental evidence:** These codes indicate that there is evidence from an experiment directly supporting the annotation of a protein. Experimental evidence codes are:
  - Inferred from Experiment (EXP)
  - Inferred from Direct Assay (IDA)
  - Inferred from Physical Interaction (IPI)
  - Inferred from Mutant Phenotype (IMP)
  - Inferred from Genetic Interaction (IGI)
  - Inferred from Expression Pattern (IEP)

  Each of the experimental evidence code have a corresponding high throughput evidence code (HTP). HTP are a type of experimental evidence that indicate that the annotation is supported by high throughput methodologies. The high throughput evidence codes are:
  - Inferred from High Throughput Experiment (HTP)
  - Inferred from High Throughput Direct Assay (HDA)
  - Inferred from High Throughput Mutant Phenotype (HMP)
  - Inferred from High Throughput Genetic Interaction (HGI)
  - Inferred from High Throughput Expression Pattern (HEP)
- **Phylogenetic evidence:** Phylogenetically inferred annotations are derived from an explicit model gain and loss of function at specific branches in a phylogenetic tree. Phylogenetic evidence codes are:
  - Inferred from Biological aspect of Ancestor (IBA)

  - Inferred from Biological aspect of Descendant (IBD)
  - Inferred from Key Residues (IKR)
  - Inferred from Rapid Divergence (IRD)

- **Computational evidence:** A computational evidence code indicates that the annotation is based on an *in silico* analysis of the protein. Computational evidence codes are:
  - Inferred from Sequence or structural Similarity (ISS)
  - Inferred from Sequence Orthology (ISO)
  - Inferred from Sequence Alignment (ISA)
  - Inferred from Sequence Model (ISM)
  - Inferred from Genomic Context (IGC)
  - Inferred from Reviewed Computational Analysis (RCA)

- **Author statements:** These codes indicate that the annotation was made on the basis of a statement made by the author(s) in the cited reference:
  - Traceable Author Statement (TAS)
  - Non-traceable Author Statement (NAS)

- **Curator statements:** These codes indicate that the annotation was made on the basis of a curatorial judgement that does not fit into any other evidence code classifications:
  - Inferred by Curator (IC)
  - No biological Data available (ND)

- **Automatically generated annotations:** The final evidence code indicates that the annotation was not manually reviewed. These annotations are ultimately based on either homology and/or other experimental or sequence information, but generally cannot be traced to an experimental source:
  - Inferred from Electronic Annotation (IEA)

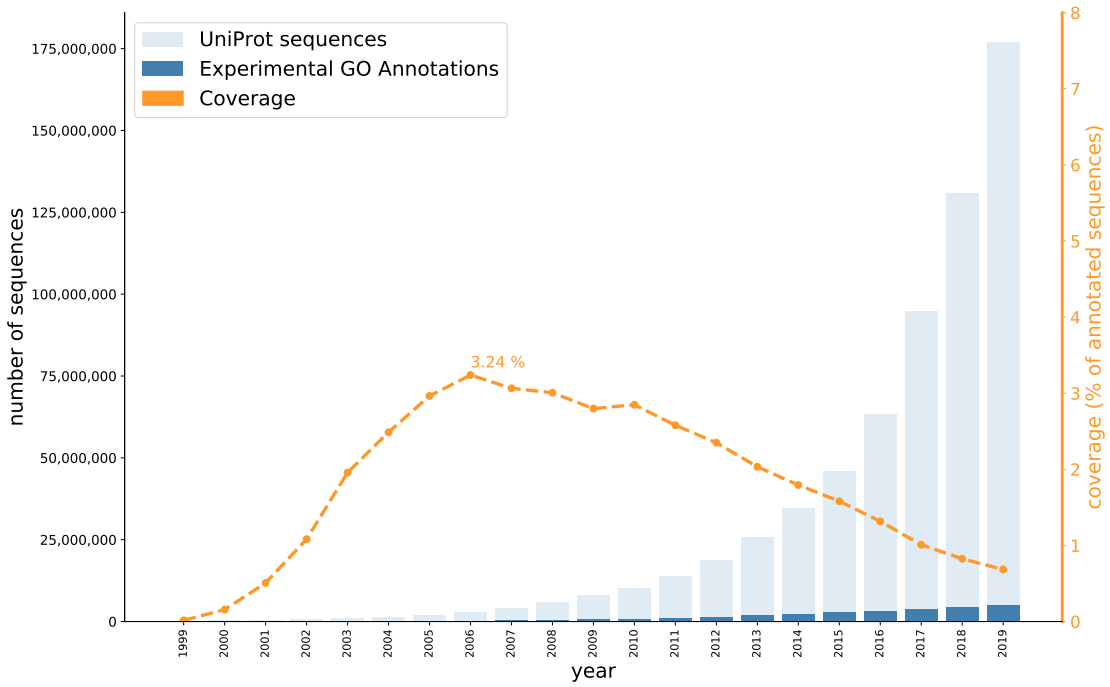


Figure 2.2 – *Evolution of the available protein sequences and the experimental GO annotations over the last 20 years. The coverage line corresponds to the axis on the right of the plot.*

To fully grasp the current situation relevant to the PFP problem, it is useful to look at some number. First, the gap between annotated and unannotated proteins is vast and it is widening steadily. Figure 2.2 depicts the evolution of available protein sequences in UniProtKB [5] in comparison to the experimental functional annotations available in the GOA database [6] since the creation of the GO in 1999. Even though the number of annotated proteins is steadily increasing, the rate at which new proteins are being sequenced is such that the percentage of annotated sequences actually decreases. In fact, the peak percentage of annotated proteins was 3.23% in 2006. This coincides with the beginning of the exponential growth made possible by NGS techniques.

Second, the growth is not evenly distributed between taxonomical superkingdoms. As can be seen in Figure 2.3, the bacterial superkingdom has been increasing in proportion almost monotonically. This means that the bacterial kingdom is the one that requires a stronger effort for its functional characterisation, as a direct consequence of the sheer number of sequenced proteins available. Consider the wide variety of technology in which of bacteria are involved. We use bacterial microorganisms for a lot of applications: food process-

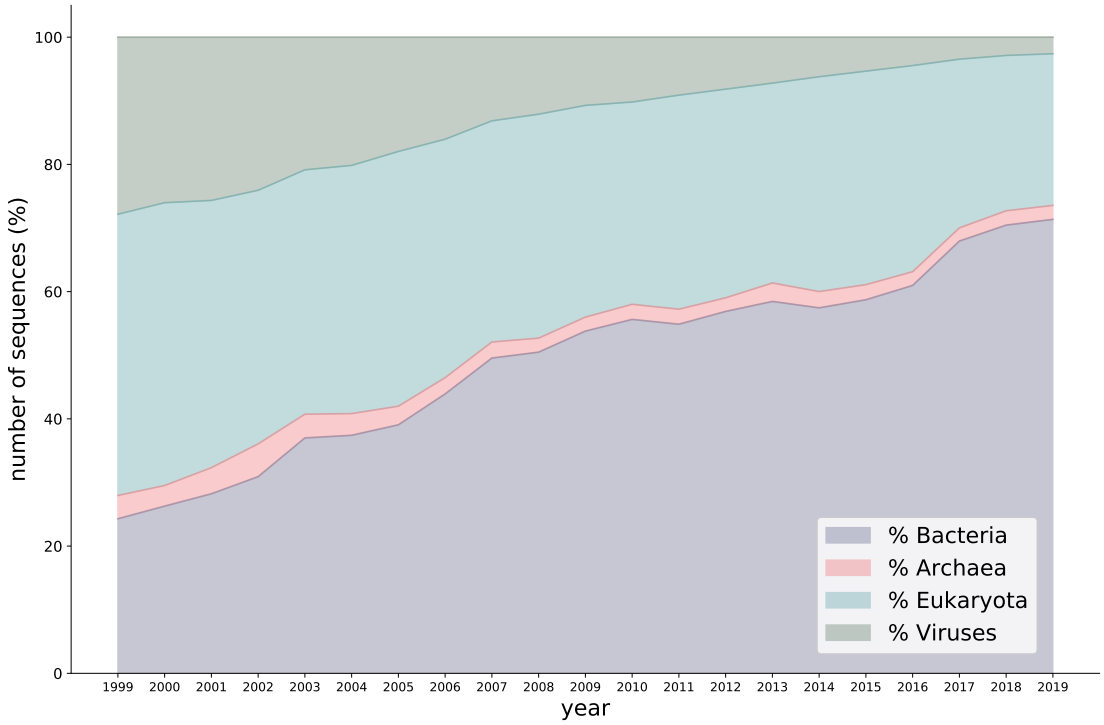


Figure 2.3 – *Distribution of available protein sequenced by superkingdom. Over the last 20 years, the proportion of bacterial proteins has dominated the other super kingdoms. Note: we include Viruses as a superkingdom even though they represent non-cellular sequences.*

ing, medicine, pest control, manufacturing cosmetics, just to name a few. These applications make of the functional characterisation crucial from an economic point of view. Additionally, some bacteria are harmful as agents of disease, and are involved in spoilage of food and other resources. Some harmful bacteria even become antibiotic-resistant, which poses an even greater challenge [7]. This is enough motivation to expand our understanding of the biological processes of bacteria. The potential gain from functionally characterising this vast collection of proteins is almost incalculable.

2.3 The current state of PFP

Over the years, several computational methods for PFP were developed, and these fall into many categories [2], I will describe the rationale in each category, and describe representative methods in the following sections.

2.3.1 Sequence-based methods

Methods in this category use an operational definition of sequence similarity to determine an homologous relation between two proteins [2]. Although these methods do not automatically predict protein function, it can be transferred from an annotated protein to a protein candidate if the similarity is above a certain threshold [8, 9]. It is a broad category, as it encompasses sub-categories such as domain-based, motif-based and feature-based methods [10, 11, 12].

The comparison between sequences aims to determine the evolutionary relationship between sequences, and infer whether they are related, i.e. determine homology between sequences. This is remarkably hard, as high sequence similarity might not be caused by genetic ancestry [2]. Two sequences might be very similar due to convergent evolution, and short sequences might be very similar due to change. On top of that, homologous proteins can have notably low sequence similarity, e.g. remote homologs with early evolutionary branching [13]. Although the collection of PFP methods that rely on sequence similarity-based transference of function is vast, I will limit my review to the tools that will be directly involved in the methods I propose in the next chapters.

BLAST

The indisputably dominant tool for sequence alignment is the Basic Local Alignment Search Tool (BLAST) [8]. It is a sequence alignment tool that will detect biologically significant similarities between sequences. It is famously over 50 faster than earlier sequence alignments, although not as accurate. This is due the use of a fast heuristic based on dynamic programming. Here I explain only the calculation of the BLAST e-value, as it is arguably the value that will allow me to describe operational definitions of homology in later chapters.

BLAST assesses the statistical significance of its score by exploiting the Gumbel extreme value distribution (EVD) as it is proved that the distribution of the Smith-Waterman alignment [14] between two random sequences follows the Gumbel EVD. Under this assumption, the probability of observing a score $S \geq x$ is given by:

$$p(S \geq x) = 1 - \exp(-e^{-\lambda(x-\mu)})$$

where

$$\mu = \frac{\log(Kmn)}{\lambda}$$

The statistical parameters λ and K are estimated by fitting the distribution of the scores before adding gaps to the alignment, of the query sequence and a lot of shuffled versions of a database sequence to the Gumbel EVD, the values of these parameters depend upon the substitution matrix, gap penalties and amino acid (or nucleotide) composition. m and n are the effective lengths of the query and database sequences, respectively. The expected score E is the number of times that an unrelated database sequence would obtain a score $S \geq x$ by change. obtained in a search for a database of D sequences is:

$$E \approx 1 - e^{-p(S \geq x)D}$$

HMMER

Pairwise sequence comparison methods (such as BLAST) assume that all amino acid positions have the same importance. However, we know that this is not the case, and a great deal of position-specific information is available for a protein or protein family. Multiple alignments of proteins, for instance, show residues that are more conserved than others, and the places of frequent deletions or insertions [15]. A profile HMM will model such multiple alignments, providing states for insertions, deletions and "emission" states. There is a collection of these three types of state equal to the length of the multiple alignment. Each "emission" state is additionally associated a distribution over the possible amino acids that it could emit. Finally, transition probabilities between states are added to the HMM, which complete the profile. With a collection of profiles, it is possible to calculate how likely a given sequence matches the set of sequences described by the profile HMM, and therefore a database search with sequence alignments can be produced.

HMMER [9] is a tool to search sequence databases for homologs of protein or DNA, and to make sequence alignments. It builds a profile hidden Markov model that assigns a position-specific scoring for substitutions, insertions, and deletions. When compared to BLAST and other sequence alignment scoring methods, HMMER is aimed to be more accurate and able to better detect re-

mote homologs. This is possible because of the strength of the underlying profile HMMs. Moreover, the current version (3.2) is as fast as BLAST for database search. As with alignment-based sequence methods, function will be transferred if a threshold for the alignment is met.

InterPro

By a wide margin, the largest source for automatic annotation of sequences in UniProtKB is InterPro [16]. It compiles the predictions from 13 specialised databases that rely on different strategies to assign potential function to proteins. The collective expertise of the specialised databases provides a wide range of functional characterisation of proteins. In addition to the database, InterProScan [17] is a tool that allows the user to input their own sequences to be scanned against InterPro.

Here, I provide a simple overview of the member databased that InterPro aggregates:

- **CATH-Gene3D [18]:** A pair of databases of globular domain annotations for millions of available protein sequences. The database uses experimentally determined three-dimensional structures from the Protein Data Bank (PDB) to determine protein domains using a mixture of automatic methods and manual curation. The domains are then classified within the CATH structural hierarchy (Class, Architecture, Topology, Homologous superfamily). Gene3D then takes CATH domain families and assigns them to the millions of sequences with no PDB structures using HMMER. This process provides both structural and functional insight into the proteins classified in the CATH hierarchy.
- **Conserved Domain Database (CDD) [12]:** This database consists of a collection of well-annotated multiple sequence alignment models for ancient domains and full-length proteins. CDD provides these as position-specific score matrices that can be used to identify conserved domains. CDD includes NCBI-curated domains that use 3D structure information to define domain boundaries, as well as domains imported from external databases.
- **HAMAP [19]:** The High-quality Automated and Manual Annotation

of Proteins (HAMAP) provides annotations of the same quality as UniProtKB/Swiss-Prot, using manually curated profiles for protein family classification and rules for functional annotation of family members. It focuses on an automatic annotation pipeline that is only applied where it can produce the same quality as manual annotation would. Originally developed for the annotation of proteins from completely sequenced bacteria, archaea, and plastids, they now produce and integrate HAMAP rules and profiles that target eukaryotic and viral protein families.

- **PANTHER [20]:** The Protein ANalysis THrough Evolutionary Relationships (PANTHER) database contains comprehensive evolutionary and functional information of protein-coding genes from 104 completely sequenced genomes. It is founded on a comprehensive set of phylogenetic trees that attempt to reconstruct the evolutionary events that led to the current family members. The trees are used to predict orthologs, paralogs, and xenologs, as well as protein families. Hidden Markov models (HMMs) are built for each family and subfamily. From a functional perspective, the trees enable inferences through both expert biocurators for experimentally-supported annotations, and through inheritance from its ancestors for uncharacterised sequences.
- **Pfam [21]:** Pfam is a database of protein families. Each entry in this database is comprised of a seed alignment, which forms the basis to build a profile HMM using HMMER. This profile is then queried against a sequence database called *pfamseq*. All matches scoring above a threshold chosen to avoid the inclusion of any know false positive are aligned back to the profile HMM to generate the full alignment. Entries are coupled with functional annotations from the literature. *pfamseq* is derived from the reference proteomes available in UniProtKB.
- **PIRSF [22]:** The PIRSF (PIR SuperFamily) classification system is defined as "a network classification system based on evolutionary relationships of whole proteins". It introduces several levels of curation for the classification of proteins in superfamilies, families, and subfamilies[1]. With respect

[1] A protein family is a group of evolutionarily-related proteins. Proteins in a family descend from a common ancestor and typically have similar function and structure. A superfamily is the biggest of such grouping of proteins and a subfamily is the smallest grouping (although different methods will have a different way of defining exactly what constitutes a subfamily, family or superfamily) [23]

to function, the PIRSF system is used to provide standardised and rich annotations for UniProtKB entries. GO annotations are considered in the "text annotation" category of the PIRSF system, with emphasis on proteins in families and subfamilies that share common functions and contain sufficient numbers of experimentally verified members.

- **PRINTS [24]:** This database is a compendium of protein fingerprints. These are defined as groups of conserved motifs that characterise a protein family. The advantages of having these fingerprints are two-fold. First, new sequences can be scanned against PRINTS to get possible clues about structure or function. Second, to categorise sequences into superfamilies, families and subfamilies characterised by common fingerprints.
- **ProDom [10]:** This database holds a comprehensive collection of protein domain families generated from the global comparison of all available protein sequences. It is constructed using iterative PSI-BLAST searches that yields an automated clustering of homologous domains into families. Multiple alignments are generated for each domain family. The possible applications of ProDom are, for instance, analysing protein domain relationships, and selecting candidate proteins for structural genomic projects.
- **PROSITE [25]:** This is a database of documentation entries describing protein families, domains, functional sites, and associated patterns and profiles to identify them. The database contributes to InterPro with patterns and profiles. PROSITE patterns are regular expressions matching short sequence motifs that hold biological meaning, they are qualitative, as they either match or not, although it is possible to evaluate their statistical significance. PROSITE profiles (or weight matrices) are quantitative, they are sequence-like linear structures consisting of alternating match and insert positions. A match position corresponds to a domain position, typically occupied by a single amino acid. A PROSITE profile provides weights for each residue type occupying this position plus a deletion penalty.
- **SMART [26]:** The Simple Modular Architecture Research Tool (SMART) allows the identification and annotation of genetically mobile domains[2] and the analysis of domain architectures. The tool compares sequences

[2]A genetically mobile domain is one that can be found associated with different domain combinations in different proteins [27]

and multiple alignments while concurrently identifying compositionally biased regions such as signal peptide, transmembrane and coiled coil segments. Each alignment is curated to assign appropriate domain boundaries and ensure its quality, and each domain is annotated extensively with respect to cellular localisation, species distribution, functional class, tertiary structure, and functionally important residues.

- **SFLD [28]:** The Structure Function Linkage Database (SFLD) is a manually curated hierarchical classification of enzymes relating specific sequence-structure features to specific chemical capabilities. It classifies evolutionarily related enzymes according to shared chemical functions that are then mapped to conserved active site features. The hierarchy is comprised of superfamilies, which are subdivided into subgroups and families.
- **SUPERFAMILY [29]:** This database holds structural and functional annotations for all proteins and genomes. These annotations are based on a collection of HMMs, which represent structural protein domains at the superfamily level. A superfamily groups several domains that have an evolutionary relationship. The database is constructed by scanning protein sequences belonging to completely sequenced genomes against the HMMs.
- **TIGRFAMs [30]:** This is a database of protein family definitions. Each entry includes a seed alignment of trusted representative sequences, a HMM built from that alignment, cut-off scores to decide which proteins are members, and annotations for transfer onto member proteins.

InterProScan [17] will scan a collection of sequences submitted by the user against all of these methods, and will be classified into to families, domains, and also functional categories. Due to the wide range of methodologies that the member databases follow to classifying proteins, InterPro provides a very profound insight into the functional characterisation of the provided sequence, which is focused on being of high precision and quality.

2.3.2 Function prediction based on Genomic Context

Methods in this category are based on the knowledge that the location of the coding gene provides important information that can be used for function prediction. Gene neighbourhood[3] and gene fusion[4] based methods fall into this category. The general idea is that DNA with an advantageous organisation of its genes will be conserved over DNA with less advantageous organisation. For instance, operons (groups of genes transcribed and regulated as one unit) will be relevant for performing a particular function, since it makes evolutionary sense to efficiently transcribe these genes for the same task [2].

A good example of this category is shown by Korbel et al. [33]. In this work, they use gene expression data to demonstrate two functional implications of genome organisation. First, chromosomal proximity indicates gene coregulation in prokaryotes independent of relative gene orientation. Second, adjacent bidirectional transcribed genes (i.e. "divergently" organised coding regions) with conserved gene orientation are strongly coregulated[5]. The authors exploit the fact that the organisation of divergently transcribed gene pairs (DT-pairs) is widely conserved in prokaryotes, whereas convergently transcribed pairs are rapidly lost in evolution.

Another example of functional associations based on genomic context is SNAP [34]. This tool predicts function based on the conservation of gene order[6]. Even though a genome-wide gene order is poorly conserved between phylogenetically distant species [35], short conserved strings of genes appear to be widespread [36]. This enables SNAP and methods based on conservation of gene order to look for small clusters of functionally related genes that might not be very close in terms of sequence-similarity. SNAP, in particular, uses a combination of neighbourhood and similarity relationships. Similarity relationships (S-relationships) are established by computing the sequence similarity between proteins, and Neighbourhood relationships (N-relationships) can

[3]Genomic neighbourhood refers to genes that occur repeatedly in close neighbourhood [31]

[4]A gene fusion event refers to a chromosomal rearrangement event where two genes fuse together to form a hybrid gene [32]

[5]Two genes are co-regulated if they are regulated by the same mechanism. This is typically reflected on a strong correlation in gene expression, and suggest the two genes are involved in the same processes, and therefore they perform similar function.

[6]Gene order refers to the possible permutations in the genome of a set of genes.

be found between pairs of genes that are adjacent in the genome. S-relationships and N-relationships are then used to build a graph that spans several genomes, in which the nodes are genes, and the links are S- and N-relationships. The hypothesis in SNAP is that cycles of SN-paths (a path that starts in a protein and that, following S- and N-relationships can end in the same protein after several jumps) found in this graph will highlight functional operons conserved over several genomes. The key assumption is that the combination of the two types of relationship will help establish functional links undetectable by either type alone.

The genomic context category also includes methods that exploit gene fusion events. The assumption that genes involved in fusion events — when two separated genes in a genome are merged or fused in another one — are expected to be functionally related. Also known as Rosetta Stone proteins, fused proteins often present separate domains that are homologous to separate but functionally related proteins. Marcotte et al. [37] developed a statistical measure for the significance of predicted functional linkage between a pair of proteins. The authors highlight the limitations of looking only for orthologous proteins to look for meaningful Rosetta Stone proteins, and turn to the broader concept of homology for this task. Then, they develop an association scoring function based on the hypergeometric distribution that measures the probability of a given number of fusion events between a given pair of proteins.

2.3.3 Function prediction based on phylogenetic trees and profiles

Methods in this category exploit evolutionary relationships between organisms to detect functional similarities between genes.

Some of the methods in this category exploit the concept of a phylogenetic profile — a binary vector that stores the presence or absence of a particular gene in a genome. These methods follow the hypothesis that proteins that participate in the same pathway or molecular complex in the cell are under pressure to evolve together to preserve their function, and therefore the comparison of phylogenetic profiles will be informative for functional characterisation. Wu et

al. [38] present a method for relaxing a previous condition that required that only identical profile pairs would be used for inference. This is done by calculating the probability distribution of a given number of chance co-occurrences of a pair of non-homologous orthologs across a set of genomes, i.e. without biological pressure. When inferring functional groups with this probability measure, they show a 30-fold increase in coverage at the same confidence level when compared to restricting the inference to identical profiles. Enault et al. [39] proposed extending the definition of phylogenetic profiles to include real values. The new vectors now encode the normalised BLAST score, denoting the bast match for a protein in a genome. The new encoding is effectively a relaxation of the profile to cases in which an exact match cannot be found in a genome, but perhaps relatively close genes are indeed present. This real-valued representation provides better performance.

Another group of methods that rely on phylogenomics make use of phylogenetic trees. There are some difficulties associated with using trees, however. First, the comparison between trees is very intricate. Second, the way to "correctly" build a phylogenetic tree is still being debated, and methods that rely on trees must decide which algorithm to use to define the tree [2]. Nevertheless, some attempts at using phylogenetic trees to elucidate function exist. Jonathan Eisen [40] reminds us that despite homology being closely related to sequence similarity, they are not the same. He suggests using phylogenetic trees in order to identify likely gene duplication, which allows the division between orthologs and paralogs. Uncharacterised genes can be assigned a likely function if the function of any ortholog is known. Furthermore, tree reconstruction techniques can be used to infer the function of uncharacterised genes by identifying the evolutionary scenario that requires the fewest functional changes over time. By reconstructing the tree in such a fashion, the underlying assumption is that the most likely evolutionary path is the one that requires the fewest functional changes [40], suggesting the "path of least resistance". As another example, SIFTER [41] builds a phylogenetic tree of homologs, identifying duplication events in the process. Then, it overlays GO annotations on this tree and it propagates the annotations to the root. Finally, it propagates the annotations to the leaves, effectively inferring function for the uncharacterised genes.

Phylogenetic profiles and trees can be used together. Vert [42] uses Support

Vector Machines (SVMs) to learn protein function from phylogenetic profiles, using the phylogenetic tree to define a kernel that calculates profile similarity. Narra et al. [43] achieved better performance by extending the profiles with new bits corresponding to the internal nodes of the trees, which allows the direct use of e-values instead of having to set cutoffs to derive binary profiles.

2.3.4 Function prediction based on protein structure

Analogously to sequence-based methods, in this category the function is predicted by establishing a structural similarity between proteins. The idea is that the structure of a protein is under more biological pressure than the sequence to preserve function. This is evidenced by the presence of remote homologs, which have been shown in the 1960s to have different sequences, but similar structure [44]. Similarity can be calculated by comparing the two structures in their entirety or only in parts.

In similar fashion to methods based on sequence similarity, many of the structure-based methods attempt to provide a score to a three-dimensional alignment between two proteins. Alignment methods attempt to maximise the number of residues in the alignment while minimising the distance or similarity measure. Some of the most popular alignment-based methods are SALIGN [45], SSM [46], MAMMOTH [47], CE [48], SSAP [49], VAST [50], SARF2 [51], and DALI [52]. For function prediction, the I-TASSER suite [53] provides structure based functional annotations, which are derived from a structural generative model in four general steps: threading template identification, iterative structure assembly simulation, model selection and refinement, and finally functional annotation. The structure models with the highest confidence scores are matched against the BioLiP [54] database of ligand-protein interactions to detect homologous function templates.

Performing a three-dimensional structural alignment-based database search is computationally very expensive, especially for large collections of proteins. This motivated the development of alternative approaches that do not rely on alignment. Wohlers et al. [55] propose a metric to compare two protein structures based on their contact map representation. The contact map represents the

distance between all possible amino acids pairs of a three-dimensional protein structure using a bi-dimensional matrix. The authors use these matrix representation to establish structural similarity [55]. The core of the idea is to exploit the representation of the structure to optimise database searches by filtering out structures that are unlikely to share functional features with a query that does not involve three-dimensional alignment. The main advantage is that a large corpus (a big database of structures) can be queried using bi-dimensional alignment techniques after reducing the candidates to a manageable size. The structural representation is crucial for filtering methods, as it will determine which metrics can be used to perform the filtering. For example, ProtDex2 [56] constructs feature vectors of the relationships between secondary structure elements of all the 3D structures in the database, and then use these feature vectors to query their database efficiently. FragBag [57] is inspired on the bag-of-words from natural language processing. It represents the protein structure as a bag-of-fragments — a vector that counts the number of occurrences of each fragment — and measures the similarity between two structures by the similarity between their vectors. YAKUSA [58] creates a library of substructures based on the proteins backbone internal coordinates (α angles) to describe protein structures as sequences of symbols. Proteins are then rapidly queried by searching for the longest common substructures. PRIDE [59] compares protein structures via an algorithm based on the distribution of inter-atomic distances.

Another group of approaches use three-dimensional substructures rather than the structure of the whole protein. A substructure will be conformed of a particular three-dimensional arrangement of atoms that conforms a part of the 3D structure of the entire molecule. Because some substructures occur often in many proteins, this fact can be used to search for similarities between proteins based on their substructures. Structural motifs are used in a similar fashion to sequence motifs. The idea is to identify common structural components between sets of functionally related proteins. Then, structure-function signatures can be collected and proteins are assigned the function of the function-known motifs present in the protein. Examples of resources that deal with structural motifs are the Database of Structural Motifs in Proteins (DSMP) [60] and the Structural Motifs of Superfamilies (SMoS) [61].

With the increasing number of available structural information about pro-

teins, an increasing number of methods that exploit this data are being created. Computational efficiency remains a challenge that is bound to increase as the available data increases. This situation is a motivating factor to keep focusing research efforts into mapping the sequence to the function without the need to analyse the three-dimensional structure. Hopefully, future computational capabilities and improved algorithms will allow the efficient analysis of protein structure.

2.3.5 Function prediction based on protein-protein interactions

Relationships between proteins present a unique opportunity for function prediction methods. Particularly, when the data is represented in the form of protein-protein interaction (PPI) networks where the nodes represent proteins and the links represent binding between the proteins [63]. Graph theoretical concepts are applied to these networks to predict the function of a query protein based on its location on the network, a principle known as "guilt by association" (GBA) [64].

Experimentally, direct binding between proteins can be tested at high throughput via the yeast two-hybrid system (Y2H) or affinity purification coupled with mass spectrometry [2, 65]. Several databases that compile associations found by such experiments are available in resources such as DIP [66], BioGRID [67], IntAct [68], MINT [69]. Other resources exist that not only compile experimental interactions, but also predicted ones. The Michigan molecular interactions (MiMI) [70] and STRING [31] databases are prime examples of these.

A simple notion that can be exploited to predict protein function in networks is the transference of function to interacting neighbours. An early example of PFP predictions based on neighbourhood interactions was published by Schwikowski et al. [71]. They demonstrate that for a yeast PPI network of 2709 proteins 63% of the interacting proteins have a common functional assignment, and 76% were found in the same subcellular compartment. A more recent example, Guilty by Association in STRING (GAS) [72] performed well in CAFA. In general, neighbourhood based predictions use diverse metrics to transfer one or more functional assignments from the neighbourhood, the latter being strictly

limited to the 1-neighbourhood[7], or in some cases a slight relaxation of that limitation. For instance, Chua et al. [73] extend the GBA principle to level-2 neighbours. Wang et al. [74] propose an iterative algorithm that also exploits unannotated proteins and their interactions for the prediction. Gillis et al. [75] predict function using indirect connections on a gene co-expression network that is extended using self-multiplication. This extended co-expression network is then used to estimate the number of paths of a certain length connect a given pair of nodes in the network. A neighbourhood-based models that takes into account the scale-free property of the PPI is the Preferential Attachment based common Neighbor Distribution (PAND) method [76], in which a probability distribution of a neighbour-sharing event between any pair of nodes in a network is calculated based on the assumption that neighbour-sharing is constrained by the preferential attachment property [77]. This distribution is then shown to be very correlated to the observed probability in simulations of scale-free networks, and was used to construct new networks with more functionally reliable links than PPIs.

Another big group of PFP methods use clustering in protein networks to identify functional modules, which represent protein complexes. The idea behind clustering methods for PFP is to simply assign the most popular function to members of the cluster. The corpus of clustering algorithms applied to biological networks is vast. Here, I briefly describe a representative set of clustering algorithms involved in functional characterisation by protein complex identifications. MCODE [78] uses vertex weighting based on the clustering coefficient to measure the likelihood that the neighbourhood of a node is a clique. The Markov cluster (MCL) [79] clusters a protein network in which the edges represent the sequence similarity between pairs of proteins. This matrix is then used to simulate random walks by alternating the "expansion" and "inflation" operations. The "expansion" operator coincides with taking the power of a stochastic matrix using the normal matrix product (i.e. matrix squaring). The "inflation" operation corresponds with taking the power entrywise, and then scaling it so that the matrix is stochastic again [79]. Frey et al. [80] propose affinity propaga-

[7] 1-neighbourhood refers to the set of proteins that are directly connected to a protein i in a PPI network – i.e., they are one jump away from protein i. Similarly, 2-, 3-, ..., and n-neighbourhoods refer to the set of proteins that can be found by doing up to 2, 3, ..., or n jumps in the PPI network starting from protein i.

tion, which takes as input measures of similarities between nodes. Real-valued messages are exchanged between nodes until a high-quality set of exemplars and corresponding clusters gradually emerges. The Restricted Neighbourhood Search Clustering Algorithm (RNSC) [81], which searches for a low-cost clustering by first composing an initial random clustering, and then moving nodes to different clusters in a randomised fashion to improve the cost. The cost function that assigns a cost to each cluster can be user-defined, and the authors show good results with an integer-valued cost called "naive cost function" followed by a real-valued "scaled cost function" [81]. CFinder [82] uses the clique percolation method to locate the *k*-clique percolation cluster that it interprets as protein complexes, and is able to cope with overlapping modules. The clustering based on maximal cliques method (CMC) [83] first generates all the maximal cliques from the PPI networks, and then removes or merges highly overlapped clusters based on their interconnectivity. Highly interconnected clusters will be merged together using a user-defined threshold to decide whether at which level to merge the clusters. Repeated Random Walks (RRW) [84] uses random walks with restart for finding the highest affinity protein to a given cluster, and linearly combine precomputed random walks to reduce the computational complexity for large clusters. ClusterONE [85] builds on the concept of the cohesiveness score and uses a greedy growth process[8] to find overlapping groups of proteins in a PPI. It starts by growing groups with high cohesiveness from selected seed proteins, initially selecting the protein with the highest degree. Whenever the growth process finishes (adding any protein to the cluster lowers the cohesiveness score), the algorithm selects the next seed by considering all the proteins that are not included in a cluster, and selecting the one with the highest degree. The process finishes when there are no proteins remaining to consider.

