UNIVERSITÄT BONN
Rheinische Friedrich-Wilhelms-Universität Bonn

INRES
Institut für Nutzpflanzenwissenschaften und Ressourcenschutz

AHRD: Automatically Annotate Proteins with Human Readable Descriptions and Gene Ontology Terms

Dissertation
zur
Erlangung des Doktorgrades (Dr. rer. nat.)
der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Florian Boecker

aus

Köln

Bonn, 2021

Angefertigt mit Genehmigung der
Mathematisch-Naturwissenschaftlichen Fakultät der
Rheinischen Friedrich-Wilhelms-Universität Bonn

1. Gutachter: Prof. Dr. Heiko Schoof

2. Gutachter: Prof. Dr. Martin Hofmann-Apitius

Tag der Promotion: 14. Juli 2021

Erscheinungsjahr: 2021

Summary

In the postgenomic era it is impossible to annotate the majority of new proteins in any other way than with computational methods. Our tool AHRD automatically annotates proteins with human readable descriptions and Gene Ontology (GO) terms on a genomic scale. It does so by performing a lexical analysis modeled on the decision process of a human curator investigating the protein descriptions of homologous proteins found by sequence similarity.

The central questions of this thesis are how GO annotations can be accurately evaluated and how the annotation performance of AHRD can be increased.

To this end we firstly generated an unbiased ground truth set of high quality protein annotations with minimal redundancy. It contains many proteins that are difficult to annotate and thus facilitates contrasting annotation methods. Secondly, we implemented and tested three evaluation metrics for the congruence of GO term annotations. The third metric, which employs the structure of the Gene Ontology and the commonness of GO terms to determine the semantic similarity of GO annotations, is able to perform the most nuanced and consistent evaluation. In addition to a preexisting simulated annealing-based approach a genetic algorithm-based machine learning method was implemented to use the aforementioned evaluation metrics to optimize AHRD's input parameters. Although the genetic algorithm was only able to provide small improvements, they were statistically significant and parameter optimization proved to be necessary to achieve optimal annotation performance. In the style of the lexical analysis of candidate descriptions a new GO term-based analysis for candidate annotations was created. This was able to improve AHRD's GO annotation performance and also enabled the incorporation of new quality indicators such as GO term information content and annotation evidence codes which improved the performance further. It also facilitated the annotation with newly combined sets of GO terms instead of only fixed sets obtained from reference proteins. However, this approach proved to be not viable as it resulted in a significant regression of annotation performance. Using our evaluation method we were able to show that AHRD is able to predict description and GO annotations better and at a greater coverage than most of its competitors. Despite the fact that AHRD is tailored for the application to whole proteomes from all branches of life and for ease of use, in the CAFA3 challenge, a community-driven evaluation of GO annotation methods that often do not have these benefits, AHRD was able to show satisfactory results in most categories.

In conclusion, we demonstrated a reliable GO annotation evaluation method and used it to develop AHRD's GO annotation from an afterthought to a mature feature. We showed that AHRD is not only successful at the annotation of descriptions but also at GO terms, while staying applicable in whole genome projects.

Contents

List of Figures

List of Tables

List of Equations

Part I
Introduction

1 Background

Emerging technologies enable the sequencing of a human genome in under 24h [1]. In 2018 the cost for sequencing dropped below $1000 per genome [2]. With the time and cost efficiency of high throughput sequencing rapidly rising the amount of publicly available sequence data has grown massively [3]. As a consequence, biological databases such as the UniProtKB are flooded with new entries [4]. But there is an ever increasing discrepancy between the number of new functionally unknown entries and the number of actually known and well-annotated proteins [4]. Even for a model organism such as *Arabidopsis thaliana* where in 2013 94% of proteins had at least one GO annotation, only 39% of proteins had a functional annotation with experimental evidence [5]. To overcome this divide, protein functions can be predicted by computer-driven, automatic classification methods. But no automatic method always works correctly. And because existing annotations are often the source for the prediction of new ones, annotation errors can propagate through databases further and further [6, 7].

1.1 Describing Protein Function

Although human readable descriptions of proteins use scientific prose they are still made using natural language. This makes them easy to parse by humans but hard to compare computationally. On one hand computationally viewed, strings of characters can appear very similar but still carry semantically distinct concepts. On the other hand in text form, the same protein function can be expressed in many synonymic ways seen as completely different by a computer.

The Gene Ontology (GO) [8] is a controlled vocabulary of classes (GO terms) with relations forming a directed acyclic graph (DAG). These GO terms fall in one of three biological knowledge domains: Biological Process (a biological "program" facilitated by multiple molecular activities), Molecular Function (activity of a protein on the molecular level) and Cellular Component (part of the cell where the protein performs its function). Besides relations such as "regulates", which can span from one domain to another one, the most important ones are "is a" and "part of". These cannot go from one knowledge domain to the other and convey clear superclass-subtype (parent-child) relationships.

The Functional Catalogue (FunCat) [9] has a similar structure. But other than in the Gene Ontology, here every child can only have a single direct parent class.

MapMan [10] bins are biological concepts arranged in a hierarchical tree structure. They are part of the MapMan framework which was created specifically for plant genomics. MapMan bins encompass metabolic and regulatory processes, transcription factors, signaling pathways and stress response to biotic as well as abiotic factors.

The InterPro database [11] integrates signatures from 13 member databases. InterPro entries can be used to classify proteins into families and assign domains to them. These domains provided by InterPro are protein function archetypes that are annotated with human readable descriptions, GO terms and EC numbers.

EC (Enzyme Commission) numbers [12] are enzyme classifications. Their numerical classification scheme is based on recommendations by the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology (IUBMB). Thus, the components of an EC number can be directly "translated" into the name of an enzyme class.

KO (KEGG Orthology) numbers are the primary identifiers of the KEGG (Kyoto Encyclopedia of Genes and Genomes) [13] database. Through assignment to a particular KO number proteins are associated to enzymatic pathways, chemical reactions and more.

1.2 Protein Function Prediction

1.2.1 Sequence Similarity / Homology-Based Function Prediction

Homologous proteins share a common evolutionary history. When two sequences are more similar than what can be expected by chance, homology can be inferred. Under selective pressure, protein function is evolutionarily conserved. The function of a protein is dependent on its structure which in turn is determined by its amino acid sequence. The selective pressure on the function consequently also leads to a conservation of the protein sequence. Thus, to a certain extent, sequence similarity (i.e. homology) is linked to similarity of protein function. But the exact amount of sequence similarity that is sufficient to guarantee a similar structure is hard to pin down [14]. And the same can be said about the relationship of protein sequence to enzymatic function [15]. Nonetheless, sequence similarity is the basis of many successful protein function prediction methods. The operating principle is to find a protein with a sequence similar to the query protein in order to transfer the function to it, assuming homology between the two.

Because of this it is paramount to be able to quantify the similarity of the two sequences. The traditional way to do this is by aligning them. For the global alignment of two biological sequences, the Needleman-Wunsch algorithm [16], published in 1970, can be used. As an application of dynamic programming, it divides the problem in many small problems and uses the solutions of these small problems to find an optimal global alignment. But it has been recognized that proteins are not monolithic. Often they are made up of several structural domains. Protein domains are units of molecular function [17] making up a repertoire continuously recombined by evolution to form the immense diversity of protein function [18]. Thus, a global alignment of protein sequences will often miss crucial similarity of some of its parts. Therefore, the Smith-Waterman algorithm [19], a modification to the Needleman-Wunsch algorithm, can be used to compute optimal local alignments. Both algorithms give optimal alignments with respect the scoring system they are provided with. These systems have scores for matches, mismatches (i.e. substitutions) and opening as well as extension of gaps. The scores for matches and mismatches between the 20 standard amino acids are stored in substitution matrices. Most commonly either PAM or BLOSUM matrices are used. PAM matrices are based on the likelihood of each particular amino acid to be replaced by an other amino acid through point

accepted mutations in an evolutionary time span [20]. BLOSUM matrices use the log-odds score of substitution pairs in observed alignments [21]. Both the Needleman-Wunsch and the Smith-Waterman algorithms have a quadratic relationship between the length of the input sequences and the time and space they require. So it is impractical at best or actually impossible to compute pairwise alignments between a query and for example all 190 million proteins in the UniProtKB [4]. For this reason the first heuristic method FASTA [22, 23] was developed. It allows rapid identification of similar sequences in a protein databases and then provides exact scores for local alignments to these results. But for the size of modern database FASTA is already much too slow. Nowadays the de facto standard for this purpose is the suit of BLAST [24] applications. Given the required computational resources BLASTP allows the search of a whole proteome worth of proteins in the current TrEMBL release in roughly a day. But large proteomes, like for example Barley with almost 40 000 proteins, can quickly push this up to the scale of weeks. More modern alternatives like DIAMOND [25] offer further speedups of up to 20 000 times.

1.2.2 Protein Structure-Based Function Prediction

The structure determines a protein's function to a high degree [26], and its amino acid sequence in turn determines said structure. So in order to predict protein function from protein sequence it makes sense to predict the structure first. This can be done based on homology to known protein structures, by ab initio modeling or by threading [27].

There is a limit to the structural motifs in the tertiary structure of proteins and in general the structure of proteins is more conserved than their sequence. So many similar sequences can be linked to the same structural elements [28]. The Protein Data Bank (PDB) [29] stores known experimentally determined structures coupled with their sequence. Sequence similarity search can be used to find homologous template sequences which are associated to the three-dimensional structure tentatively fitting the query protein. Then, a model of the query protein is assembled from structures found for fragments of its sequence. Thereafter, this model can be refined based on a statistical potential or on physics-based energy calculations [30].

For proteins without any homologous protein structures in PDB threading or fold recognition can be applied [31, 32]. It uses statistics derived from the connection of structures in PDB to their sequences. The fit of each amino acid of the query is compared to positions in template structures. Then, the template with the best fit can be used to build a model of the structure.

Ab initio protein modeling can be applied to generate a protein's structure solely based on calculations of molecular dynamics. So no previously solved structures are needed and the model is built from scratch. A correctly folded protein is in a relatively low state of free energy. To model the free energy dependent on the protein structure an energy function must be used. Typically, such energy functions are either derived from physics of molecular interactions or based on statistical knowledge of native protein conformations. Then, the conformation space of the protein is explored to find a structure with

a low-energy conformation. The length of the amino acid sequence and thus the size of the protein determines the size of the conformation space. Finding a good low-energy conformation can quickly become restricted by the high amount of computational effort it requires. This can be mitigated by leveraging the assistance of machine learning algorithms. Even so, as of 2017 ab initio protein structure prediction is still limited to under 120 residues [33]. Distributed computing projects like Rosetta@home are used to push these limitations a bit further, for example in an effort to fight the COVID-19 pandemic [34].

1.2.3 Sequence Pattern-Based Function Prediction

Proteins are made up of functional units. These units, known as domains, are conserved and continuously recombined by evolution to form new proteins [35]. Domain families contain proteins with just those or more different domains [18]. Pfam [36] is a database of multiple sequence alignments and profile hidden Markov models of domain families. For each family a functional annotation, literature references and database links are available. So query protein sequences that have been aligned to entries in Pfam can be annotated with these functions.

The functional sites of proteins experience the highest selective pressure and are thus in general highly conserved. Consequently, common motifs can be found in their sequences which can be linked to their function. InterPro [11] is a collection of libraries with such motifs. It can be used to quickly detect known motifs in a query sequence.

1.2.4 Genomic Context-Based Function Prediction

The physical organization of their genes on the genome can provide indications for proteins' functions.

Genes repeatedly observed neighboring the query can indicate functional association. An example for this are operons, first described in 1960 [37]. They are groups of genes, roughly at the same locus, that are regulated in tandem and have functions targeted at a common objective. The organization, occurrence and regulation of genes in operons are conserved. So if a pair of genes is repeatedly found in close proximity in various species, a selective pressure to keep up the gene organization can be assumed. This permits the inference of protein function.

For some protein pairs homologs can be found in other organism that are fused into one larger protein. This is a strong indication for the presence of an interaction between the two proteins and thus also enables inference of their function. Because they decipher the interaction of unknown proteins, these kind of sequences have been coined Rosetta stone sequences [38].

Both genetic neighborhood and gene fusion rely on the proximity of the proteins' genes on the genome. If this is not the case, phylogenetic profiles can be of help. These consist of the presence or absence of orthologous proteins in various species. If proteins are often gained or lost together, i.e. have similar phylogentic profiles, they are likely to be functionally related [39].

Each genomic context-based protein function prediction method has its limits, strengths and weaknesses. So combining them typically yields the best results [40].

1.2.5 Proteomic Context-Based Function Prediction

In addition to the above mentioned genome-based methods, there are also experimental techniques available to extract protein interaction data from the proteome itself. Hawkins and Kihara [27] list mass spectrometry, 2D gel electrophoresis, yeast two-hybrid methods, protein chips and high-throughput protein-protein interaction screens in their 2007 review. The interactions of proteins in the cell can be represented as graphs or networks. Nodes are proteins, and edges are their interactions. Then, the "guild by association" principle can be used: Proteins close to each other are more likely to share their function than farther apart ones. In an early method a protein simply inherited the function that was most common in its direct neighborhood [41]. Later methods increased the predictive power by transferring functions statistically abundant in a broader neighborhood [42]. A more current example application uses the STRING database [43] to predict Biological Process and Cellular Component GO terms [44]. In general, protein-protein interaction network-based function prediction works only on the broader function (i.e. Biological Process of the Gene Ontology as opposed to Molecular Function). The only exception are proteins that interact within the same protein complex. Then, similarity of the Molecular Function can be assumed as well.

Gene expression data can be obtained by utilizing high throughput experimental techniques. Historically cDNA micro arrays [45] were used. Nowadays, RNA-Seq is typically used as it offers some distinct advantages: It is not limited to known sequences, provides details about alternative splicing, has a lower background signal, a larger dynamic range and higher accuracy for quantifying expression levels, better reproducibility and easier sample preparation [46]. The expression data is used to cluster genes that show similar expression patterns under certain conditions or along a time scale. A common function can be inferred for the products of these genes. So most methods that use expression data for function prediction focus on the identification of significant gene clusters and the enrichment of GO terms in them [27].

1.2.6 Current Challenges in Protein Function Prediction

Sometimes proteins perform their function only in specific tissues or under specific conditions (stress or developmental stages [47]). This can also affect protein-protein interactions [48]. So proteins with multiple functions might not exhibit all of them when the time comes to finally confirm them experimentally. This means that ways to define the differential functions of proteins become increasingly important.

Differing research interest, cost and feasibility constrains as well as ethical restraints have lead to deviating levels of completeness of the function annotations between various species [49]. Depending on the subontology, from 86% to

88% of the experimental GO annotations in the UniProtKB are from only 10 species [50]. Plants are especially affected because only one of these 10 species comes from their kingdom: *Arabidopsis thaliana*. In the future good functions need to be provided for non-model species as well.

Complete and unbiased GO annotations are rare. So they are subject to the "open world assumption", which means that the absence of a GO term does not guarantee the absence of a function [51, 52]. This leads to a systematic overestimation of the false positive rate when predictions are evaluated in accordance to an inherently sparse ground truth. This means that annotation methods optimized to reproduce annotations similar to such a ground truth will predict too few GO terms in many cases.