Global optimisation-based methods predict protein function by considering the full topology of the network[9]. For instance, the bagging Markov random field framework (BMRF) [86] follows a maximum a posteriori principle to form a network score that considers pairwise gene interactions in PPI networks, and it

[8]A greedy algorithm uses a heuristic to determine its next step. It will make the locally optimal choice with the assumption that it will lead to a global optimum.

[9]The network topology is the arrangement of links and nodes in the network. The topology is very useful to describe the different properties of the network such as its degree distribution and regularity.

searches for subnetworks with maximal scores. Then, a bagging scheme[10] based on bootstrapping samples is implemented to statistically select high confidence subnetworks. Re et al. [87] proposed a gene ranking algorithm based on kernelised score functions that exploit the topology and structure of the graph, and also capture functional relationships between genes and provide a method to integrate a network from multiple biological sources. GeneMANIA [88] proposes a dedicated label propagation algorithm that can take advantage of negative annotations – i.e., it can use the knowledge that a protein is NOT involved in a particular function to make its prediction by down-weighting the association to a GO term for a particular protein. GeneMANIA builds a network by combining networks using a ridge regression. The COst Sensitive neural Network (COSNet) [89] predicts protein function by Hopfield networks – a special type of artificial neural network [90]. This approach is focused on dealing with the the unbalanced nature of GO annotations, with almost no negatives. The predictive capabilities of Hopfield networks are then extended in COSNetM [91] to take multifunctional genes into account. UNIPred [92] combines different biomolecular networks using a supervised algorithm that project nodes into a vectorial space in which they are linearly separated according to a function-specific score, once separated, these nodes are reintegrated into a new network. Subsequently, COSNet is used on this network to predict function. FunctionalFlow [93] uses an iterative algorithm that effectively performs a label propagation of the functional annotations in the graph.

2.3.6 Function prediction using gene expression data

Methods in this category make use of gene expression data. This data comes from experiments that measure the expression level of a gene at a given time under specific conditions. To properly study expression data, the data analyst must be aware of the challenges inherent to the nature of the experiment use during measurement. In particular, normalisation and visualisation should be

[10]Bagging is a technique used in machine learning to improve the stability and accuracy of algorithms. It is based on several bootstrapped samples that are used for learning and are subsequently aggregated. Typically, the outputs of running the learning algorithm on the bootstrapped samples are aggregated using the average for regression and the most popular class for classification.

done with care [94] to eliminate discrepancies that may arise in expression data measured across different labs and under different conditions or time frames.

When it comes to protein function, similarities between expression profiles of genes can indicate functional similarities. Walker et al. [95] did seminal work on functional characterisation of genes using GBA on the basis of a combinatoric metric association of co-expression, i.e. the authors built a binary protein-protein co-expression network based on gene expression data. They examined 40,000 human genes and discovered several previously unidentified genes associated with cancer, inflammation, steroid-synthesis and other processes, with the majority of the genes not showing sequence similarity to known genes.

For gene expression data, many clustering algorithms can be used to organise gene expression profiles, and many were specifically designed for gene expression data, such as CAST [96], which proceeds in two phases for building clusters one by one. In the first phase, elements with high affinity are added to a cluster (a protein has high affinity to a cluster if it is similar to other proteins in the cluster). In the second phase, elements with low affinity are removed from the cluster. The algorithm finishes when no changes are made to any cluster. Clusters built using this heuristic have been shown to preserve functional categories. Unannotated genes are associated with the function of the majority of genes in the cluster. Many of the developed clustering techniques deal with the usual challenges associated with clustering, e.g. how to define similarity between expression profiles and how many clusters to extract, or how to deal with overlapping clusters. For instance, Wu et al. [97] employ different clustering algorithms and annotates a cluster with the functional class with the smallest p-value, calculated from the fractions of classes of the different functional classes in the cluster, i.e. the p-value for each class is calculated taking into account its frequency in the cluster itself (how many proteins in the cluster are associated with each class), and only classes with the smaller p-value are used as predictions. Unannotated genes are then assigned the functional class of the cluster, associated with a confidence value based on the p-value of the cluster. Similarly, Swift et al. [98] uses consensus clustering, and proposes a robust clustering that seeks maximum agreement[11] between several clustering

[11]Maximum agreement between several algorithm is achieved by collecting the output of all the algorithms and picking the output that the majority of the algorithms pick.

algorithms by reporting only the co-clustered genes that are grouped together by the different algorithms. An objective function that rewards high agreement clusters and penalises low agreement ones is optimised using simulated annealing.

Although clusters of gene expression profiles can be informative about function, they might not be always coherent, as pointed out by Zhou et al. [99]. They investigate a graph-theoretic approach, in which genes are encoded as nodes, and edges connect genes with correlated expression profiles — a co-expression network. They carry on to conduct a simple experiment in which the shortest path between genes with the same GO term are analysed to check whether genes in the path belong to the same GO term, or GO terms that are ancestors or descendants in the ontology. The experiment shows high accuracy for mitochondrial and cytoplasmic genes in *S. cerevisiae*, but medium accuracy for nuclear genes.

Supervised learning exploit not only the expression data, but also already annotated genes. For instance, Brown et al, [100] compare different classifiers to learn functions from yeast gene expression data. Parzen window, Fisher's linear discriminant analysis (LDA), two decision tree classifiers (C4.5 and MOC1), and SVMs with different kernels were compared. They concluded that SVM with radial kernel performs the best. Mateos et al. [101] use multilayer perceptrons[12] for functional annotation on MIPS categories, and identify three sources of error: class size, heterogeneity, and Borges effect — the simultaneous membership of a gene in several functional classes — which highlights the difficult nature of PFP for machine learning algorithms. Recently, Makrodimitris et al. [102] developed a function prediction algorithm called Metric Learning for Co-expression (MCL), based on the hypothesis that when the purpose is to find similarly functioning genes, the co-expression of genes should not be determined on all samples but only on those samples informative of the GO term of interest. MCL was used as a baseline for CAFA-π, and outperformed all of the algorithms submitted by predictors to the challenge[13].

[12] A perceptron is a single artificial neuron. It performs a binary classification that maps its input $\mathbf{x}$ and return 1 if $\mathbf{w} \cdot \mathbf{x} + b > 0$ and 0 otherwise. $\mathbf{w}$ are the parameters or "weights" of the perceptron and b is its bias parameter.

[13] As a baseline, MCL [102] was not a competitor in CAFA-π, and had access to the experimental data available well after the prediction deadline.

2.3.7 Data Integration methods

Methods in this category exploit and integrate heterogeneous data to improve the predictions. This is a very broad category that falls under the umbrella of machine learning algorithms. I will follow the organisation suggested in the survey published by Shehu et al. [2], as data integration is inherently overlapping with the categories explained before.

An important group of methods for function prediction are based on building vectors of features from different sources of biological data. These vectors are subsequently analysed using existing machine learning techniques. Many of the methods mentioned before follow such an approach. Here, I mention some methods that also rely on a vector of features from heterogeneous sources, but that can not be properly classified above. For instance, Lobley et al. [103] predict protein function by focusing on intrinsically disordered regions[14]. Focusing on disordered regions makes a lot of sense. Their relevance for protein function comes from the fact that these regions could offer some flexibility to the 3D structure of the protein. For example, disordered stretches can allow movement between domains or can be sites of molecular attachment that become ordered on binding with another protein and give rise to function [103]. Lobley et al. investigated a total of 122 features extracted from proteins, and these cover 14 different sources of biological information about proteins. CombFunc [104] integrates sequence-based, PPI and gene co-expression features and these features are used in three different SVMs for different levels of the GO under the MF and BP subdomains.

A big group of methods involve combining classifiers. GOPred [105] combines different classifiers and evaluates the performance of different combination strategies such as majority voting, mean, weighted mean, and addition. Re and Valentini [106] integrate heterogeneous data sources with different aggregation techniques and predict function using an ensemble of SVM classifiers. They also demonstrated that simple ensemble methods are competitive and often outperform state-of-the art methods [107]. Obozinski et al. [108] propose "reconciliation" to address the drawback of making predictions for GO terms indepen-

[14]Disordered regions in proteins are defined as those which lack a stable well-defined 3D structure

dently. Several predictions are combined and calibrated into a set of predictions that is consistent with the GO structure. Schietgat et al. [109] use hierarchical multi-label decision trees that are combined via bagging, which is shown to better combine predictions from decision trees in comparison to random forests and boosting. Valentini [110] proposes exploiting the true path rule (TPR) that governs the hierarchical structure of ontologies such as GO and FunCat. The TPR ensemble technique is a hierarchical ensemble algorithm that obeys TPR. Classifiers are trained independently and are subsequently combined using an information propagation mechanism that follows a two-way information flow. Positive predictions traverse the structure recursively to the ancestors, while negative predictions affect the offspring.

2.3.8 Function prediction using text mining methods

Many other approaches exist for predicting function. Text-mining methods exploit the existing corpus of papers that study protein function to make predictions of functional associations. Renner et al. [111] use text mining to perform document clustering. Documents are compared by looking at terms that occur in them and then generating clusters of terms. The idea is that if two documents contain terms that belong to the same cluster, the documents probably describe the same phenomenon. To cluster the terms, the authors analyze their co-occurrence in documents, proceeding to build clusters starting at a term and adding terms that often co-occur with it, recursively. Eskin and Agitech [112] use an SVM classifier, combining several text and sequence kernels. Their method starts by learning a text classifier on textual annotations available for some proteins in the dataset. This classifier is used to predict the functional information of unannotated sequences. Then, a joint text-sequence classifier is trained on the expanded dataset using a kernel for both sequences and text. With this joint kernel, the classifier learns from both sequences and textual annotations, and the interactions between them.

2.4 Over-representation Analysis for protein function

The functional enrichment of sets of genes is a very important part of many methods in the PFP field [113]. These sets are very often differentially expressed genes that are up- or down-regulated under certain conditions [114]. The over-representation analysis identifies which GO terms are under- or over-represented in the selected set. The definition of over-representation varies a lot. One of the simplest definitions is simply to count the number of times a GO term occurs in the set (taking into account the duplicates, as these might contribute to the evidence of an association). Then, divide by the number of proteins in the set.

More rigorous (and arguably more useful) methods identify a GO term as over-represented only if the number of proteins associated with the term is enriched versus a background model as established by a probability distribution, i.e. whether the distribution of the associations of proteins to a given term is very different when compared to the distribution of other terms. Several background models have been proposed and are exploited by multiple bioinformatics tools for functional enrichment [115]. The most traditional strategy is to take the set of genes and iteratively test the enrichment of each GO term in a linear mode. Thereafter, the enriched annotations that pass a threshold on an enrichment p-value are reported ordered by their enrichment probability (enrichment p-value). Commonly, the enrichment p-value is calculated using well-known statistical methods such as Chi-square, Fisher's exact test, binomial, and hypergeometric probabilities. Most enrichment tools follow this strategy [116, 117, 118]. A limitation of such enrichment methods is that the relationships between terms is often diluted in the iterative process.

An increasingly popular strategy follows is the gene set enrichment analysis (GSEA) [119]. This is, to follow a "no-cutoff" strategy that takes the set and computes a maximum enrichment score (MES) from the rank order of all gene members in the GO term. Then, enrichment p-values can be obtained by matching the MES to randomly shuffled MES distributions. Tools that follow the GSEA have the nice property that it does not require the user to pre-select a set of interesting genes, and that the expression values present in the experiments are already integrated into the enrichment p-value [115]. Disadvantages of GSEA-

like methods are that for many biological studies, establishing the right score to rank the genes is not an easy task. Another possible disadvantage emerges when dealing with multiple sources of data. The GSEA strategy is driven mostly by the fold change of its ranked genes, but biologist know that small changes on some genes (e.g., those involved in regulatory processes) could lead to big downstream biological consequences. Conversely, a big change in metabolic genes may be a consequence of smaller, but important, regulatory events [115].

A third category of enrichment tools follows a modular approach. They follow the traditional motivation for enrichment as in the first strategy, but take into account gene-gene or term-term relationships, often from the hierarchical structure of the GO. New versions of widely used tools and newly released enrichment tools offer the user the option of choosing which analysis to perform [120, 121, 122]. If possible, a modular approach is preferable to other techniques, as it better captures the hierarchical nature of the GO, providing a larger biological picture with the enrichment [115]. The disadvantage of modular approaches emerge mainly with GO terms or genes without strong relationships to neighbouring GO terms/genes, as these could be left out of the analysis. Also, like in the first strategy, the quality of the pre-selected set of genes will impact the analysis [115].

Unfortunately, there is no single rule of thumb to compare or decide which strategy to use for a given project. This decision will depend on many factors such as the heterogeneity (or homogeneity) of the sources of data, whether the researcher is interested in a group of (already pre-selected) genes, and the IT expertise required for running the analysis tools correctly.

3

Sequence 2 Function

3.1 PFP for newly sequenced organisms

As mentioned in section 2.3, for newly sequenced organisms most of the sophisticated PFP methods are not applicable. This is mostly due to the unavailability of data for such organisms, which are required by the best performing methods.

Naturally, this results in predicting protein function using simpler methods that rely exclusively in finding sequence similarities. In this chapter, I show that by transferring information from well studied organisms, many sophisticated techniques can be used even for newly sequenced organisms. I focus on how to enable the use of label propagation in networks for PFP. Methods in this category have shown a great potential in many applications, including PFP [123]. In the context of protein networks, these techniques are fundamentally based on the "guilt-by-association" principle: a protein will share many molecular and phenotypic characteristics with closely connected proteins in the network. Function is considered to be such a characteristic.

For protein function prediction, a label propagation algorithm generally requires two things. First, a protein network, in which nodes are proteins, and links can be relationships between them of various types. Second, information about a subset of the proteins in the network in the form of labels associated to the proteins. These labels will be propagated from their original proteins (nodes) to other proteins using the edges of the network as the communication channel [123]. Consider the following diffusion process: a label acts as a fluid, that will flow to neighbouring proteins with a strength proportional to the weight of the edge. After some time, the diffusion process will stabilise, with the label reaching proteins that were previously unlabelled. The amount of fluid (label) that reaches a previously unlabelled protein will be related to the probability of the association between the label and the protein.

Despite being conceptually simple, the diffusion process will strongly propagate labels to proteins close to the source. Distant nodes could be reached in collaboration if the label flows from multiple sources and reaches the same protein from several different links, capturing the topological features of the network [123].

S2F (Sequence to Function), is a novel method for the characterisation of newly sequenced organisms based on network propagation. Our framework focuses on exploiting functionally relevant data available for model organisms and transferring it to newly sequenced organisms. S2F uses a novel label propagation algorithm, tackling the lack of data by inferring both the seed and the networks to propagate them. S2F provides new solutions to the problems of integrating an initial set of labels (the seed), how to build a network, and how to propagate the labels on the network.

3.2 The S2F Framework

The framework consists of three main steps:

1. **Seed Inference:** The first step is to create an initial set of predictions based solely on the sequence. This is done by combining predictions from InterPro [16] and HMMER [9].

2. **Network Inference:** In the second part, we build a protein-protein network by combining several types of interactions (links) into a single network. Links are transferred from all organisms available in the STRING database [31]. Known as interologs [4, 124], these transferred interactions are integrated into a single network.
3. **Label propagation:** The final step of the process is to propagate the initial labelling into the network. For this, we propose a new label propagation algorithm.

These steps are illustrated in Figure 3.1, and will be explained in detail in this section.

3.2.1 Building the seed

The emergence of sequence-based function prediction techniques motivated the development of large pipelines for the functional characterization of newly sequenced genomes. These include InterPro [16], HMMER [9] and GenDB [125] among with the ones specifically designed for microbial gene function characterization such as BASys [126], PUMA2 [127], MaGe [128], AGMIAL [129], IMG [130] and PIPA [131]. Among these, PIPA and InterPro adopt a meta-approach where the results from various homology and motif search techniques such as HAMAP [132] and Pfam [21] are systematically integrated to obtain a single function prediction. We use InterPro as it provides a very complete set of sequenced-based tools that can easily be integrated to our framework. InterPro [16] integrates 13 databases that give functional prediction based only on sequence data and some trained models over manually curated data. We consider the output of every model as binary, giving the user a set of terms that have been predicted with probably high precision. For every InterPro model $k \in \{1, 2, \ldots, m\}$, we have a set of predictions R^k. We use these models to obtain an initial matrix $R \in [0, 1]^{n \times t}$, where n is the number of proteins in the organism, and t is the number of GO terms. R is constructed by integrating model R^k into a single matrix

$$R_{ij} = \frac{\sum_{k=1}^{m} R_{ij}^k}{m}$$

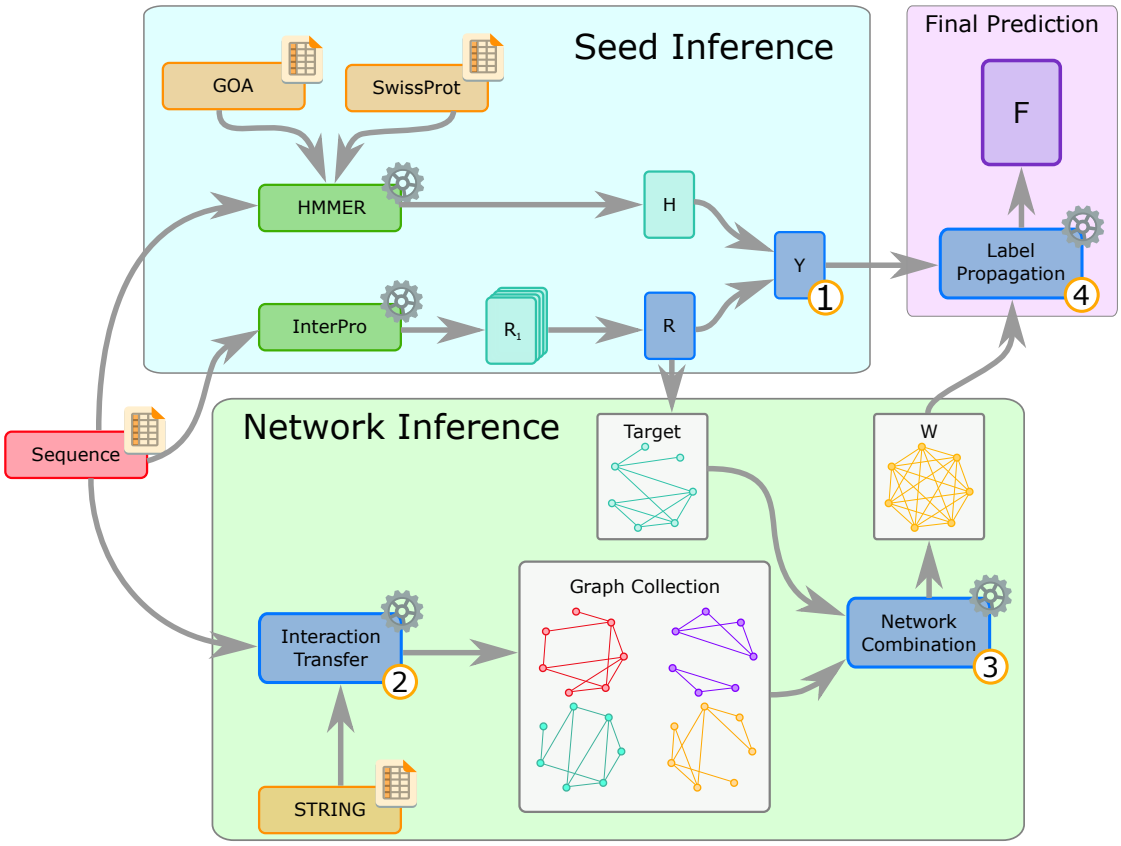


Figure 3.1 – *Diagram of the entire S2F framework. External datasets are STRING, UniProtKB/GOA, and UniProtKB, shown in orange. The leftmost element is the Input of the system: the set of amino acid sequences of the target organism. Running HMMER using sequences in UniProtKB/SwissProt with experimental annotations as the database results into the HMMER seed H. Running InterPro results in a collection of seeds that correspond to every model, this collection is aggregated and a single InterPro seed R is produced. The initial guess Y, that will be propagated later on, is calculated by a linear combination of H and IP. The lower part of the diagram shows the building of the network. A collection of networks is obtained by finding interologs between the target organism and every organism with relationships reported in the STRING database. The network collection will be combined into a single network. The resulting network W and the initial guess Y are finally fed to our label propagation algorithm that outputs the final prediction F.*

After normalising by m (the number of models for which InterPro gives a prediction for the organism), each value in R represents the likelihood that the association is present in all models in InterPro. To keep the predictions consistent with the GO structure, all matrices R^k are up-propagated using the true path rule.

The lack of availability of enough training data for some GO terms (only a few proteins annotated) means the InterPro models are not trained to output

those terms to avoid false positives. This results into many GO terms (generally the rare ones) not being predicted by InterPro.

To increase the coverage of the seeds on top of the predictions by InterPro, we combine its output with the one produced by HMMER [9]. HMMER is a method based on profile hidden Markov models. It will build profile hidden Markov models for every sequence in the query, and compare against profiles found in the database, producing an e-value of the match. We use HMMER against sequences in UniProtKB/SwissProt [5], which only contains protein sequences that have been experimentally validated. For every hit in HMMER with an e-value $\leq 1e-8$, we copy any functional annotations from the protein in UniProtKB/SwissProt to the protein in the target organism. The result, after up-propagating the assignments with the true path rule, is a matrix H of the same dimensions as R. We finally combine both seeds into a single matrix that will be propagated on the network afterwards. We do it using a simple linear combination, and the initial guess Y is therefore defined as:

$$Y = \alpha R + (1 - \alpha) H$$

In our experiments, we set $\alpha = 0.9$. This gives more importance to InterPro, a decision justified by the fact that the InterPro predictions contained in R are made by combining several models, while those in H are a similarity based on a single method. Developing a stronger analysis of the tuning of the value of α is part of my future work.

3.2.2 Building the network

The strategy to build the protein-protein network is divided in two steps: first, we transfer interactions from several organisms into several networks, and then these are combined into a single network.

Interaction Transfer

Several types of protein-protein interactions have been shown to be relevant as functional associations [31]. We consider the case when the *target* organism has

been recently sequenced, and therefore no information is available regarding any of the interactions between its proteins. Consequently, we propose here a methodology to transfer interactions from different *source* organisms for which information is available. Consider the diagram in Figure 3.2, the organism label "NEW" represents a collection of protein sequences (the nodes) without any interaction information (all links are missing). A source organism, labelled "SOURCE" in the figure, contains such protein-protein relations. The idea is to establish a rule that allows us to transfer these links from one organism to the other. We use STRING [31] as the dataset from which we transfer this information.

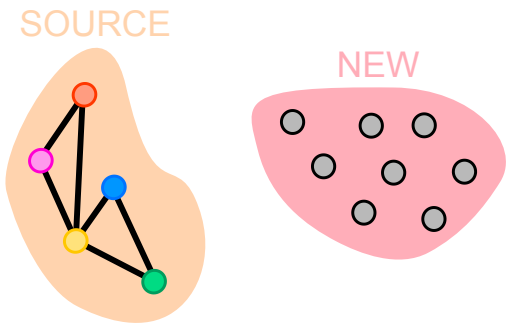


Figure 3.2 – *The network for the target organism (labelled "NEW") starts with no links, only the nodes are available. A source organism (labelled "SOURCE") from the STRING database is used to predict these links.*

To transfer interactions, we use the concept of interolog as defined by Yu et al. [4, 124]. Given a pair of interacting proteins A and B in the source organism, and a second pair A' and B' in the target organism such that A and A', and B and B' are found to be orthologs[1], we transfer the link in the source organism to the target organism – i.e., we copy the weight of the link in the source organism to the target organism (see Figure 3.3). We use this algorithm to transfer existing interaction evidence from well-studied source organisms to the newly sequenced target organism. Using organisms with experimentally derived interactions in STRING as source, and the newly sequenced organism as target, we build one transferred network W^r per interaction type r.

At the time of writing, STRING [31] provides a database that compiles several types of interaction between proteins for 5090 organisms totalling

[1]Note: For the link transference, we do not require that the pairs are experimentally validated orthologs. We simply consider a set of stringent similarity conditions (explained later in this chapter) and call orthologs to any pair of proteins that meets our similarity conditions.

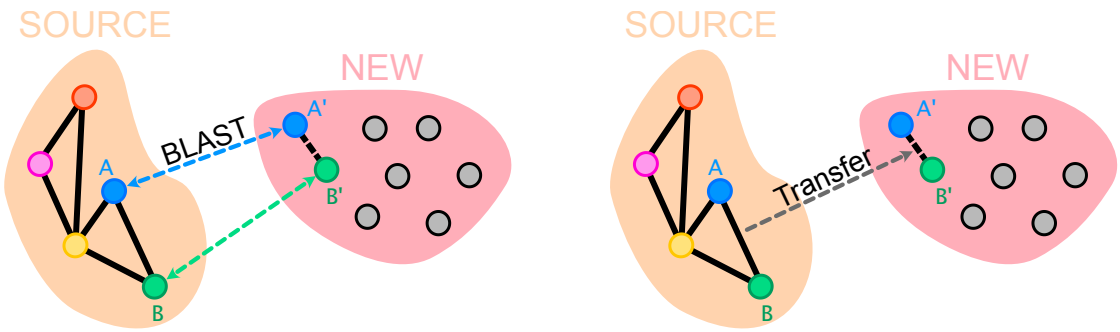


Figure 3.3 – *The interolog process. A link is transferred between two pairs of proteins are found to be orthologous. An iterative process can be used to transfer links from all organisms for which protein-protein relations have been established.*

3,123,056,667 interactions divided in 7 types: neighbourhood, fusion, co-occurrence, homology, experiments, co-expression, textmining, and database. Each interaction is annotated with a score that ranges from 0 to 1 and represents the confidence that STRING assigns to the interaction. STRING's scores represent the probability of that association which is calculated by combining the probabilities from all the sources of evidence included the database[2] [32]. For genomic neighbourhood, gene fusion events, and co-occurrence, periodic systematic genome comparisons are done against UniProtKB/SwissProt. For all other types, the confidence scores assigned to each predicted association are derived by benchmarking the performance of the predictions given a common reference set of trusted, true (experimentally validated) associations. The benchmarked scores generally correspond to the probability of finding the linked proteins within the same pathway in the Kyoto Encyclopaedia of Genes and Genomes (KEGG) [133]. We aim to transfer high-confidence interactions to the target organism, while maintaining high-connectivity in the network.

The idea that protein-protein interactions are conserved between species has been used in different methods, such as [134] and [135]. Prior to the publication of STRING, Yu et al. [4] did seminal work in the transference of networks. In their work, interactions are transferred with high precision using the criteria for interolog-mapping proposed by Walhout et al. [124]: Given an interacting pair of proteins A and B in the source organism, and two proteins A' and B' in the tar-

[2]The exact calculation adds the probabilities for each source of evidence. To each source of evidence, a "prior" has been added to account for the probability that two randomly picked proteins are interacting. This "prior" is subtracted from each score, and then added back after all the scores are combined [32]

get organism, transfer the weight between A and B to $A^{'}$ and B' if A and A', and B and B' are found to be orthologs. Operationally, orthology is considered if some conditions are met. First, mutual best hit with BLAST, and both e-values less than 1×10^{-10}. Second, at least 80% percent identity in both directions (to avoid transference between multi-domain proteins with different domain structure). Finally, an additional condition is introduced in [4] to achieve high precision: a high threshold on the geometric mean of both percent identities (highest precision above 80%), which the authors call "joint identity". The "joint identity" condition was validated with experimental measurements by Yu et al. and has proven to achieve very good accuracy [4].

Various approaches for interolog transfer use a variant of this method [136, 137] where orthologs between source and target are computed, and then weights are mapped from one organism to the other.

We use these concepts to transfer the likelihood of functional interactions between proteins reported in STRING. By transferring from several organisms, we increase the coverage of the target organism. We relax the third condition on the orthologs (highest precision score above 60%), and we have a criterion to determine the most suitable edge when multiple candidates exist for a given interolog. Algorithm 1 details our method for building a collection of transferred networks for the target organism.

To construct the collection of networks to integrate, we consider only those with at least 3 transferred edges. This is, for every interaction type k in STRING, if at least 3 edges are transferred, we count with a network W_k with interactions of type k that are collected from every organism in STRING.

Finally, we add a homology network to the collection to make sure the network is connected and make the prediction more stable. This homology network is obtained by calculating a pairwise homology score between all the proteins in our target organism. The motivation for the homology network is that the label propagation will benefit from a network with a minimal number of connected components. This is because if the network has isolated "island" components, a label starting in such an island will have limited reach. The Ho-

Let T be the target organism
foreach *type of network i, initialise* $G_i = \emptyset$ **do**
  Score[] = {}
  foreach *source organism S in the collection* **do**
    Find pairs of orthologs proteins $\langle S_a, T_a \rangle$ and $\langle S_b, T_b \rangle$ using mutual best hit by BLAST subject to:
      - e-values $\leq 1 \times 10^{-6}$
      - mutual percent identity $\geq 80\%$
      - geometric mean between percent identities $\geq 60\%$

    if $\exists$ weight$(S_a, S_b) \in$ *network i in S* **then**
      NewScore = max(BLAST(S_a, T_a), BLAST(T_a, S_a), BLAST(S_b, T_b), BLAST(T_b, S_b))
      if $\nexists$ Score$[(T_a, T_b)]$ **or** NewScore < Score(T_a, T_b) **then**
        Score$[(T_a, T_b)]$ = NewScore
        Set edge(T_a, T_b) with value weight(S_a, S_b) in G_i
  Add G_i to the network collection of organism T

Algorithm 1: S2F interolog transfer

mology matrix W^h is defined as:

$$W^h_{ij} = -\log\left(\frac{\text{e-value}_{ij}}{11}\right)$$

where e-value$_{ij}$ is the BLAST e-value between proteins i and j. The reason to divide e-value$_{ij}$ by 11 is that we consider 11 to be an unreasonably high e-value, and so it is a safe "maximum e-value" for our purpose. The – log operator allow us to turn the W^h matrix from having higher values for more similar proteins and low values for proteins with low similarity.

The homology matrix is important, as it will bring the homology relation (which has been extensively used in PFP) to the label propagation procedure. Also, by including this matrix to the combination procedure explained in the next section, we ensure that the final network will be connected, even if it is by a very weak link. This will give S2F the change to propagate labels to every part of the network.