1.3 Evaluation of GO Annotations

The most basic comparison of GO annotations can be performed by simply considering the presence and absence of the GO terms in question. Only exact GO ID matches are considered and the structure of the GO is completely omitted. This method treats the comparison of GO annotations like a simple multi-label classification problem but in reality it is a *hierarchical* multi-label classification problem (HMC).

In contrast, semantic similarity-based approaches are able to quantify how similar two different GO terms are. Edge- (or path-) based approaches do this by calculating the number of edges (or nodes) on the shortest path between terms [53]. But most current methods derive the semantic similarity of GO terms from their information content. If the information content of GO terms is calculated using only the topology of the GO (e.g. [54, 55, 56]), no additional information is needed and calculated similarity values remain stable with respect to a particular version of the GO. However, when the information content of GO terms is derived from their frequencies in a corpus of annotated proteins [57, 58], the similarity values can be more nuanced but become dependent on said corpus. Recent similarity-based approaches often use a combination of the aforementioned methods [59].

2 Automated Assignment of Human Readable Descriptions (AHRD)

Under the guidance of Prof. Dr. Heiko Schoof, AHRD was originally developed by Girish Srinivas in 2009 [60]. At its core AHRD performs a lexical analysis of candidate descriptions retrieved by sequence similarity search methods like BLAST [24, 61]. This central workflow was written in the Perl programming language [62]. It was supplemented by GO terms acquired through an extended version of SIFTER [63, 64] and domain names retrieved with InterProScan [65].

In 2014 Asis Hallab and Kathrin Klee ported AHRD to Java [66] and made it version controlled in Git [67] as well as publicly available on GitHub (`https://github.com/groupschoof/AHRD`). Its code is organized in the "package by layer" principle and is developed in a test driven manner. The prediction of GO terms was moved to an R [68] workflow called PhyloFun (`https://github.com/groupschoof/PhyloFun`), which algorithmically propagates GO terms inside a phylogenetic tree [69]. AHRD has already been used in many genome projects and for annotations in some databases (see supplement 28.5 for details).

The main functionality of AHRD is still to use the descriptions of candidate proteins to predict annotations for unknown proteins. To do so, one must generate the main input AHRD needs first: the results of a similarity search of the query amino acid sequences in one or better several protein databases. The lexical analysis of the candidate descriptions AHRD performs works as follows: AHRD removes descriptions that contain indicators for a previous annotation transfer because the description should be as close to the primary source as possible. After that, the descriptions are deconstructed into their words, from here on referred to as tokens. Tokens known to be common to all kinds of protein descriptions and generally considered as uninformative are ignored. All others are scored by their abundance in descriptions from proteins with a high bit-score search result, a good alignment overlap and an origin in a trusted database. Then, the description candidates can be ranked according to their tokens and the top result can be used for annotation transfer to the query. Along with the human readable description GO terms can be transferred as well. Because of the Gene Ontology's structure, prediction of GO terms is a hierarchical multi-label classification (HMC) problem. This needs be taken in consideration when GO annotations are predicted as well as when predictions are evaluated in comparison to ground truth annotations. Both of which are central to the work presented in this thesis.

3 Aim

The overarching objective is to increase AHRD's ability to predict concise protein function annotations — i.e. its annotation performance. This immediately leads to the problem of quantification of said annotation performance. The cornerstone of evaluating protein annotations is a set of known, well-annotated proteins. AHRD's primary use case is the functional annotation of an organism's

proteome in the course of a genome project. There are well-annotated proteomes of model organisms but these have species-specific biases and AHRD is meant to be species agnostic. Protein databases have a wealth of well-annotated proteins but are also rife with redundant proteins, erroneous annotations and artificial biases towards proteins that are either easily examined or of relevance in better supported research fields. So the first major task is to find a way to generate an unbiased, non-redundant set of well-annotated ground truth proteins. These properties should hold true for the set's human readable description as well as GO annotations.

The next question that arises about annotation performance evaluation is how to exactly quantify the congruence of protein function annotations. Concretely, we need to find a nuanced and consistent scoring method appropriate for GO annotation predictions. First, the usefulness of a direct GO ID overlap-based score is to be determined. The second candidate that we want to examine facilitates the structure of the Gene Ontology by extension of the GO annotations to all of their parental terms. Furthermore, the added benefit of considering the pervasiveness of GO terms in a corpus of typical annotations, is to be investigated.

With a proper ground truth and a robust performance evaluation method the foundation for using machine learning is laid. We want to find out if this is a necessary step to achieve optimal protein annotation performance with AHRD.

In the past AHRD only examined the descriptions of proteins in order to rank reference proteins that are candidates for a transfer of the function to a query protein. An open question is whether a separate algorithm, which instead examines the GO annotations of these candidates, can be used to make better predictions for the GO terms of query proteins. The consideration of candidate GO annotations opens up a few new avenues to determine the value of possible annotations. Then, the homologs AHRD uses for annotation transfer are subject to different, new rankings, each of which needs to be examined to determine its usefulness.

Human readable descriptions cannot be mixed and matched without difficult natural language processing because of the complex interactions of words within them. In contrast each GO term is a concept in itself and defined in its entirety elsewhere (in the Gene Ontology). Thus, recombining multiple terms from different sources without disruption of meaning is possible. Accordingly we want to determine whether it is feasible to annotate query proteins with GO terms picked from multiple different reference proteins instead of transferring a fixed set from the top ranking homolog.

Lastly, the now optimized AHRD is to be compared to other programs commonly used for protein function prediction. There is the need to look at the quality as well as the quantity of the predicted descriptions and GO terms determined with our own evaluation methods. As a result of our participation in the CAFA 3 challenge, their evaluation method has been applied to AHRD's GO term predictions as well and is to be examined too.

Part II
Methods

4 Public Data and Software

4.1 Selection of Reference Databases as Annotation Source

The UniProtKB currently contains the sequences and annotations of over 190 million proteins [4]. These proteins are from species across all branches of the tree of life. It is split into Swiss-Prot for expert curated proteins and TrEMBL for uncurated proteins. These properties make the UniProtKB a good general annotation source for AHRD. Especially because AHRD can use different priorities for annotations retrieved from Swiss-Prot or from TrEMBL. If AHRD is used to annotate a proteome of a particular organism, a curated database from a related organism can be useful (e.g. TAIR [70] in case of a plant). But niche databases that contain useful information have often already been merged into Swiss-Prot anyway (as is the case for TAIR). And because the proteins in our ground truth test set (section 5) are also scattered across all groups of organisms, the two UniProtKB databases will be a good match.

Since our ground truth test set is sourced from Swiss-Prot, we created a custom version of it, which is missing all proteins found in the set. This makes sure that no self-matches will obfuscate the performance analysis of AHRD and its competitors. As we tested AHRD (section 7.6) using either the second or third version of our ground truth set (section 5), which are based on the UniProtKB's 10th release from 2016 and 9th release from 2018 respectively, we always used the appropriate UniProtKB version to remove the ground truth set from and use as reference search space.

4.2 Reference Proteomes

To show examples of AHRD's annotation performance in its typical use case — a proteome of a newly sequenced species — we compare the annotation fraction (coverage) of two established proteomes prior to and after AHRD's use for both descriptions and GO terms [8, 71]. First, we use the proteome of *Hordeum vulgare* (subsp. *vulgare*) commonly known as domesticated barley [72, 73]. And secondly we use the proteome of *Blumeria graminis* (f. sp. *hordei*) commonly known as barley powdery mildew [74]. In many cases annotations from these genome projects have made it into the two UniProtKB-Databases. In an effort to make the conditions more realistic we removed the proteins of all barley subspecies from the UniProtKB (Release 2019_09) to create the reference databases to use for AHRD in this test. Analogously, we removed the proteins of all subspecies to create the reference databases for the annotation of *Blumeria graminis*.

Different genome projects have different nomenclatures to signal annotation failure for their proteins. To catch all common cases we removed all descriptions with "undescribed protein", "unknown function", "unknown protein" or "uncharacterized protein" at the start. Thus, proteins with these descriptions are treated as not annotated.

Similarly we removed the three GO root terms (GO:0003674, GO:0008150 and GO:0005575) from the GO annotations so that proteins with only root terms were properly recognized as unannotated.

4.3 Other Protein Annotation Programs — AHRD's Competitors

4.3.1 Maximum Attainable

AHRD does not build compound annotations but always transfers the complete annotation of a reference protein from one of its target databases. But there is not always a perfect annotation for every query protein. Sometimes even the best choice will lead to an F-score smaller than one. But knowing the ground truth, we can always find the best possible choice AHRD could have made. Similarly the annotations to achieve a maximum possible precision or recall can be determined. The scores for this mock annotation method are reported as "Maximum Attainable".

4.3.2 "Best BLAST Swiss-Prot" (BBsprot) and "Best BLAST TrEMBL" (BBtrembl)

BLAST [24] reports the bit-score as a measure of the alignment quality of each hit in a target database. The hit with the highest bit-score is called the "Best BLAST" result. Transferring the "Best BLAST" result's annotation to the query protein is a straight forward yet, naive annotation method. We evaluated it alongside AHRD and its other competitors.

4.3.3 Blast2GO

Blast2GO [75] is a commercial desktop application utilizing similarity search results to annotate proteins with GO terms and descriptions. Activated with a free basic account key, it must be run interactively (batch mode is a pro feature). First, we merged Swiss-Prot (excluding the ground truth proteins) and TrEMBL into one big database and performed a BLAST search of our protein set on it. The BLAST results were imported into Blast2GO (version 5.2.5) specifying that up to 300 results should be considered per query with an HSP length cutoff of 10. It took roughly 10 hours to annotate our test set of 2244 proteins (section 5).

4.3.4 NetGO

NetGO [76] is an updated version of GOLabeler, an annotation method that topped many categories in the CAFA3 challenge [77]. NetGO uses offline (pre-computed) machine learning to integrate sequence and network information of reference proteins to annotate queries with GO terms in conjunction with a confidence score. The NetGO online service is limited to 1000 proteins at a time and takes 1 to 2 hours to complete annotation. Larger jobs either have to be

divided into chunks or submitted via email. We chose a confidence threshold of 0.5 to evaluate NetGO alongside AHRD.

4.3.5 EggNOGmapper

The EggNOGmapper [78] uses a database of precomputed clusters and phylogenies to assign orthology-based GO terms and descriptions to query proteins. We used EggNOGmapper (version: emapper-1.0.3-35-g63c274b; emapper DB: 2.0) with its default settings, online at `http://eggnog-mapper.embl.de/` on the 26th of September of 2019. It took roughly 50 minutes to annotate the 2244 proteins of our test set (section 5).

4.4 Programming Languages and Software Packages

- **AHRD — Automated Assignment of Human Readable Descriptions**
  Software that provides functional annotation for novel protein sequences in the form of human readable descriptions and GO terms. AHRD is the central topic of this thesis, written in Java, version controlled with git and freely available on GitHub (`https://github.com/groupschoof/AHRD`).

- **git**
  Git [67] is a distributed version control system. It is well-suited for multiple developers collaboratively working on multiple branches of a software project. After changes to the source code snapshots are saved as so-called commits, de facto versions of the software which are identifiable by unique (SHA-1) hash codes generated based on the commit's code.

- **Java 8**
  Java [66] is an object-oriented general purpose programming language. It is compiled to bytecode which then can be run on all architectures that have a java runtime environment (JRE) installed. This makes Java programs largely platform independent. AHRD previously only required Java 7 but since commit b6aca4d (introducing the owlapi) it requires Java 8.

  - **JUnit 4.9b2**
    JUnit [79] is a unit testing framework for Java. It allows us to develop AHRD in a test-driven manor.
  - **YamlBeans 1.06**
    YamlBeans [80] is a Java library that AHRD uses to load its settings from YML files.
  - **OWL API 4.5.4**
    The OWL API [81] is a Java library used by AHRD to parse a OWL formatted download of the Gene Ontology database.

- **Apache Ant**
  Ant [82] is a build automation software developed by the Apache Software Foundation. We use it to build AHRD's .jar file (i.e. compiling the Java source code to bytecode and packaging it into a single file).

- **Eclipse**
  Eclipse [83] is the IDE (integrated development environment) that was used to develop AHRD. Features of Eclipse often utilized include its object browser, version control with the git plugin *EGit*, a debugger and of course the Java editor. Among other things the Java editor facilitates syntax highlighting, code assist and code refactoring.

- **R 3.6.2**
  R [68] is an interpreted programming language focusing on statistical computing and graphics. It is freely available under the GNU General Public License.

  - **RStudio**
    RStudio [84] is an IDE for R.
  - **ggplot2 3.2.1**
    Ggplot2 [85] is an R package for data visualization.
  - **gridExtra 2.3**
    We used the gridExtra [86] R package to arrange multiple graphs made with ggplot2 in a single plot.

- **BLAST**
  The Basic Local Alignment Search Tool [24] is a program that can compare biological sequence data (amino-acid, DNA or RNA). Given a query, its primary use is the search of sequences with a high degree of similarity from large databases. BLAST uses a stochastic model to estimate the similarity of sequences. Thus an optimal alignment of the query and target sequence cannot be guaranteed but it is much more time efficient than the Smith-Waterman algorithm.

- **DIAMOND**
  DIAMOND [25] is a modern alternative to BLAST offering a significant reduction in computation time while maintaining a similar sensitivity.

5 Creating a Ground Truth Set With Low Functional Redundancy

To train and evaluate AHRD a set of proteins with well-known functions is needed — a ground truth set. Because UniProtKB/Swiss-Prot contains only curated proteins it is a good source in principle.

First, we extracted all proteins with at least one experimentally verified GO annotation. Experimentally verified GO annotations (expGOA) are annotations

with evidence codes listed under "Experimental Evidence Codes" and "High Throughput Experimental Evidence Codes" in supplemental section 28.4 as well as "TAS: Traceable Author Statement" and "IC: Inferred by Curator". This step is important to eliminate proteins whose functions are likely to have been transferred from other proteins. We also filtered out proteins with description lines matching any entry in AHRD's description line blacklist. In effect, this step uses the descriptions as another avenue to gather evidence indicative of a prior annotation transfer.

Additionally, all proteins were removed whose descriptions start with "Protein" followed by a single word. This single word is very often just a gene identifier useless for description of protein function and thus undesirable.

An ideal ground truth set for general function prediction contains an equal number of proteins for different protein families and different phylogentic taxa. But the composition of Swiss-Prot reflects trends in biological research instead of biological abundance of proteins and thus shows biases towards certain protein families and taxa. These biases can lead to an overspecialization of AHRD's training and skewed evaluation results, which both need to be avoided. We thus iterated over all proteins and kept each new protein only if all words in its description had not been seen in any of the previous descriptions. This removes redundancy from the set based on the descriptions. In this step it was necessary to use AHRD's token blacklist to ignore unspecific words commonly found in protein descriptions (such as "protein", "gene" or "family") or else to many otherwise dissimilar proteins would have been discarded.

In the next step, we clustered the remaining protein sequences with the CD-HIT algorithm [87] at a sequence identity threshold of 40%. Through keeping only one protein of each cluster we further lowered redundancy in the set based on sequence similarity.

The first version of this set was based on Swiss-Prot from the UniProtKB release 2016_10 and dubbed "nrSprotExpGOAv1" (3357 sequences). Because we found that proteins with useless descriptions like "Protein ABC" (where ABC stands for a gene ID) were enriched in our set, we added a filter for these cases and created the second version "nrSprotExpGOAv2" (2251 sequences). A third version "nrSprotExpGOAv3" (2244 sequences) was created by updating the source data to the UniProtKB release 2018_09.