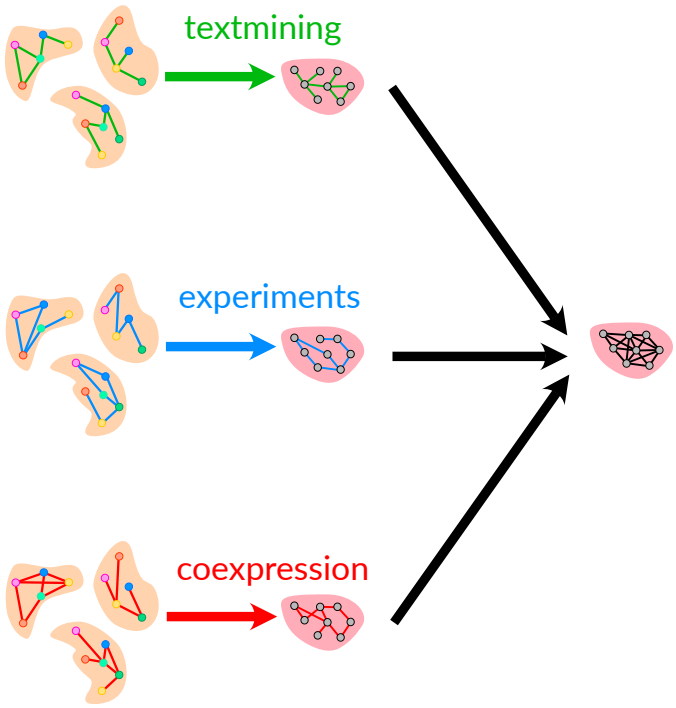


Figure 3.4 – *Overview of the network combination. Several types of relations are transferred using the method described in the previous section. With the collection of networks available to the target organism, our network combination procedure is used to build a single network.*

Network Combination

With several networks available, we now face the task of combining them into one cohesive network in which to diffuse the seeds, as depicted in Figure 3.4. This is not a simple task and many methods exist that integrate multiple networks [138, 88, 139]. For example, GeneMANIA [88] integrates its network by taking a weighted average of the individual functional association networks, and learns the weights of its combination using a ridge regression on the initial set of known labels. Since we have no prior functional information, we propose here a new solution for combining multiple functional networks that does not rely on a set of known labels. It is important to note that this network combination is necessary because we are working under the assumption that no experimental link has been discovered for the target organism. Nevertheless, when this assumption is not true, it is highly likely that performing the combination described in this section will increase the predicting power of S2F. Importantly, our network combination is not limited to the networks available in STRING.

Any protein-protein network can be added to the collection easily.

Given m networks $W^d \in R^{n\times n}$ $(d = 1, 2, \ldots, m)$, where the (i, j)-th entry of W^d represents the strength of the interaction between proteins i and j in network W^d, we aim to build a combined network W that integrates every W^d by a linear combination. To determine which coefficient to use for each network, we use a target network T, and optimise the coefficients of the combination so that the difference between T and W is minimal, in a similar fashion to GeneMANIA. Our target network, however, is built taking into account all the GO terms included in the R matrix (InterPro seed) simultaneously. We define the target network as the Jaccard coefficient of R after applying a threshold τ:

$$T_{ij} = \frac{\left|N_i \cap N_j\right|}{\left|N_i \cup N_j\right|}$$

where $N_i = \{k|R_{\mathrm{ik}} > \tau\}$ and $N_j = \left\{k\middle|R_{\mathrm{jk}} > \tau\right\}$ are the sets of all GO terms associated to proteins i and j respectively. The threshold τ on the R matrix allows us to consider the associations that were found in many InterPro models.

We compute the combined network W by a linear regression, minimising the following objective function:

$$\left(\widehat{c}, \widehat{b}\right) = \underset{c,b}{\operatorname{argmin}} \sum_{i,j} \left(b + \sum_{d=1}^{m} c_d W^d_{\mathrm{ij}} - T_{\mathrm{ij}} \right)^2$$

where b is used to remove the bias in T. This linear regression can be solved efficiently, and we can interpret $\widehat{c}_d$ as representing how much model d contributes to the combination. To avoid sparsity, a homology network is added to the collection of networks used in the combination.

The target network T can be interpreted as representing functional similarity between proteins. Note that T may lack information about the functional relationship of proteins, since it contains only information from InterPro and two proteins that do not have any prediction in R will have a functional similarity of zero in the target network. Nevertheless, we expect that the combined network will correct for this since the transferred interactions will inevitably make the functional similarity different from zero.

3.2.3 Label propagation

The final step in the framework is our label propagation procedure. Label propagation is at the heart of the procedure. The potential of this technique is made evident by the number of available methods that have been developed [123]. However, the unavailability of initial information (functional annotations) and a suitable network (a reasonably well connected protein-protein network) has limited its use on newly sequenced organisms that do not have either of those components available until many experiments are performed.

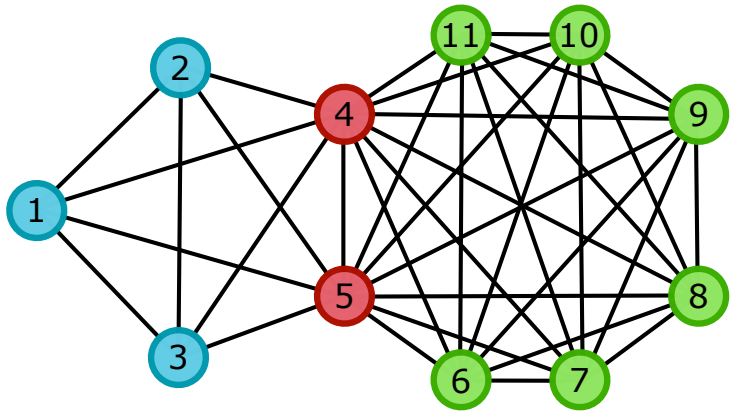
We propose a novel label propagation method inspired by the Consistency Method (CM) [140]. Our development upon the original CM method is motivated on what we call the "problem with overlapping communities", described in the following section.

The problem with overlapping communities

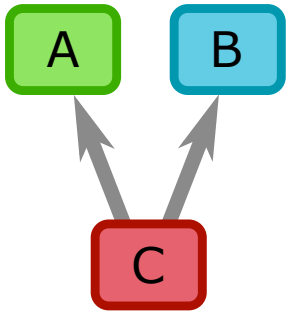
Protein functions are ultimately performed by protein complexes. These complexes can be found in a protein-protein network as communities that often overlap [85]. Intuitively, one might expect that proteins in the intersection of two communities share many properties when compared to their neighbours. Consider the network in Figure 3.5A, where proteins 4 and 5 are the intersection of the communities formed by nodes 1-5 and 4-11. Notably, nodes 4 and 5 share more communities with each other than any other pair of nodes in the network (i.e. they conform an "overlapping community"). To our knowledge, our label propagation method is the first to model overlapping communities, which avoids the problem of over- or under-propagating labels when communities overlap.

To demonstrate the problem, we ran three diffusion methods in the toy network depicted in 3.5A with a fixed smoothness parameter $\lambda = 1$. Here, λ is a parameter that is used in these label propagation methods to determine how strong is the "smoothness constraint" of the objective function (see next section for a full explanation). These methods are the consistency method (CM) [140], GeneMANIA [88], and S2F. Due to the lack of true negatives in our data (i.e., GO annotations with the NOT modifier), we use GeneMANIA without its bias

A



B



C

Method	10 → 11	10 → 5	4 → 5	4 → 11	4 → 1
CM	0.0622	0.0558	0.0619	0.0558	0.0632
CM-GeneMANIA	0.0646	0.0497	0.0488	0.0497	0.0619
S2F	0.0677	0.0498	0.0572	0.0498	0.0518

Figure 3.5 – *The problem with overlapping communities. A) A simple network that features two overlapping communities. Proteins 1-5 conform the first cluster, and proteins 4-11 the second. Our intuition is that in the case in which only protein 4 has a known function, then an ideal diffusion algorithm should assign a bigger score to protein 5 than any others because they are the intersection of the communities. In case protein 10 is the only one with known function, the algorithm should give a bigger score to protein 11 than to protein 5. All edges are assigned a weight of* 1 *B) The relations between the labels assigned to the network C) The resulting scores of running the different diffusion methods on the toy network, each column represents the direction of the diffusion in which* $A \rightarrow B$ *represents that protein A is the source of the label, and the scores for B are reported. CM and CM-GeneMANIA give unintuitive results in the case in which protein 4 is the only labelled protein (highlighted in yellow). Notice how they both give* $4 \rightarrow 5$ *a lower score than* $4 \rightarrow 1$ *and* $4 \rightarrow 11$ *respectively, which is not consistent with the communities arising from the topology of the network.*

setting, and we refer to the modified version as CM-GeneMANIA. Let us assign a function A to the cluster composed by proteins 1-5, a function B to the cluster made by proteins 4-11, and a more specific function C to proteins in the intersection (4 and 5). Additionally, to better model a plausible situation in GO, C is a descendant term from both A and B as depicted in Figure 3.5B. We consider two situations:

1. If we hide the function of every protein except 4, an ideal diffusion method should assign protein 5 a higher score compared to other proteins.
2. If we hide the function of every protein except 10, the ideal method should assign protein 11 a higher score than the one assigned to protein 5.

As shown in Figure 3.5C, we can see that both CM and CM-GeneMANIA produce unintuitive results in the case when an overlapping protein is the only one with the label. We observe then that both CM and CM-GeneMANIA suffer from the problem of ignoring the effect of overlapping communities, as they assign a greater score to proteins 1 and 11 respectively, and only S2F assigns the greater score to protein 5 (which shares a greater number of communities with protein 4).

Objective function of the S2F label propagation

To model the community effect in the interaction network, we use the Jaccard coefficient in a probabilistic setting:

$$J_{ij} = \frac{\sum_k W_{ik} W_{jk}}{\sum_k W_{ik} + \sum_k W_{jk} - \sum_k W_{ik} W_{jk}}$$

An element J_{ij} in matrix J relates to how much elements i and j belong to the same community in network W. Considering the W in the toy problem above, we observe some interesting cases[3]: $J_{12} = 1$, $J_{14} = \frac{5}{11}$, $J_{45} = 1$, $J_{19} = \frac{2}{11}$, $J_{89} = 1$. When the pair belongs exactly to the same community, they have a 1, (J_{12}, J_{45}, and J_{89}); if the pair share a community, but they do not exclusively belong to it, the value is smaller (J_{14}); if the pair does not share any community, the value becomes significantly small (J_{19}).

Formally, our cost function is:

$$Q(F) = \sum_{i=1}^{n} (F_i - Y_i)^2 + \frac{\lambda}{2} \sum_{i=1}^{n} \frac{1}{d_i} \sum_{j=1}^{n} J_{\text{ij}} W_{\text{ij}} \left(F_i - F_j\right)^2$$

where $J_{\text{ij}} W_{\text{ij}}$ models the community effect — the more i and j are connected by their neighbours, the more the diffusion rate;

$$\frac{1}{d_i} = \frac{1}{\sum_j J_{\text{ij}} W_{\text{ij}}}$$

[3]Note that we consider that the diagonal of matrix W is filled with 1 for the calculation.

is a normalisation factor that gives every protein the same ability of affecting others. The closed form solution that minimises $Q(F)$ is:

$$F^* = (I + \lambda L)^{-1} Y$$

where $L = D_{S2F} - W_{S2F}$ is the Laplacian of W_{S2F},

$$W_{\mathrm{S2F}_{ij}} = \frac{1}{2}\left(\frac{1}{d_i} + \frac{1}{d_j}\right) J_{ij} W_{ij}$$

and D_{S2F} is a diagonal matrix where the i-th element is $D_{\mathrm{S2F}_{ii}} = \sum_j W_{S2F\,ij}$. For organisms for which no functional data is available, we use the initial guess matrix R as Y.

This label propagation algorithm has the nice property that unlike other methods, it gives intuitive predictions when dealing with overlapping communities. And as the Consistency Method [140], it does not suffer from the inconsistency problem. Finally, a property of our diffusion method is its consistency, i.e. if $Y_j \geq Y_i$, then $F_j \geq F_i$.

Theorem 1. *If $Y_j \geq Y_i$, then $F_j \geq F_i$.*

Proof. First, we show that the matrix $(I + L)^{-1}$ is nonnegative. It is obvious that $B = I + L = I + D - A$ satisfies $B_{ii} > 0$ for each i and $B_{ij} \leq 0$. Also, for an all-one vector x, we have

$$Bx = (I + D - A)x > 0.$$

$(I + L)^{-1}$ is nonnegative (see Theorem 2.1 (2e,2f) in "A survey on M-matrices" by Poole and Boullion [141]).

Now we can use this property to show the conclusion. Assuming that $Y_j \geq Y_i$, we have $(I + L)^{-1}(Y_j - Y_i) \geq 0$, which means $F_j - F_i \geq 0$. □

In a similar way, it can be shown that our model and GeneMANIA have the same property. Furthermore, it can be shown that CM satisfies the property by that fact that the matrix

$$(I - \alpha S)^{-1} = \sum_{i=0}^{\infty} (\alpha S)^i$$

is nonnegative.

Let term j be an ancestor of term i in the GO, because we apply the up-propagation procedure to both InterPro and HMMER, we always have that $Y_j \geq Y_i$. By this theorem, our predictions about term i and term j satisfy $F_j \geq F_i$, and thus they are consistent.

Another important property of the diffusion model is that if $\text{sum}(Y_j) > \text{sum}(Y_i)$, then $\text{sum}(F_j) > \text{sum}(F_i)$. This is important because in both IP and HMMER, there are many false positive GO terms. However, a false positive GO term i may receive less annotation in both IP component models and HMMER – i.e. the overall initial guess for i, reflected in the number $\text{sum}(Y_i)$, is small. On the contrary, a true positive GO term j may receive more votes, and so $\text{sum}(Y_j)$ is larger. Therefore $\text{sum}(F_j) > \text{sum}(F_i)$ according to this property, which means that with a higher probability our prediction put j prior to i – this usually results in a better per-gene performance.

3.3 Evaluation

There are two main testing scenarios for protein function prediction. First, given a gene, provide a set of functions associated to that gene (*per gene* prediction). Conversely, given a function, provide a set of genes which perform that function (*per term* prediction). The first case is useful when the objective of the prediction is to characterise an entire genome. This will give a panoramic view of the functional landscape of the genome. The second case may be more interesting while studying a particular function, and the objective is to identify which proteins are involved in performing such function in the studied organism.

3.3.1 Organism Selection

As highlighted in section 2.2, the overwhelming majority of available sequence is from the bacterial superkingdom. Thus, we selected 10 bacteria that comply with the following criterion:

- The selected organism must have at least 10 functional annotations with

an experimental evidence code. This helps us to properly assess the performance of the method in a *per gene* setting.

- We need a reasonable diversity (i.e., annotations for multiple proteins in multiple GO domains) of GO terms to reliably estimate the performance in a *per term* setting. The second condition is that for every subdomain: biological process (BP), molecular function (MF), and cellular component (CC), we need at least 8 *popular terms*, i.e. terms that are annotated to at least 3 genes. This criterion allows us to be fair in determining the performance of S2F on every subdomain.

We checked our criteria using the set of annotations made available by UniProtKB/GOA [6]. The final list of selected organisms for evaluation is shown in Table 3.1.

NCBI ID	Name	Genes	Annotated genes	Popular BP	Popular MF	Popular CC
1111708	*Synechocystis sp.*	2413	137	103	31	38
122586	*Neisseria meningitidis B*	1405	16	58	19	16
208964	*Pseudomonas aeruginosa*	4403	918	719	224	51
223283	*Pseudomonas syringae pv. tomato*	5063	40	71	44	24
224308	*Bacillus subtilis*	3393	340	307	114	28
246200	*Silicibacter pomeroyi*	4094	65	29	42	11
272624	*Legionella pneumophila Philadelphia 1*	2070	18	33	8	10
83332	*Mycobacterium tuberculosis*	3287	2204	887	279	63
83333	*Escherichia coli*	3858	3313	1572	705	139
99287	*Salmonella typhimurium*	3695	110	190	42	34

Table 3.1 – *List of organisms that match the selection criteria selected from UniProtKB/GOA (downloaded on May 2018). Note that the number in popular terms may be bigger than the number in "annotated genes" as this criterion is applied after up-propagation.*

3.3.2 Black List

For evaluation purposes, and in order to remove any trace of experimental information associated to the target genome under evaluation, and therefore simulate it as a newly sequenced genome, we remove experimental data related to that particular genome from the input databases. Not only data for that particular specie is removed, but also for phylogenetically close species. Operationally,

we considered "close" organisms, all of those that have a NCBI taxonomy identifier that are descendants of the same "grandfather" node of our target organism. This is, if our target organism is a, we traverse the phylogenetic tree two levels up, and remove all descendants of the parent of the parent of a. A table with the blacklist for every organism in Microsoft Excel format can be found online at `https://paccanarolab.org/s2f`

3.3.3 Evaluation Metrics

There are many ways to evaluate the performance of predictions of protein function, which requires measuring the performance using several metrics [142]. We use these metrics to compare our predictions with a gold standard. For the selected bacteria, we use the available experimental annotations as the ground truth (see GOA files in Supplementary Data). We follow the analysis used in the CAFA challenge [3] and measure the performance of S2F using the $F_{\max}$, $S_{\min}$, the area under the receiver operating characteristic (ROC) curve (AUC-ROC), and the area under the precision recall curve (AUC-PR) metrics.

$F_{\max}$, and $S_{\min}$ are described in terms of the precision-recall and uncertainty-misinformation curves respectively.

The $F_{\max}$ metric is defined as:

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

where $\mathrm{pr}(\tau)$ and $\mathrm{rc}(\tau)$ are the precision and recall metrics respectively, when considering a threshold of τ for the prediction.

The minimum semantic distance $S_{\min}$ is defined:

$$S_{\min} = \min_{\tau} \left\{ \sqrt{\mathrm{ru}(\tau)^2 + mi(\tau)^2} \right\}$$

where $\mathrm{ru}(\tau)$ and $\mathrm{mi}(\tau)$ are the remaining uncertainty and misinformation metrics respectively, when considering a threshold τ for the prediction.

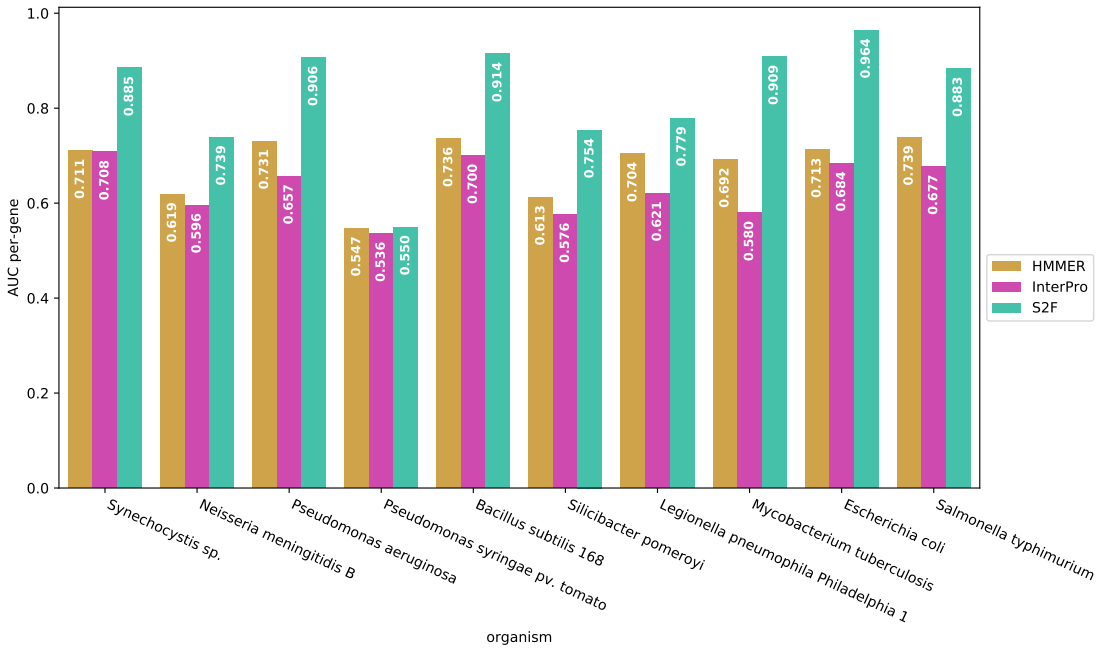


Figure 3.6 – *AUC-ROC for every organism (mean in a per-gene setting). Methods are HMMER, InterPro, and S2F (HMMER + InterPro + diffusion). For every organism, S2F gives the best score, suggesting that the functional knowledge transferred from other organisms and the network propagation are effective for predicting function of newly sequenced organism.*

3.3.4 Performance

We compare S2F against InterPro and HMMER, two sequence-based methods widely used in the community for the initial characterisation of newly sequenced organisms, as was pointed out in section 2.3.1. As can be seen in Figure 3.6, S2F is effective for predicting protein function when no information is available. Notice how for some organisms the improvement in terms of AUC-ROC when using the network propagation has significantly more impact than for other organisms. A possible explanation for this is the fact that the functional characterisation of organisms is not evenly distributed – i.e. Some organisms will be closer to "well characterised" organisms than others. For those organisms, it is more likely that S2F can transfer a generous amount of information to be used for the prediction. Contrastingly, an organism located in a "desolate" area will receive less information, and the predictions will be closer to the initial seed.

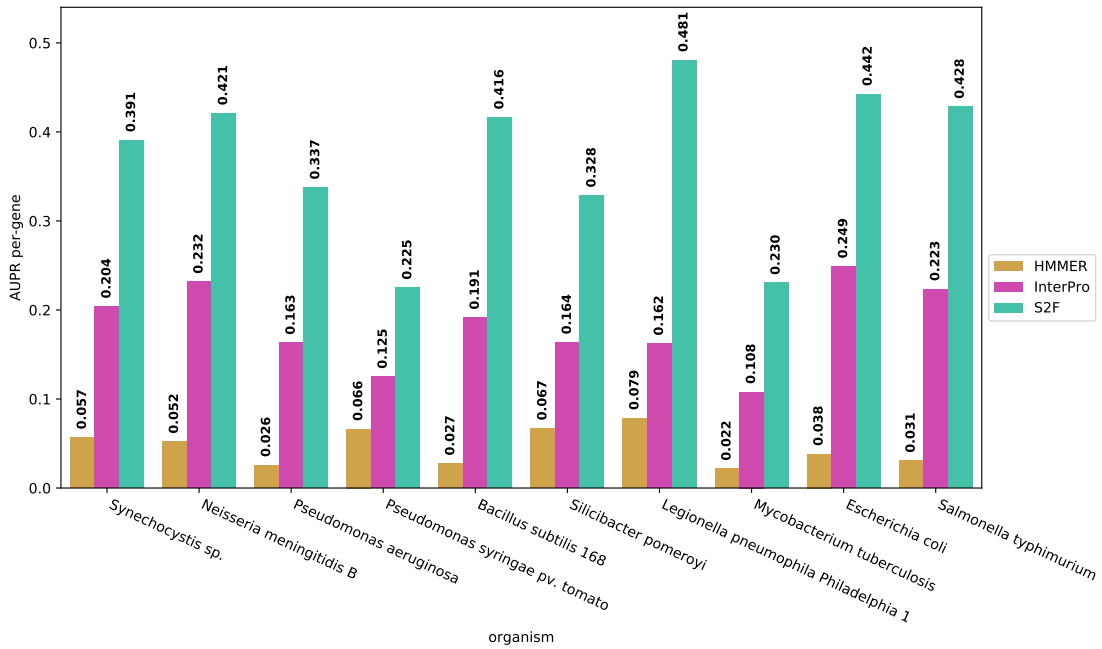


Figure 3.7 – *AUC-PR for every organism (mean in a per-gene setting). Methods are HMMER, InterPro, and S2F (HMMER + InterPro + diffusion). As for the ROC-AUC metric, S2F gives the best performance.*

Figure 3.7 shows the *per-gene* setting of the evaluation using AUC-PR. We observe that in both settings we outperform traditional methods. Figure 3.8 A and B show the performance evaluation using the $F_{\max}$ metric in *per-gene* and *per-term* settings respectively. Similarly, Figure 3.9 A and B show the evaluation using the $S_{\min}$ metric. In this case, the impact of the low coverage of InterPro becomes apparent in the *per-term* setting. A possible explanation for this impact in performance is that because InterPro predictions are very focused on precision, its low coverage results in a greater remaining uncertainty. An important thing to notice is that the network combination and label propagation amplify the predictive power of the original seed. This is reflected in the fact that S2F performs better for those organism in which HMMER and InterPro also perform better when compared to other organisms. Finally, in Figure 3.10 we can see a comparison of our label propagation and the Consistency Method. More results available in Appendix A.2 or, preferably, an interactive data explorer that is available on the project's website: `http://www.paccanarolab.org/s2f`.

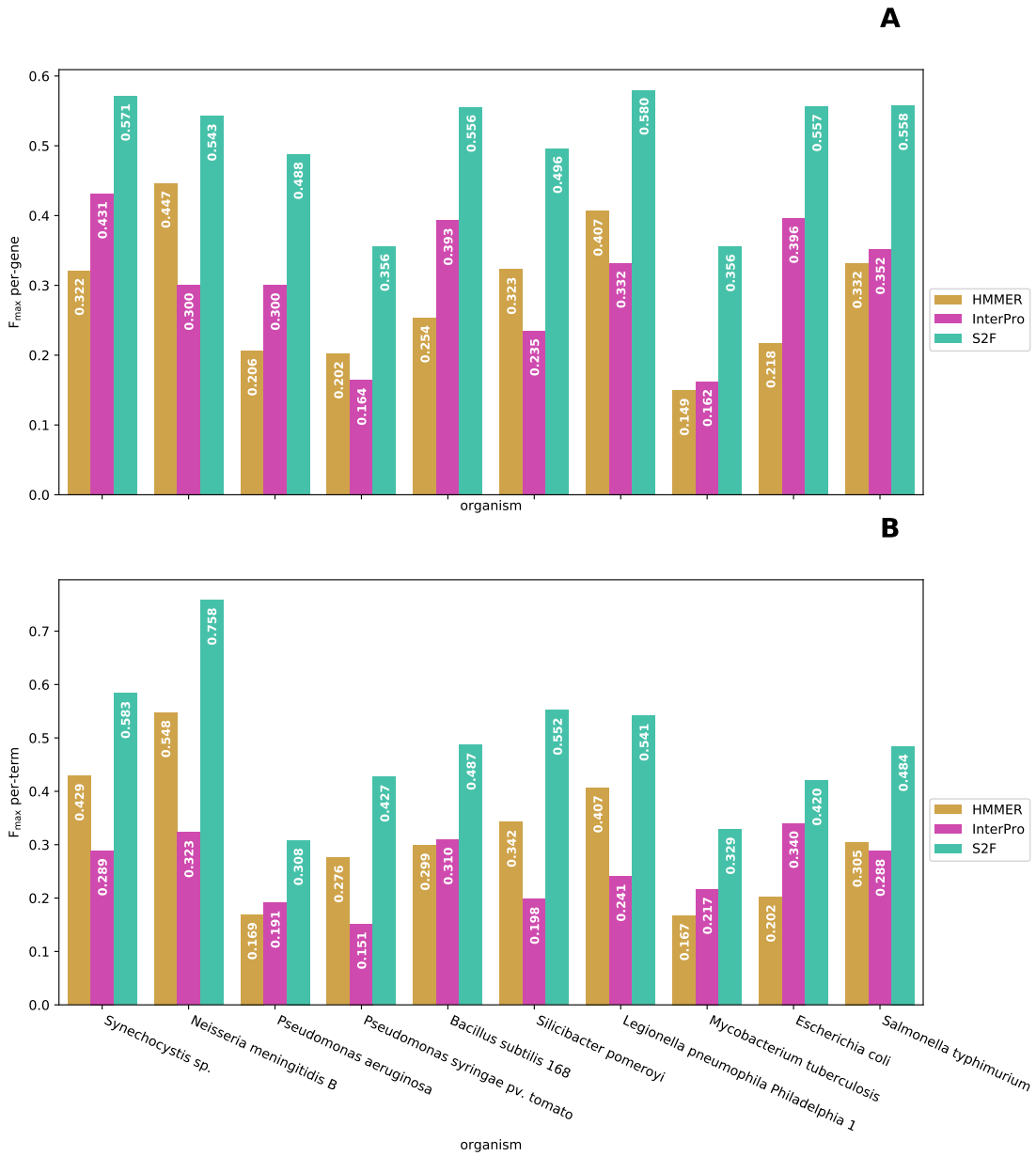


Figure 3.8 – *Performance comparison using the $F_{\max}$ metric A) per-gene setting B) per-term setting. We observe how S2F performs better for all selected bacteria.*

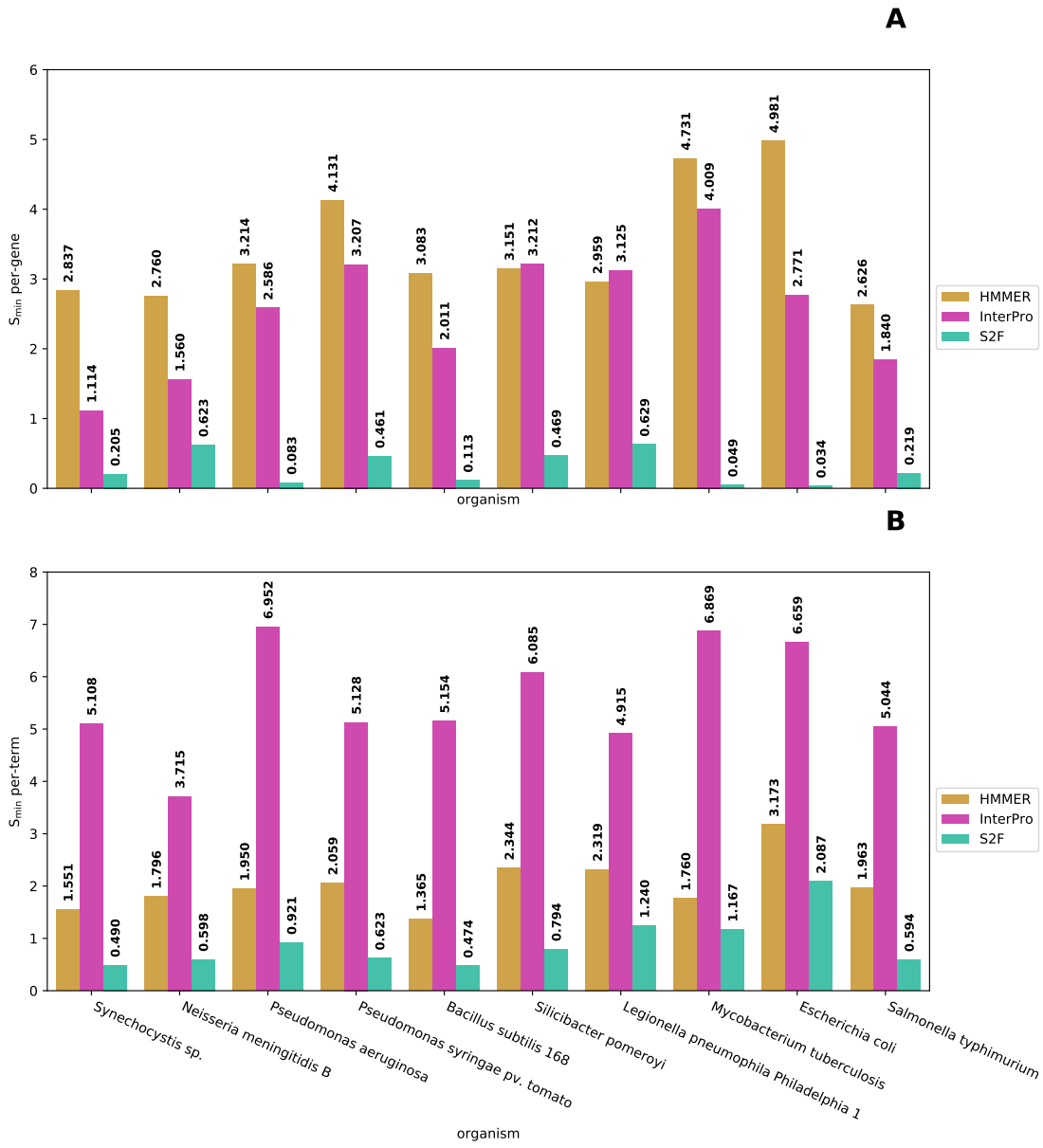


Figure 3.9 – *Performance comparison using the $S_{\min}$ metric (lower is better) A) per-gene setting B) per-term setting. S2F performs better in every case.*

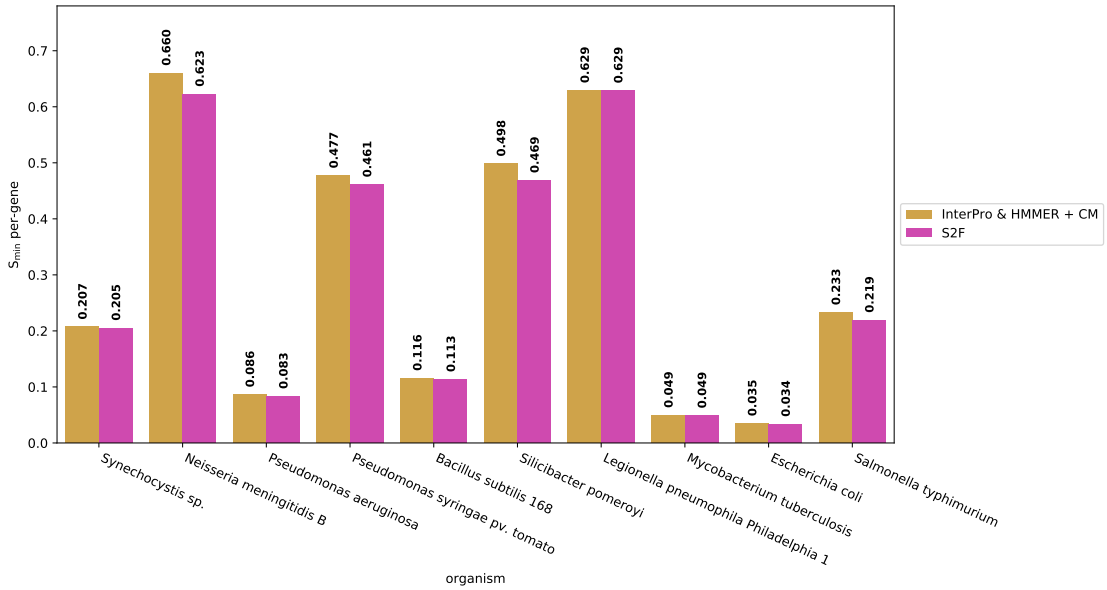


Figure 3.10 – *$S_{\min}$ performance comparison of S2F (our label propagation) and CM. The diffusion was run using our transferred and combined network, and the same initial seed (InterPro + HMMER). For this measure, the performance of our label propagation is always equal or better than CM for predicting protein function. The difference in performance on these 10 organisms is not statistically significant however, with a p-value of 0.92 using an independent two-sample t-test.*

3.3.5 Network combination coefficients

A label propagation algorithm can only perform well if the underlying network has meaningful connections – i.e., links between proteins that are likely to share function. In our case this depends entirely on the network transference and combination procedures. Figure 3.11 shows the coefficients learned by the combination procedure. Unsurprisingly, the homology coefficient is the highest in almost every case. This is expected, since the homology relation is something that can be calculated for every pair of proteins without further experiments once the sequences are obtained. Having a value for every pair results into a fully connected homology matrix. Therefore, this matrix is the most "malleable" one, i.e. it will provide many more links than the other networks in the collection.