6 AHRD's Capabilities Prior to This Work

When the function of a new amino acid sequence needs to be determined, the result of a sequence similarity search is often the starting point. AHRD's algorithm combines the search results from multiple databases and emulates the decision process of a human curator to determine a reference protein that is most likely to provide a good functional description and a GO annotation for a given query sequence (section 6.1). The emphasis AHRD places on numerous factors indicating the quality of a reference protein's annotations is controlled by a number of parameters. AHRD's annotation performance, as a function of a certain set of parameter values, can be evaluated by the comparison of its

predictions to ground truth annotations (section 5). We evaluate human readable descriptions based on word overlap (section 6.2). Then, the so computed evaluation score can be used as the objective function of a heuristic parameter optimization approach based on simulated annealing (section 6.3).

6.1 Annotation of Proteins With Human Readable Descriptions

AHRD's annotation procedure for human readable descriptions was originally implemented by Asis Hallab [69], Kathrin Klee and Girish Srinivas under the supervision of Heiko Schoof.

6.1.1 Description Annotation Workflow

Protein databases contain many proteins with functional descriptions transferred from other proteins. These transfer methods are often automatic and can create and propagate annotation errors. To weed out these cases AHRD first discards proteins with telltale signs of previous annotation transfers (words like "similar to", "probable", "putative", "predicted" and so forth found in the protein descriptions; see supplement 28.2.1).

Protein descriptions often contain additional database-specific information like the organism or a clone ID the protein belongs to (supplement 28.2.2). This information will be no longer relevant after description transfer and is thus removed by AHRD beforehand.

Words common to many protein descriptions, with little to no information conveyed (e.g. "protein", "gene", "family", "product"; see supplement 28.2.3), are ignored in the following scoring process but, in case of a top scoring target sequence, transferred to the query nonetheless.

The remaining tokens (character sequences with a collective meaning) are scored in accordance to three major criteria: Alignment quality of the query to the target sequence, confidence in the quality of the database and abundance in high scoring targets.

The target proteins are ranked according to their tokens while a normalization step keeps AHRD from preferring longer descriptions over shorter ones.

Together with the top scoring description GO terms can be transferred to the query protein. If the focus is on GO terms, only targets that also have GO terms associated with them will be considered.

An example of AHRD's decision process is shown in figure 1.

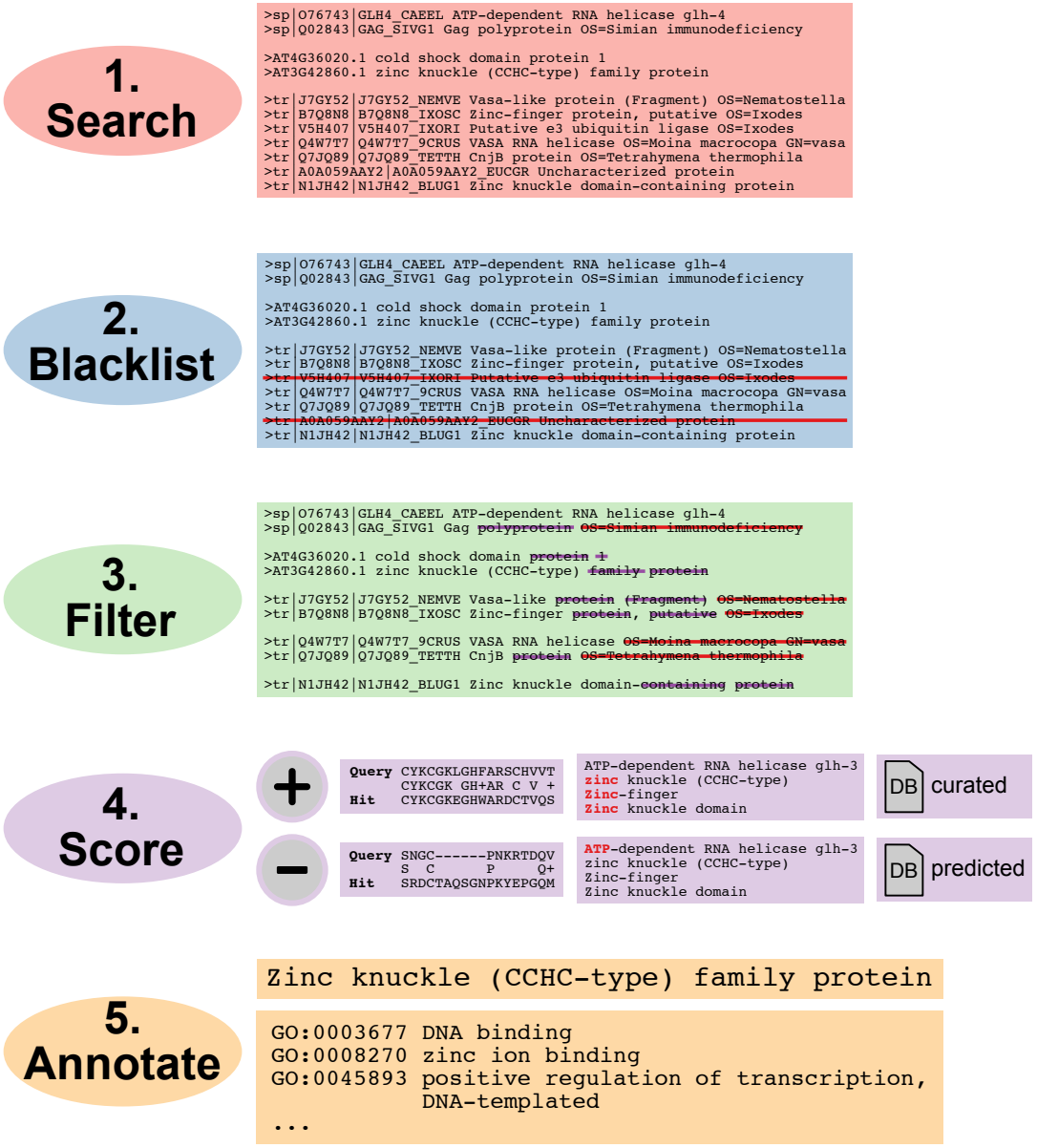
1. Search
>sp|O76743|GLH4_CAEEL ATP-dependent RNA helicase glh-4
>sp|Q02843|GAG_SIVG1 Gag polyprotein OS=Simian immunodeficiency
>AT4G36020.1 cold shock domain protein 1
>AT3G42860.1 zinc knuckle (CCHC-type) family protein
>tr|J7GY52|J7GY52_NEMVE Vasa-like protein (Fragment) OS=Nematostella
>tr|B7Q8N8|B7Q8N8_IXOSC Zinc-finger protein, putative OS=Ixodes
>tr|V5H407|V5H407_IXORI Putative e3 ubiquitin ligase OS=Ixodes
>tr|Q4W7T7|Q4W7T7_9CRUS VASA RNA helicase OS=Moina macrocopa GN=vasa
>tr|Q7JQ89|Q7JQ89_TETTH CnjB protein OS=Tetrahymena thermophila
>tr|A0A059AAY2|A0A059AAY2_EUCGR Uncharacterized protein
>tr|N1JH42|N1JH42_BLUG1 Zinc knuckle domain-containing protein
2. Blacklist
>sp|O76743|GLH4_CAEEL ATP-dependent RNA helicase glh-4
>sp|Q02843|GAG_SIVG1 Gag polyprotein OS=Simian immunodeficiency
>AT4G36020.1 cold shock domain protein 1
>AT3G42860.1 zinc knuckle (CCHC-type) family protein
>tr|J7GY52|J7GY52_NEMVE Vasa-like protein (Fragment) OS=Nematostella
>tr|B7Q8N8|B7Q8N8_IXOSC Zinc-finger protein, putative OS=Ixodes
>tr|V5H407|V5H407_IXORI Putative e3 ubiquitin ligase OS=Ixodes
>tr|Q4W7T7|Q4W7T7_9CRUS VASA RNA helicase OS=Moina macrocopa GN=vasa
>tr|Q7JQ89|Q7JQ89_TETTH CnjB protein OS=Tetrahymena thermophila
>tr|A0A059AAY2|A0A059AAY2_EUCGR Uncharacterized protein
>tr|N1JH42|N1JH42_BLUG1 Zinc knuckle domain-containing protein
3. Filter
>sp|O76743|GLH4_CAEEL ATP-dependent RNA helicase glh-4
>sp|Q02843|GAG_SIVG1 Gag polyprotein OS=Simian immunodeficiency
>AT4G36020.1 cold shock domain protein 1
>AT3G42860.1 zinc knuckle (CCHC-type) family protein
>tr|J7GY52|J7GY52_NEMVE Vasa-like protein (Fragment) OS=Nematostella
>tr|B7Q8N8|B7Q8N8_IXOSC Zinc-finger protein, putative OS=Ixodes
>tr|Q4W7T7|Q4W7T7_9CRUS VASA RNA helicase OS=Moina macrocopa GN=vasa
>tr|Q7JQ89|Q7JQ89_TETTH CnjB protein OS=Tetrahymena thermophila
>tr|N1JH42|N1JH42_BLUG1 Zinc knuckle domain-containing protein
4. Score
+
Query CYKCGKLGHFARSCHVVT
CYKCGK GH+AR C V +
Hit CYKCGKEGHWARDCTVQS
ATP-dependent RNA helicase glh-3
zinc knuckle (CCHC-type)
Zinc-finger
Zinc knuckle domain
DB curated
−
Query SNGC------PNKRTDQV
S C P Q+
Hit SRDCTAQSGNPKYEPGQM
ATP-dependent RNA helicase glh-3
zinc knuckle (CCHC-type)
Zinc-finger
Zinc knuckle domain
DB predicted
5. Annotate
Zinc knuckle (CCHC-type) family protein
GO:0003677 DNA binding
GO:0008270 zinc ion binding
GO:0045893 positive regulation of transcription, DNA-templated
...

Figure 1: AHRD's Central Workflow: The Decision Process to Find the Best Functional Description for a Query Protein

1. Search:
   Sequence similarity searches are carried out on chosen protein databases. The first two reference proteins are from Swiss-Prot, the section of the UniProtKB [4] which exclusively consists of curated entries. The next two proteins are from TAIR [70]. TAIR contains only *Arabidopsis thaliana* proteins but also offers a high level of curation. 13 proteins (7 shown) are found in TrEMBL. This is the uncurated but much larger section of the UniProtKB.

2. Blacklist:
   Reference proteins with descriptions matched in a blacklist are considered uninformative and are therefore discarded (crossed out in red).

3. Filter:
   Parts of the descriptions that are specific to the database are removed from them (crossed out in red). Unspecific words common to protein descriptions are kept but excluded from the following scoring step (crossed out in purple).

4. Score:
   Candidate proteins are scored in accordance to three major quality indicators. The first row shows cases that lead to a score increase (indicated by +) while the second row shows reasons for lowering the score (indicated by -). The sequence similarity, measured by the score and the overlap of the alignment between query and reference, is the first major factor to distinguish candidates (first column). The frequency of its words in all hits (indicated in red) is the next differentiation factor for description candidates (second column). The confidence generally put into the quality of annotations in the database is the third factor to discriminate candidate descriptions (third column).

5. Annotate:
   The reference protein with the top scoring candidate description is used as description and GO annotation source for the query protein.

Courtesy of Kathrin Klee and Heiko Schoof.

6.1.2 Scoring of Candidate Descriptions

Each query protein receives a set of description candidates from the sequence similarity searches on multiple reference protein databases.

Overlap score
For each description candidate an overlap score is calculated. This is the average fraction of the query and hit sequence that is covered by the local alignment.

$$o_i = \frac{(QueryEnd - QueryStart + 1) + (SubjectEnd - SubjectStart + 1)}{QueryLength + SubjectLength} \quad (1)$$

where:

$$\begin{aligned} i &:= \text{index over all description candidates} \\ Query &:= \text{query protein's amino acid sequence} \\ Subject &:= \text{found hit protein's sequence} \\ Start \text{ and } End &:= \text{the respective sequence position in the BLAST alignment} \\ Length &:= \text{the respective sequence length} \end{aligned}$$

Token score
The description candidates for all queries are split at every occurrence of a dash, a back or fourth slash, a semicolon, a comma, a colon sign, any quotation mark, a period sign, a pipe symbol, a parenthesis and of course white space. If it passes the token blacklist (supplement 28.2.3) each distinct token is assigned a token score. The token score is calculated as a linear combination of sequence similarity, database weight and sequence overlap. The sequence similarity is calculated as the sum of bit scores of all description candidates containing the token divided by the sum of bit scores of all description candidates. The database weight is calculated as the sum of database weights of all description candidates that contain the token divided by the sum of database weights of all description candidates. And the sequence overlap is calculated as the sum of overlap scores of all description candidates that contain the token divided by the sum of the overlap scores of all description candidates. The importance of each of these three terms can be re balanced by configuration of the three token score weights, which always have to sum up to one.

$$ts(t) = \beta \frac{\sum_k b_k}{\sum_i b_i} + \omega \frac{\sum_k w_k}{\sum_i w_i} + \sigma \frac{\sum_k o_k}{\sum_i o_i} \tag{2}$$

where:

$t \in T$
$t :=$ a token (word) in a description candidate
$T :=$ set of all destinct tokens of all description candidates
$i :=$ index over all description candidates
$k :=$ index over the description candidates that contain t
$b :=$ bit score of a description candidate's alignment
$w :=$ weight of the database a description candidate is from
$o :=$ overlap score of a description candiate's sequence with the query sequence
$\beta :=$ token score bit score weight (configurable)
$\omega :=$ token score database score weight (configurable)
$\sigma :=$ token score overlap score weight (configurable)
$\beta + \omega + \sigma = 1.0$

Adjusted token score
We use half of the maximum token score over all description candidates of a query as threshold to distinguish informative tokens from non-informative ones. Then, the threshold value is subtracted from the token scores of the non-informative tokens. This adjusted token score is calculated to further penalize undesirable tokens.

$$T_{ifr} = \left\{ t_l \mid ts(t_l) \geq \max_m ts(m) * itt \right\} \tag{3}$$

$$T_{non} = \left\{ t_l \mid ts(t_l) < \max_m ts(m) * itt \right\} \tag{4}$$

$$ts_{adjusted}(t_l) = \begin{cases} ts(t_l), & t_l \in T_{ifr} \\ ts(t_l) - \max_m ts(m) * itt, & t_l \in T_{non} \end{cases} \tag{5}$$

where:

$t_l \in d_i$
$l :=$ index over all tokens in a description candidate
$d_i :=$ set of tokens in a description candidate
$m \in T$
$itt :=$ informative token threshold ($= 0.5$)
$T_{ifr} :=$ set of informative tokens
$T_{non} :=$ set of non-informative tokens

In AHRD commit a5bccd1 the *itt* (informative token threshold) was made configurable between 0 and 1.