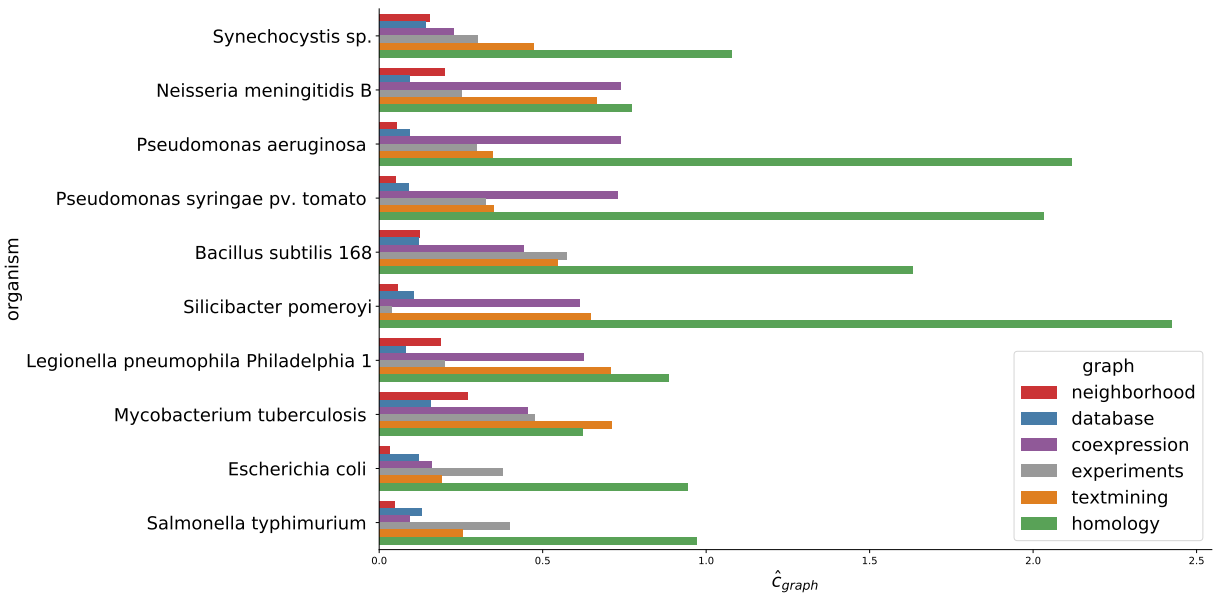


Figure 3.11 – *The learned coefficients for every organism.*

3.3.6 S2F for organisms with experimental data

I have shown that S2F is a powerful framework for the functional characterisation of newly sequenced organisms. It is not limited, however, only to these organisms, and can be used to complement already available experiments. In the following section, I show the performance of S2F in the context of the CAFA Challenge.

3.4 The CAFA Challenge

The focus of S2F is on newly sequenced organisms, and thus we compared it only against sequenced-based methods (InterPro and HMMER). It is, however, very interesting to compare it against other PFP methods that exploit data other than the sequence. Although we have not made this comparison ourselves, a very fair comparison can be found in the context of the Critical Assessment for Functional Annotation (CAFA) [143, 3, 144].

3.4.1 Structure of the challenge

CAFA is a community-wide effort to assess the status of computational functional annotation methods. It is run as a timed challenge. First, a collection of experimentally unannotated proteins are made public by the organisers and teams of predictors are given time to submit their predictions to the organisers. Then, there is a waiting period while annotations accumulate in databases such as UniProtKB/Swiss-Prot [5] and GOA [6]. After this waiting period, new experimental annotations are collected, analysed and finally published.

This structure allows CAFA to fairly compare all submissions on the same playing field. Because no training data is provided by the organisers, the ranking implicitly evaluates the ability to collect and properly exploit any available information.

3.4.2 Modifications to S2F

For the challenge, S2F is modified in simple but very important ways. These changes are motivated by eliminating the assumption that the organism has been recently sequenced. For our framework, this is a major change, since there is more data available in every step of the way and, crucially, available GO annotations. We now modify the framework in three ways:

1. **GOA-Clamping:** Due to the availability of GO annotations in the GOA database [6], the seed is now built, and subsequently "clamped" with these experimental annotations. This means that after combining the seed, all available (GO-term, protein) pairs that can be found in the annotations are set to 1. This clamping is also imposed after running the network propagation procedure.
2. **Network:** Similarly, due to the availability of protein-protein interactions, the networks are built giving more importance to the experimentally reported links already available to the target organism
3. **Blacklist:** Finally, we do not use the blacklist in this scenario, as we are not validating the framework, and allow S2F to transfer information from

every source.

The impact of these simple changes has important implications in the predictions. First, because CAFA involves fairly well-studied organisms, the amount of available data is absolutely not negligible. The available experimental GO annotations are deeply involved into providing S2F with prediction power, as they are used not only to build the seed, but also to build the target for the network combination. S2F is able to use the available annotations on top of those included in the original seed (InterPro and HMMER) through the GOA clamping step, improving the seeds. Then, it is able to use the existing experimentally validated PPI on top of its predicted network. This, on top of not using a blacklist, ensures that S2F is using one of the most complete networks possible for the prediction.

3.4.3 Modifications for CAFA π

In addition to the modifications we made for CAFA 2, we introduced a modification to the pipeline for CAFA-π, that maps the raw S2F score to the available set of scores that are allowed for a submission. This modification was motivated by the rules of this specific version of the challenge, summarised below.

CAFA π Rules

The rules were published by the CAFA organisers [145]. The goal of the prediction task is to predict what genes/proteins in a certain organism are associated with a given function, expressed as a GO term. There are two organisms and two GO-terms, summarised in Table 3.2.

Organism	GO-terms
Pseudomonas aeruginosa.	Biofilm formation GO:0042710, Motility: GO:0001539
Candida albicans.	Biofilm formation GO:0042710

Table 3.2 – *Prediction setting for CAFA-π.*

A participant is allowed a maximum of 3 models, i.e. up to 3 different scores for every protein and GO term. Each model can include one or more of the 3

possible set of predictions. The submission file should contain a list of protein targets that the team think are associated with the function designated in the filename, followed by a probabilistic estimate of the association (score). The score must be in the interval (0.00, 1.00] and contain two significant figures.

Score transformations

In order to assess the proper way to transform the scores, we inspected the distribution of the S2F prediction scores. Figure 3.14 shows the histogram of the raw S2F scores for motility on Pseudomonas. In this particular scenario, only 28 out of the 100 possible scores allowed for submission would be used. This is not ideal, since our method would be penalised for the gaps in the scores, being assigned a true positive only when reaching lower scores. The idea of transforming the scores is motivated by the fact that the evaluation metrics will be affected by the distribution of annotations. This may not be obvious at a first glance, since the relative order of the predictions appears to be maintained. This is not true when the precision of the scores is restricted as it is in CAFA. To demonstrate this, we created a simple array of 100 scores, and then rounded it to 2 and 1 decimal places, and calculated their respective AUC-ROC and AUC-PR values. Figures 3.12 and 3.13 show the effects of simply rounding the scores to the closest reduced precision value.

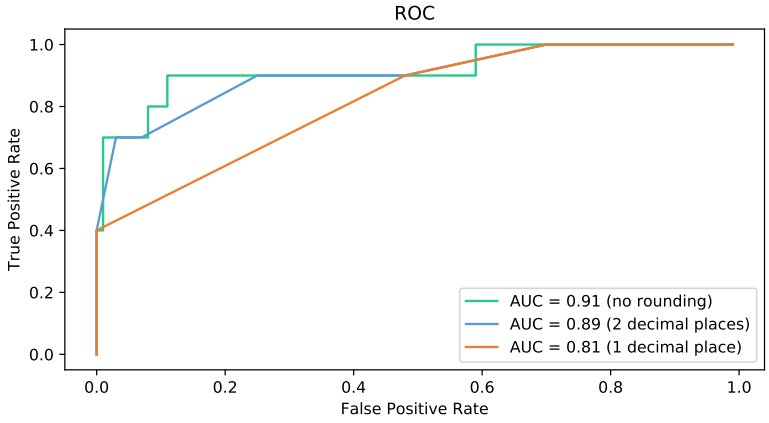


Figure 3.12 – *ROC curve for a toy prediction problem. The green curve is calculated using the scores without any rounding. The blue curve is calculated by rounding the original scores to 2 decimal places, and the orange curve by rounding the original scores to 1 decimal place.*

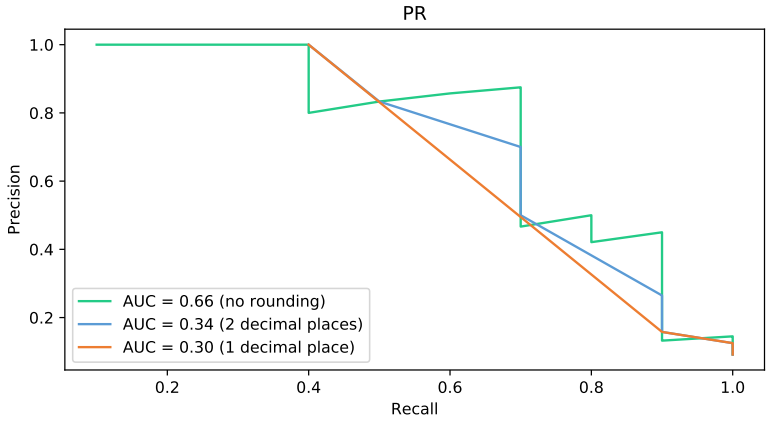


Figure 3.13 – *PR curve for a toy prediction problem. The green curve is calculated using the scores without any rounding. The blue curve is calculated by rounding the original scores to 2 decimal places, and the orange curve by rounding the original scores to 1 decimal place.*

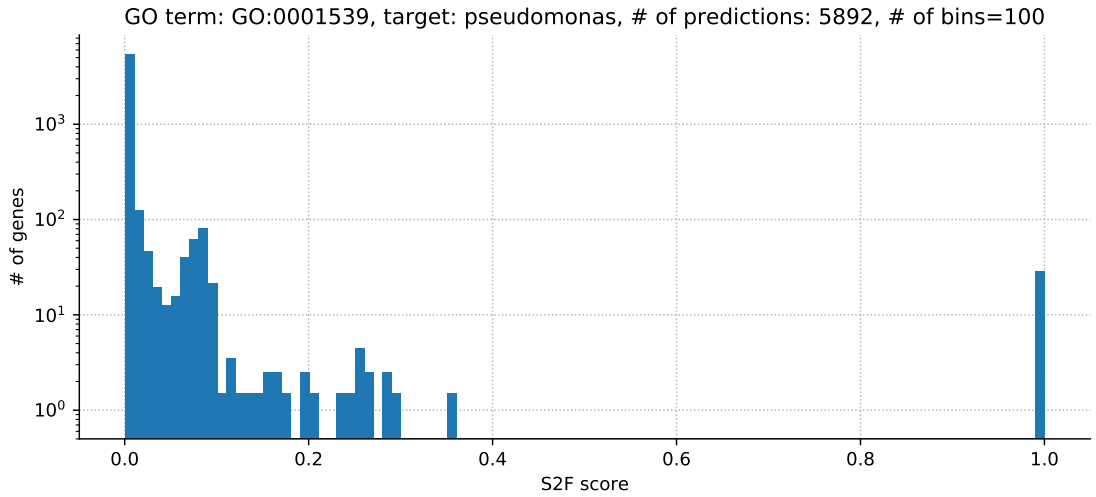


Figure 3.14 – *A histogram depicting the distribution of raw S2F scores for Motility (GO:0001539) on Pseudomonas genes. 100 bins were used, which could be mapped one-to-one to the CAFA scores. This would be a waste, however, since most of the scores would not be used. This would have an impact in the evaluation of the prediction algorithm, because for most of the thresholds used to compute the true positives, the "wasted" scores would not contribute. Only 28 out of the 100 possible scores would have an assignment if we consider the distribution of the figure.*

Modifying the score is not straightforward. Assigning a score that is too high is associated with the risk of a false positive very early in the tests. Thus, we decided to use a different binning strategy for the three models we could submit.

1. **Ignore Lowest:** This strategy assigns the lowest score (0.01) to the l lowest

scoring genes, the highest score (1.00) to the experimental annotations and any annotation that is already scored with a 1, and distributing the rest of the genes on the remaining 98 scores (0.02 to 0.99).

2. **Split Data A:** As in the previous strategy, this strategy assigns 0.01 to the l lowest scoring genes, and maintains the genes with a score of 1.00. The remaining scores are divided in 2 groups, the first group of 79 scores (0.21 to 0.99) is distributed between the h_{split} highest-scoring genes, and the second group of 19 scores (0.02 to 0.20) is distributed on the remaining genes.
3. **Split Data B:** This strategy is identical to the previous one, but we increase the number of genes assigned to the group of 79 scores by increasing the value of h_{split}.

Table 3.3 shows the parameters we used for our submissions. The values of l were selected so that most of the scores already below 0.01 will be assigned to the lowest score, and to distribute the remaining scores to the highest-scoring genes. Figure 3.15 shows how each strategy affects the original score. In this section, results for predicting Motility (GO:0001539) on Pseudomonas are included, as this is the category in which our method was included in the top-5 on the CAFA π manuscript [146]. Other histograms and score distributions can be found in appendix A.1.

Organism	l	h_{split}	h_{upper}
Pseudomonas	4713	300	600
Candida	10,000	500	1000

Table 3.3 – *Parameters used for our submissions to CAFA-π.*

Of all the strategies, the one that was highly-ranked on CAFA π was the "ignore lowest" strategy. This suggests that the original scoring of S2F is rather conservative, which supports statements in the CAFA 2 case study, in which S2F predicted the correct leaf GO term, but below its F_{max} threshold.

3.4.4 Performance on CAFA

Our submission for CAFA 2, named "PaccanaroLab", ranked first for the Biological Process GO domain on the overall evaluation using the F_{max} measure

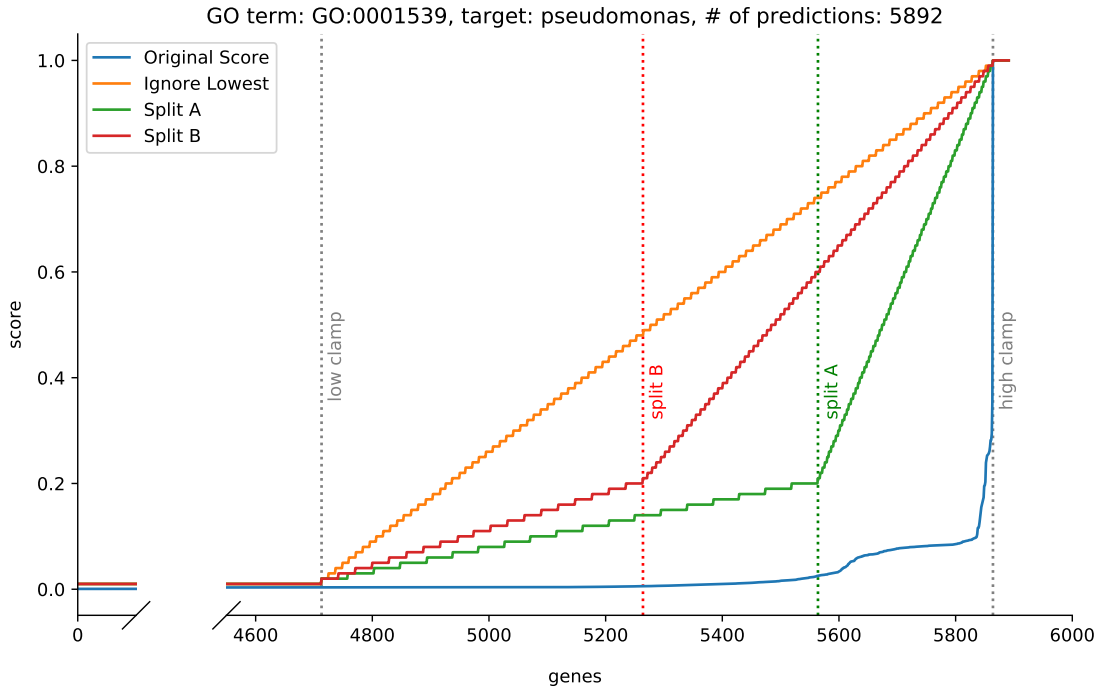


Figure 3.15 – *Score comparison of all strategies for motility (GO:0001539) on Pseudomonas.*

(see Figure 3.16). On CAFA-π our model, named "PaccanaroLab 1" ranked third on the AUC for predicting motility for Pseudomonas (see Figure 3.17).

It is relevant to put our method in context with other methods in the CAFA challenge. Although the exact parameters of every method in CAFA are not available, their analysis on the keywords of the methods reveals that the integration of information other than the sequence does outperform by a considerable margin when compared to sequence alignment. Moreover, for biological process overall, where we ranked first in CAFA 2, our method is not similar to other top ranked methods. This suggests that S2F is doing something different which leads to better predictions in this category.

3.5 Discussion

We have shown that S2F is capable of leveraging label propagation techniques and the "guilt-by-association" principle for predicting functionally uncharacterised organisms. We propose a framework that can pick up patterns that are

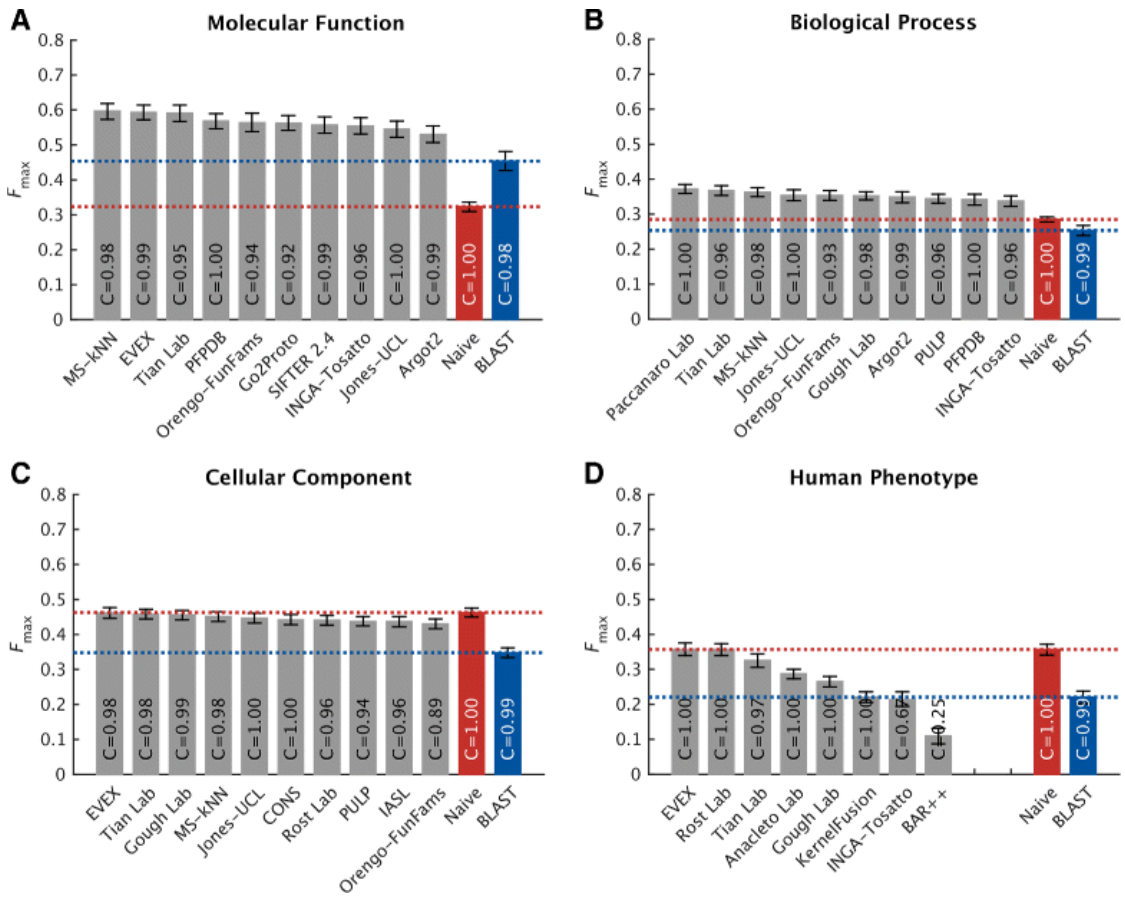


Figure 3.16 – *Overall evaluation using the maximum F measure, F_{max}. Evaluation was carried out on no-knowledge benchmark sequences in the full mode. The coverage of each method is shown within its performance bar. A perfect predictor would be characterized with $F_{max} = 1$. Confidence intervals (95 %) were determined using bootstrapping with* 10,000 *iterations on the set of benchmark sequences. For cases in which a principal investigator participated in multiple teams, the results of only the best-scoring method are presented.* figure and caption taken from: [3]

relevant for the functional characterisation of proteins by integrating data from already studied organisms. By focusing our efforts on *de novo* prediction, we expect to assist with the annotation process. The main purpose of S2F is to narrow down the daunting amount of possibilities to consider for experimental annotation. With the proposed framework, we show that we can considerably improve the predictions of current methods for *de novo* prediction.

An early version of S2F which is optimised to use existing evidence for the target organism was submitted to the CAFA2 and CAFA-π challenges [3, 146], where it ranked as a top performing method. This shows the potential of S2F to predict function for already characterised organisms, improving the existing

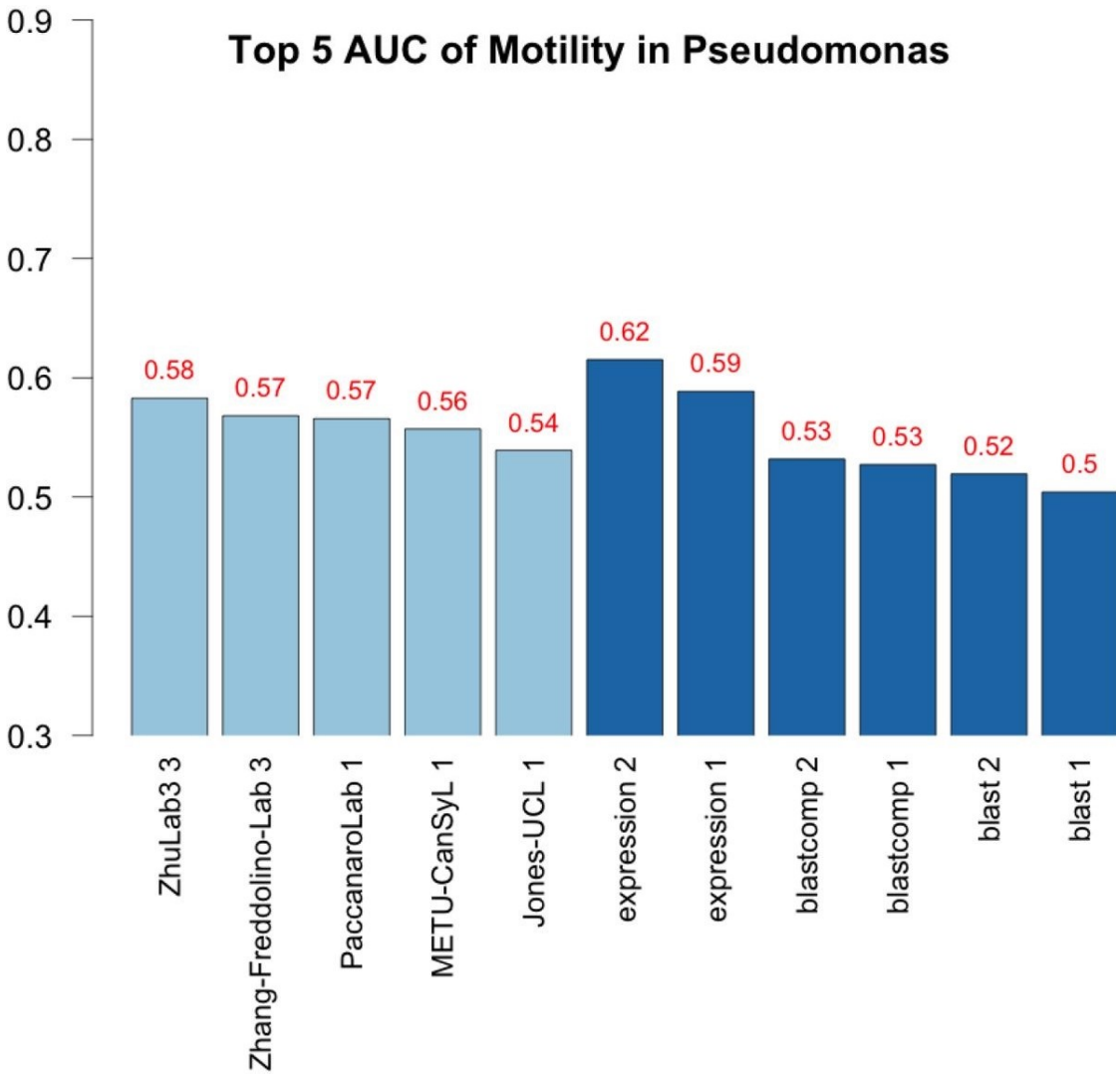


Figure 3.17 – *AUROC of top 5 teams in CAFA-π. The best performing model from each team is picked for the top five teams, regardless of whether that model is submitted as model 1.* figure and caption taken from: [146]

annotations. An easy-to-run version of this is also provided in the software linked above.

There is room for improvement for S2F. We are aware that our evaluation is made possible due to the availability of high-quality data for the bacteria we selected. Evaluation with organisms with less data is pending. As mentioned before, we believe that the impact of our framework will be different for organisms depending on their closeness to well characterised organisms. Predicting such impact could be done by assessing the distance to well-studied organisms and assessing the quality of the transferred network and the seeds. Both of these

improvements are not obvious, as there is still no consensus on how to measure how "well annotated" an organism is. For instance, if we have a metric that can tell us how "complete" a seed is, we would have an objective way to set our α parameter and incorporating other sequence-based models to the seed. If, on top of that, a "functional distance" between organisms was available it would allow us to assign a confidence score on the predictions for every organism, which would be really useful to use to guide new experiments. Also pending are trying other methods for combining the functional network, as well as the addition of a functional similarity network. We intend to try adding a functional protein-protein network generated using GOssTo [147, 148].

Additionally, further insight into other kingdoms remains to be done. In this study we focused or efforts on Bacteria. We expect the impact of S2F to be similar for other kingdoms. Expanding this analysis will likely expand our understanding of the landscape of functionally relevant information throughout species.

3.6 Implementation

From a software engineering point of view, S2F is a very modular, extensible and versatile tool. It is programmed in Python 3.6, and with the exception of the external tools it 100% platform independent. Although the main command: `predict` requires a UNIX based system, virtually every other command can be used on Windows if the seed files are provided.

3.6.1 Software Design

The code in which S2F is programmed can be seen as a framework with several utilities that are not evident from the point of view of the user, but that makes the software easier to maintain, install, and run.

The architecture divides the software into several *components*, which are made available to the user by a collection of *commands* that can be called from the command prompt.

At the time of writing, the available commands are:

- `install`: It manages the setup of the environment required by S2F. With configurable options, it will download the latest version of STRING, UniProtKB, and GOA, and will interactively ask the user to input the location of the binaries required for the main prediction command.
- `predict`: It runs S2F on a given FASTA file. It makes use of the vast majority of available components in a coordinated way. Highly configurable in terms of verbosity, and with the ability to resume the computation from safe points in the pipeline in case of an interrupt. This is the default S2F pipeline.
- `combine`: It runs the S2F graph combination algorithm on an arbitrary collection of graphs provided by the user. This is very useful when the user has protein networks that might want to add to the ones in STRING. A seed must be provided, that will be used to build the target network.
- `diffuse`: It diffuses a user provided seed onto a user provided network. At the moment, two diffusion algorithms are available: The consistency method, and S2F.
- `hmmer-seed`: Given the output of a HMMER search, it will generate a seed file compatible with S2F.
- `combine-seeds`: It combines an arbitrary collection of seeds. The user can provide the coefficients. By default, this command will simply linearly combine the seeds with equal weights.
- `build-clamp`: Provided a FASTA file and a list of evidence codes, it will extract annotations from the GOA database downloaded during installation into a file which can be used later with the `predict` command.

The components are in general internal classes that support the functionality of the commands, but some of them have value on their own, and can be used in other PFP related programs. Here, I list the ones I consider most useful:

- `GOTool`: This is a collection of utility classes written for S2F that are neatly wrapped into as a python library. It provides the importable `GeneOntology` module, that provides parsers for OBO and GAF files,

it builds the Gene Ontology structure from an OBO file into memory, and the same structure can be used efficiently in memory to load several GAF files, which will be handled separately by the same object. This allows very versatile analysis of the annotations, with utilities to up-propagate annotations, compute metrics such as the information content, and export the annotations to a variety of formats including the very useful `pandas.DataFrame` serialised using the `pickle` algorithm. I intend to make `GOTool` available as a stand-alone tool in the future.

- `FancyApp`: This importable module is inherited by the majority of the components in S2F. It provides the programmer with a collection of very useful tools for debugging and notifying the user of the current progress of the program in an elegant command line interface. Among the utilities provided are a progress bar, an automatic logger without the need of previous configuration with different levels of verbosity and colours. Finally, a twitter notification script can be configured within `FancyApp` to send direct messages to the users, alerting them of milestones that can happen over many hours or even days of computation time.
- `Configuration`: A simple utility to manage configuration files with great versatility, it provides a very fast parser and a singleton loader of the configuration environment that allows the programmer to query any configured variable in independent runs. It provides functionalities for easily setting and updating configurations on disk, as well as as friendly interface with the user, that will prevent a critical component to run in case of a misconfiguration.