Lexical Score
All token scores of a description candidate are summed up to calculate its lexical score. To avoid a bias toward long descriptions the lexical score is corrected by the highest token score in all description candidates and the proportion of informative tokens in the description candidate in question.

$$ls(d_i) = \frac{\mid T_{ifr} \mid}{\mid T_{ifr} \mid + \mid T_{non} \mid} \cdot \frac{\sum_{t_l \in d_i} ts_{adjusted}(t_l)}{\max\limits_{m} ts(t_m)} \tag{6}$$

where:

$$\begin{aligned} \mid T_{ifr} \mid &:= \text{cardinality of the informative tokens of } d_i \\ \mid T_{non} \mid &:= \text{cardinality of the non-informative tokens of } d_i \end{aligned}$$

Description score
Finally, a description score can be assigned to each candidate. It is a combination of the lexical score and the blast score influenced by a configurable weight. The blast score is calculated as the fraction of the description candidates bit score of the maximum bit score of all hits for the particular query.

$$ds(d_i) = ls(d_i) + \delta \frac{b_i}{\max\limits_{n} b_n} \tag{7}$$

where:

$$\begin{aligned} \delta &:= \text{description score bit score weight (configurable)} \\ n &:= \text{index over all description candidates (like } i) \\ b &:= \text{bit score of a description candidate's alignment} \end{aligned}$$

The description scores are used to rank all description candidates and the query is annotated with the highest scoring candidate's description.

6.2 Evaluation of Protein Annotations: Word Overlap-Based Comparison of Human Readable Descriptions

For the evaluation, the ground truth description and the predicted description are split into tokens in the same way it is performed in preparation of the description candidate scoring (section 6.1.2). The tokens are also filtered by the token blacklist (supplement 28.2). By ignoring tokens common to all kinds of protein description we increase AHRD's ability to differentiate between wrong and correct descriptions. This is something that becomes especially important

when the evaluation of AHRD's predicted descriptions is used to optimize its parameters (sections 7.4 and 6.3).

The intersection of the ground truth and prediction constitutes the set of true positive tokens. The size of this set can be compared to the size of the prediction to calculate the precision and to the size of the ground truth to determine the recall.

$$precision_{HRD}(GT_{HRD}, Pred_{HRD}) = \frac{\mid GT_{HRD} \cap Pred_{HRD} \mid}{\mid Pred_{HRD} \mid} \tag{8}$$

$$recall_{HRD}(GT_{HRD}, Pred_{HRD}) = \frac{\mid GT_{HRD} \cap Pred_{HRD} \mid}{\mid GT_{HRD} \mid} \tag{9}$$

where:

$$\begin{aligned} precision_{HRD} &:= \text{precision of a human readable description} \\ recall_{HRD} &:= \text{recall of a human readable description} \\ GT_{HRD} &:= \text{set of ground truth tokens} \\ Pred_{HRD} &:= \text{set of predicted tokens} \\ \mid \cdot \mid &:= \text{cardinality of a set} \end{aligned}$$

The F_β-score is calculated as the weighted mean of precision and recall.

$$F_\beta(precision, recall) = (1 + \beta^2) \cdot \frac{precision \cdot recall}{\beta^2 \cdot precision + recall} \tag{10}$$

where:

$$\beta := \text{weighting parameter between precision and recall}$$

To give equal weight to precision and recall the β-parameter was always kept at 1, thus calculating the F_1-score, the *harmonic* mean of precision and recall. But AHRD supports other values as well. Modification to β can especially influence AHRD's training (sections 7.4 and 6.3) to yield parameter sets better suited to make precise predictions ($0 < \beta < 1$) or to make more comprehensive predictions ($\beta > 1$).

6.3 Parameter Optimization via Simulated Annealing

AHRD's parameter optimzation via simulated annealing was originally implemented by Asis Hallab and described in his PhD thesis [69].

Simulated annealing models the process of heating a material to reduce defects (i.e. decrease the energy in the system) while cooling it down slowly [88]. This probabilistic approach is meant to approximate the global optimum (i.e. a maximum evaluation score) in the optimization problem of finding the best set of parameters. To do so, the algorithm moves through the parameter space by trying one change to the parameters at a time. The choice which parameter to change is influenced by the so-called "hill climbing probability" p_h. It is the probability to modify the same parameter again after the last change was beneficial.

$$p_h := \frac{\exp(-(1-d)) + s}{\exp(0) + s} \tag{11}$$

where:

$d :=$ increase in mean F_β score achieved in the last step
$\exp() :=$ the exponential function
$s :=$ configurabe scaling factor (default: 0.7)
$p_h :=$ probability to modify the same parameter again

A parameter randomly chosen to be changed is equally likely to have its value increased or decreased (within allowed value margins). The magnitude m of the change is based on a standard normally distributed random value r ($\mu = 0$ and $\sigma = 1$) transformed by two configurable parameters for the standard deviation c_1 and the mean c_2.

$$m = r * c_1 + c_2 \tag{12}$$

where:

$r :=$ Gaussian distributed random value
$c_1 :=$ parameter for the standard deviation (default: 0.15)
$c_2 :=$ parameter for the mean (default: 0.25)
$m :=$ value to change a parameter by

A positive change in the resulting evaluation score always means the new parameters will be accepted. Otherwise both the amount of change and the temperature are used to calculate p_{acpt}: the probability of accepting parameters resulting in worse evaluation scores.

$$p_{acpt} = \exp\left(-(F_\beta(a) - F_\beta(c)) \cdot \frac{k}{T_c}\right) \tag{13}$$

where:

$F_\beta(x) :=$ mean F_β score of a parameter set x
$a :=$ accepted parameter set
$c :=$ currently evaluated parameter set
$T_c :=$ current temperature
$k :=$ scale parameter
$\exp() :=$ the exponential function
$p_{acpt} :=$ accepting probability

In order to avoid being trapped in local optima of the parameter space, the algorithm starts at a high temperature. As the temperature is decreased in small steps at each iteration, the probability of accepting big negative changes to the evaluation score decreases as well.

With the option "remember_simulated_annealing_path" set to TRUE, the evaluation of already encountered parameter sets can be omitted, thus saving compute time at the cost of additional memory use.

The procedure finishes when the temperature reaches 0. The best set of parameter values encountered during the optimization procedure is considered its result.

7 The Improvements Made to AHRD

Increasing the GO annotation performance of AHRD necessitates the ability to assess said annotation performance in the first place. This can simply be based on term overlap (section 7.2.1). But we also implemented more sophisticated GO evaluation metrics based on term ancestry overlap (section 7.2.2) and semantic similarity (section 7.2.3). These hinge on the availability of information about the hierarchical relationships of the terms in the Gene Ontology as well as information about the typical commonness of a given GO term in protein annotations (section 7.1). Using the description-based method to choose a candidate protein for the transfer of its annotations to the query (section 6.1) is an indirect way to perform GO term annotation, which can cost annotation performance. We thus implemented a new algorithm that works in a similar way but focuses on the GO annotations of the candidate proteins (section 7.3). In addition to the already available simulated annealing-based approach (section 6.3) we implemented another heuristic procedure to improve AHRD predictions by employing a genetic algorithm (section 7.4).

7.1 Parsing the Gene Ontology and Swiss-Prot to Build AHRD's Internal GO Term Database

AHRD needs to access its internal GO database when the annotation frequency (equation 15) is needed to calculate the information content score (equation 27) for a GO annotation candidate, when the ancestry of a GO term is needed to calculate the ancestry overlap-based F-score of a GO prediction (section 7.2.2) or when the maximum common information content of GO term pairs (equation 22) needs to be determined for the calculation of semantic similarity-based F-scores (section 7.2.3). In these instances a serialized object will be quickly loaded from disk if the GO database has been computed in a previous execution of AHRD. But if it is needed for the first time, AHRD has to build its GO database from scratch:

Parsing the Gene Ontology
First, the Gene Ontology is downloaded as a single OWL file and parsed using the owlapi java library. In this step each GO term is saved with its ID, label, namespace (MFO, PBO or CCO) and obsolescence status. Additionally, alternative GO IDs are linked to their primary instance and the ancestry of each term is determined.

Counting Swiss-Prot annotations
A file containing all of Swiss-Prot is retrieved from the UniProtKB FTP server and filtered for the GO annotation lines. Each annotation of a certain GO term found in Swiss-Prot is counted towards the term itself and each term in its ancestry. Consequently, the annotation count of a GO term $\eta(go)$ is the sum of its direct annotations $\mid Annot(gt) \mid$ and its indirect annotations through direct annotations of its child terms $\sum_{ct \in C(gt)} \mid Annot(ct) \mid$.

$$\eta(gt) = \begin{cases} \mid Annot(gt) \mid, & \text{if } gt \text{ is a leaf} \\ \mid Annot(gt) \mid + \sum_{ct \in C(gt)} \mid Annot(ct) \mid, & \text{otherwise} \end{cases} \tag{14}$$

where:

$$\begin{aligned} gt &:= \text{a GO term} \\ Annot(gt) &:= \text{set of all reference annotations that contain } gt \\ \mid \cdot \mid &:= \text{cardinality of a set} \\ ct &:= \text{a child GO term} \\ C(gt) &:= \text{set of child terms of GO term } gt \\ \eta(gt) &:= \text{annotation count of GO term } gt \end{aligned}$$

Annotation frequency
The frequency f of a GO term gt in Swiss-Prot can be calculated by dividing its annotation count $\eta(gt)$ with the annotation count of the root term $\eta(root(gt))$ of the ontology (MFO, BPO or CCO) it belongs to. This works because the root term is implicitly annotated with every annotation of any term in the particular ontology and thus has the sum of annotations for the ontology as its annotation count. Consequently, the root terms own annotation frequency turns out to be 1. GO terms that have no direct or indirect annotations in Swiss-Prot are marked with an annotation frequency of 0.

$$f(gt) = \frac{\eta(gt)}{\eta(root(gt))} \tag{15}$$

where:

$$\begin{aligned} gt &:= \text{a GO term} \\ \eta(gt) &:= \text{annotation count of GO term } gt \\ root(gt) &:= \text{root term of the subontology GO term } gt \text{ belongs to} \\ f(gt) &:= \text{the annotation frequency of GO term } gt \end{aligned}$$

Information content
The information content ic of a GO term gt is calculated by taking the negative natural logarithm of its annotation frequency $f(gt)$. This results in a value of 0 for the three root terms and infinity for GO terms without annotations in Swiss-Prot.

$$ic(gt) = -\ln(f(gt)) \tag{16}$$

where:

$$\begin{aligned} gt &:= \text{a GO term} \\ f(gt) &:= \text{the annotation frequency of GO term } gt \\ ic(gt) &:= \text{the information content of GO term } gt \end{aligned}$$

As of the 18th of February 2020 the lowest non-zero information content is assigned to the CCO term "cellular anatomical entity"(GO:0110165). It has been (in-/directly) annotated in Swiss-Prot 681 717 times. The number of annotations in the cellular component ontology is not much higher at 795 054. Its information content is thus $-\ln(681717/795054) \approx 0.15$
2 811 GO terms are annotated only once in Swiss-Prot. With 1 155 153 the BPO has the most annotations and thus its single annotation terms have the highest finite information content at $-\ln(1/1155153) \approx 13.96$.

7.2 Evaluation of Protein Annotations: Gene Ontology Annotation Evaluation

7.2.1 Term Overlap-Based Evaluation

The most simple way to compare predicted GO annotations to ground truth GO annotations is to determine the fraction of overlapping GO terms relative to the number of GO terms in the prediction and in the ground truth.

$$precision_{simpleGO}(GT_{GO}, Pred_{GO}) = \frac{\mid GT_{GO} \cap Pred_{GO} \mid}{\mid Pred_{GO} \mid} \tag{17}$$

$$recall_{simpleGO}(GT_{GO}, Pred_{GO}) = \frac{\mid GT_{GO} \cap Pred_{GO} \mid}{\mid GT_{GO} \mid} \tag{18}$$

where:

$$\begin{aligned} GT_{GO} &:= \text{set of ground truth GO terms} \\ Pred_{GO} &:= \text{set of predicted GO terms} \\ \mid \cdot \mid &:= \text{cardinality of a set} \\ precision_{simpleGO} &:= \text{"simple" precision of a GO annotation} \\ recall_{simpleGO} &:= \text{"simple" recall of a GO annotation} \end{aligned}$$

We call the F_β-score (equation 10) calculated from the $precision_{simpleGO}$ and $recall_{simpleGO}$ the "simple GO score".

7.2.2 Term Ancestry Overlap-Based Evaluation

The "simple GO term score" (section 7.2.1) only considers exact matches between a prediction and the ground truth. Two distinct GO terms with a very similar meaning are very close to each other in the DAG (directed acyclic graph) but still have completely different GO IDs. However, such a pair of GO terms will usually have a major part of their ancestry in common. So the mutual overlap of their two ancestries in relation to the distinct parts of their ancestries can be used as a measure for the similarity of two terms in the GO. Here we propose an F-score with this information included. After the sets of predicted and ground truth GO terms are expanded with their respective ancestries the precision and recall are calculated in the same way as the "simple GO score".

$$precision_{ancestryGO}(GT_{GO}, Pred_{GO}) = \frac{\mid A(GT_{GO}) \cap A(Pred_{GO}) \mid}{\mid A(Pred_{GO}) \mid} \tag{19}$$

$$recall_{ancestryGO}(GT_{GO}, Pred_{GO}) = \frac{\mid A(GT_{GO}) \cap A(Pred_{GO}) \mid}{\mid A(GT_{GO}) \mid} \tag{20}$$

where:

$$\begin{aligned} A(gt) &:= \text{set of ancestors of the GO term } gt \text{ (including } gt \text{ itself)} \\ GT_{GO} &:= \text{set of ground truth GO terms} \\ Pred_{GO} &:= \text{set of predicted GO terms} \\ \mid \cdot \mid &:= \text{cardinality of a set} \\ precision_{ancestryGO} &:= \text{ancestry-based precision of a GO annotation} \\ recall_{ancestryGO} &:= \text{ancestry-based recall of a GO annotation} \end{aligned}$$

We coined the resulting F_β-score (equation 10) the "ancestry GO score".

7.2.3 Term Semantic Similarity-Based Evaluation

If the prediction of a set of GO terms is evaluated against a set of ground truth GO terms, only by a simple check of the presence or absence of GO terms ("simple GO score" in section 7.2.1), most, if not all of the nuance in DAG of the Gene Ontology is missed. On the one hand, if a predicted GO term is close to a ground truth term in the tree, the prediction should not be considered to be completely wrong ("ancestry GO score" in section 7.2.2). On the other hand, the exact reproduction of a ground truth term can be insignificant if the term is very common (e.g. "protein binding" GO:0005515) or even meaningless if it is a root term. We thus propose the semantic similarity of GO term sets as an evaluation metric. It is based on the common information content of GO term pairs.

Implementation

The information content ic of a GO term gt is calculated as the negative natural logarithm of its annotation frequency f in Swiss-Prot (section 7.1 equations 15

and 16).

$$ic(gt) = -\ln\left(\frac{f(gt)}{f(root(gt))}\right) \tag{21}$$

where:

$gt :=$ a GO term
$f(gt) :=$ annotation frequency of a GO term
$root(gt) :=$ root term of the subuntology GO term gt belongs to
$ic(gt) :=$ information content of a GO term

Theoretically the information content a GO term that is not (yet) annotated in any protein in Swiss-Prot is infinite ($-ln(0) = \infty$). In practice we use the highest finite information content in its ancestry as a surrogate.