Current efforts of S2F involve the addition of the CAFA-π specific re-scoring, commands for InterPro seeds and the integration of the testing and plot generation scripts used in this chapter and the upcoming publication. A very thorough documentation of the software and libraries is currently being written and is available at `https://github.com/paccanarolab/s2f/wiki`, where also issues and questions regarding the software are actively monitored.

4

ConSAT

Gene duplication, divergence and rearrangement have been predominantly responsible for the expansion of a species' protein complement during evolution [149]. Consequently, several PFP methods exploit protein domains identified by resources either looking for conserved similar protein sequences or similar structural units. From a functional perspective, however, the function of a protein might not be the union of the individual member domain functions. The combination of several domains might add, eliminate, or modify the functions [150, 151]. Therefore, the elucidation of the domain arrangement of a protein, its *domain architecture*, is critical in deciphering its role at the biological, molecular, and cellular levels. The Consensus Architecture Annotation Tool (ConSAT) consists of both a tool and web application for protein function annotation for genome projects of any scale. The annotation method in this case relies on protein domain architectures to assign function to proteins.

Obtaining the domain architecture of a protein is difficult due to the divergent agreement of different methods on the domains they detect and their positions in the sequence [11]. ConSAT aims at identifying *consensus domain*

architectures; that is, a *unified* domain arrangement derived from the different domain detection methods in a way that avoids representing the same underlying domain with assignments from two or more methods at the same time, and where no overlap is present, except for domain insertions (see Figure 4.1 for a schematic of a consensus architecture and its construction). Important aspects that are taken into account are the N terminal to C terminal order of identified domains, the accurate relative organisation in complicated cases of non contiguous domain arrangements, and finally a computer parsable representation of such domain arrangement. To our knowledge, there is no single method that addresses this issue. Typically, non contiguous domain arrangements (domain insertion, circular permutation, etc.) are considered problematic and excluded or ignored.

Once the consensus domain architectures are obtained, we assign functions to every architecture. Directly, by associating the domain with the over-represented GO terms of its component domains. Indirectly, by "annotation transfer" from the set of proteins associated with the architecture, where the over-represented GO terms are associated to the architecture itself. Finally, we associate English keywords to the architectures by mining the PubMed abstracts associated to proteins associated with the architecture. These complementary approaches provide a wide variety of functional assessments of every architecture, not only in the context of GO terms, but also the natural language commonly used to the characterisation of the domains involved.

4.1 Obtaining the ConSAT architectures

Consensus architectures are obtained through a two-step process. First, we estimate preliminary domain architectures through a modified version of the GFam algorithm [152]. Second, these preliminary architectures are broadened by using the ConSAT Putative Protein Domain (CPPD) database. In this section, I describe the two steps in detail.

4.1.1 Preliminary Architectures

The method starts with the set of domains that InterPro assigns to a sequence and creates a preliminary consensus architecture. This is done by incrementally adding domains from the different models in InterPro. We allow a minimal level of overlap between domains to overcome the fact that the different models might diverge in the location of the boundaries for the same domain. By default, the allowed overlap is 20 residues. We also allow domain insertions (a domain inside another domain) to account for cases in which a domain assigned by a model *A* is found into a domain found by model *B*.

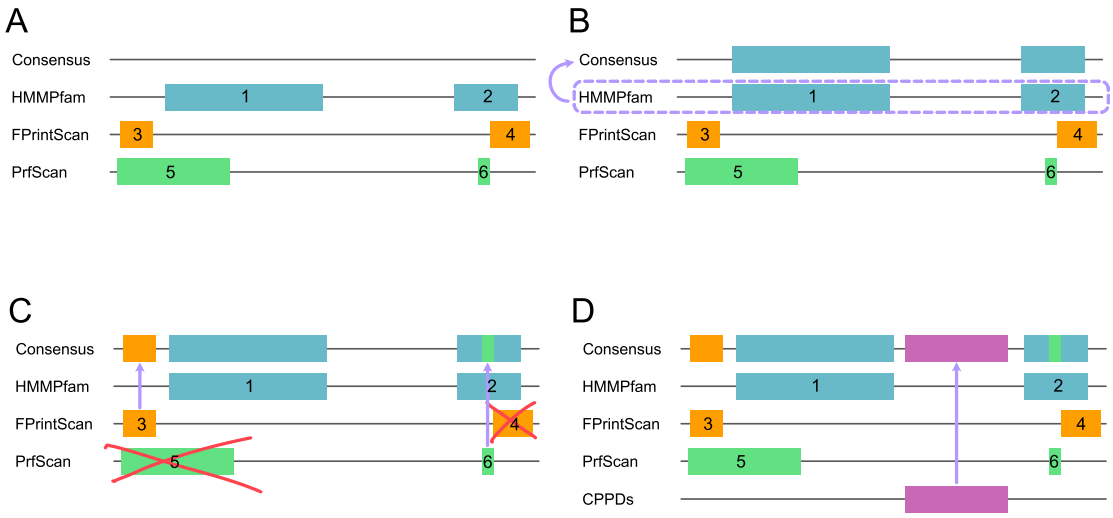


Figure 4.1 – *The detection of consensus architectures. A) Three InterPro sources (HMMPfam, FPrintScan and PrfScan) find domains (1 to 6) in the sequence; the consensus architecture is empty. B) Domains from HMMPfam (1 and 2) are added to the consensus architecture as they cover more residues than any other source. C) The remaining domains are processed in decreasing order of length (5, 4, 3, and 6). Domains 5 and 4 are discarded as they overlap with 1 and 2, respectively. Domain 3 is added to the consensus architecture. Finally, domain 6 is added as an insertion in 2, completing the preliminary consensus domain architecture. D) After the scan with the CPPD data source, a new domain (purple) is added, completing the final consensus architecture.*

Figure 4.1 shows the different phases of the procedure for a protein. Only three InterPro models (HMMPfam, FPrintScan and PrfScan) are shown to make the process easy to visualise. For a given protein, we begin by choosing the InterPro model that covers the most residues in the sequence (Figure 4.1B). The remaining domains are integrated into the architecture in descending order of length. Some domains that would largely overlap with the architecture built

so far are discarded. The size of the allowed overlap as well as the e-value thresholds for the different methods can be defined by the user. Default values for these parameters are 20 residues for the overlap and 1×10^{-3} for e-values, with the exception of Superfamily, HMMPanther, Gene3D, and HMMPIR for which the default e-value threshold is used.

4.1.2 Refinement of the architectures

A central part of ConSAT is its database of Putative Protein Domains (CPPDs). To build this resource, we run the first stage of ConSAT (building the preliminary consensus domain architectures) over the reviewed UniProtKB/Swiss-Prot. Unassigned protein fragments of length greater than or equal to 30 residues (based on the distribution of domain lengths of the included domain databases) are then extracted and clustered using a procedure similar to that of GFam [152].

First, a pairwise BLAST [8] alignment is run on all the unassigned sequence fragments. Only significant hits are kept (BLAST e-value <= 1×10^{-3}, percent identity of at least 45%, minimum normalised alignment of at least 0.7). This results in a binary graph where the nodes represent the fragments, and the links denote a high quality pairwise similarity between the fragments it connects.

This binary graph is used to derive a new weighted graph, where the nodes are still the sequence fragments and a link between two nodes is weighted by the Jaccard coefficient between the sets of neighbours of those two nodes in the binary graph. This operation has the effect of reducing the noise in the single link between the two nodes in the binary version by "averaging" it over the nodes' neighbourhoods. This network is subsequently binarised by keeping only the links with a weight greater than $\frac{2}{3}$. The Connected Components Algorithm (CCA) is the applied to the new binary graph to obtain clusters containing fragments from at least four different protein sequences. Each cluster is then labelled as a ConSAT Putative Protein Domain (CPPD).

Finally, we run Clustal Omega [153] to obtain a multiple alignment of the sequence contained in a putative domain. This alignment is used to build a Hidden Markov Model with HMMER [154]. Those models constitute the CPPD

database.

In the second stage of ConSAT, the aim is to refine the preliminary consensus domain architecture and increase its coverage by dealing with sequence fragments that were not covered by any InterPro model. First, low complexity regions are estimated using the SEG algorithm [155] and discarded. The remaining fragments are scanned against the CPPD database of putative domains using HMMER [154] and all hits are added to the consensus domain architecture. This stage is depicted as the purple domain transferred in Figure 4.1D. It is important to note that for cases in which no InterPro model assigns domains to a protein, putative domains are still available. This increases the number of characterised sequences.

4.2 Functional assignment

ConSAT uses the consensus domain architectures to assign function to proteins. This is, functional terms are assigned to the architecture, and proteins are associated with all the functional terms of its architecture. Two types of functional terms are assigned: GO terms, and English words.

GO terms are assigned using two approaches:

- **Direct method:** We obtain GO terms associated with each domain in the architecture using InterPro2GO [156]. Then, we perform an over representation analysis followed by a multiple hypothesis correction (Benjamini-Hochberg [157]) to obtain a p-value for each GO term.
- **Indirect method:** Go terms are assigned by "annotation transfer". We start from the set of proteins associated with the architecture. Then, all the GO terms associated with this set are retrieved from UniProt-GOA. Importantly, we restrict these associations to the ones with experimental evidence codes, plus Traceable Author Statement and Inferred by Curator. Finally, an over representation analysis of these terms and the Benjamini-Hochberg correction are used to obtain a p-value for each term.

4.2.1 GO terms, direct method

This method uses the set of GO terms associated to each of the individual domains of the architecture. InterPro2GO [156] provides a set of GO terms for most domains. We augment these assignments by up-propagating them using the true-path rule.

Following Cho et al. [158], we run a binomial test find statistically significant GO terms associated to an architecture. For each GO term T we compute its probability $p_T = \frac{n_T}{M}$, where n_T is the number of domains mapped to T in InterPro2GO, and M the total number of domains in InterPro2GO. For a given architecture A, let K be the total number of domains (considering repeats) and let k_T be the number of domains annotated with term T. The p-value for assigning term T to architecture A is the probability of observing k_T domains out of K. Under the null hypothesis, the number of domains mapped to T follows a binomial distribution with parameter p_T, and thus the p-value for overrepresented terms is given by:

$$\text{p-value} = \sum_{x=k_t}^{K} \binom{n}{x} {p_T}^x (1 - p_T)^{K-x}$$

Given an architecture, we carry out a test for each GO term, and therefore correcting for multiple hypothesis testing is needed. We apply the Benjamini-Hochberg correction method [157], and terms with corrected p-value below 0.05 are finally assigned to the architecture.

4.2.2 GO terms, indirect method

This method uses the set of GO terms associated with proteins that are covered by the architecture for at least 80%. First, we identify these proteins and retrieve their associated GO terms from GOA [6]. The associations we consider are restricted to those with experimental evidence codes, plus those inferred by curators or with traceable authors. As for the direct method, we up-propagate these associations with the true path rule. We then test whether a GO term T found k_T times in the annotations of a set of n proteins is statistically significant. Let N be the total number of proteins, of which K_T are annotated with term T.

The p-value for term T is calculated as the probability of finding each GO term at least k_T times out of n draws (hypergeometric distribution):

$$\text{p-value} = \sum_{x=k_T}^{K_T} \frac{\binom{K_T}{x}\binom{N-K_T}{n-x}}{\binom{N}{n}}$$

As in the previous case, we correct p-values with the Benjamini-Hochberg method and terms with corrected p-value below 0.05 are assigned to the architecture.

4.2.3 Combined p-value

Our system provides the p-values obtained by the direct and indirect method separately. It also provides a combined p-value for the combination of functional assignments, which is computed using Fishers method [159]. A test statistic is computed using:

$$-2(\log(p_1) + \log(p_2))$$

where p_1 is the p-value obtained by the direct method and p_2 is the one given by the indirect method. The combined p-value is obtained knowing that the previous statistic has a chi-squared distribution with $2n$ degrees of freedom (where n is the number of combined p-values, in this case $n = 2$).

4.2.4 English Keywords

For each protein, we can retrieve textual information from both UniProtKB and PubMed. UniProtKB provides a natural language description of each protein, and PubMed provides the abstracts in which each protein is explicitly mentioned. We put together these texts to assign a set of weighted words to each architecture.

Our procedure begins by tokenising, then removing non-alphabetical symbols, stop words (e.g. prepositions, articles), numerals, and applying Porter's stemming algorithm [160]. We then represent each protein as a bag of words [161], that is a high-dimensional vector where each keyword represents a dimension. Vector coordinates are computed following a tf-idf scheme [162]: the

value of term i for protein j is the product of the absolute frequency of that term in the text that describes the protein, tf_{ij}, and its inverse document frequency, idf_i, defined as $\log(N/N_i)$, where N is the total number of protein sequences and N_i is the number of sequences containing that keyword. For each architecture A, the vectors of the associated proteins are then summed up into a single vector. The (unnormalised) weight w_{iA} for keyword i and architecture A is given by:

$$w_{iA} = \sum_{j \in \text{proteins}(A)} tf_{ij} \cdot idf_i = \sum_{j \in \text{proteins}(A)} tf_{ij} \cdot \log\left(\frac{N}{N_i}\right)$$

Following Joachims [163], the vector is then normalised by their Euclidean length in order to obtain weights that are comparable across architectures. The weight for keyword i and architecture A, $\bar{w}_{iA}$ is:

$$\bar{w}_{iA} = \frac{w_{iA}}{\sqrt{\sum_{j=1}^{n} w_{jA}^2}}$$

Only the top 100 words are shown on the web server, while we keep the top 500 for each architecture in our database.

4.3 Notation for protein domain architectures

We have developed a notation to uniquely represent the domain ordering of an architecture using a simple string of text. We use it both in the stand-alone as well as in the web application, alongside with the graphical depiction to denote the architectures. With this notation it is possible to characterise any possible complex domain arrangement constituted by any combination of the two operations of domain juxtaposition (a domain places after another) and domain insertion. Juxtaposed domains are represented by separating them with semicolons:

`IPR002885;IPR013498`

represents an architecture composed of domain `IPR002885` followed by domain `IPR013498`. Domain insertions are represented using curly braces:

`IPR002885{IPR013498}`

represents that the domain IPR013498 is inserted within domain IPR002885. In this notation, nesting of insertions is allowed by different levels of curly braces:

IPR002885{IPR013498;IPR012456{CPPD00001}}

means that the architecture is composed of the domain IPR002885, which has inserted juxtaposed domains IPR013498 and IPR012456, where the latter has domain CPPD00001 inserted.

4.4 The ConSAT web server

We have run ConSAT on the entire UniProtKB [5] protein sequences database. The results are available through the ConSAT web server. This web application provides a user friendly interface to all the functional predictions as well as the protein architectures for all UniProtKB sequences. Importantly, the web server also allows users to submit their own set of sequences, which makes of ConSAT a fully-fledged online function prediction system.

In addition to the web application, ConSAT is available as an easy-to-run stand-alone command-line Python application. The entire source code is released under the GPLv3 licence. processing pipelines. The entire software together with its manuals is available at https://www.paccanarolab.org/consat_resource.

5

ICrep

Another way to approach the functional characterisation of an organism is by studying its protein complexes. The complexes are after all the structures that ultimately perform these functions. In this chapter, I present ICrep: a protein interaction and complexes repository that encompasses every sequenced organism. In similar fashion to S2F, ICrep builds on top of the interolog concept [124, 4] to infer protein protein interaction networks. The goal is different, however, as the main focus is not to predict protein function, but to infer protein complexes. A functional interpretation of the complexes is provided by an over-representation analysis.

5.1 The ICrep idea

The core idea of ICrep is to build a resource that compiles protein complexes for all available organisms. At a first glance, this idea seems fairly simple. It gets complicated, however, when making decision about the particular details of the

implementation such as the operational definition of orthology, and which set of genes to use for a particular organism. The idea can be broken down into 3 major steps for every organism:

1. Compile a protein-protein network of experimental and predicted interactions.
2. Identify protein complexes, leveraging the PPI network.
3. Provide a functional interpretation to every identified complex.

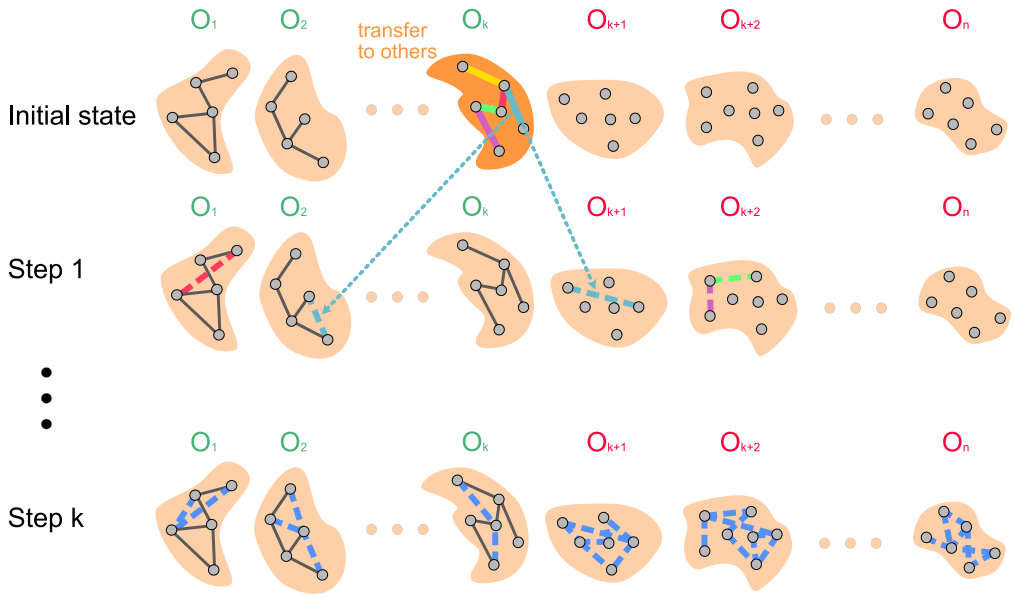


Figure 5.1 – *The ICrep core concept is to transfer links between organisms. Most organisms do not have experimentally reported interactions (Red $\{O_{k+1} - O_n\}$). We transfer all experimental interactions between organisms, even to those that already count with interactions ($\{O_1 - O_k\}$). In the end, we expect to have one PPI network per organism, in which some will be experimental (solid links), and some will be interologs (dashed links). The figure shows k steps, which correspond to the k organisms that count with experimental interactions. In every step, a different organism is used as "source", and every other organism is considered a "target".*

Figure 5.1 illustrates the first step of the pipeline. Here, we transfer information from all organisms with experimental PPIs to every other organism. This allowed us to build a database of PPI networks for virtually every sequenced organism. Figure 5.2 shows the process of identifying a protein complex. The ClusterONE [85] algorithm is used to cluster every PPI, resulting on a database of predicted complexes.

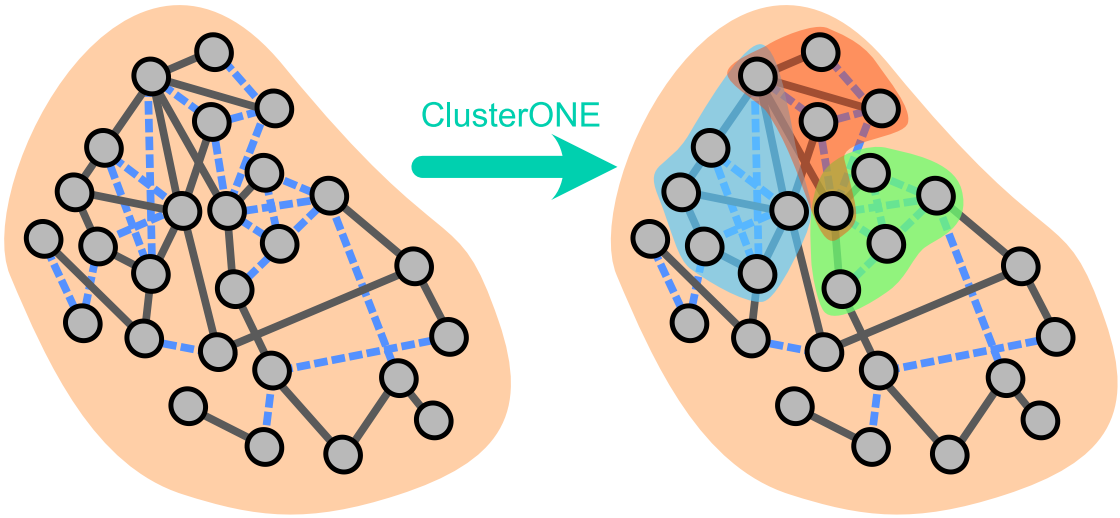


Figure 5.2 – *Given a PPI network, we cluster it to find protein complexes using ClusterONE. This will allow us to find protein complexes, even if they overlap.*

The final step of the process is to provide functional context to the predicted complexes. This is done by doing an over-representation analysis of GO terms of the member proteins. In the following sections, I go in detail about each one of these steps, finalising with some screenshots of the web interface that allows for easy navigation of the database.

5.2 Interaction transfer

As a source for experimental interactions, we collected PPIs from the following datasets:

- BioGRID [67]: The Biological General Repository for Interaction Datasets. It is a public database of genetic and protein interaction data for model organisms and human. Currently holding over 1,400,000 interactions curated from other datasets and the literature.
- IntAct [68]: A molecular interaction database holding over 785,000 interactions. Curated by the European Bioinformatics Institute (EBI) and collaborators.
- MINT [69]: The Molecular INTeraction database, focused on experimentally verified protein-protein interactions mined from the scientific litera-

ture by expert curators. Currently holds over 125,000 high quality interactions.

- DIP [66]: The Database of Interacting Proteins. It catalogues experimentally determined protein-protein interactions, currently holding over 81,000 interactions.

The starting point of inferring PPIs is to put together the interaction datasets. Because of the several existing methods of detecting protein-protein interaction, not every reported PPI has the same quality. Therefore, we filter the initial datasets using the Molecular Interaction Ontology (PSI-MI) [164]. We allow only interactions that are in both the "molecular association" and "experimental interaction detection" categories and their respective children categories. The complete list of valid terms is available in the Downloads page of the tool: `https://paccanarolab.org/icrep/downloads/`.

To transfer the interactions, we use the method proposed by Yu et al. [4]. In this case, we have a small difference when compared to the S2F transference. For ICrep, we use less stringent conditions for homology, setting a threshold of 1×10^{-4} on the e-value reported by BLAST [8]. The motivation behind this relaxed condition is that since we are not focusing exclusively on recently sequenced organisms, we prioritise a more "complete" picture of the PPI networks. Furthermore, we are not transferring STRING interactions, but experimental PPI from 4 databases that, if transferred, will probably map to high-quality interologs. Another difference in comparison to the S2f interolog is that instead of using the geometric mean of the percent identities as a condition, we use it to assing a "quality score" of the interolog. This is:

$$\sqrt{p_{A,A'}\, p_{B,B'}}$$

where $p_{A,A'}$ and $p_{B,B'}$ are the percent sequence identities of pairs of proteins (A, A') and (B, B') respectively. This quality score can be used by the users to filter ICrep, which effectively allows to restrict ICRep to using only interologs with the desired quality or better. The interologs for ICrep are computed on both UniProtKB/SwissProt and UniProtKB/TrEMBL, this is to allow the users to decide whether to include elements from the curated and automatically annotated sections of UniProtKB.

5.3 Complex prediction

From a networks perspective, a protein complex can be seen as a community in a PPI network. We used the ClusterONE [85] algorithm to identify such communities. This algorithm works very well with weighted networks, and is capable of detecting overlapping communities. To build the network, we assigned a probability to each interolog by using the resulting mapping from joint identity and percentage of verified interologs by Yu et al. [4]. Figure 5.3 shows the curve we use to assign the weights to the network. This curve represents the percentage of experimentally verified interactions with respect to the joint identity of transferred interologs. The mapping is between interactions in *S. cerevisiae*, *C. elegans*, *D. melanogaster*, and *H. pylori* onto the *S. cereviciae* genome, assessed against a gold standard [4]. We have used only this mapping for ICRep. Using a different mapping that is better suited for each organism would probably improve the quality of the trasnferred interactions, but to our knowledge no such resource exists at the time of writing.

5.4 Over-representation analysis

The final step of the ICrep procedure is to provide a functional context to each predicted complex. We do so by conducting a GO over-representation analysis on the components of each complex. We check each term using a hypergeometric test to determine whether it is overrepresented within the annotations available for the members of the complex.

5.5 Web tool

Once all the components of the database are computed, we compile them into a PostgreSQL database with a user friendly web interface. The resource is available at `https://paccanarolab.org/icrep/`. From the website, all the data can be downloaded in text format, and is totally browsable by searching for the organism of interest.

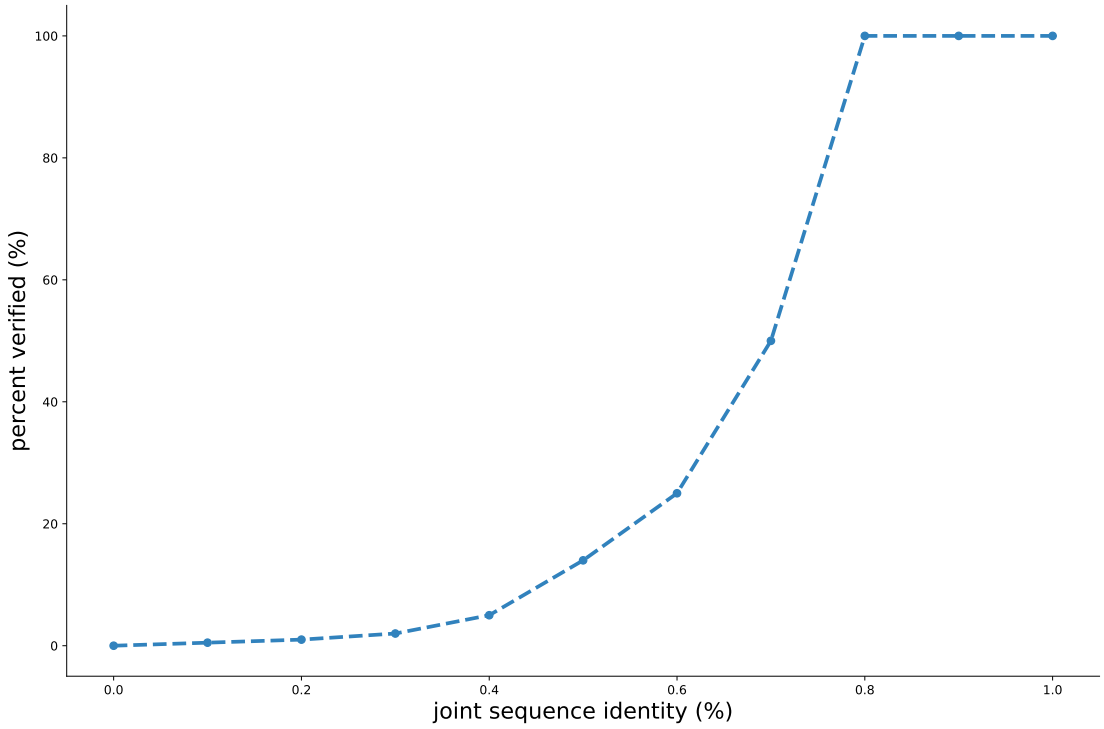


Figure 5.3 – *Recreation (the data to create this chart is taken from the interolog paper by Yu et al. [4], and used to create a bigger, vectorised chart.) of the joint sequence identity and percentage of verified interactions mapping by Yu et al. [4]. For ICrep, we use this curve to determine the weights of the network before the clustering procedure. The joint sequence identity is defined as the geometric mean of the percent identities. The joint identity is used solely as a confidence level or measure of the "quality" of the transferred interaction. The weight of the link itself is still the one reported in the STRING database.*

5.5.1 Organisation of the website

The website is organised in pages that links to entries relevant from a particular element. For instance, the user might be interested on a particular interaction, or a particular organism. The ease of use makes it simple to the user to start from an element (e.g., an organism) and inspect it in details, moving to the interactions, interologs, complexes, and even organisms that are related through the interactions or interologs.

The home page features a simple organism search box with auto-completion. This is to easily navigate to what we consider the most informative starting point. On the top of the website, the user has easy access to the Download page, in which all the data can be downloaded in text format.

Organism page

The organism page is the most central part of ICrep. It compiles all the interactions, interologs, and complexes related to the queried proteome. By default, the web interface lists experimental interactions, interologs, and protein complexes involving proteins from both UniProtKB/Swiss-Prot and UniProtKB/TrEMBL. A toggle button is provided on the top of the organism page to include only reviewed proteins from UniProtKB/Swiss-Prot.

In each section of the organism page, there is a download button that will download the relevant data in text format.

The interaction and interolog pages

We provide a particular interface to explore entries related to a particular experimental interaction. On this page, the user will find a link to the relevant article in PubMed, and the interologs that are a result of this interactions, that may belong to other organisms. We also report "sibling interactions", which are those experimental interactions who could have been the source of the interaction being looked at. This is, if evidence was not available, this would have been an interolog coming from the sibling interaction. Finally, the interaction page links to all predicted complexes that include this interaction.

In a similar way, the interolog has a page to allow easy navigation to the related source experimental interaction, and the predicted complexes in which it is involved. The interolog quality is reported in this page as well.

The protein complex page

The central piece of the resource is the protein complex page. It links to every included protein, all experimental interactions, all interologs, and a list of over-represented GO terms. It also provides a downloadable figure of the network. A screenshot of the complex page can be seen in Figure 5.4.

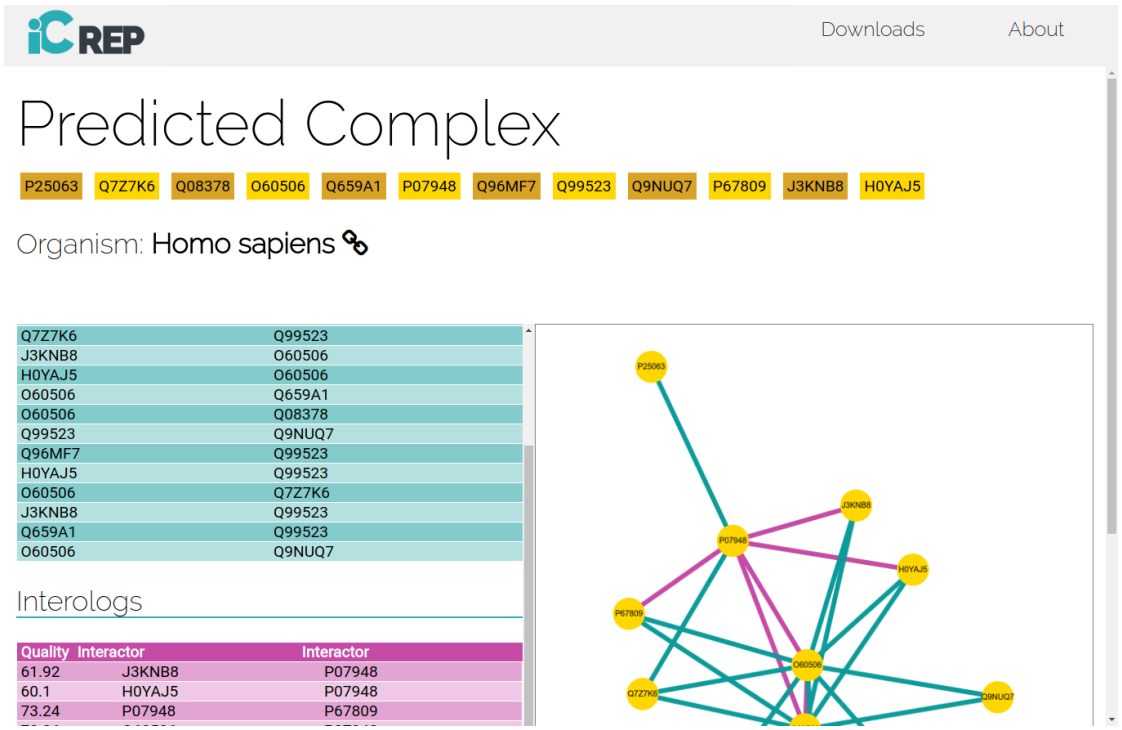


Figure 5.4 – *A Screenshot of ICRep showing the GUI when exploring a predicted complex. For every complex, an interactive image of the network is built, which also allows easy navigation to the relevant PPIs. The image is colour-coded according to the type of interaction (pink for interolog or blue for experimental interactions)*

5.6 Discussion

ICRep is, to our knowledge, the most complete resource for predicted interactions and complexes. Table 5.1 shows a breakdown of the available data with respect to the number of organisms in the UniProtKB non-redundant list of proteomes. To have an idea of the magnitude of the task, it is useful to look at other numbers. ICrep features a grand total of 711,892 experimental interactions compiled from all the datasets, 501,529,053 interologs, and 2,747,765 predicted protein complexes. All of these are distributed over 16,742 proteomes. Importantly, we have identified 57,635,339 homologs with our criterion, and although these are not directly available for download on ICrep, the full BLAST matrix will be made available. This matrix is a very useful resource for network-based computational biology, and due to the immense computational cost to produce such a matrix, a future work is to keep an updated pairwise BLAST matrix available for download.

	Number of organisms in UniProtKB (non-redundant proteomes)	Number of organisms with experimental PPIs (from BioGRID, MINT, IntAct, DIP)	Number of organisms in ICRep with predicted or experimental interactions	Number of organisms in ICRep with predicted complexes
Eukaryota	1494	67	1470	1439
Bacteria	25,893	153	22,491	12,612
Archaea	953	18	953	610

Table 5.1 – *Information contained in the ICrep database for organisms in the UniProtKB non-redundant proteomes.*

6

Future work and collaborations

In this chapter, I will summarise collaborations that are related to my work in protein function prediction. Also, I will introduce the preliminary work done in current projects.

6.1 Chlamydomonas reinhardtii cell cycle

A collaboration project with the Biology Department at Royal Holloway is aimed at understanding the behaviour of Chlamydomonas reinhardtii cells over time. Proteomics data was sampled every 2 hours for one day with 6 replicates per sample. One cell line is used as control, and the second cell line is exposed to Rapamycin. Please refer to our collaborators' paper [165] for details on the experimental setup.

My involvement in the current stage of the project is to provide both a functional analysis over the time points. In particular, the analysis that I have performed are:

1. Protein function prediction of all proteins in the organism.
2. Functional analysis of periodic proteins clustered by their expression profiles.
3. Prediction of a PPI network.
4. Prediction of protein complexes.
5. Analysis of the effect of Rapamycin on the complexes.

Function prediction was performed using S2F. All experimental annotations were retrieved from GOA [6] and Phytozome [166]. After running the S2F framework, we keep the top-10 predictions for the rest of the analysis, which involves the periodic proteins. The periodicity of these proteins is determined by a colleague using the Perseus Framework [167]. I performed an over-representation analysis to determine the functional context of each group of periodic proteins. Figure 6.1 shows the over-representation results for the biological process subdomain. Figures for the other domains are shown at the end of this section.

For the PPI and complexes prediction, we follow the procedure used for ICRep: we compute align all proteins using BLAST, and we transfer interologs to Chlamydomonas reinhardtii. Then, we cluster the inferred PPI using ClusterONE. Finally, every predicted complex is used as a group of genes for a GO over-representation analysis. Furthermore, we created a score that is intended to measure the difference in the expression profile of a whole complex in a control versus Rapamycin setting. The score takes into account the connections in the inferred PPI and the relative "amount" of the protein found in the proteomics data. Given a weighted PPI network w, the score is defined as:

$$\frac{\sum_E A_i A_j w_{ij}}{|E|}$$

where E is the set of edges in network w, A_i is the amount of protein i and w_{ij} is the weight between proteins i and j in the network.

While inferring the complexes, we do not take into account the gene expression data, and this creates a small complication to the interpretation of the score. Some complexes have among its component proteins for which we do not have the expression amounts. This reduces the number of complexes for which we

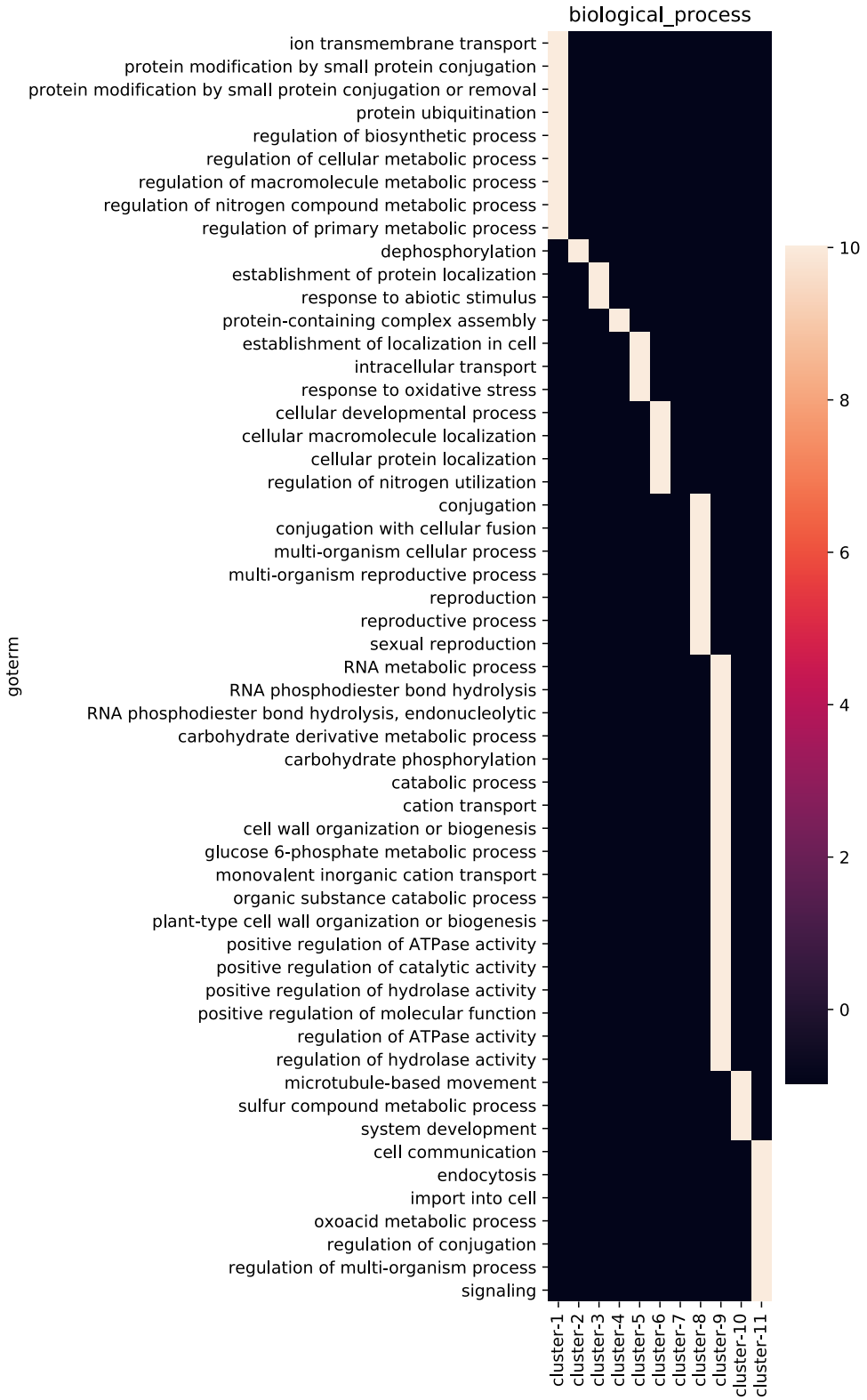


Figure 6.1 – *Over-representation analysis for the periodic proteins. A highly overrepresented function is assigned a clearer score. The over-representation score is similar to that of ICRep, with significant associations set at p-value* > 0.05

can accurately compute the score without making assumptions over the real distribution of the data.

Furthermore, This score might be modified in the future to better reflect topological properties of the graph. For instance, dividing by $|E|$ might be too stringent. Using this denominator hints at the presumption that all of these edges are binary, which is not the case due to the presence of interologs in the set of edges. A possible improvement is to redefine the score as:

$$\frac{\sum_E A_i A_j w_{ij}}{\sum_E w_{ij}}$$

This change implies that the topology of the network considers the weights of the edges and not merely the number of links within the complex. We are currently working on properly interpreting such modifications to the score.

6.2 prot2vec

In many fields such as natural language processing (NLP) and network science, a high-dimensional embedding has been useful to describe relevant components such as words or nodes [168, 169]. These dense embeddings, if done correctly, allow the use of arithmetic manipulation of the vectors that represent such elements.

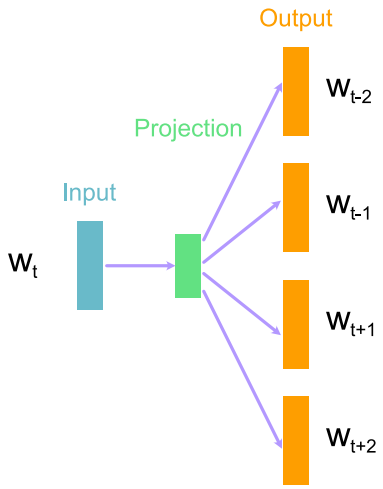


Figure 6.2 – *The skip-gram architecture. The training objective is to learn vectors that are good at predicting the nearby context. Originally, the input would be a word, and the context will be the surrounding words in the sentence. node2vec changed the configuration so that the context are close nodes in the network.*

Word2Vec [168, 170] introduces the skip-gram, a neural network architecture that has proven very useful for embeddings in NLP (Figure 6.2). With this architecture, they embed words in a way that words with similar meaning are group together, and they discovered that the dense embeddings encode many linguistic regularities and patterns.

node2vec [169] extends the use of this architecture to a network setting, in which they embed nodes. A very important concept when dealing with skip-grams is the context of the object being embedded. In NLP, this is naturally the surrounding words in a sentence. In node2vec, the context is determined by a random walk in the graph that can be tuned to favour a breadth-first (BFS)[1] or depth-first (DFS)[2] manner, or a combination of these two.

An application of node2vec is the prediction of PPI. It is important to notice that node2vec will use only the topological information of current PPI, or will need a modification to the meaning of the edges in the network to integrate other characteristic of the proteins. In other words, from the point of view of node2vec, every node is an abstract object that is characterised solely by its place on the network. We think that adding information from the sequence to the node2vec embedding can improve the performance by adding more features to the nodes.

We extended the skip-gram architecture so that the input and output layers take into account the sequence of amino acids of the proteins involved. This means that the architecture is expanded. First, the input layer now starts from a one-hot encoded representation of the sequence. Then, the output layer now attempts to recover the sequence of the neighbouring proteins. These changes are depicted in Figure 6.3. We follow the node2vec method to build the context of every protein.

Preliminary results show that our modifications to the architecture make a slight improvement to the performance of predicting PPI. However, many aspects remain to be tested. In particular, we are testing which search strategy

[1]BFS is a traversing algorithm that will visit every node in a "breadth first" basis. This is, starting from a node a, visit the first child of node a, and then visit subsequent children of a, and only carry on to "deeper" nodes after every children of a was visited.

[2]DFS is a traversing algorithm that will visit nodes in a "depth first" basis. This is, starting from a node a, it will visit the first child of a, and then the first child of that node, and it will carry on until it finds a leaf node before carrying on to the secong child of a.

(BFS or DFS) works better for this application. Also, what is a possible interpretation of vector operations done in the embedding space, and their potential applications. The functional characterisation of proteins could be achieved in this vectorial space, as not only the topological features of the network would be encoded, but also the sequence itself. The rationale behind expecting functional groups of proteins to be embedded together is that the local context will be "seen" by prot2vec, reinforcing the unique features that are related to the collection of sequences that form a community that can be topologically similar to a different community, but that has a unique set of members, which should set them apart.

Other areas of explorations for prot2vec have to do with the possibility of extending the information associated with the proteins to be embedded by the system. As evidenced by a lot of well-performing algorithms, extra information about the genes such as expression profiles and domain architectures can increase the performance of a predictor. I think that such side information, if encoded and treated correctly, can greatly contribute to a general characterisation of the genes in a vectorial space. As a first step, homology relations could be somewhat encoded by prot2vec by simply training it with proteins from multiple organisms. Similar sequences will tend to cluster together in the vectorial space, and their interacting neighbours too. This will also make of prot2vec a promising tool for predicting protein-protein interactions.

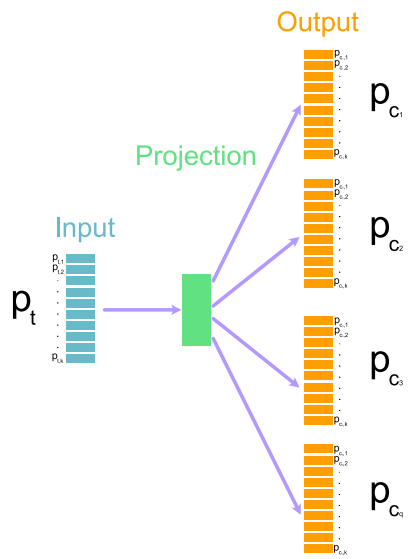


Figure 6.3 – *The prot2vec architecture in its current form. In comparison to the original skip-gram architecture, the input and output layers are expanded to accommodate for the sequence of aminoacids of a target protein p_t and its context $\{p_{c_1}, p_{c_2}, \cdots, p_{c_q}\}$. Context proteins are chosen using in the same way node2vec chooses context nodes.*

6.2.1 Implementation

The current implementation of prot2vec is done using the tensorflow library and the keras API. The code structure shares the same versatility of S2F, as the same conventions and tools are used. The current stage of the implementation is still in its early evaluation phase. Several modifications to the architecture are still in need of exploration:

- Use a recurrent unit such as the long short term memory (LSTM) to better model the order of the sequence.
- See if performing data augmentation on the padding improves performance. This is, instead of padding all the sequences to the right, create new samples with different padding.
- Evaluate prot2vec on other tasks that are not related to protein function such as PPI prediciton and network combination.

A new module developed specifically for prot2vec is `FASTATool`. In similar fashion to `GOTool`, this library provides a fast parser for FASTA files an

provides useful utilities to dealing with this format in a deep learning context. FASTA files can be translated into one-hot encoded versions of the sequences, with a vocabulary that comprises every amino acid and a *stop* token. It also provides a generator compatible with the keras API, that can be used for efficiently training a tensorflow model without the need to pre-process files in bulk. I intend to release `FASTATool` as a stand-alone tool alongside `GOTool`.

The source code will be made available upon publication.

A

S2F Appendix

A.1 CAFA π histograms and mappings

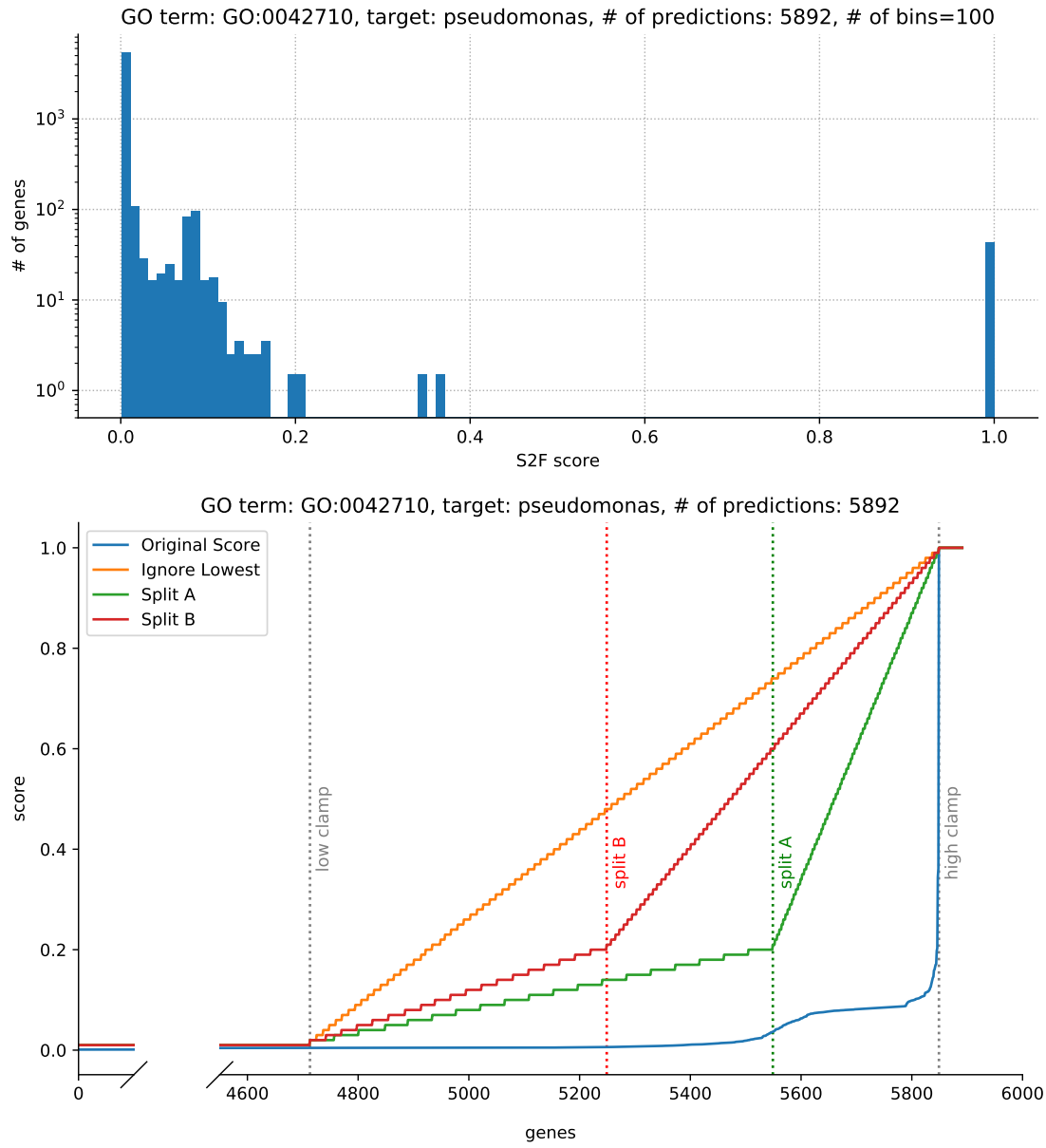


Figure A.1 – *Histogram of S2F scores and comparison of all strategies for biofilm formation (GO:0042710) on Pseudomonas.*

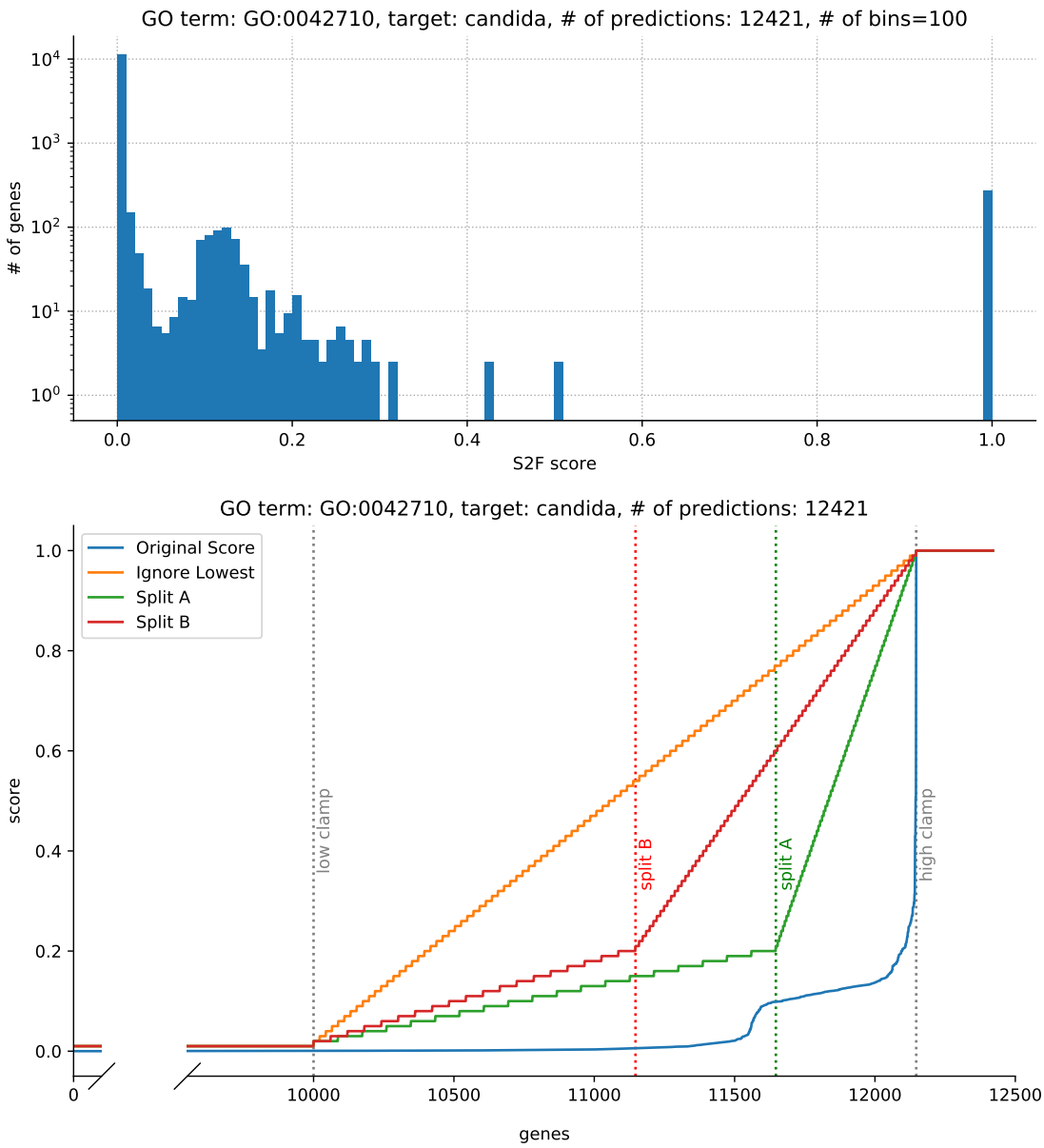


Figure A.2 – *Histogram of S2F scores and comparison of all strategies for biofilm formation (GO:0042710) on Candida.*