The highest information content of their mutual ancestry determines the maximum common information content $mcic$ of two GO terms gt_a and gt_b [57].

$$mcic(gt_a, gt_b) = max(ic(gt) : gt \in A(gt_a) \cap A(gt_b)) \tag{22}$$

where:

$gt :=$ a GO term
$A(gt) :=$ set of ancestors of the GO term gt (including gt itself)
$ic(gt) :=$ information content of a GO term
$mcic(gt, gt) :=$ maximum common information content of the two GO terms

The sum of the highest maximum common information content of each ground truth term and all prediction terms is divided by the sum of the information content of the ground truth term set to calculate a semantic similarity-based recall. The recall thus measures the fraction of the ground truth's information content that the prediction was able to reproduce.

$$recall_{semsimGO}(GT_{GO}, Pred_{GO}) = \frac{\sum\limits_{gt_a \in GT_{GO}} max(mcic(gt_a, gt_b) : gt_b \in Pred_{GO})}{\sum\limits_{gt_x \in GT_{GO}} ic(gt_x)} \tag{23}$$

where:

$gt :=$ a GO term
$GT_{GO} :=$ set of ground truth GO terms
$Pred_{GO} :=$ set of predicted GO terms
$ic(gt) :=$ information conent of a GO term
$mcic(gt, gt) :=$ maximum common information content of the two GO terms
$recall_{semsimGO} :=$ semantic similarity-based recall

The semantic similarity-based precision is calculated analogously. The sum of the highest maximum common information content of each term in the prediction set and all ground truth terms is divided by the sum of the information content of the predicted terms. It thus represents the fraction of the prediction's information content that was actually part of the ground truth.

$$precision_{semsimGO}(GT_{GO}, Pred_{GO}) = \frac{\sum_{gt_b \in Pred_{GO}} max(mcic(gt_b, gt_a) : gt_a \in GT_{GO})}{\sum_{gt_x \in Pred_{GO}} ic(gt_x)} \tag{24}$$

where:

$$\begin{aligned} gt &:= \text{a GO term} \\ GT_{GO} &:= \text{set of ground truth GO terms} \\ Pred_{GO} &:= \text{set of predicted GO terms} \\ ic(gt) &:= \text{information conent of a GO term} \\ mcic(gt, gt) &:= \text{maximum common information content of the two GO terms} \\ prcision_{semsimGO} &:= \text{semantic similarity-based precision} \end{aligned}$$

Finally, the F_β-score is calculated as the weighted mean of recall and precision (equation 10). We use the term "semantic similarity (SemSim) GO score" to refer to this evaluation metric.

Consequences

Because the UniProtKB/Swiss-Prot annotation counts (equation 14) are propagated up to the roots of each ontology (i.e. the root terms are always considered to be annotated indirectly), their annotation count ends up equal to the number of annotations in the ontology over all. A root terms annotation frequency (equation 15) is thus calculated to be 1. And therefore, their information content (equation 16) turns out to be 0, just like intuition suggests.

If two terms from the same ontology are so dissimilar that their shared ancestry consists of only the root term, their maximum common information content is consequently 0. In such a case the semantic similarity-based recall (the maximum common information content divided by the information content of the ground truth) is also 0. Similarly the semantic similarity-based precision (the maximum common information content divided by the information content of the prediction) is determined to be 0 as well in such cases.

If the predicted term has the ground truth term in its ancestry, their maximum common information content is the information content of the ground truth term. Thus, the semantic similarity-based recall is 1 in such a case while the precision will be between 0 and 1 and reflect a penalty for the over-specific prediction.

Analogously, if the ground truth term has the predicted term in its ancestry, their maximum common information content is the information content of the predicted term. Consequently, in such a case the semantic similarity-based precision will be calculated to be 1, but the recall ends up between 0 and 1 reflecting the missing information content in the prediction.

Edge cases
Even in such a ground truth set as our own "nrSprotExpGOAv3.fasta" (section 5), consisting exclusively of curated proteins with at least one experimentally verified GO annotation each, sometimes no information is available for one of the three subontologies. In theory this means all predictions in this subontology are over-specific and are thus penalized with a lower precision score. But if there is no knowledge about a particular aspect of a protein's function, the prediction can also not be known to be wrong. After all, all proteins must have at least one molecular function, must participate in at least one biological process and must appear in some cellular component. To take this into account, predictions made in a subontology for which no prior knowledge exists (i.e. the ground ground truth is empty) are ignored by the scoring algorithm.

Originally in the absence of any annotation in the prediction or ground truth, both the recall and precision were just set to 0. Consequently, the portion of proteins annotated — an annotation method's coverage — had a strong influence on the average F_β-score. We decided that separating these two quality measures can give a greater insight in the different strengths and weaknesses of various annotation methods. Consequently, from AHRD's commit 1c316ea onward we set the recall and precision both to NaN (not a number) in case a ground truth protein is entered in the evaluation without any annotations at all. If no prediction was made the recall is set to 0 and the precision is set to NaN. Both of these cases also result in a NaN value for the "semantic similarity F_β-score". It is important to be able to distinguish these cases from bad predictions that simply have so little in common with the ground truth that the resulting score is 0. Then, NaN values are excluded when calculating average F_β-scores. So instead they have influence on the fraction of proteins with a non-NaN F_β-score, which can additionally be provided as "coverage" in the output.

7.3 Annotation of Proteins With Gene Ontology Terms

Previous versions of AHRD transferred GO annotations from reference proteins scored solely on the characteristic of their human readable descriptions. To increase AHRD's GO term annotation performance we implemented a candidate protein scoring procedure based directly on GO annotations. But for the prediction of GO terms it is just as important to avoid electronically transferred protein annotations as it is for the description prediction. We thus subject the candidate reference proteins for the annotation with GO terms to the same filtering steps performed on the description candidates (section 6.1.2). So AHRD's GO term annotation procedure benefits from quality indicators derived from human readable descriptions.

7.3.1 AHRD's Scoring of Candidate GO Term Annotations

GO term score
For each unique GO term *gt* of all candidate annotations a GO term score *gts* is calculated. Among other factors we also tested different versions of the GO term score (section 7.6.1). Here only the full fledged version incorporating both the information content score *ics* (equation 27) and the evidence code score *ecs* (equation 28) in addition to the GO term abundance *gtas* (equation 26) is shown.

$$gts(gt) = gtas(gt) \cdot ics(gt) \cdot ecs(gt) \tag{25}$$

where:

$$\begin{aligned} gt &:= \text{a GO term} \\ gtas(gt) &:= \text{the GO term abundancy score of a GO term} \\ ics(gt) &:= \text{the information content score of a GO term} \\ ecs(gt) &:= \text{the evidence code score of a GO term} \\ gts(gt) &:= \text{GO term score of a GO term} \end{aligned}$$

GO term abundancy score
The GO term abundancy score *gtas* is analogous to the token score for the scoring of candidate description tokens (section 6.1.2). It is a linear combination of sequence similarity, database weight and sequence overlap. The sequence similarity is calculated as the sum of bit scores of all annotation candidates containing the GO term ($\sum_k b_k$) divided by the sum of bit scores of all annotation candidates ($\sum_i b_i$). The database weight is calculated as the sum of database weights of all annotation candidates that contain the GO term ($\sum_k w_k$) divided by the sum of database weights of all annotation candidates ($\sum_i w_i$). And the sequence overlap is calculated as the sum of overlap scores of all annotation candidates that contain the GO term ($\sum_i o_k$) divided by the sum of the overlap scores of all annotation candidates ($\sum_i o_i$). The importance of each of these three terms can be rebalanced by configuration of the three token score weights β, ω and σ, which always have to sum up to one. The weights' names are recycled from the calculation of the token scores for the description annotation.

$$gtas(gt) = \beta \frac{\sum_k b_k}{\sum_i b_i} + \omega \frac{\sum_k w_k}{\sum_i w_i} + \sigma \frac{\sum_k o_k}{\sum_i o_i} \tag{26}$$

where:

$gt \in GT$
$gt :=$ a GO term
$GT :=$ set of all distinct GO terms of all candidate annotations
$i :=$ index over all candidate annotations
$k :=$ index over the candidate annotations that contain gt
$b :=$ bit score of a candidate's sequence alignment
$w :=$ weight of the database a candidate annotation is from
$o :=$ overlap score of a candiate's sequence with the query sequence
$\beta :=$ token score bit score weight (configurable between 0 and 1)
$\omega :=$ token score database score weight (configurable between 0 and 1)
$\sigma :=$ token score overlap score weight (configurable between 0 and 1)
$\beta + \omega + \sigma = 1.0$

GO term information content score
The information content score *ics* leverages a different source than the sequence similarity search results as a quality measure of a candidate GO term: Its expected occurrence in the annotation of a typical protein in Swiss-Prot (section 7.1), i.e. its annotation frequency f. The impact of the information contents score on the GO term score (gts) can be configured by changing the GO term information content weight (ι).

$$ics(gt) = 1 - \iota \cdot f(gt) \tag{27}$$

where:

$gt :=$ a GO term
$ics(gt) :=$ the information content score of a GO term
$\iota :=$ GO term information content weight (configurable between 0 and 1)
$f(gt) :=$ the annotation frequency of a GO term in Swiss-Prot

GO term evidence code score
Consider a GO annotation $p \xrightarrow{ec} gt$. For a given GO term gt there is a set of candidate proteins with annotations $Annot(gt)$ for the GO term. For each annotation $a \in Annot$ an evidence code ec is provided to give an indication how the association to the GO term is supported. Using configurable weights for each evidence code an average can be computed over all annotations of a GO term. Augmented by another weighting parameter, this average is used to determine the evidence code score ecs.

$$ecs(gt) = 1 - \epsilon \cdot \left(1 - \frac{\sum_{a \in Annot(gt)} w(ec(a))}{\mid Annot(gt) \mid} \right) \tag{28}$$

where:

$$\begin{aligned}
gt &:= \text{a GO term} \\
a &:= p \xrightarrow{ec} gt := \text{a GO annotation} \\
a &\in Annot(gt) \\
Annot(gt) &:= \text{set of all candidate annotations that contain } gt \\
\mid \cdot \mid &:= \text{cardinality of a set} \\
ec(a) &:= \text{the evidence code of an annotation} \\
w(ec) &:= \text{the weight of an evidence code (configurable between 0 and 1)} \\
\epsilon &:= \text{GO term evidence code score weight (configurable between 0 and 1)} \\
ecs(gt) &:= \text{the evidence code score of a GO term}
\end{aligned}$$

If not specified, the evidence code weights default to a value of 1 for all evidence codes encountered. Thus, the influence of the evidence code score is negated and AHRD behaves in an evidence code agnostic manner. The evidence code weights we used in our experiments are listed in supplement 28.4

Adjusted GO term score
The informative token threshold itt — an adjustable parameter — is used to distinguish informative GO terms GT_{ifr} from non-informative GO terms GT_{non} by comparison of their GO term scores gts to the maximum GO term score. The adjusted GO term score $gts_{adjusted}$ is calculated by subtraction of the threshold value from all non-informative GO term scores. This further penalizes unwanted GO terms and aims at increasing the discriminatory power of the GO term score.

$$GT_{ifr} = \left\{ gt_l \mid gts(gt_l) \geq \max_{m \in GT} gts(m) \cdot itt \right\} \tag{29}$$

$$GT_{non} = \left\{ gt_l \mid gts(gt_l) < \max_{m \in GT} gts(m) \cdot itt \right\} \tag{30}$$

$$gts_{adjusted}(gt_l) = \begin{cases} gts(gt_l), & gt_l \in GT_{ifr} \\ gts(gt_l) - \max_{m \in GT} gts(m) \cdot itt, & gt_l \in GT_{non} \end{cases} \tag{31}$$

where:

$gt :=$ a GO term
$l :=$ index over all distinct GO terms in all candidate annotations
$gts(gt) :=$ the GO term score of a GO term
$GT :=$ set of all distinct GO terms in all candidate annotations
$itt :=$ informative token threshold (configurable between 0 and 1)
$GT_{ifr} :=$ set of informative GO terms
$GT_{non} :=$ set of non-informative GO term
$gts_{adjusted}(gt) :=$ the adjusted GO term score of a GO term

Lexical GO annotation score

For each GO annotation candidate D_i a lexical score $lgas$ is calculated analogously to the lexical score for description candidates (equation 6): The sum of the adjusted GO term scores $gts_{adjusted}$ of all GO terms gt of the GO annotation D_i is divided by the highest GO term score of all annotation candidates. Many uninformative GO terms (with negative adjusted GO term scores) will already result in a low lexical score, but to mitigate a bias towards long annotations even further the fraction of informative tokens in all tokens of the GO annotation candidates is used to scale the final score value.

$$lgas(D_i) = \frac{\mid GT_{ifr} \cap D_i \mid}{\mid D_i \mid} \cdot \frac{\sum_{l \in D_i} gts_{adjusted}(l)}{\max_{m \in GT} gts(m)} \tag{32}$$

where:

$D_i :=$ set of GO terms that is the GO annotation of one protein
$\mid GT_{ifr} \cap D_i \mid :=$ cardinality of the informative GO terms of D_i
$\mid GT_{non} \cap D_i \mid :=$ cardinality of the non-informative GO terms of D_i
$\mid D_i \mid = \mid GT_{ifr} \cap D_i \mid + \mid GT_{non} \cap D_i \mid :=$ cardinality of D_i
$gts_{adjusted}(gt) :=$ adjusted GO term score of a GO term
$GT :=$ set of all distinct GO terms in all candidate annotations
$lgas(D) :=$ lexical GO annotation score of a GO term set

GO annotation score

Finally, the GO annotation score *gas* for a GO annotation candidate D_i is calculated as the sum of its lexical GO annotation score *lgas* and its relative bit score. The relative bit score of an annotation candidate is the fraction of its bit score b_i and the maximum bit score of all annotation candidates weighted by a configurable parameter δ.

$$gas(D_i) = lgas(D_i) + \delta \frac{b_i}{\max\limits_n b_n} \tag{33}$$

where:

D_i := set of GO terms that is the GO annotation of a candidate protein
$lgas(D)$:= lexical GO annotation score of a GO term set
δ := description score bit score weight (configurable between 0 and 1)
n := index over all candidate proteins (like i)
b := bit score of a candidate protein's alignment
$gas(D)$:= GO annotation score of the GO annotation of a candidate protein

In the end, the highest scoring candidates GO annotation is transferred to the query protein.

7.3.2 Annotation of GO Slim Terms

GO slims are versions of the Gene Ontology that have been pruned to contain only broad terms. They are used to give an overview of a body of more detailed annotations. GO slims can be custom made but the Gene Ontology consortium also maintains standard sets and provides them in the OBO format.

AHRD can output GO slim terms in addition to its normal GO annotations. This is triggered by providing an OBO-formatted file of the GO slim using the "go_slim" YML-key. Instead of properly parsing the OBO-file AHRD simply matches all lines that start with "id: GO:" and have nothing but seven digits afterwards. Besides being crude but effective this "parsing" method enables the input of custom GO slims with files that simply provide the GO terms in lines that match this pattern.

After the normal GO annotation is completed, AHRD searches for the GO slim terms in the ancestry of each protein's annotations. GO slims can have terms whose sets of descendants overlap. If this leads to a case of multiple GO slim terms for one original (detailed) GO term, only the GO slim term with the highest information content (section 7.1 equation 16) will be annotated.