A.2 Detailed performance results

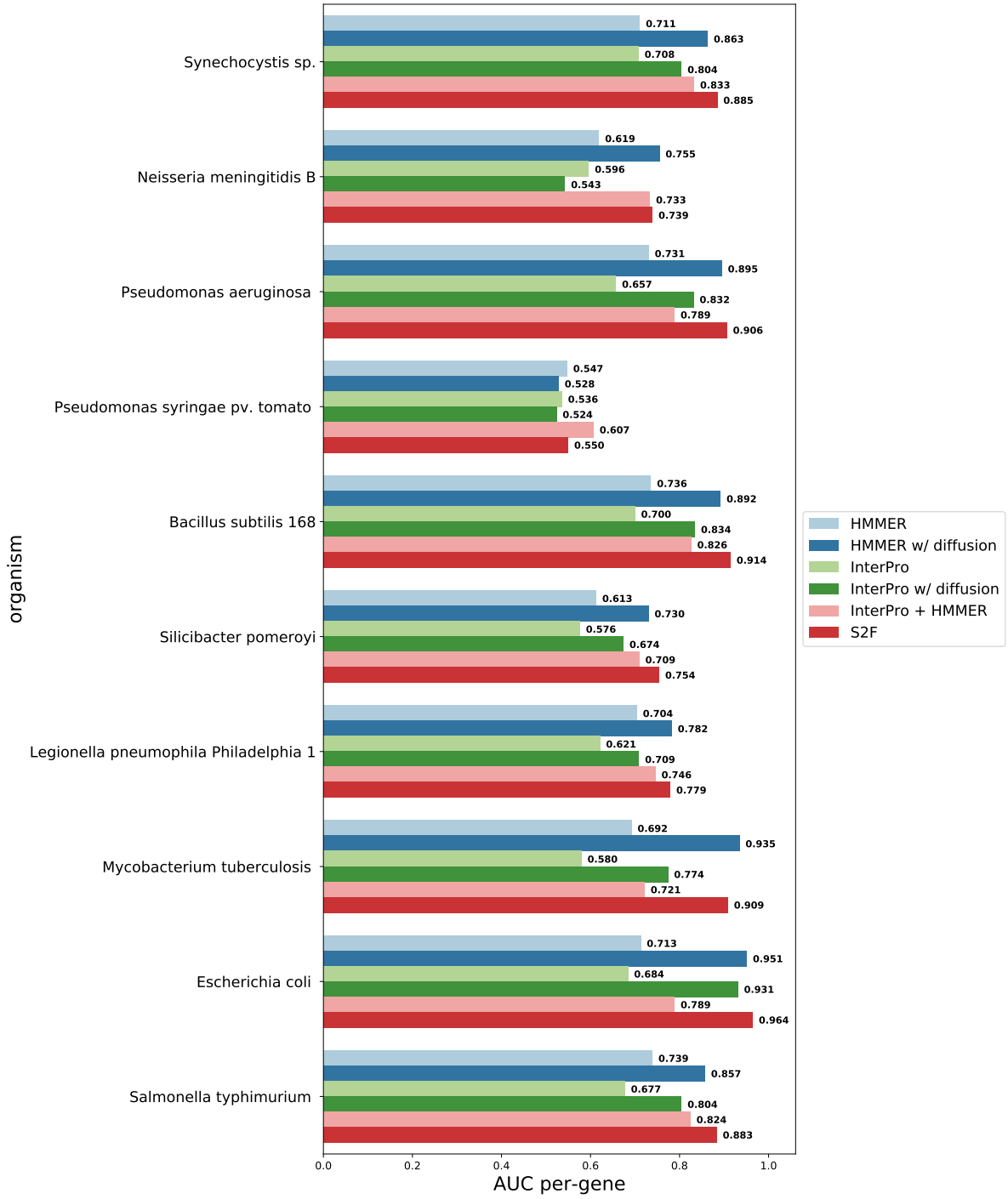


Figure A.3 – *AUC* per-gene

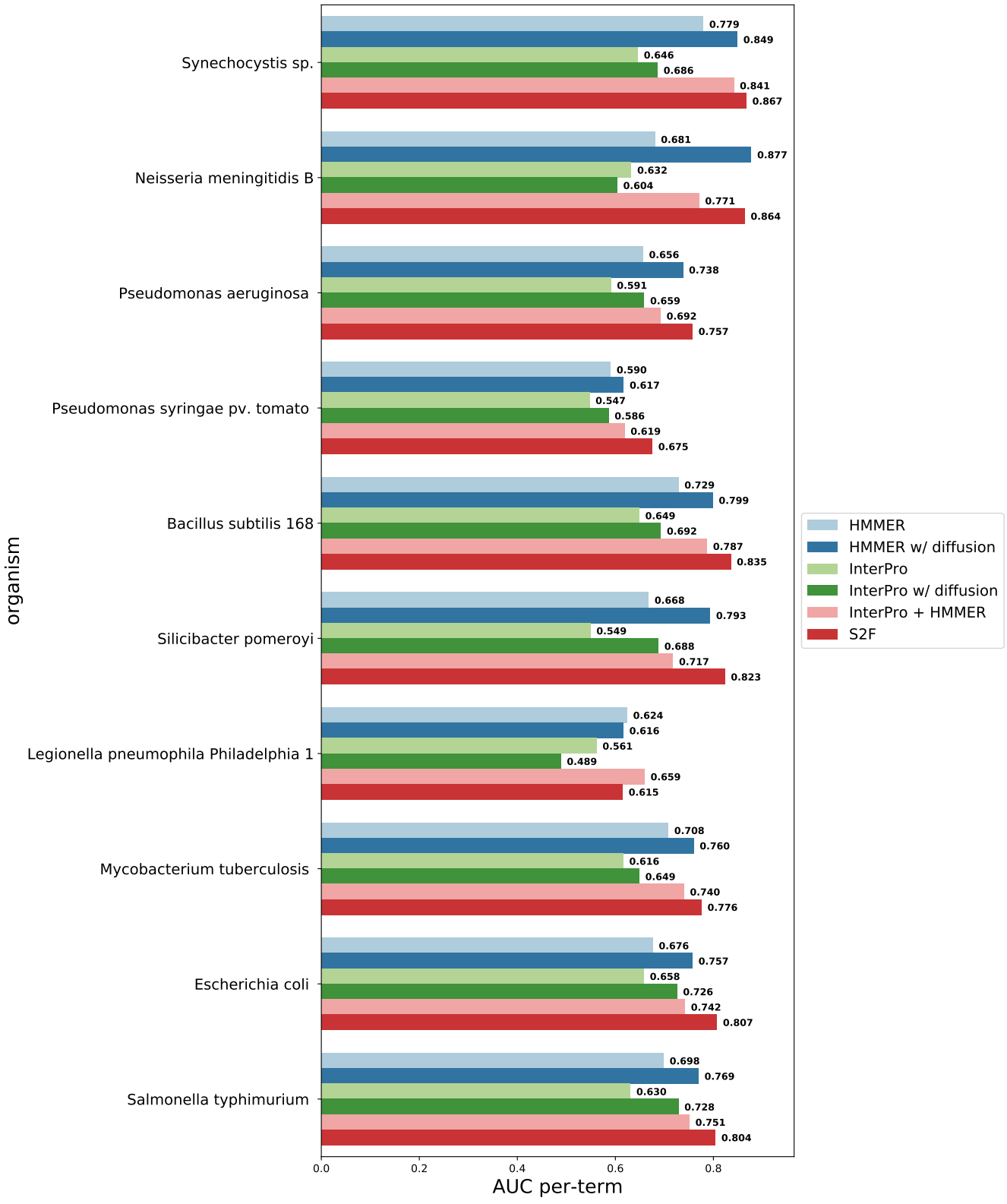


Figure A.4 – *AUC* per-term

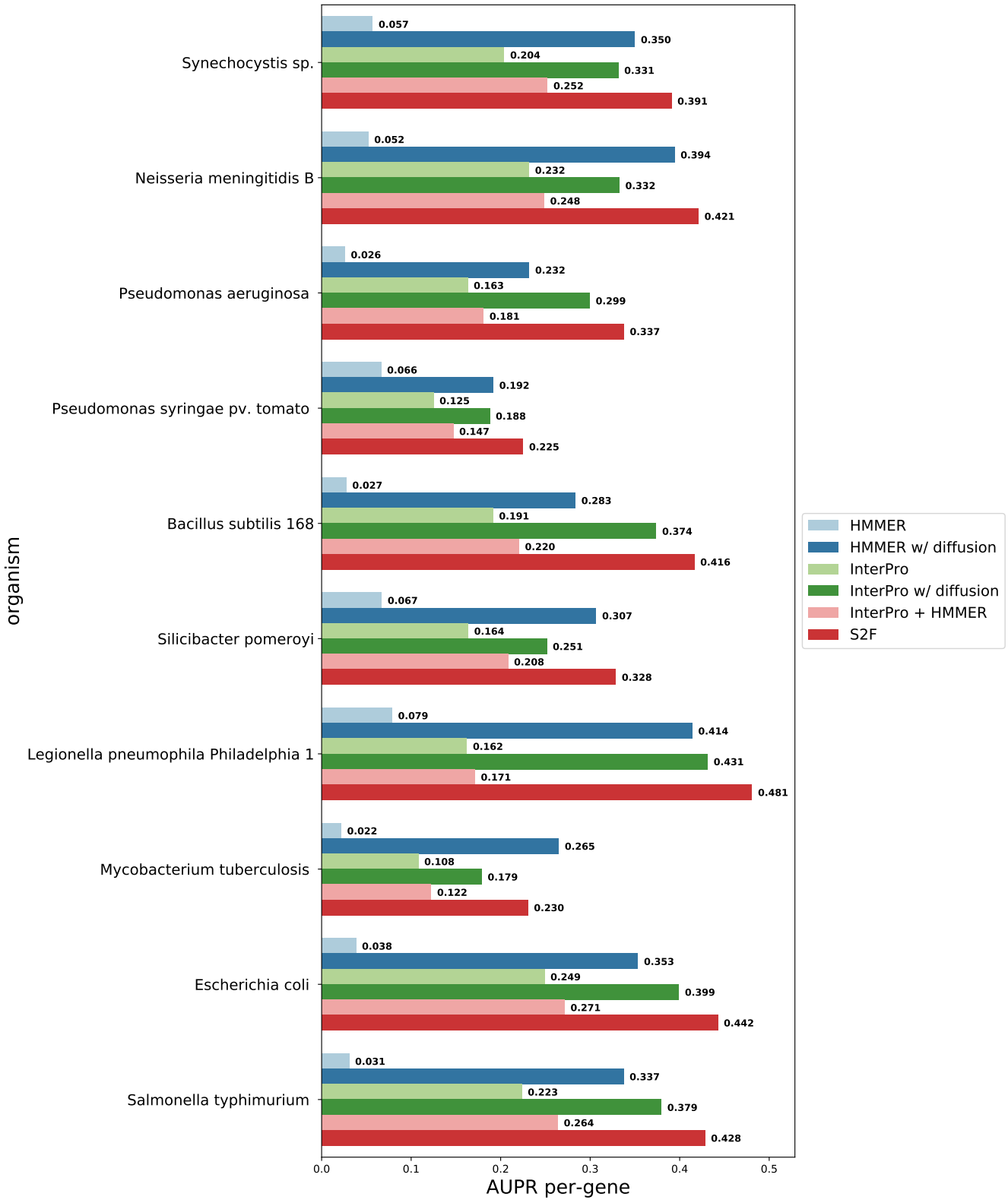


Figure A.5 – *AUPR* per-gene

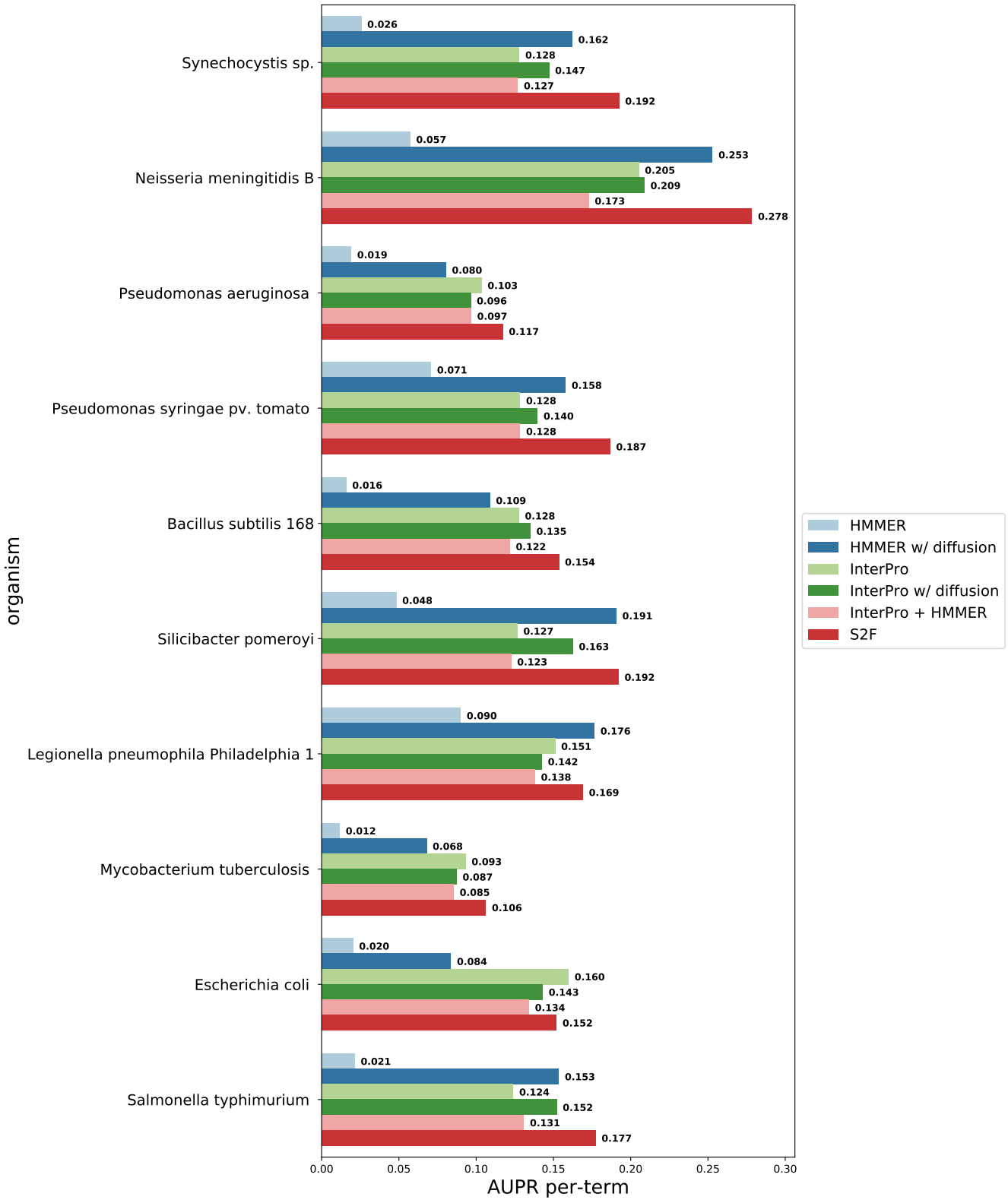


Figure A.6 – *AUPR* per-term

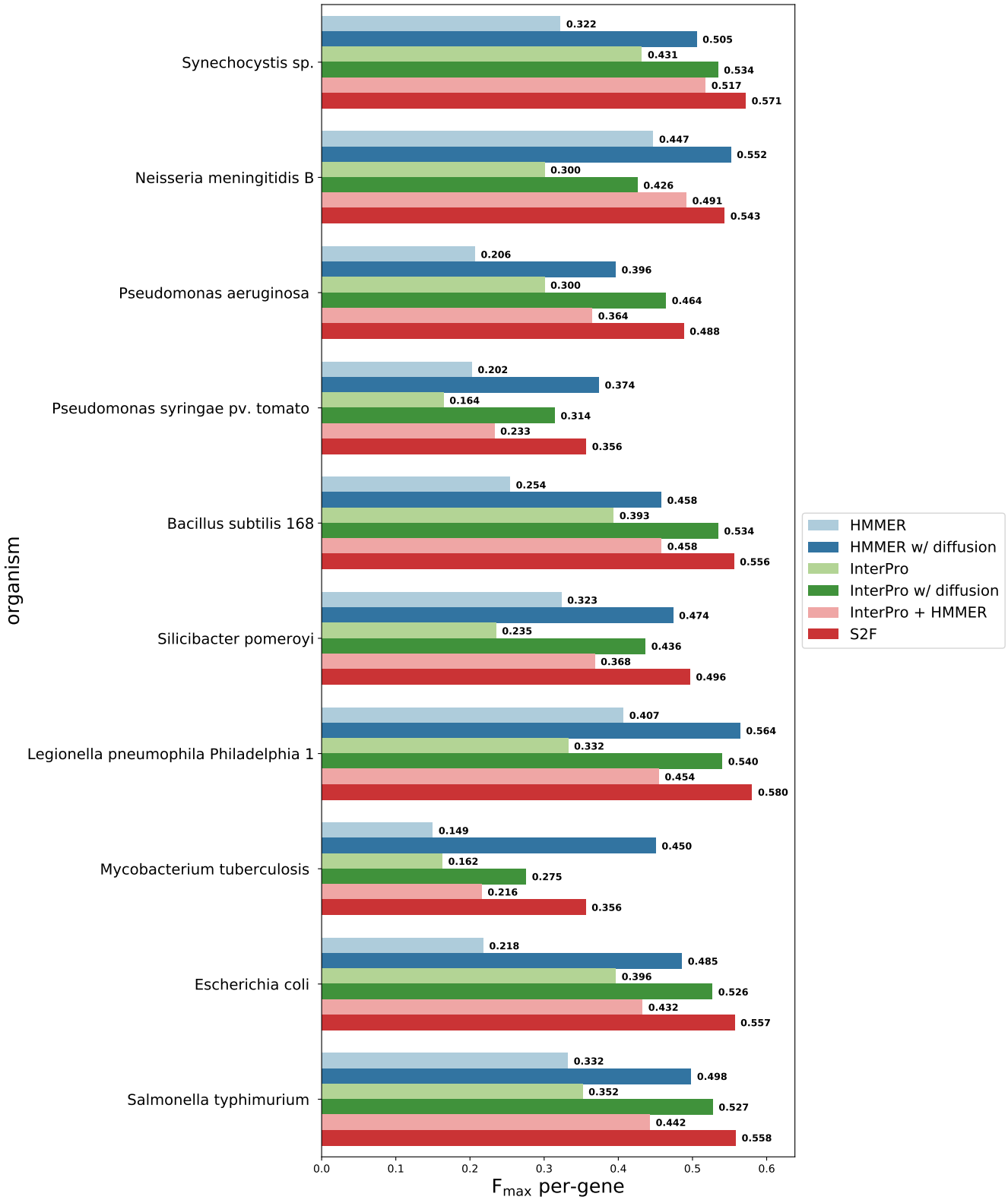


Figure A.7 – $F_{\max}$ per-gene

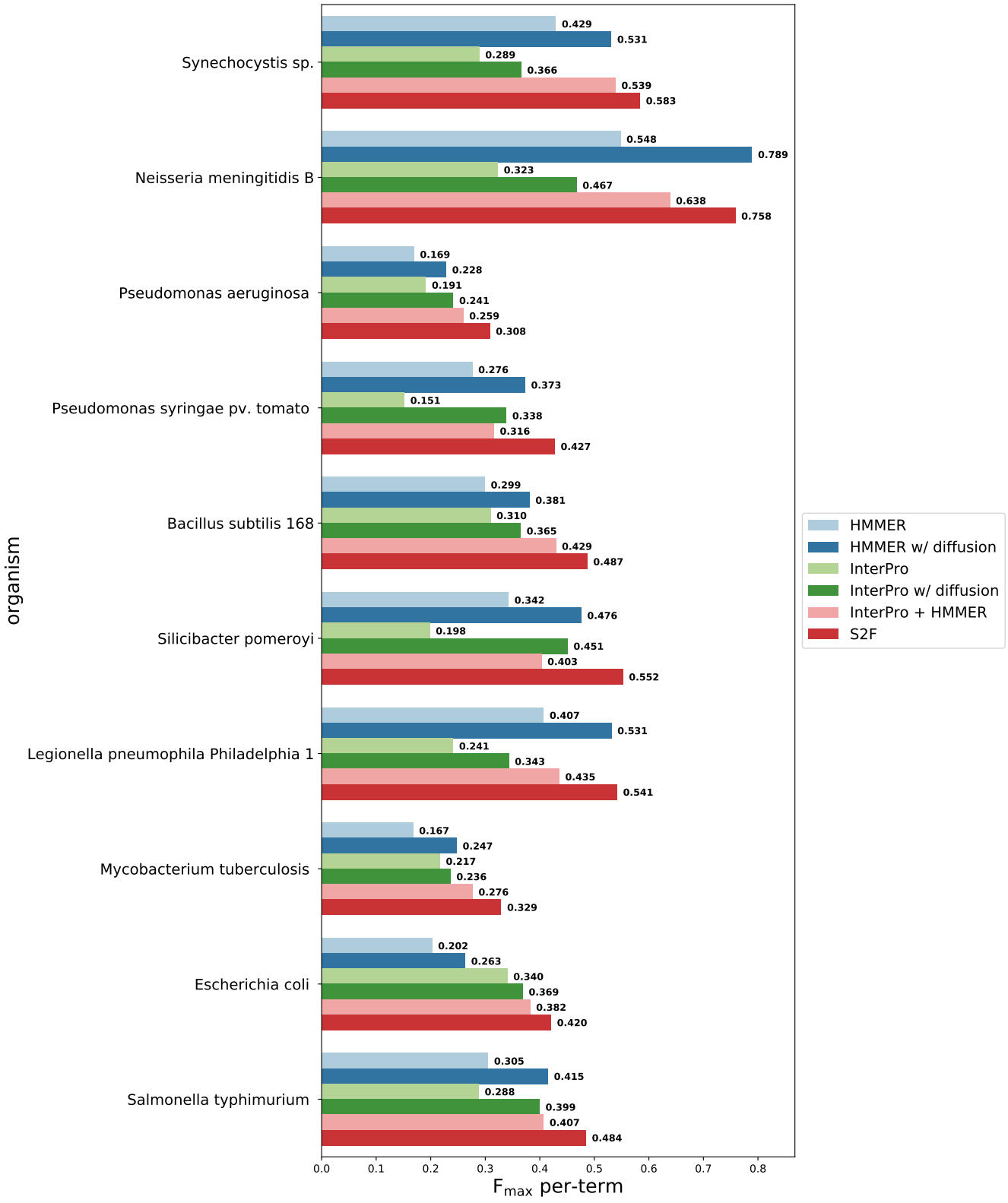


Figure A.8 – $F_{\max}$ per-term

Bibliography

[1] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock, "Gene Ontology: tool for the unification of biology," *Nature genetics*, vol. 25, pp. 25–29, May 2000.

[2] A. Shehu, D. Barbar, and K. Molloy, "A Survey of Computational Methods for Protein Function Prediction," in *Big Data Analytics in Genomics* (K.-C. Wong, ed.), pp. 225–298, Cham: Springer International Publishing, 2016.

[3] Y. Jiang, T. R. Oron, W. T. Clark, A. R. Bankapur, D. DAndrea, R. Lepore, C. S. Funk, I. Kahanda, K. M. Verspoor, A. Ben-Hur, D. C. E. Koo, D. Penfold-Brown, D. Shasha, N. Youngs, R. Bonneau, A. Lin, S. M. E. Sahraeian, P. L. Martelli, G. Profiti, R. Casadio, R. Cao, Z. Zhong, J. Cheng, A. Altenhoff, N. Skunca, C. Dessimoz, T. Dogan, K. Hakala, S. Kaewphan, F. Mehryary, T. Salakoski, F. Ginter, H. Fang, B. Smithers, M. Oates, J. Gough, P. Trnen, P. Koskinen, L. Holm, C.-T. Chen, W.-L. Hsu, K. Bryson, D. Cozzetto, F. Minneci, D. T. Jones, S. Chapman, D. BKC, I. K. Khan, D. Kihara, D. Ofer, N. Rappoport, A. Stern, E. Cibrian-Uhalte, P. Denny, R. E. Foulger, R. Hieta, D. Legge, R. C. Lovering, M. Magrane, A. N. Melidoni, P. Mutowo-Meullenet, K. Pichler, A. Shypitsyna, B. Li, P. Zakeri, S. ElShal, L.-C. Tranchevent, S. Das, N. L. Dawson, D. Lee, J. G. Lees, I. Sillitoe, P. Bhat, T. Nepusz, A. E. Romero, R. Sasidharan, H. Yang, A. Paccanaro, J. Gillis, A. E. Sedeo-Corts, P. Pavlidis, S. Feng, J. M. Cejuela, T. Goldberg, T. Hamp, L. Richter, A. Salamov, T. Gabaldon, M. Marcet-Houben,

F. Supek, Q. Gong, W. Ning, Y. Zhou, W. Tian, M. Falda, P. Fontana, E. Lavezzo, S. Toppo, C. Ferrari, M. Giollo, D. Piovesan, S. C. Tosatto, A. del Pozo, J. M. Fernndez, P. Maietta, A. Valencia, M. L. Tress, A. Benso, S. Di Carlo, G. Politano, A. Savino, H. U. Rehman, M. Re, M. Mesiti, G. Valentini, J. W. Bargsten, A. D. J. van Dijk, B. Gemovic, S. Glisic, V. Perovic, V. Veljkovic, N. Veljkovic, D. C. Almeida-e Silva, R. Z. N. Vencio, M. Sharan, J. Vogel, L. Kansakar, S. Zhang, S. Vucetic, Z. Wang, M. J. E. Sternberg, M. N. Wass, R. P. Huntley, M. J. Martin, C. ODonovan, P. N. Robinson, Y. Moreau, A. Tramontano, P. C. Babbitt, S. E. Brenner, M. Linial, C. A. Orengo, B. Rost, C. S. Greene, S. D. Mooney, I. Friedberg, and P. Radivojac, "An expanded evaluation of protein function prediction methods shows an improvement in accuracy," *Genome Biology*, vol. 17, p. 184, Sept. 2016.

[4] H. Yu, N. M. Luscombe, H. X. Lu, X. Zhu, Y. Xia, J.-D. J. Han, N. Bertin, S. Chung, M. Vidal, and M. Gerstein, "Annotation Transfer Between Genomes: ProteinProtein Interologs and ProteinDNA Regulogs," *Genome Research*, vol. 14, pp. 1107–1118, June 2004.

[5] T. U. Consortium, "UniProt: a worldwide hub of protein knowledge," *Nucleic Acids Research*, vol. 47, pp. D506–D515, Jan. 2019.

[6] R. P. Huntley, T. Sawford, P. Mutowo-Meullenet, A. Shypitsyna, C. Bonilla, M. J. Martin, and C. O'Donovan, "The GOA database: gene Ontology annotation updates for 2015," *Nucleic Acids Research*, vol. 43, pp. D1057–1063, Jan. 2015.

[7] E. Tacconelli, E. Carrara, A. Savoldi, S. Harbarth, M. Mendelson, D. L. Monnet, C. Pulcini, G. Kahlmeter, J. Kluytmans, Y. Carmeli, M. Ouellette, K. Outterson, J. Patel, M. Cavaleri, E. M. Cox, C. R. Houchens, M. L. Grayson, P. Hansen, N. Singh, U. Theuretzbacher, N. Magrini, A. O. Aboderin, S. S. Al-Abri, N. Awang Jalil, N. Benzonana, S. Bhattacharya, A. J. Brink, F. R. Burkert, O. Cars, G. Cornaglia, O. J. Dyar, A. W. Friedrich, A. C. Gales, S. Gandra, C. G. Giske, D. A. Goff, H. Goossens, T. Gottlieb, M. Guzman Blanco, W. Hryniewicz, D. Kattula, T. Jinks, S. S. Kanj, L. Kerr, M.-P. Kieny, Y. S. Kim, R. S. Kozlov, J. Labarca, R. Laxminarayan, K. Leder, L. Leibovici, G. Levy-Hara, J. Littman, S. Malhotra-Kumar, V. Manchanda, L. Moja, B. Ndoye, A. Pan, D. L. Paterson, M. Paul, H. Qiu, P. Ramon-Pardo, J. Rodrguez-Bao, M. Sanguinetti, S. Sengupta, M. Sharland, M. Si-Mehand, L. L. Silver, W. Song, M. Steinbakk, J. Thomsen, G. E. Thwaites, J. W. van der Meer, N. Van Kinh, S. Vega, M. V. Villegas, A. Wechsler-Frds, H. F. L. Wertheim, E. Wesangula, N. Woodford, F. O. Yilmaz, and A. Zorzet, "Discovery, research, and development of new antibiotics: the

WHO priority list of antibiotic-resistant bacteria and tuberculosis," *The Lancet Infectious Diseases*, vol. 18, pp. 318–327, Mar. 2018.

[8] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *Journal of Molecular Biology*, vol. 215, pp. 403–410, Oct. 1990.

[9] "Hmmer." `http://hmmer.org/`.

[10] C. Bru, E. Courcelle, S. Carrre, Y. Beausse, S. Dalmar, and D. Kahn, "The ProDom database of protein domain families: more emphasis on 3d," *Nucleic Acids Research*, vol. 33, pp. D212–D215, Jan. 2005.

[11] A. Heger and L. Holm, "Exhaustive Enumeration of Protein Domain Families," *Journal of Molecular Biology*, vol. 328, pp. 749–767, May 2003.

[12] A. Marchler-Bauer, Y. Bo, L. Han, J. He, C. J. Lanczycki, S. Lu, F. Chitsaz, M. K. Derbyshire, R. C. Geer, N. R. Gonzales, M. Gwadz, D. I. Hurwitz, F. Lu, G. H. Marchler, J. S. Song, N. Thanki, Z. Wang, R. A. Yamashita, D. Zhang, C. Zheng, L. Y. Geer, and S. H. Bryant, "CDD/SPARCLE: functional classification of proteins via subfamily domain architectures," *Nucleic Acids Research*, vol. 45, pp. D200–D203, Jan. 2017.

[13] E. V. Koonin and M. Y. Galperin, "Evolutionary Concept in Genetics and Genomics," in *Sequence Evolution Function: Computational Approaches in Comparative Genomics* (E. V. Koonin and M. Y. Galperin, eds.), pp. 25–49, Boston, MA: Springer US, 2003.

[14] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences," *Journal of Molecular Biology*, vol. 147, pp. 195–197, Mar. 1981.

[15] S. R. Eddy, "Hidden Markov models," *Current Opinion in Structural Biology*, vol. 6, no. 3, pp. 361 – 365, 1996.

[16] A. L. Mitchell, T. K. Attwood, P. C. Babbitt, M. Blum, P. Bork, A. Bridge, S. D. Brown, H.-Y. Chang, S. El-Gebali, M. I. Fraser, J. Gough, D. R. Haft, H. Huang, I. Letunic, R. Lopez, A. Luciani, F. Madeira, A. Marchler-Bauer, H. Mi, D. A. Natale, M. Necci, G. Nuka, C. Orengo, A. P. Pandurangan, T. Paysan-Lafosse, S. Pesseat, S. C. Potter, M. A. Qureshi, N. D. Rawlings, N. Redaschi, L. J. Richardson, C. Rivoire, G. A. Salazar, A. Sangrador-Vegas, C. J. A. Sigrist, I. Sillitoe, G. G. Sutton, N. Thanki, P. D. Thomas,

S. C. E. Tosatto, S.-Y. Yong, and R. D. Finn, "InterPro in 2019: improving coverage, classification and access to protein sequence annotations," *Nucleic Acids Research*, vol. 47, pp. D351–D360, Jan. 2019.

[17] P. Jones, D. Binns, H.-Y. Chang, M. Fraser, W. Li, C. McAnulla, H. McWilliam, J. Maslen, A. Mitchell, G. Nuka, S. Pesseat, A. F. Quinn, A. Sangrador-Vegas, M. Scheremetjew, S.-Y. Yong, R. Lopez, and S. Hunter, "InterProScan 5: genome-scale protein function classification," *Bioinformatics*, vol. 30, pp. 1236–1240, May 2014.

[18] T. E. Lewis, I. Sillitoe, N. Dawson, S. D. Lam, T. Clarke, D. Lee, C. Orengo, and J. Lees, "Gene3d: Extensive prediction of globular domains in proteins," *Nucleic Acids Research*, vol. 46, pp. D435–D439, Jan. 2018.

[19] I. Pedruzzi, C. Rivoire, A. H. Auchincloss, E. Coudert, G. Keller, E. de Castro, D. Baratin, B. A. Cuche, L. Bougueleret, S. Poux, N. Redaschi, I. Xenarios, and A. Bridge, "HAMAP in 2015: updates to the protein family classification and annotation system," *Nucleic Acids Research*, vol. 43, pp. D1064–D1070, Jan. 2015.

[20] H. Mi, X. Huang, A. Muruganujan, H. Tang, C. Mills, D. Kang, and P. D. Thomas, "PANTHER version 11: expanded annotation data from Gene Ontology and Reactome pathways, and data analysis tool enhancements," *Nucleic Acids Research*, vol. 45, pp. D183–D189, Jan. 2017.

[21] S. El-Gebali, J. Mistry, A. Bateman, S. R. Eddy, A. Luciani, S. C. Potter, M. Qureshi, L. J. Richardson, G. A. Salazar, A. Smart, E. L. L. Sonnhammer, L. Hirsh, L. Paladin, D. Piovesan, S. C. E. Tosatto, and R. D. Finn, "The Pfam protein families database in 2019," *Nucleic Acids Research*, vol. 47, pp. D427–D432, Jan. 2019.

[22] A. N. Nikolskaya, C. N. Arighi, H. Huang, W. C. Barker, and C. H. Wu, "PIRSF Family Classification System for Protein Functional and Evolutionary Analysis," *Evolutionary Bioinformatics*, vol. 2, p. 117693430600200033, 2006.

[23] C. H. Wu, A. Nikolskaya, H. Huang, L.-S. L. Yeh, D. A. Natale, C. R. Vinayaka, Z.-Z. Hu, R. Mazumder, S. Kumar, P. Kourtesis, R. S. Ledley, B. E. Suzek, L. Arminski, Y. Chen, J. Zhang, J. L. Cardenas, S. Chung, J. CastroAlvear, G. Dinkov, and W. C. Barker, "PIRSF: family classification system at the Protein Information Resource," *Nucleic Acids Research*, vol. 32, pp. D112–D114, Jan. 2004.

[24] T. K. Attwood, A. Coletta, G. Muirhead, A. Pavlopoulou, P. B. Philippou, I. Popov, C. Rom-Mateo, A. Theodosiou, and A. L. Mitchell, "The PRINTS database: a fine-grained protein sequence annotation and analysis resourceits status in 2012," *Database*, vol. 2012, Jan. 2012.

[25] C. J. A. Sigrist, E. de Castro, L. Cerutti, B. A. Cuche, N. Hulo, A. Bridge, L. Bougueleret, and I. Xenarios, "New and continuing developments at PROSITE," *Nucleic Acids Research*, vol. 41, pp. D344–D347, Jan. 2013.

[26] I. Letunic and P. Bork, "20 years of the SMART protein domain annotation resource," *Nucleic Acids Research*, vol. 46, pp. D493–D496, Jan. 2018.

[27] T. Doerks, R. R. Copley, J. Schultz, C. P. Ponting, and P. Bork, "Systematic Identification of Novel Protein Domain Families Associated with Nuclear Functions," *Genome Research*, vol. 12, pp. 47–56, Jan. 2002.

[28] E. Akiva, S. Brown, D. E. Almonacid, A. E. Barber, A. F. Custer, M. A. Hicks, C. C. Huang, F. Lauck, S. T. Mashiyama, E. C. Meng, D. Mischel, J. H. Morris, S. Ojha, A. M. Schnoes, D. Stryke, J. M. Yunes, T. E. Ferrin, G. L. Holliday, and P. C. Babbitt, "The StructureFunction Linkage Database," *Nucleic Acids Research*, vol. 42, pp. D521–D530, Jan. 2014.

[29] A. P. Pandurangan, J. Stahlhacke, M. E. Oates, B. Smithers, and J. Gough, "The SUPERFAMILY 2.0 database: a significant proteome update and a new webserver," *Nucleic Acids Research*, vol. 47, pp. D490–D494, Jan. 2019.

[30] D. H. Haft, J. D. Selengut, R. A. Richter, D. Harkins, M. K. Basu, and E. Beck, "TIGRFAMs and Genome Properties in 2013," *Nucleic Acids Research*, vol. 41, pp. D387–D395, Jan. 2013.

[31] D. Szklarczyk, J. H. Morris, H. Cook, M. Kuhn, S. Wyder, M. Simonovic, A. Santos, N. T. Doncheva, A. Roth, P. Bork, L. J. Jensen, and C. von Mering, "The STRING database in 2017: quality-controlled protein-protein association networks, made broadly accessible," *Nucleic Acids Research*, vol. 45, no. D1, pp. D362–D368, 2017.

[32] C. von Mering, L. J. Jensen, B. Snel, S. D. Hooper, M. Krupp, M. Foglierini, N. Jouffre, M. A. Huynen, and P. Bork, "STRING: known and predicted proteinprotein associations, integrated and transferred across organisms," *Nucleic Acids Research*, vol. 33, pp. D433–D437, Jan. 2005.

[33] J. O. Korbel, L. J. Jensen, C. v. Mering, and P. Bork, "Analysis of genomic

context: prediction of functional associations from conserved bidirectionally transcribed gene pairs," *Nature Biotechnology*, vol. 22, pp. 911–917, July 2004.

[34] G. Kolesov, H. W. Mewes, and D. Frishman, "SNAPping up functionally related genes based on context information: a colinearity-free approach1 1edited by J. Thornton," *Journal of Molecular Biology*, vol. 311, pp. 639–656, Aug. 2001.

[35] T. Itoh, K. Takemoto, H. Mori, and T. Gojobori, "Evolutionary instability of operon structures disclosed by sequence comparisons of complete microbial genomes.," *Molecular Biology and Evolution*, vol. 16, pp. 332–346, Mar. 1999.

[36] A. K. Bansal, "An automated comparative analysis of 17 complete microbial genomes," *Bioinformatics*, vol. 15, pp. 900–908, Nov. 1999.

[37] C. J. V. Marcotte and E. M. Marcotte, "Predicting functional linkages from gene fusions with confidence.," *Applied bioinformatics*, vol. 1, no. 2, pp. 93–100, 2002.

[38] J. Wu, S. Kasif, and C. DeLisi, "Identification of functional links between genes using phylogenetic profiles," *Bioinformatics*, vol. 19, pp. 1524–1530, Aug. 2003.

[39] F. Enault, K. Suhre, C. Abergel, O. Poirot, and J.-M. Claverie, "Annotation of bacterial genomes using improved phylogenomic profiles," *Bioinformatics*, vol. 19, pp. i105–i107, July 2003.

[40] J. A. Eisen, "Phylogenomics: Improving Functional Predictions for Uncharacterized Genes by Evolutionary Analysis," *Genome Research*, vol. 8, pp. 163–167, Mar. 1998.

[41] B. E. Engelhardt, M. I. Jordan, J. R. Srouji, and S. E. Brenner, "Genome-scale phylogenetic function annotation of large and diverse protein families," *Genome Research*, vol. 21, pp. 1969–1980, Nov. 2011.

[42] J.-P. Vert, "A tree kernel to analyse phylogenetic profiles," *Bioinformatics*, vol. 18, pp. S276–S284, July 2002.

[43] K. Narra and L. Liao, "Use of extended phylogenetic profiles with e-values and support vector machines for protein family classification,"

[44] J. C. Kendrew, R. E. Dickerson, B. E. Strandberg, R. G. Hart, D. R. Davies, D. C. Phillips, and V. C. Shore, "Structure of Myoglobin: A Three-Dimensional Fourier Synthesis at 2 . Resolution," *Nature*, vol. 185, pp. 422–427, Feb. 1960.

[45] H. Braberg, B. M. Webb, E. Tjioe, U. Pieper, A. Sali, and M. S. Madhusudhan, "SALIGN: a web server for alignment of multiple protein sequences and structures," *Bioinformatics*, vol. 28, pp. 2072–2073, Aug. 2012.

[46] E. Krissinel and K. Henrick, "Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions," *Acta Crystallographica Section D: Biological Crystallography*, vol. 60, pp. 2256–2268, Dec. 2004.

[47] A. R. Ortiz, C. E. M. Strauss, and O. Olmea, "MAMMOTH (matching molecular models obtained from theory): an automated method for model comparison," *Protein Science: A Publication of the Protein Society*, vol. 11, pp. 2606–2621, Nov. 2002.

[48] I. N. Shindyalov and P. E. Bourne, "Protein structure alignment by incremental combinatorial extension (CE) of the optimal path.," *Protein Engineering, Design and Selection*, vol. 11, pp. 739–747, Sept. 1998.

[49] C. A. Orengo and W. R. Taylor, "SSAP: Sequential structure alignment program for protein structure comparison," in *Methods in Enzymology*, vol. 266 of *Computer Methods for Macromolecular Sequence Analysis*, pp. 617–635, Academic Press, Jan. 1996.

[50] T. Madej, C. J. Lanczycki, D. Zhang, P. A. Thiessen, R. C. Geer, A. Marchler-Bauer, and S. H. Bryant, "MMDB and VAST+: tracking structural similarities between macromolecular complexes," *Nucleic Acids Research*, vol. 42, pp. D297–D303, Jan. 2014.

[51] N. N. Alexandrov, "SARFing the PDB," *Protein Engineering, Design and Selection*, vol. 9, pp. 727–732, Sept. 1996.

[52] L. Holm and C. Sander, "Protein Structure Comparison by Alignment of Distance Matrices," *Journal of Molecular Biology*, vol. 233, pp. 123–138, Sept. 1993.

[53] J. Yang, R. Yan, A. Roy, D. Xu, J. Poisson, and Y. Zhang, "The I-TASSER Suite: protein structure and function prediction," *Nature Methods*, vol. 12, pp. 7–8, Jan. 2015.

[54] J. Yang, A. Roy, and Y. Zhang, "BioLiP: a semi-manually curated database for biologically relevant ligandprotein interactions," *Nucleic Acids Research*, vol. 41, pp. D1096–D1103, Jan. 2013.