7.4 Parameter Optimization With a Genetic Algorithm

Given ground truth annotations, AHRD can calculate an evaluation score based on a particular set of parameters. The evaluation score is calculated by taking the mean F_β-score over all query proteins in AHRD's prediction run. If AHRD is only supplied with the data to annotate the queries with human readable descriptions, the evaluation score will be based on the overlap of words in ground truth and prediction. But if an annotation of GO terms was performed the evaluation score is based on the semantic similarity (section 7.2.3) of ground truth GO terms and predicted GO terms. We use this evaluation score as objective function in both a simulated annealing approach (section 6.3) and a genetic algorithm to optimize AHRD's parameters.

Our genetic algorithm [89] works by revising a "population" of parameter sets for a certain number of "generations". It uses the biology inspired processes of selection, recombination and mutation to arrive at a high performing set of parameters. The first generation can be seeded by parameters known to perform well but we use a naive set instead. The rest of the first generation is filled with randomly generated parameters. After an evaluation has been performed based on each parameter set, the population is ranked in accordance to the parameter sets' scores. This ranking is used to create the next generation of parameter sets. The best 20% survive and are transferred as is. To form an offspring the parameter sets of two random survivors are combined randomly so that on average half of the parameter values come from the first parent and the other half come from the second parent. These make up the next 20% of the new generation. Another 20% is filled with mutants of random survivors. A mutant is a parameter set with one parameter value changed. To create a mutated parameter set, the java method for parameter modification from the simulated annealing trainer (implemented by Asis Hallab [69]) was reused. The random selection of survivors for the creation of both offspring and mutants is based on their ranking and has a strong bias towards the very best performing parameter sets (equation 34). The rest (40%) of the new generation is filled with new randomly generated sets of parameters. After the last generation is evaluated, the top ranking parameter set is considered the result of the training procedure. In figure 2 the resulting probabilities for 20 parameter sets are plotted against the rank.

$$R = \lceil abs(r * n/3) \rceil \tag{34}$$

where:

$$\begin{aligned} r &:= \text{Gaussian distributed random value } (\mu = 0 \text{ and } \sigma = 1) \\ n &:= \text{size of the survivor set} \\ abs() &:= \text{absolute value of a number} \\ \lceil \cdot \rceil &:= \text{ceiling function} \\ R &:= \text{Selected Rank } \{R \in \mathbb{N} : 1 \leq R_{select} \leq n\} \end{aligned}$$

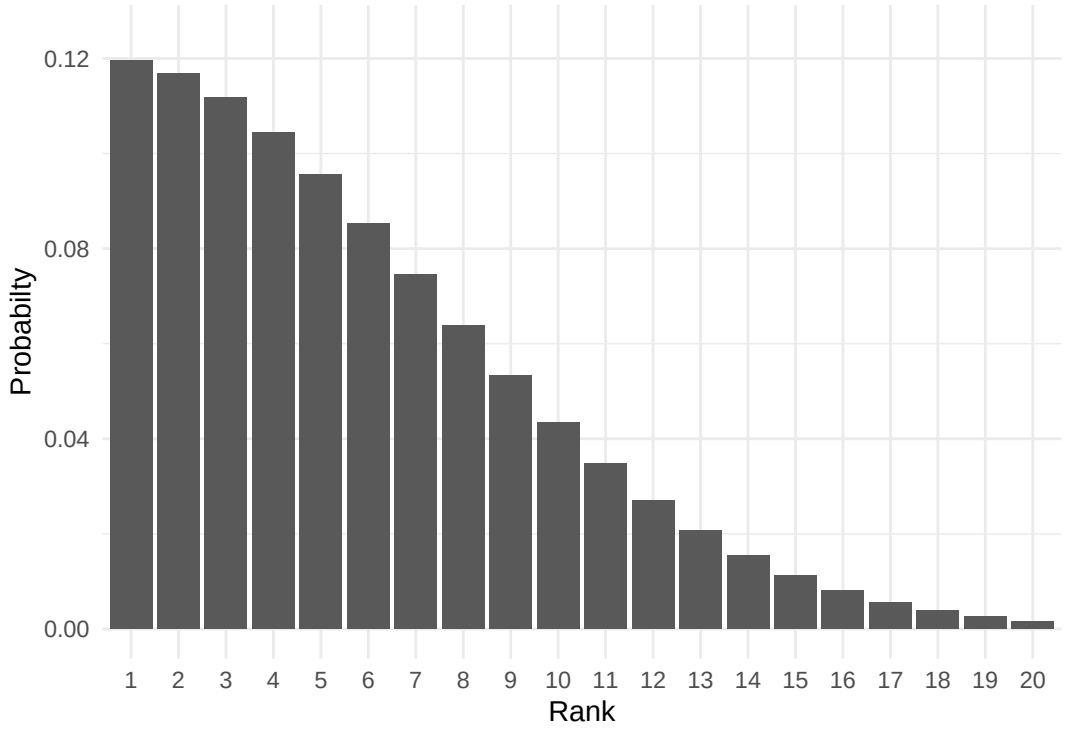

Figure 2: Rank-Dependent Parameter Set Selection Probability
The genetic algorithm selects the best performing parameter sets to survive to the next generation. The survivors are the basis for the generation of new parameter sets by crossover and mutation. Which particular survivors are used is determined by random selection based on their function prediction performance ranking (equation 34). Here we show the resulting probability for a rank to be selected in a typical example of a survivor set size of 20 (population size of 100 parameter sets).

7.5 Prediction and Evaluation of Term-Centered GO Annotations

In certain scenarios there can be value to taking the typical research question of "What are the functions of a certain protein?" and turning it around to ask "What are the proteins with a certain function?". In the context of GO terms this translates to asking "Which proteins are to be annotated with a particular GO term?" instead of "With which GO terms is a single protein to be annotated?". This question is usually brought up in the context of an organism's whole proteome and a few selected GO terms, just like in the CAFA-π challenge (section 8.2). To participate, we created a version of AHRD that can predict and evaluate the associations of a proteome's proteins to one or more GO terms.

Prediction
In the CAFA-π challenge predictions were evaluated with a sliding annotation confidence threshold. Therefore, we based the term-centric version of AHRD on the separate GO term annotation version of AHRD (section 7.3.1), because it calculates GO term scores (equation 25), which can be used to quantify the confidence in GO terms separately. First, for each protein, all reference annotations from the sequence similarity search results are stored alongside their GO term scores. If the GO term of interest is found in the ancestry of the annotations for a protein, the protein is considered to be associated with the term. The GO term score of the annotation that has the GO term of interest in its ancestry is used as the confidence score for the association to the protein. Of course the GO term of interest can have multiple children (or grandchildren and so forth) in the annotations that are found for the protein. In these cases the maximum of the GO terms scores is used. For a given GO term all proteins are investigated in this manner. The confidence scores are scaled by their maximum value over all proteins to use the full "space" between 0 and 1.

Evaluation
Using the confidence score as a discrimination factor in the binary classification question of GO term association, a receiver operator curve can be generated for a parameter set of AHRD. So the evaluation procedure iterates over 101 small confidence score threshold steps from 0 to 1. At each threshold value the proteins with greater confidence scores are considered to be predicted. If the GO term of interest also appears in the ancestry of the ground truth GO annotations of a protein, the protein is considered a true positive and otherwise a false positive. A protein that has the GO term of interest in its ancestry but has not been associated with it, is considered a false negative. Thus, for each threshold value the precision, recall and F_β-score can be calculated. The maximum F_β-score of all confidence score threshold values is used as performance metric for the current parameter set. AHRD can perform the term-centric annotation for multiple GO terms at the same time. In this case the F_β-score is averaged over the GO terms.

Training
Parameter optimization based on GO term-centric annotations can be performed

using the GO term-centric F_β-score (see above) as objective function. Both simulated annealing (section 6.3) and the genetic algorithm (section 7.4) can be used.

7.6 Testing AHRD

Through the course of developing AHRD, different ideas to improve its annotation performance have been pursued. Naturally, a central point in this endeavor was the ability to quantify the impact of these ideas. The most important component of any prediction evaluation is a good ground truth set. Our procedure to extract a set of proteins from UniProtKB/Swiss-Prot with experimentally verified GO annotations and a low sequence redundancy is described in section 5. The UniProtKB was also used as sequence similarity search space and annotation reference. So because the ground truth proteins were part of the search space, self-matches needed to be avoided. Usually some proteins from the ground truth might appear in the search results of other ground truth proteins and consequently the self-matches should be filtered on a case by case basis. But because our ground truth set has a very low sequence redundancy, it is be very unlikely to find ground truth proteins in each other's search results. Thereby, it was possible to avoid self-matching at an earlier stage of the work flow by simply filtering the complete set of ground truth proteins from Swiss-Prot.

AHRD's parameter optimization procedures described in sections 6.3 and 7.4 are designed to heuristically find an optimum in the parameter space that performs as closely as possible to the global optimum. Each time we introduce a significant change to AHRD's annotation prediction algorithm the parameter space changes too. It was thus necessary to retrain and reevaluate AHRD each time changes were made to assess the impact of these changes.

For the evaluation a set of 1000 proteins from our ground truth set were taken aside. The remaining proteins (e.g. 1244 from nrSprotExpGOAv3) were used for the training. This split of the ground truth set was performed at random and repeated 10 times. This 10 times splitting procedure was performed only once. Thus, all subsequent analysis steps use the same 10 training and test sets. This facilitates a better comparison between AHRD versions as the variation between splits can be excluded as a contributing factor for differences between the mean evaluation scores.

All parameter optimization was performed with the genetic algorithm we implemented for AHRD (section 7.4). Although we generally observed satisfactory training results after 5000 parameter set evaluations (e.g. 50 generations of 100 parameter sets), we chose to perform 100 generations at a population size of 200 to be sure. Depending on the complexity (only descriptions or also GO annotations), our training took between 4 and 8 hours on a single core of a server grade CPU. After that, the resulting parameter set was used for the function prediction of the remaining 1000 proteins, which were evaluated in light of their ground truth annotations. The mean value of the 1000 so obtained evaluation scores was calculated to express the annotation performance in a single number

for each ground truth set split.

As mentioned in section 7.2.3, in AHRD's git commit 1c316ea we implemented the separate evaluation of quality and coverage of a method's annotations. That means the average F_β-score is only calculated from proteins that have annotations predicted. And in addition, the coverage, i.e. the number of annotated proteins divided by the size of the protein test set, can be obtained.

7.6.1 Evaluate Different GO Prediction Approaches for AHRD

We used UniProtKB version 2016_10 (both Swiss-Prot and TrEMBL) as the search space for reference annotations. The protein set that was used is the second version of our ground truth set (section 5). It was extracted from Swiss-Prot (UniProtKB 2016_10) and called nrSprotExpGOAv2.

Variation in training
Because the training procedure makes liberal use of pseudo-randomly generated numbers the outcome, i.e. the mean evaluation score of the protein test set, can vary for repeated executions of the same training. Therefore, it was necessary to assess the magnitude of the variation in the resulting mean score based solely on the variation of the training. To do so we used AHRD commit 87166d5, which is functionally identical to 24a82dc (see below), as an example to repeat the training and evaluation procedure for every of the 10 ground truth set splits 1000 times, and we recorded the resulting evaluation scores.

Variation in splitting
As previously mentioned, to evaluate the influence of a change in AHRD's GO prediction algorithm, the ground truth set was split 10 times into training and test set. Because the evaluation scores for the proteins can vary wildly from each other, the exact composition of the protein splits can also influence the mean evaluation score of the test set. It was thus also necessary to quantify the variation in the mean evaluation score based on protein set splitting. To do so, we pooled the evaluation scores of all proteins in the 10 test sets after one round of training and evaluation with example AHRD commit 87166d5. Samples of the same size as the test set (1000 proteins) were randomly (without replacement) drawn from the pool 1000 times. The mean evaluation score was calculated each time. When proteins where inevitably found in more than one test set, the evaluation score for the particular protein was chosen at random from one of the sets.

Pairwise one-sided t-tests of evaluation scores from different AHRD versions
The performance of AHRD's different prediction algorithms can be assessed by comparing their ground truth protein reannotation-based evaluation scores. The validity of hypotheses regarding the in- or decrease of these scores must be ascertained with statistical testing. Figure 5 in section 11 demonstrates that the difference of mean evaluations scores can be expected to be much higher between ground truth set splits than what is caused by the random nature of AHRD parameter optimization. We observed that the variation between the

ground truth set splits is also higher than the difference between various prediction algorithms. Therefore we chose to perform the hypothesis testing in a pairwise fashion on a per protein basis. The evaluation scores (F_1, precision and recall) are limited to the interval between 0 and 1. Furthermore, in many cases the scores will be the same between different prediction algorithms, which leads to a high number of difference values equal to 0. Consequently, the distribution of pairwise differences can not be expected to exhibit normality. Non-parametric tests could be used but their increased robustness comes at a cost to statistical power. Thanks to the central limit theorem, parametric tests comparing the mean of an outcome variable, such as the t-test, are robust to departures from the normality assumption if sample sizes are large enough [90]. With 10 000 score pairs (1000 proteins $\times$ 10 splits) per test, this requirement is easily met in our case.

AHRD's GO prediction approaches

- **Conventional description-based:**
  Feature code: `- - - -`
  Commit: 24a82dc
  The description score (equation 7) is used to choose the candidate protein, whose GO annotations are transferred to the query.

  - The adjustment of the token scores (equations 3 – 5) is performed with an informative token threshold fixed to 0.5.
  - No GO term score is calculated so the GO term abundance cannot be considered.
  - No GO term score is calculated so the information content of the GO terms cannot be considered.
  - No GO term score is calculated so the evidence codes of the candidate GO annotations cannot be considered.

- **Description-based with variable informative token threshold:**
  Feature code: `itt - - -`
  Commit: dab2244
  The description score (equation 7) is used to choose the candidate protein, whose GO annotations are transferred to the query.

  `itt` The adjustment of the token scores (equations 3 – 5) is performed with a variable informative token threshold (`itt`).
  - No GO term score is calculated so the GO term abundance cannot be considered.
  - No GO term score is calculated so the information content of the GO terms cannot be considered.
  - No GO term score is calculated so the evidence codes of the candidate GO annotations cannot be considered.

- **Basic GO-based:**
  Feature code: `- gtas - -`
  Commit: b8d95c6
  The GO annotation score (equation 33) is used to choose the candidate protein, whose GO annotations are transferred to the query.

  `-` The adjustment of the GO term scores (equations 29 – 31) is performed with an informative token threshold fixed to 0.5.

  `gtas` The GO term score (equation 25) incorporates the abundance of GO terms among the candidate proteins (equation 26).

  `-` The GO term score (equation 25) ignores the information content of the GO term.

  `-` The GO term score (equation 25) ignores the evidence codes of the candidate annotations.

- **GO-based with information content score:**
  Feature code: `- gtas ics -`
  Commit: b250834
  The GO annotation score (equation 33) is used to choose the candidate protein, whose GO annotations are transferred to the query.

  `-` The adjustment of the GO term scores (equations 29 – 31) is performed with an informative token threshold fixed to 0.5.

  `gtas` The GO term score (equation 25) incorporates the abundance of GO terms among the candidate proteins (equation 26).

  `ics` The GO term score (equation 25) incorporates the information content of the GO term (equation 27).

  `-` The GO term score (equation 25) ignores the evidence codes of the candidate annotations.

- **GO-based with evidence code score:**
  Feature code: `- gtas - ecs`
  Commit: 510656a
  The GO annotation score (equation 33) is used to choose the candidate protein, whose GO annotations are transferred to the query.

  `-` The adjustment of the GO term scores (equations 29 – 31) is performed with an informative token threshold fixed to 0.5.

  `gtas` The GO term score (equation 25) incorporates the abundance of GO terms among the candidate proteins (equation 26).

  `-` The GO term score (equation 25) ignores the information content of the GO term.

  `ecs` The GO term score (equation 25) incorporates the evidence codes of the candidate annotations (equation 28).

- **GO-based with information content score and evidence code score:**
  Feature code: `- gtas ics ecs`
  Commit: f0a2e61
  The GO annotation score (equation 33) is used to choose the candidate protein, whose GO annotations are transferred to the query.

  `-` The adjustment of the GO term scores (equations 29 – 31) is performed with an informative token threshold fixed to 0.5.

  `gtas` The GO term score (equation 25) incorporates the abundance of GO terms among the candidate proteins (equation 26).

  `ics` The GO term score (equation 25) incorporates the information content of the GO term (equation 27).

  `ecs` The GO term score (equation 25) incorporates the evidence codes of the candidate annotations (equation 28).

- **GO-based with variable informative token threshold:**
  Feature code: `itt gtas - -`
  Commit: 3ebdb8b
  The GO annotation score (equation 33) is used to choose the candidate protein, whose GO annotations are transferred to the query.

  `itt` The adjustment of the token scores (equations 3 – 5) is performed with a variable informative token threshold.

  `gtas` The GO term score (equation 25) incorporates the abundance of GO terms among the candidate proteins (equation 26).

  `-` The GO term score (equation 25) ignores the information content of the GO term.

  `-` The GO term score (equation 25) ignores the evidence codes of the candidate annotations.

- **GO-based with variable informative token threshold and information content score:**
  Feature code: `itt gtas ics -`
  Commit: 9bfb756
  The GO annotation score (equation 33) is used to choose the candidate protein, whose GO annotations are transferred to the query.

  `itt` The adjustment of the token scores (equations 3 – 5) is performed with a variable informative token threshold.

  `gtas` The GO term score (equation 25) incorporates the abundance of GO terms among the candidate proteins (equation 26).

  `ics` The GO term score (equation 25) incorporates the information content of the GO term (equation 27).

  `-` The GO term score (equation 25) ignores the evidence codes of the candidate annotations.

- **GO-based with variable informative token threshold and evidence code score:**
  Feature code: `itt gtas - ecs`
  Commit: 557b902
  The GO annotation score (equation 33) is used to choose the candidate protein, whose GO annotations are transferred to the query.

  `itt` The adjustment of the token scores (equations 3 – 5) is performed with a variable informative token threshold.

  `gtas` The GO term score (equation 25) incorporates the abundance of GO terms among the candidate proteins (equation 26).

  `-` The GO term score (equation 25) ignores the information content of the GO term.

  `ecs` The GO term score (equation 25) incorporates the evidence codes of the candidate annotations (equation 28).

- **Fully featured GO-based with variable informative token threshold, information content score and evidence code score:**
  Feature code: `itt gtas ics ecs`
  Commit: fdfbdea
  The GO annotation score (equation 33) is used to choose the candidate protein, whose GO annotations are transferred to the query.

  `itt` The adjustment of the token scores (equations 3 – 5) is performed with a variable informative token threshold.

  `gtas` The GO term score (equation 25) incorporates the abundance of GO terms among the candidate proteins (equation 26).

  `ics` The GO term score (equation 25) incorporates the information content of the GO term (equation 27).

  `ecs` The GO term score (equation 25) incorporates the evidence codes of the candidate annotations (equation 28).

7.6.2 Test Competitors Alongside AHRD

The third version of our non-redundant ground truth protein set (section 5) was annotated by some of AHRD's best known competitors (section 4.3). We used the AHRD commit 651fd2b of the "sem_sim_go_OWL" branch to create our own descriptions and GO annotations. To highlight the differences in the predictions, we sorted the evaluation scores and plotted line plots (figures 12 and 14). The differences were also calculated on a by-protein basis and plotted in two-sided histograms (figures 13 and 15).

8 The CAFA Challenge

The CAFA (Critical Assessment of Functional Annotation) challenge is a recurring community-wide contest to test competing computational protein function prediction tools [91].

8.1 CAFA3

In 2017 we participated in the third installment of the challenge [77] with AHRD under the name "schoofcropbiobonn". We submitted around 1.5 million annotations for roughly 126 000 test proteins. After the submission deadline in February new experimentally verified annotations accumulated for some of the test proteins until the final benchmark collection was performed in November. The organizers used evaluation metrics established in the previous challenge [92, 93]:

- F_{max} is the F_1-score of the point on the precision-recall curve that maximized the harmonic mean of both precision and recall.
- The weighted version of F_{max} (wF_{max}) takes the information content of GO terms into account.
- The remaining uncertainty (ru) is defined as the average sum of the information content of ground truth terms missing from the prediction (i.e. the false negatives). The misinformation (mi) is the average sum of the information content of predicted terms not found in the ground truth (i.e. the false positives). S_{min} minimizes the Euclidean distance of the remaining uncertainty and the misinformation to the origin of a ru-mi-graph (at which: $ru = 0$ and $mi = 0$).
- A normalized version of S_{min} (nS_{min}) brings the score values between 0 and 1. To achieve this the information content sums are divided by the sum of the information content of the union of prediction and ground truth (i.e. false positives + true positives + false negatives).

In addition the evaluation was divided into multiple aspects:

- Ontology
  - Biological process (BPO)
  - Molecular function (MFO)
  - Cellular component (CCO)
- Species
  - all
    - eukarya
      - *Rattus norvegicus* (RAT)
      - *Candida albicans* SC5314 (CANAX)
      - *Homo sapiens* (HUMAN)
      - *Arabidopsis thaliana* (ARATH)
      - *Dictyostelium discoideum* (DICDI)
      - *Mus musculus* (MOUSE)
      - *Drosophila melanogaster* (DROME)
      - *Schizosaccharomyces pombe* (SCHPO)
      - *Danio rerio* (DANRE)
    - prokarya
      - *Escherichia coli* K-12 (ECOLI)
      - *Salmonella typhimurium* (SALTY)
      - *Bacillus subtilis* (BACSU)
- Type
  - No Knowledge: Only proteins with no experimentally verified annotations at the time of the submission deadline are evaluated.
  - Limited Knowledge: Proteins that, at the time of the submission deadline, had experimentally verified annotations in one or two ontologies (but not in the ontology that is benchmarked) are evaluated.
- Mode
  - Full: Evaluation of all proteins (penalizes not making predictions).
  - Partial: Only proteins for which predictions were made are evaluated.

8.2 CAFA-π

Shortly after CAFA3 (section 8.1) a smaller interim challenge, coined CAFA-π, was held. It focused exclusively on term-centered annotations (GO terms are annotated with the parts of an organism's proteome that they are associated with). The GO term "biofilm formation" (GO:0042710) was to be annotated with proteins from *Candida albicans* (strain SC5314) and *Pseudomonas aeruginosa* (strain UCBPP-PA14) and the GO term "cilium or flagellum-dependent cell motility" (GO:0001539) was to be annotated with *Pseudomonas aeruginosa* proteins. In order to evaluate the CAFA-π submissions, the organizers performed genome-wide screens of the GO terms in question in the aforementioned model organisms.

To make the predictions for our submission we used the term-centered version of AHRD presented in section 7.5. Because the challenge is quite species-specific, we tried to increase AHRD's annotation performance by training it in the respective taxonomic niche. So we used 1 518 Swiss-Prot proteins of the *Candida* genus as ground truth and 33 025 Swiss-Prot proteins as well as ca. 7.6 million TrEMBL proteins of the Fungi kingdom as search space for AHRD's training before the *C. albicans* predictions. Similarly, we used used 3 501 Swiss-Prot proteins of the *Pseudomonas aeruginosa* species group as ground truth and 14 482 Swiss-Prot proteins as well as ca. 5 million TrEMBL proteins of the Pseudomonadales order as search space for AHRD's training before the *P. aeruginosa* predictions. The term-centered version of AHRD is able to predict and consequently also able to be trained on multiple GO terms at once. But to achieve the best performance AHRD is capable of, the training for *P. aeruginosa* was performed for the two GO terms of interest separately.

8.3 CAFA4

The fourth CAFA experiment was announced in late 2019. We participated with over 1.5 million annotations for around 99.7% of the 97 999 target protein sequences. To speed up the preparations necessary for AHRD, we used DIAMOND [25] for the sequence similarity searches on UniProtKB/TrEMBL [4].

Part III
Results

9 Comparison of the Low Redundancy Ground Truth Set to Random Proteins

Section 5 describes the workflow for the creation of a set of ground truth proteins with a low level of function and sequence redundancy based on Swiss-Prot proteins with experimentally verified GO annotations. In figure 3 AHRD's annotation performance on this set is compared to the annotation performance on random proteins from the whole pool of Swiss-Prot proteins with experimentally verified GO annotations. The performance was determined using AHRD's git commit 7389bc5 and adhering to the training and test procedures laid out in section 7.6. Figure 3 also shows the annotation performance of the "Best BLAST" results (section 4.3.2) and the maximum theoretically possible performance (section 4.3.1).

When the non-redundant set is annotated, the average scores are lower for all methods, be it for descriptions (HRD) or GO terms. The amount of proteins annotated with descriptions (HRD coverage) remains almost unchanged. The annotation coverage with GO terms is only marginally lower for AHRD but clearly lower for the "Best BLAST" results.

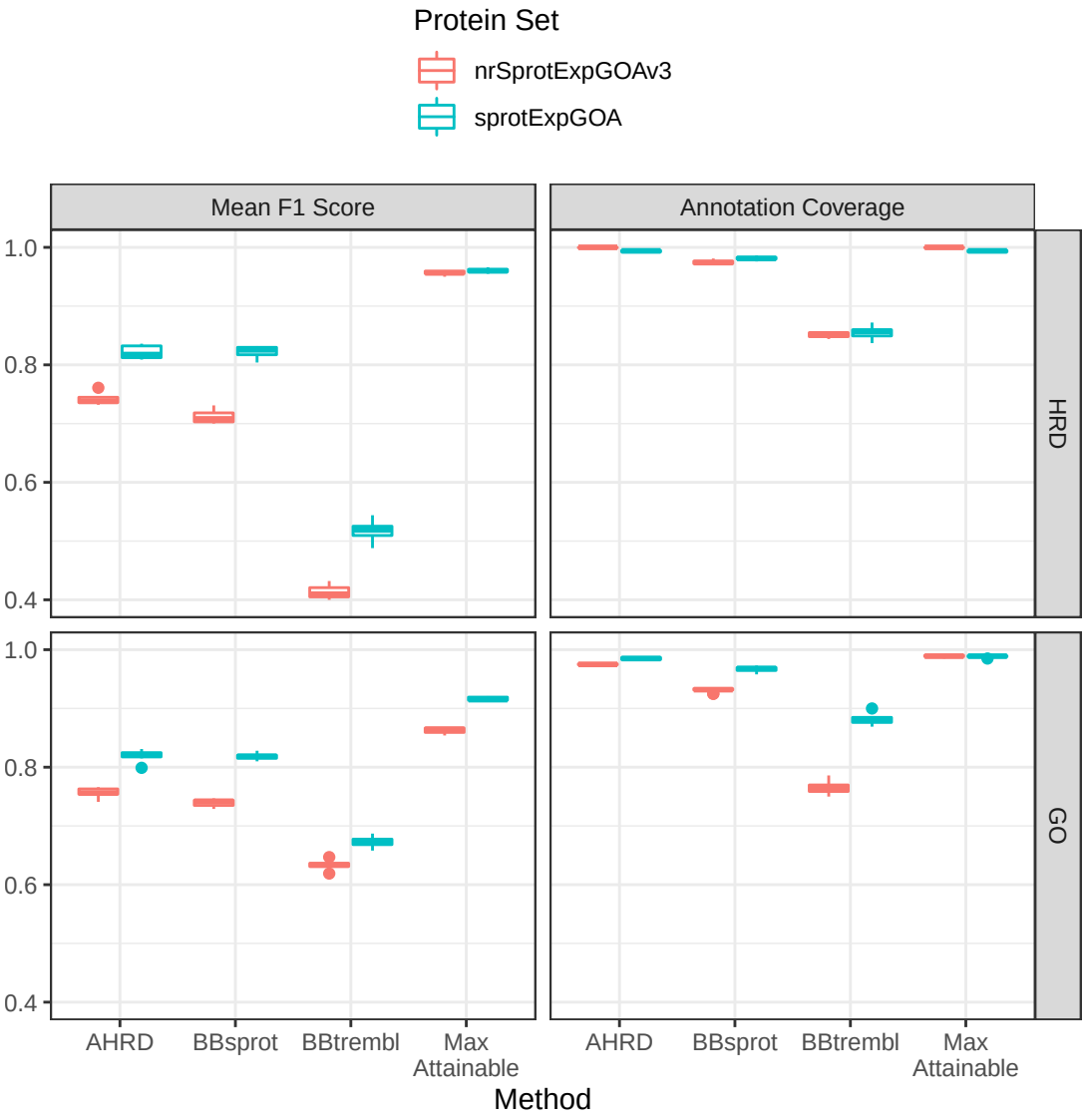

Figure 3: Annotation Performance on a Low Redundancy Protein Set Compared to the Annotation Performance on Random Swiss-Prot Proteins

In order to test AHRD and other protein annotation programs a good set of ground truth proteins is needed. Here the third version of such a set created in accordance to the workflow described in section 5 is compared to random sets of proteins extracted from the same Swiss-Prot release. The non-redundant set of Swiss-Prot proteins with experimentally verified GO annotations (nrSprotExpGOAv3) is shown in pink. Random sets of Swiss-Prot proteins with experimentally verified GO annotations (sprotExpGOA) are shown in turquoise. Next to the annotation performance of AHRD, the performance of the BLAST result with the highest bitscore from Swiss-Prot or TrEMBL is shown (section 4.3.2). To find the highest possible annotation performance all candidate annotations were evaluated against the ground truth. The score averaged over the top candidates of every protein in the set is shown as "Max Attainable" (section 4.3.1). Description-based evaluations are in the top row and GO-based evaluations are in the bottom row. The mean F_1-scores of the annotated proteins are shown in the first column while the second column shows the fraction of annotated proteins in the set. All tests were repeated with randomly selected proteins 10 times.

10 Difference of GO Evaluation Methods

Figure 4 compares the three evaluation methods AHRD (commit 44f1374) provides for GO annotations (section 7.2).

The "Simple" method (section 7.2.1) uses the number of GO terms overlapping in prediction and ground truth compared to the size of these sets to determine precision and recall, which are used to calculate an F_1-score. The "Ancestry" method (section 7.2.2) adds all parental GO terms to the prediction and the ground truth before the sizes of both sets and their overlap are determined. For the "SemSim" method (section 7.2.3) the maximum common information content of the GO terms in prediction and ground truth is compared to calculate a semantic similarity-based precision, recall and subsequently an F_1-score.