[55] I. Wohlers, M. Le Boudic-Jamin, H. Djidjev, G. W. Klau, and R. Andonov, "Exact Protein Structure Classification Using the Maximum Contact Map Overlap Metric," in *Algorithms for Computational Biology* (A.-H. Dediu, C. Martn-Vide, and B. Truthe, eds.), Lecture Notes in Computer Science, (Cham), pp. 262–273, Springer International Publishing, 2014.

[56] Z. Aung and K.-L. Tan, "Rapid 3d protein structure database searching using information retrieval techniques," *Bioinformatics*, vol. 20, pp. 1045–1052, May 2004.

[57] I. Budowski-Tal, Y. Nov, and R. Kolodny, "FragBag, an accurate representation of protein structure, retrieves structural neighbors from the entire PDB quickly and accurately," *Proceedings of the National Academy of Sciences*, vol. 107, pp. 3481–3486, Feb. 2010.

[58] M. Carpentier, S. Brouillet, and J. Pothier, "YAKUSA: A fast structural database scanning method," *Proteins: Structure, Function, and Bioinformatics*, vol. 61, no. 1, pp. 137–151, 2005.

[59] K. Molloy, M. J. Van, D. Barbara, and A. Shehu, "Exploring representations of protein structure for automated remote homology detection and mapping of protein structure space," *BMC Bioinformatics*, vol. 15, p. S4, July 2014.

[60] K. Guruprasad, M. S. Prasad, and G. R. Kumar, "Database of Structural Motifs in Proteins," *Bioinformatics*, vol. 16, pp. 372–375, Apr. 2000.

[61] S. Chakrabarti, K. Venkatramanan, and R. Sowdhamini, "SMoS: a database of structural motifs of protein superfamilies," *Protein Engineering, Design and Selection*, vol. 16, pp. 791–793, Nov. 2003.

[62] Y. Hou, W. Hsu, M. L. Lee, and C. Bystroff, "Efficient remote homology detection using local structure," *Bioinformatics*, vol. 19, pp. 2294–2301, Nov. 2003.

[63] A. Vazquez, A. Flammini, A. Maritan, and A. Vespignani, "Global protein function prediction from protein-protein interaction networks," *Nature Biotechnology*, vol. 21, pp. 697–700, June 2003.

[64] S. Oliver, "Guilt-by-association goes global," *Nature*, vol. 403, pp. 601–602, Feb. 2000.

[65] P. Legrain, J. Wojcik, and J.-M. Gauthier, "Proteinprotein interaction maps: a lead towards cellular functions," *Trends in Genetics*, vol. 17, pp. 346–352, June 2001.

[66] I. Xenarios, D. W. Rice, L. Salwinski, M. K. Baron, E. M. Marcotte, and D. Eisenberg, "DIP: the Database of Interacting Proteins," *Nucleic Acids Research*, vol. 28, pp. 289–291, Jan. 2000.

[67] C. Stark, B.-J. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers, "BioGRID: a general repository for interaction datasets," *Nucleic Acids Research*, vol. 34, pp. D535–D539, Jan. 2006.

[68] H. Hermjakob, L. MontecchiPalazzi, C. Lewington, S. Mudali, S. Kerrien, S. Orchard, M. Vingron, B. Roechert, P. Roepstorff, A. Valencia, H. Margalit, J. Armstrong, A. Bairoch, G. Cesareni, D. Sherman, and R. Apweiler, "IntAct: an open source molecular interaction database," *Nucleic Acids Research*, vol. 32, pp. D452–D455, Jan. 2004.

[69] A. Zanzoni, L. Montecchi-Palazzi, M. Quondam, G. Ausiello, M. Helmer-Citterich, and G. Cesareni, "MINT: a Molecular INTeraction database," *FEBS Letters*, vol. 513, no. 1, pp. 135–140, 2002.

[70] V. G. Tarcea, T. Weymouth, A. Ade, A. Bookvich, J. Gao, V. Mahavisno, Z. Wright, A. Chapman, M. Jayapandian, A. zgr, Y. Tian, J. Cavalcoli, B. Mirel, J. Patel, D. Radev, B. Athey, D. States, and H. V. Jagadish, "Michigan molecular interactions r2: from interacting proteins to pathways," *Nucleic Acids Research*, vol. 37, pp. D642–D646, Jan. 2009.

[71] B. Schwikowski, P. Uetz, and S. Fields, "A network of proteinprotein interactions in yeast," *Nature Biotechnology*, vol. 18, pp. 1257–1261, Dec. 2000.

[72] D. Piovesan, M. Giollo, C. Ferrari, and S. C. E. Tosatto, "Protein function prediction using guilty by association from interaction networks," *Amino Acids*, vol. 47, pp. 2583–2592, Dec. 2015.

[73] H. N. Chua, W.-K. Sung, and L. Wong, "Exploiting indirect neighbours and topological weight to predict protein function from proteinprotein interactions," *Bioinformatics*, vol. 22, pp. 1623–1630, July 2006.

[74] D. Wang and J. Hou, "Explore the hidden treasure in proteinprotein interaction networks An iterative model for predicting protein functions," *Journal of Bioinformatics and Computational Biology*, vol. 13, p. 1550026, Sept. 2015.

[75] J. Gillis and P. Pavlidis, "The role of indirect connections in gene networks in predicting function," *Bioinformatics*, vol. 27, pp. 1860–1866, July 2011.

[76] H. Li, P. Tong, J. Gallegos, E. Dimmer, G. Cai, J. J. Molldrem, and S. Liang, "PAND: A Distribution to Identify Functional Linkage from Networks with Preferential Attachment Property," *PLOS ONE*, vol. 10, p. e0127968, July 2015.

[77] A.-L. Barabsi, "Scale-Free Networks: A Decade and Beyond," *Science*, vol. 325, pp. 412–413, July 2009.

[78] G. D. Bader and C. W. Hogue, "An automated method for finding molecular complexes in large protein interaction networks," *BMC Bioinformatics*, vol. 4, p. 2, Jan. 2003.

[79] A. J. Enright, S. Van Dongen, and C. A. Ouzounis, "An efficient algorithm for large-scale detection of protein families," *Nucleic Acids Research*, vol. 30, pp. 1575–1584, Apr. 2002.

[80] B. J. Frey and D. Dueck, "Clustering by Passing Messages Between Data Points," *Science*, vol. 315, pp. 972–976, Feb. 2007.

[81] A. D. King, N. Prulj, and I. Jurisica, "Protein complex prediction via cost-based clustering," *Bioinformatics*, vol. 20, pp. 3013–3020, Nov. 2004.

[82] B. Adamcsek, G. Palla, I. J. Farkas, I. Dernyi, and T. Vicsek, "CFinder: locating cliques and overlapping modules in biological networks," *Bioinformatics*, vol. 22, pp. 1021–1023, Apr. 2006.

[83] G. Liu, L. Wong, and H. N. Chua, "Complex discovery from weighted PPI networks," *Bioinformatics*, vol. 25, pp. 1891–1897, Aug. 2009.

[84] K. Macropol, T. Can, and A. K. Singh, "RRW: repeated random walks on genome-scale protein networks for local cluster discovery," *BMC Bioinformatics*, vol. 10, p. 283, Sept. 2009.

[85] T. Nepusz, H. Yu, and A. Paccanaro, "Detecting overlapping protein complexes in protein-protein interaction networks," *Nature Methods*, vol. 9, pp. 471–472, May 2012.

[86] L. Chen, J. Xuan, R. B. Riggins, Y. Wang, and R. Clarke, "Identifying protein interaction subnetworks by a bagging Markov random field-based method," *Nucleic Acids Research*, vol. 41, pp. e42–e42, Jan. 2013.

[87] M. Re, M. Mesiti, and G. Valentini, "A Fast Ranking Algorithm for Predicting Gene Functions in Biomolecular Networks," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, pp. 1812–1818, Nov. 2012.

[88] S. Mostafavi, D. Ray, D. Warde-Farley, C. Grouios, and Q. Morris, "GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function," *Genome Biology*, vol. 9, no. Suppl 1, p. S4, 2008.

[89] M. Frasca, A. Bertoni, M. Re, and G. Valentini, "A neural network algorithm for semi-supervised node label learning from unbalanced data," *Neural Networks*, vol. 43, pp. 84–98, July 2013.

[90] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences*, vol. 79, pp. 2554–2558, Apr. 1982.

[91] M. Frasca, "Automated gene function prediction through gene multifunctionality in biological networks," *Neurocomputing*, vol. 162, pp. 48–56, Aug. 2015.

[92] M. Frasca, A. Bertoni, and G. Valentini, "UNIPred: Unbalance-Aware Network Integration and Prediction of Protein Functions," *Journal of Computational Biology*, vol. 22, pp. 1057–1074, Sept. 2015.

[93] E. Nabieva, K. Jim, A. Agarwal, B. Chazelle, and M. Singh, "Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps," *Bioinformatics*, vol. 21, pp. i302–i310, June 2005.

[94] Z. Bar-Joseph, A. Gitter, and I. Simon, "Studying and modelling dynamic biological processes using time-series gene expression data," *Nature Reviews Genetics*, vol. 13, pp. 552–564, Aug. 2012.

[95] M. G. Walker, W. Volkmuth, E. Sprinzak, D. Hodgson, and T. Klingler, "Prediction of Gene Function by Genome-Scale Expression Analysis: Prostate Cancer-Associated Genes," *Genome Research*, vol. 9, pp. 1198–1203, Dec. 1999.

[96] A. Ben-Dor, R. Shamir, and Z. Yakhini, "Clustering Gene Expression Patterns," *Journal of Computational Biology*, vol. 6, pp. 281–297, Oct. 1999.

[97] L. F. Wu, T. R. Hughes, A. P. Davierwala, M. D. Robinson, R. Stoughton, and S. J. Altschuler, "Large-scale prediction of Saccharomyces cerevisiae gene function using overlapping transcriptional clusters," *Nature Genetics*, vol. 31, pp. 255–265, July 2002.

[98] S. Swift, A. Tucker, V. Vinciotti, N. Martin, C. Orengo, X. Liu, and P. Kellam, "Consensus clustering and functional interpretation of gene-expression data," *Genome Biology*, vol. 5, p. R94, Nov. 2004.

[99] X. Zhou, M.-C. J. Kao, and W. H. Wong, "Transitive functional annotation by shortest-path analysis of gene expression data," *Proceedings of the National Academy of Sciences*, vol. 99, p. 12783, Oct. 2002.

[100] M. P. S. Brown, W. N. Grundy, D. Lin, N. Cristianini, C. W. Sugnet, T. S. Furey, M. Ares, and D. Haussler, "Knowledge-based analysis of microarray gene expression data by using support vector machines," *Proceedings of the National Academy of Sciences*, vol. 97, pp. 262–267, Jan. 2000.

[101] A. Mateos, J. Dopazo, R. Jansen, Y. Tu, M. Gerstein, and G. Stolovitzky, "Systematic Learning of Gene Functional Classes From DNA Array Expression Data by Using Multilayer Perceptrons," *Genome Research*, vol. 12, pp. 1703–1715, Nov. 2002.

[102] S. Makrodimitris, M. J. T. Reinders, and R. C. H. J. van Ham, "Metric learning on expression data for gene function prediction," *Bioinformatics*, July 2019.

[103] A. Lobley, M. B. Swindells, C. A. Orengo, and D. T. Jones, "Inferring Function Using Patterns of Native Disorder in Proteins," *PLOS Computational Biology*, vol. 3, p. e162, Aug. 2007.

[104] M. N. Wass, G. Barton, and M. J. E. Sternberg, "CombFunc: predicting protein function using heterogeneous data sources," *Nucleic Acids Research*, vol. 40, pp. W466–W470, July 2012.

[105] . S. Sara, V. Atalay, and R. Cetin-Atalay, "GOPred: GO Molecular Function Prediction by Combined Classifiers," *PLOS ONE*, vol. 5, p. e12382, Aug. 2010.

[106] M. Re and G. Valentini, "Integration of heterogeneous data sources for gene function prediction using decision templates and ensembles of learning machines," *Neurocomputing*, vol. 73, pp. 1533–1537, Mar. 2010.

[107] M. R and G. Valentini, "Simple ensemble methods are competitive with state-of-the-art data integration methods for gene function prediction," in *Proceedings of the third International Workshop on Machine Learning in Systems Biology* (S. Deroski, P. Guerts, and J. Rousu, eds.), vol. 8 of *Proceedings of Machine Learning Research*, (Ljubljana, Slovenia), pp. 98–111, PMLR, 05–06 Sep 2009.

[108] G. Obozinski, G. Lanckriet, C. Grant, M. I. Jordan, and W. S. Noble, "Consistent probabilistic outputs for protein function prediction," *Genome Biology*, vol. 9, p. S6, June 2008.

[109] L. Schietgat, C. Vens, J. Struyf, H. Blockeel, D. Kocev, and S. Deroski, "Predicting gene function using hierarchical multi-label decision tree ensembles," *BMC Bioinformatics*, vol. 11, p. 2, Jan. 2010.

[110] G. Valentini, "True Path Rule Hierarchical Ensembles for Genome-Wide Gene Function Prediction," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 8, pp. 832–847, May 2011.

[111] A. Renner and A. Aszdi, "High-throughput functional annotation of novel gene products using document clustering," in *Biocomputing 2000*, pp. 54–68, World Scientific, 1999.

[112] E. Eskin and E. Agichtein, "Combining text mining and sequence analysis to discover protein functional regions," in *Biocomputing 2004*, pp. 288–299, World Scientific, 2003.

[113] G. L. Holliday, R. Davidson, E. Akiva, and P. C. Babbitt, "Evaluating Functional Annotations of Enzymes Using the Gene Ontology," in *The Gene Ontology Handbook* (C. Dessimoz and N. kunca, eds.), Methods in Molecular Biology, pp. 111–132, New York, NY: Springer New York, 2017.

[114] C. Webber, "Functional Enrichment Analysis with Structural Variants: Pitfalls and Strategies," *Cytogenetic and Genome Research*, vol. 135, no. 3-4, pp. 277–285, 2011.

[115] D. W. Huang, B. T. Sherman, and R. A. Lempicki, "Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists," *Nucleic Acids Research*, vol. 37, pp. 1–13, Jan. 2009.

[116] U. Raudvere, L. Kolberg, I. Kuzmin, T. Arak, P. Adler, H. Peterson, and J. Vilo, "g:Profiler: a web server for functional enrichment analysis and conversions of gene lists (2019 update)," *Nucleic Acids Research*, vol. 47, pp. W191–W198, July 2019.

[117] Q. Zheng and X.-J. Wang, "GOEAST: a web-based software toolkit for Gene Ontology enrichment analysis," *Nucleic Acids Research*, vol. 36, pp. W358–W363, July 2008.

[118] R. Z. Vłncio and I. Shmulevich, "ProbCD: enrichment analysis accounting for categorization uncertainty," *BMC Bioinformatics*, vol. 8, p. 383, Oct. 2007.

[119] A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and J. P. Mesirov, "Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles," *Proceedings of the National Academy of Sciences*, vol. 102, p. 15545, Oct. 2005.

[120] M. V. Kuleshov, M. R. Jones, A. D. Rouillard, N. F. Fernandez, Q. Duan, Z. Wang, S. Koplev, S. L. Jenkins, K. M. Jagodnik, A. Lachmann, M. G. McDermott, C. D. Monteiro, G. W. Gundersen, and A. Ma'ayan, "Enrichr: a comprehensive gene set enrichment analysis web server 2016 update," *Nucleic Acids Research*, vol. 44, pp. W90–W97, July 2016.

[121] D. W. Huang, B. T. Sherman, and R. A. Lempicki, "Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources," *Nature Protocols*, vol. 4, pp. 44–57, Jan. 2009.

[122] D. V. Klopfenstein, L. Zhang, B. S. Pedersen, F. Ramrez, A. W. Vesztrocy, A. Naldi, C. J. Mungall, J. M. Yunes, O. Botvinnik, M. Weigel, W. Dampier, C. Dessimoz, P. Flick, and H. Tang, "GOATOOLS: A Python library for Gene Ontology analyses," *Scientific Reports*, vol. 8, pp. 1–17, July 2018.

[123] L. Cowen, T. Ideker, B. J. Raphael, and R. Sharan, "Network propagation: a universal amplifier of genetic associations," *Nature Reviews Genetics*, vol. 18, pp. 551–562, Sept. 2017.

[124] A. J. Walhout, R. Sordella, X. Lu, J. L. Hartley, G. F. Temple, M. A. Brasch, N. Thierry-Mieg, and M. Vidal, "Protein interaction mapping in C. elegans using proteins involved in vulval development," *Science (New York, N.Y.)*, vol. 287, pp. 116–122, Jan. 2000.

[125] F. Meyer, A. Goesmann, A. C. McHardy, D. Bartels, T. Bekel, J. Clausen, J. Kalinowski, B. Linke, O. Rupp, R. Giegerich, and A. Phler, "GenDB–an open source genome annotation system for prokaryote genomes," *Nucleic Acids Research*, vol. 31, pp. 2187–2195, Apr. 2003.

[126] G. H. Van Domselaar, P. Stothard, S. Shrivastava, J. A. Cruz, A. Guo, X. Dong, P. Lu, D. Szafron, R. Greiner, and D. S. Wishart, "BASys: a web server for automated bacterial genome annotation," *Nucleic Acids Research*, vol. 33, pp. W455–W459, July 2005.

[127] N. Maltsev, E. Glass, D. Sulakhe, A. Rodriguez, M. H. Syed, T. Bompada, Y. Zhang, and M. D'Souza, "PUMA2grid-based high-throughput analysis of genomes and metabolic pathways," *Nucleic Acids Research*, vol. 34, pp. D369–D372, Jan. 2006.

[128] D. Vallenet, L. Labarre, Z. Rouy, V. Barbe, S. Bocs, S. Cruveiller, A. Lajus, G. Pascal, C. Scarpelli, and C. Mdigue, "MaGe: a microbial genome annotation system supported by synteny results," *Nucleic Acids Research*, vol. 34, pp. 53–65, Jan. 2006.

[129] K. Bryson, V. Loux, R. Bossy, P. Nicolas, S. Chaillou, M. van de Guchte, S. Penaud, E. Maguin, M. Hoebeke, P. Bessires, and J.-F. Gibrat, "AGMIAL: implementing an annotation strategy for prokaryote genomes as a distributed system," *Nucleic Acids Research*, vol. 34, pp. 3533–3545, June 2006.

[130] V. M. Markowitz, K. Mavromatis, N. N. Ivanova, I.-M. A. Chen, K. Chu, and N. C. Kyrpides, "IMG ER: a system for microbial genome annotation expert review and curation," *Bioinformatics*, vol. 25, pp. 2271–2278, Sept. 2009.

[131] C. Yu, N. Zavaljevski, V. Desai, S. Johnson, F. J. Stevens, and J. Reifman, "The development of PIPA: an integrated and automated pipeline for genome-wide protein function annotation," *BMC Bioinformatics*, vol. 9, p. 52, Jan. 2008.

[132] T. Lima, A. H. Auchincloss, E. Coudert, G. Keller, K. Michoud, C. Rivoire, V. Bulliard, E. de Castro, C. Lachaize, D. Baratin, I. Phan, L. Bouguel-

eret, and A. Bairoch, "HAMAP: a database of completely sequenced microbial proteome sets and manually curated microbial protein families in UniProtKB/Swiss-Prot," *Nucleic Acids Research*, vol. 37, pp. D471–D478, Jan. 2009.

[133] M. Kanehisa and S. Goto, "KEGG: Kyoto Encyclopedia of Genes and Genomes," *Nucleic Acids Research*, vol. 28, pp. 27–30, Jan. 2000.

[134] A. Ben-Hur and W. S. Noble, "Kernel methods for predicting proteinprotein interactions," *Bioinformatics*, vol. 21, pp. i38–i46, June 2005.

[135] R. Sharan, S. Suthram, R. M. Kelley, T. Kuhn, S. McCuine, P. Uetz, T. Sittler, R. M. Karp, and T. Ideker, "Conserved patterns of protein interaction in multiple species," *Proceedings of the National Academy of Sciences*, vol. 102, pp. 1974–1979, Feb. 2005.

[136] H. Gu, P. Zhu, Y. Jiao, Y. Meng, and M. Chen, "PRIN: a predicted rice interactome network," *BMC Bioinformatics*, vol. 12, p. 161, May 2011.

[137] M. Lin, X. Shen, and X. Chen, "PAIR: the predicted Arabidopsis interactome resource," *Nucleic Acids Research*, vol. 39, pp. D1134–1140, Jan. 2011.

[138] V. Gligorijevi, M. Barot, and R. Bonneau, "deepNF: deep network fusion for protein function prediction," *Bioinformatics*, vol. 34, pp. 3873–3881, Nov. 2018.

[139] H. Cho, B. Berger, and J. Peng, "Compact Integration of Multi-Network Topology for Functional Analysis of Genes," *Cell Systems*, vol. 3, pp. 540–548.e5, Dec. 2016.

[140] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schlkopf, "Learning with Local and Global Consistency," p. 8, 2004.

[141] G. Poole and T. Boullion, "A Survey on M-Matrices," *SIAM Review*, vol. 16, pp. 419–427, Oct. 1974.

[142] W. T. Clark and P. Radivojac, "Information-theoretic evaluation of predicted ontological annotations," *Bioinformatics*, vol. 29, pp. i53–i61, July 2013.

[143] P. Radivojac, W. T. Clark, T. R. Oron, A. M. Schnoes, T. Wittkop, A. Sokolov, K. Graim, C. Funk, K. Verspoor, A. Ben-Hur, G. Pandey, J. M.

Yunes, A. S. Talwalkar, S. Repo, M. L. Souza, D. Piovesan, R. Casadio, Z. Wang, J. Cheng, H. Fang, J. Gough, P. Koskinen, P. Trnen, J. Nokso-Koivisto, L. Holm, D. Cozzetto, D. W. A. Buchan, K. Bryson, D. T. Jones, B. Limaye, H. Inamdar, A. Datta, S. K. Manjari, R. Joshi, M. Chitale, D. Kihara, A. M. Lisewski, S. Erdin, E. Venner, O. Lichtarge, R. Rentzsch, H. Yang, A. E. Romero, P. Bhat, A. Paccanaro, T. Hamp, R. Kaner, S. Seemayer, E. Vicedo, C. Schaefer, D. Achten, F. Auer, A. Boehm, T. Braun, M. Hecht, M. Heron, P. Hnigschmid, T. A. Hopf, S. Kaufmann, M. Kiening, D. Krompass, C. Landerer, Y. Mahlich, M. Roos, J. Bjrne, T. Salakoski, A. Wong, H. Shatkay, F. Gatzmann, I. Sommer, M. N. Wass, M. J. E. Sternberg, N. kunca, F. Supek, M. Bonjak, P. Panov, S. Deroski, T. muc, Y. A. I. Kourmpetis, A. D. J. van Dijk, C. J. F. t. Braak, Y. Zhou, Q. Gong, X. Dong, W. Tian, M. Falda, P. Fontana, E. Lavezzo, B. Di Camillo, S. Toppo, L. Lan, N. Djuric, Y. Guo, S. Vucetic, A. Bairoch, M. Linial, P. C. Babbitt, S. E. Brenner, C. Orengo, B. Rost, S. D. Mooney, and I. Friedberg, "A large-scale evaluation of computational protein function prediction," *Nature Methods*, vol. 10, pp. 221–227, Mar. 2013.

[144] I. Friedberg and P. Radivojac, "Community-Wide Evaluation of Computational Function Prediction," in *The Gene Ontology Handbook* (C. Dessimoz and N. kunca, eds.), Methods in Molecular Biology, pp. 133–146, New York, NY: Springer New York, 2017.

[145] C. Organisers, "Critical assessment for functional annotation pi - submission rules." `https://www.synapse.org/#!Synapse:syn11533497/wiki/497640`, 2017.

[146] N. Zhou, Y. Jiang, T. R. Bergquist, A. J. Lee, B. Z. Kacsoh, A. W. Crocker, K. A. Lewis, G. Georghiou, H. N. Nguyen, M. N. Hamid, L. Davis, T. Dogan, V. Atalay, A. S. Rifaioglu, A. Dalkiran, R. Cetin-Atalay, C. Zhang, R. L. Hurto, P. L. Freddolino, Y. Zhang, P. Bhat, F. Supek, J. M. Fernndez, B. Gemovic, V. R. Perovic, R. S. Davidovi, N. Sumonja, N. Veljkovic, E. Asgari, M. R. Mofrad, G. Profiti, C. Savojardo, P. L. Martelli, R. Casadio, F. Boecker, I. Kahanda, N. Thurlby, A. C. McHardy, A. Renaux, R. Saidi, J. Gough, A. A. Freitas, M. Antczak, F. Fabris, M. N. Wass, J. Hou, J. Cheng, J. Hou, Z. Wang, A. E. Romero, A. Paccanaro, H. Yang, T. Goldberg, C. Zhao, L. Holm, P. Trnen, A. J. Medlar, E. Zosa, I. Borukhov, I. Novikov, A. Wilkins, O. Lichtarge, P.-H. Chi, W.-C. Tseng, M. Linial, P. W. Rose, C. Dessimoz, V. Vidulin, S. Dzeroski, I. Sillitoe, S. Das, J. G. Lees, D. T. Jones, C. Wan, D. Cozzetto, R. Fa, M. Torres, A. W. Vesztrocy, J. M. Rodriguez, M. L. Tress, M. Frasca, M. Notaro, G. Grossi, A. Petrini, M. Re, G. Valentini, M. Mesiti, D. B. Roche, J. Reeb, D. W. Ritchie, S. Aridhi, S. Z. Alborzi, M.-D. Devignes, D. C. E. Koo, R. Bonneau, V. Gligorijevi,

M. Barot, H. Fang, S. Toppo, E. Lavezzo, M. Falda, M. Berselli, S. C. Tosatto, M. Carraro, D. Piovesan, H. U. Rehman, Q. Mao, S. Zhang, S. Vucetic, G. S. Black, D. Jo, D. J. Larsen, A. R. Omdahl, L. W. Sagers, E. Suh, J. B. Dayton, L. J. McGuffin, D. A. Brackenridge, P. C. Babbitt, J. M. Yunes, P. Fontana, F. Zhang, S. Zhu, R. You, Z. Zhang, S. Dai, S. Yao, W. Tian, R. Cao, C. Chandler, M. Amezola, D. Johnson, J.-M. Chang, W.-H. Liao, Y.-W. Liu, S. Pascarelli, Y. Frank, R. Hoehndorf, M. Kulmanov, I. Boudellioua, G. Politano, S. D. Carlo, A. Benso, K. Hakala, F. Ginter, F. Mehryary, S. Kaewphan, J. Bjrne, H. Moen, M. E. E. Tolvanen, T. Salakoski, D. Kihara, A. Jain, T. muc, A. Altenhoff, A. Ben-Hur, B. Rost, S. E. Brenner, C. A. Orengo, C. J. Jeffery, G. Bosco, D. A. Hogan, M. J. Martin, C. ODonovan, S. D. Mooney, C. S. Greene, P. Radivojac, and I. Friedberg, "The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens," *bioRxiv*, p. 653105, May 2019.

[147] H. Yang, T. Nepusz, and A. Paccanaro, "Improving GO semantic similarity measures by exploring the ontology beneath the terms and modelling uncertainty," *Bioinformatics*, vol. 28, pp. 1383–1389, May 2012.

[148] H. Caniza, A. E. Romero, S. Heron, H. Yang, A. Devoto, M. Frasca, M. Mesiti, G. Valentini, and A. Paccanaro, "GOssTo: a stand-alone application and a web tool for calculating semantic similarities on the Gene Ontology," *Bioinformatics*, vol. 30, pp. 2235–2236, Aug. 2014.

[149] T. Ohta, "Role of gene duplication in evolution," *Genome*, vol. 31, pp. 304–310, Jan. 1989.

[150] S. Koide, "Generation of new protein functions by nonhomologous combinations and rearrangements of domains and modules," *Current Opinion in Biotechnology*, vol. 20, pp. 398–404, Aug. 2009.

[151] M. Bashton and C. Chothia, "The Generation of New Protein Functions by the Combination of Domains," *Structure*, vol. 15, pp. 85–99, Jan. 2007.

[152] R. Sasidharan, T. Nepusz, D. Swarbreck, E. Huala, and A. Paccanaro, "GFam: a platform for automatic annotation of gene families," *Nucleic Acids Research*, vol. 40, pp. e152–e152, Oct. 2012.

[153] F. Sievers, A. Wilm, D. Dineen, T. J. Gibson, K. Karplus, W. Li, R. Lopez, H. McWilliam, M. Remmert, J. Sding, J. D. Thompson, and D. G. Higgins, "Fast, scalable generation of high-quality protein multiple sequence align-

ments using Clustal Omega," *Molecular Systems Biology*, vol. 7, p. 539, Jan. 2011.

[154] J. Mistry, R. D. Finn, S. R. Eddy, A. Bateman, and M. Punta, "Challenges in homology search: HMMER3 and convergent evolution of coiled-coil regions," *Nucleic Acids Research*, vol. 41, pp. e121–e121, July 2013.

[155] J. C. Wootton and S. Federhen, "[33] Analysis of compositionally biased regions in sequence databases," in *Methods in Enzymology*, vol. 266 of *Computer Methods for Macromolecular Sequence Analysis*, pp. 554–571, Academic Press, Jan. 1996.

[156] E. B. Camon, D. G. Barrell, E. C. Dimmer, V. Lee, M. Magrane, J. Maslen, D. Binns, and R. Apweiler, "An evaluation of GO annotation retrieval for BioCreAtIvE and GOA," *BMC Bioinformatics*, vol. 6, no. Suppl 1, p. S17, 2005.

[157] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: a practical and powerful approach to multiple testing," *Journal of the Royal statistical society: series B (Methodological)*, vol. 57, no. 1, pp. 289–300, 1995.

[158] R. J. Cho and M. J. Campbell, "Transcription, genomes, function," *Trends in Genetics*, vol. 16, no. 9, pp. 409–415, 2000.

[159] F. Mosteller and R. A. Fisher, "Questions and Answers," *The American Statistician*, vol. 2, no. 5, pp. 30–31, 1948.

[160] M. F. Porter, "An algorithm for suffix stripping," *Program*, Mar. 1980.

[161] R. Baeza-Yates and B. Ribeiro-Neto, *Modern information retrieval*, vol. 463. 1999.

[162] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, pp. 513–523, Jan. 1988.

[163] T. Joachims, "Text categorization with Support Vector Machines: Learning with many relevant features," Machine Learning: ECML-98, (Berlin, Heidelberg), pp. 137–142, Springer Berlin Heidelberg, 1998.

[164] K. Van Roey, S. Orchard, S. Kerrien, M. Dumousseau, S. Ricard-Blum,

H. Hermjakob, and T. J. Gibson, "Capturing cooperative interactions with the PSI-MI format," *Database*, vol. 2013, Jan. 2013.

[165] J. Jppner, U. Mubeen, A. Leisse, C. Caldana, A. Wiszniewski, D. Steinhauser, and P. Giavalisco, "The target of rapamycin kinase affects biomass accumulation and cell cycle progression by altering carbon/nitrogen balance in synchronized Chlamydomonas reinhardtii cells," *The Plant Journal*, vol. 93, no. 2, pp. 355–376, 2018.

[166] D. M. Goodstein, S. Shu, R. Howson, R. Neupane, R. D. Hayes, J. Fazo, T. Mitros, W. Dirks, U. Hellsten, N. Putnam, and D. S. Rokhsar, "Phytozome: a comparative platform for green plant genomics," *Nucleic Acids Research*, vol. 40, pp. D1178–D1186, Nov. 2011.

[167] S. Tyanova, T. Temu, P. Sinitcyn, A. Carlson, M. Y. Hein, T. Geiger, M. Mann, and J. Cox, "The Perseus computational platform for comprehensive analysis of (prote)omics data," *Nature Methods*, vol. 13, p. 731, June 2016.

[168] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *arXiv:1301.3781 [cs]*, Jan. 2013. arXiv: 1301.3781.

[169] A. Grover and J. Leskovec, "node2vec: Scalable Feature Learning for Networks," *arXiv:1607.00653 [cs, stat]*, July 2016. arXiv: 1607.00653.

[170] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," p. 9.