Over half of the predictions consist of eight GO terms or less while the ground truth has a median of 11 GO terms. In many cases this leads to a quantization of the "Simple" methods scores to common small fractions such as $\frac{2}{5}$, $\frac{1}{2}$ or $\frac{2}{3}$.

Expanding the ground truth and prediction sets to include their whole ancestry increases the number of possible set sizes and thus results in more fine-grained score values of the ancestry-based F_1-score. Predictions that match the ground truth exactly are expanded with their ancestry in precisely the same way the ground truth is and thus keep the perfect score of 1.0. Predictions with false negative or false positive terms that do not share parts of their ancestry with the respective other set, result in increasingly lower scores than the "simple GO score" the more ancestors they have. However, false negative or false positive terms that do share many ancestral terms with the terms of the respective other set result in increased F_1-scores in comparison to the "simple GO score". This results in a net increase of the mean F_1-score (e.g. 0.691 up from 0.651).

With increasing distance to the root the information content of GO terms can never decrease and typically increases (see section 7.1 for reference). Because the information content does not always increase proportionally to the depth in the ontology, the ancestry-based GO evaluation score and the semantic similarity-based GO evaluation score can differ from each other. Two extreme examples found in Swiss-Prot release 2020_05 combined with the 2020-10 release of the Gene Ontology are the terms "BP: glycolytic process" (GO:0006096) and "CC: symplast" (GO:0055044). GO:0006096 has an ancestry of 47 terms but has been found in Swiss-Prot 5 209 times resulting in an information content of only 5.4. In contrast to that, GO:0055044 has only three ancestors but because it was also annotated only twice (sp|Q94AN2|CHER1_ARATH and sp|O80928|DOF24_ARATH), it has an information content of 12.9. So the "SemSim" approach increases the weight of rare terms comparatively close to the root and lowers the importance of common terms — even if they are further away from the root and have accordingly large ancestries. Table 6 shows the evaluation of an example protein annotation that illustrates this. Additionally, the semantic similarity-based GO evaluation offers greater consistency in a variety of edge cases. Root terms have an information content of 0.0 and are thus not considered as a valid annotation (tables 1 and 5). Predictions for ontologies not

represented in the ground truth are not treated as false positives as they should not be considered to be wrong if no knowledge is available for the particular functional aspect of the protein (table 11).

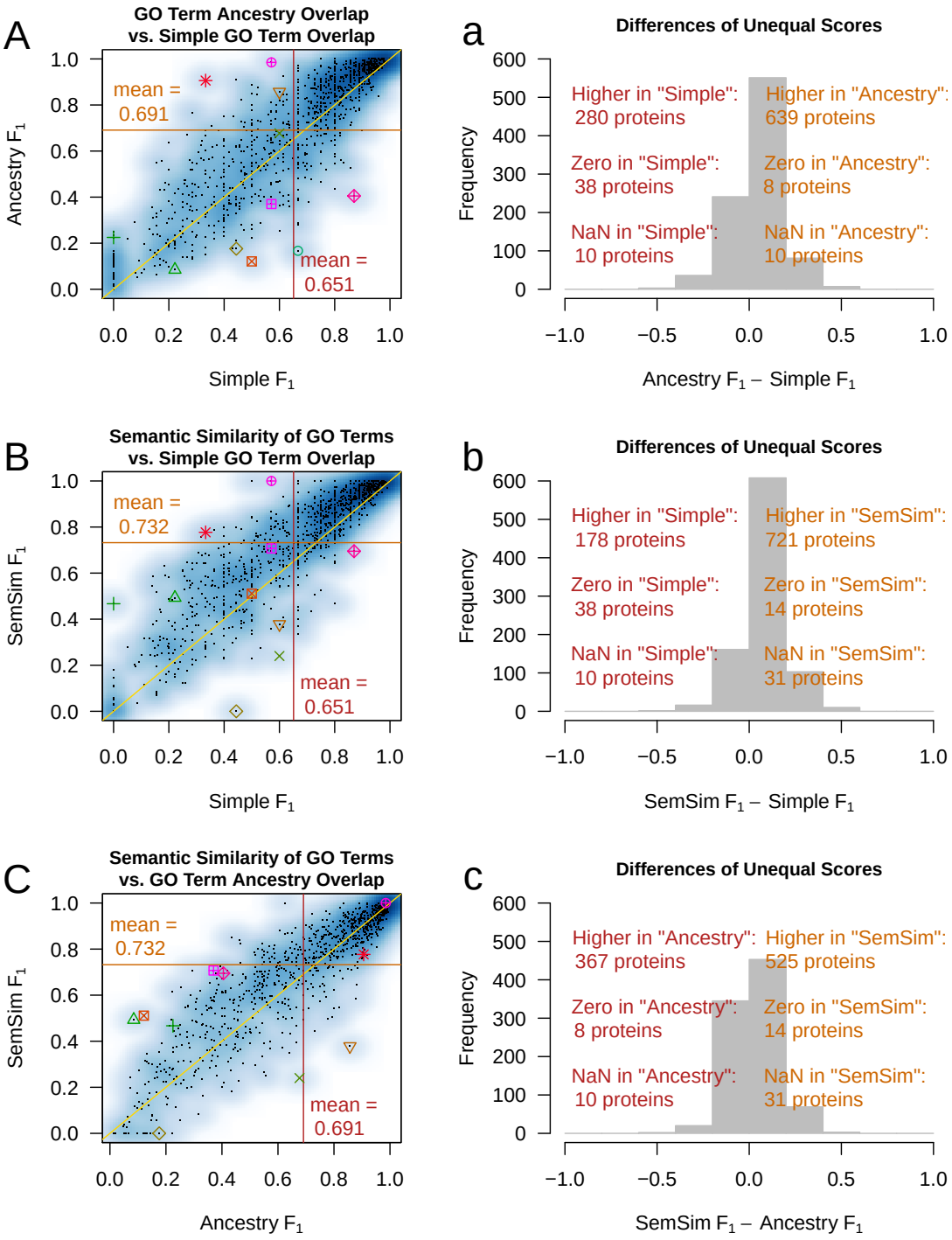
A
GO Term Ancestry Overlap
vs. Simple GO Term Overlap
mean =
0.691
mean =
0.651
Ancestry F1
Simple F1
a
Differences of Unequal Scores
Higher in "Simple":
280 proteins
Higher in "Ancestry":
639 proteins
Zero in "Simple":
38 proteins
Zero in "Ancestry":
8 proteins
NaN in "Simple":
10 proteins
NaN in "Ancestry":
10 proteins
Frequency
Ancestry F1 – Simple F1
B
Semantic Similarity of GO Terms
vs. Simple GO Term Overlap
mean =
0.732
mean =
0.651
SemSim F1
Simple F1
b
Differences of Unequal Scores
Higher in "Simple":
178 proteins
Higher in "SemSim":
721 proteins
Zero in "Simple":
38 proteins
Zero in "SemSim":
14 proteins
NaN in "Simple":
10 proteins
NaN in "SemSim":
31 proteins
Frequency
SemSim F1 – Simple F1
C
Semantic Similarity of GO Terms
vs. GO Term Ancestry Overlap
mean =
0.732
mean =
0.691
SemSim F1
Ancestry F1
c
Differences of Unequal Scores
Higher in "Ancestry":
367 proteins
Higher in "SemSim":
525 proteins
Zero in "Ancestry":
8 proteins
Zero in "SemSim":
14 proteins
NaN in "Ancestry":
10 proteins
NaN in "SemSim":
31 proteins
Frequency
SemSim F1 – Ancestry F1

Figure 4: Comparison of AHRD's GO Annotation Evaluation Methods
AHRD's GO Annotations for 1000 proteins from the third version of our ground truth set (section 5) were evaluated in accordance with three different methods.

Because the prediction and ground truth sets can sometimes be quite small, the "Simple" F_1-scores can become quantized. This can be seen in panel A and B as vertical lines of data points at values of e.g. $\frac{2}{5}$, $\frac{1}{2}$ and $\frac{2}{3}$. The "Ancestry" and "SemSim" methods do not show similar lines which can be attributed to more fine-grained score values.

Based on their mean F_1-scores, ancestry-based scores are generally higher than "Simple" scores (panel A) while "SemSim" scores are even higher still (panel C). This is reaffirmed by the number of proteins where the "Ancestry" or "SemSim" scores are higher than the "Simple" score (panels a and c).

The "Simple" method scores 38 proteins with a 0.0 which implies no overlap between prediction and ground truth. But the "Ancestry" method can find common parental GO terms in 30 of these cases (see table 3 for the example protein O13926). Although these predictions still score very low (see the vertical line at the origin of the graph in panel A) this shows how the "Ancestry" method can give a more nuanced evaluation. With the "SemSim" method the number of predictions scored with a 0.0 increases again (panel c). This can be traced back to the methods recognition of root GO terms which have an information content of 0.0 (see table 5 for the example protein Q557B8).

AHRD was unable to assign any GO terms to 10 of the query proteins. These consequently received NaN (Not a Number) scores in all three methods (panel a and b). The "SemSim" method gave NaN scores to an additional 21 proteins (panel c). In these cases there were GO terms predicted, but none in any (sub-)ontology present in the ground truth (see table 1 for the example protein Q3E7D1).

All example proteins:
○ **Q3E7D1** (table 1): Only root term predicted
△ **Q10478** (table 2): Predicted GO term covers significant part of the information content in various ground truth terms
\+ **O13926** (table 3): Prediction entails a significant part of the ground truth's information content
× **P24559** (table 4): High-information-content-terms completely missing in prediction
◇ **Q557B8** (table 5): Overlap of only root terms
▽ **Q9S851** (table 6): Prediction with many ancestors but low information content
⊠ **P20962** (table 7): GO term with few ancestors but high information content
✳ **Q9P4R5** (table 8): Prediction of a crucial term with many ancestors and high information content
⌖ **P08148** (table 9): Prediction recalls most of the ground truth but misses the most important terms
⊕ **D4A770** (table 10): Root terms in prediction
⊞ **Q9UBM4** (table 11): Ontology missing in ground truth

Table 1: Example Protein ○ Q3E7D1: Only Root Term Predicted
The only knowledge available about this protein is in the CCO: It is located in the chloroplast (GO:0009507). But the root term of the BPO is also present in the ground truth and the prediction.

The "simple GO score" does not take the different importance of these terms into account. Thus, half of the ground truth is considered reproduced, resulting in a recall of $1/2 = 0.5$.

The "ancestry GO score" weights the "chloroplast" term much higher but does not dismiss the BPO root terms as irrelevant. So its recall is set to $1/11 = 0.\overline{09}$.

When the "SemSim GO score" is calculated, the BPO root term's information content of 0.0 makes the prediction effectively empty and thus prevents the precision to be calculated. In essence, the protein is considered unannotated.

Ground Truth:	sp|Q3E7D1|FB131_ARATH						Ancestors:	IC:
CC: GO:0009507 chloroplast							10	4.54
BP: GO:0008150 biological_process							1	0.00
Prediction:	sp|Q3EBI7|FB130_ARATH						Ancestors:	IC:
BP: GO:0008150 biological_process							1	0.00
	Simple			Ancestry			SemSim	
F_1	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall
0.667	1.0	0.5	0.167	1.0	0.091	NaN	NaN	0.0
One of two terms			GO:0009507 has nine ancestors. So one of 11 terms have been recalled.			Root terms are ignored, so effectively no prediction was made.		

Table 2: Example Protein △ Q10478: Predicted GO Term Covers Significant Part of the Information Content in Various Ground Truth Terms
Of six CC terms and one term in BP and MF each, only one CC term has been predicted.

Because none of the three GO evaluation scores find any false positives, the precision is determined to be 1.0 with all the methods.

Weighting all GO terms equally, the "simple GO score" computes a recall of $1/8 = 0.125$.

On the one hand the ancestries of the CC terms in the ground truth overlap on many of their terms. But on the other hand the predicted CC term "outer membrane" (GO:0019867) reproduces only three of these ancestral CC terms. Even worse: The BP term "protein insertion into mitochondrial outer membrane" (GO:0045040) has the most ancestors of any ground truth term, and all of these are considered false negatives. Thus, the "ancestry GO score" calculates the recall to be only ≈ 0.044.

The overlap of the ground truth CC term's ancestries is also reflected in the combined information content of ground truth for this ontology. It is far lower than the sum of the shown information content numbers. Additionally, despite its many ancestors the BP term "protein insertion into mitochondrial outer membrane" (GO:0045040) has not a proportionally high information content. The predicted CC term is thus considered to reproduce a significant amount of the ground truth's information content, resulting in a "SemSim GO recall" of a comparatively high value of ≈ 0.327

Ground Truth:	sp|Q10478|SAM50_SCHPO	Ancestors:	IC:
CC: GO:0001401	SAM complex	14	9.64
CC: GO:0005739	mitochondrion	9	4.34
CC: GO:0005741	mitochondrial outer membrane	15	6.65
CC: GO:0016020	membrane	2	1.86
CC: GO:0016021	integral component of membrane	4	2.26
CC: GO:0019867	outer membrane	3	5.41
BP: GO:0045040	protein insertion into mitochondrial outer membrane	38	8.57
MF: GO:0003674	molecular_function	1	0.00
Prediction:	tr|E9D731|E9D731_COCPS	Ancestors:	IC:
CC: GO:0019867	outer membrane	3	5.41

Simple			Ancestry			SemSim		
F_1	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall
0.222	1.0	0.125	0.085	1.0	0.044	0.493	1.0	0.327
Only one out of eight terms			GO:0001401, GO:0005741 and GO:0045040 each have a large number of ancestors.			GO:0019867 recalls a significant part of the information content of the other CC terms.		

Table 3: Example Protein + O13926: Prediction Entails a Significant Part of the Ground Truth's Information Content

As there is no direct overlap between the ground truth and the prediction GO terms, the "simple GO score" is set to 0.0.

Because "methylation" (GO:0032259) is in the ancestry of both "peptidyl-lysine methylation" (GO:0018022) and "regulation of cytoplasmic translation" (GO:2000765) a small part of the BPO is reproduced. Also, both terms predicted in the MFO (GO:0008168 "methyltransferase activity" and GO:0016740 "transferase activity") are ancestors of the only ground truth term in the MFO (GO:0016279 "protein-lysine N-methyltransferase activity"). Consequently, the "ancestry GO precision" is determined to be 1.0 and the recall low but not 0.0 with a value of ≈ 0.127.

The information content of GO terms with many ancestors is compared to the information content of terms with fewer ancestors. But in this case the information contented does not scale proportionally with the size of the ancestry. The information content of the prediction is thus found to cover a significantly greater part of the ground truth's information content than its ancestry is able to reproduce of the ground truth's ancestry. Therefore, with a value of ≈ 0.304 the "SemSim GO recall" is higher as well.

Ground Truth:	sp|O13926|YF66_SCHPO	Ancestors:	IC:
CC: GO:0005634	nucleus	8	3.05
CC: GO:0005737	cytoplasm	4	1.62
CC: GO:0005829	cytosol	5	3.41
BP: GO:0018022	peptidyl-lysine methylation	22	7.54
BP: GO:2000765	regulation of cytoplasmic translation	20	8.13
MF: GO:0016279	protein-lysine N-methyltransferase activity	11	7.64
Prediction:	tr|S9R925|S9R925_SCHOY	Ancestors:	IC:
BP: GO:0032259	methylation	3	5.51
MF: GO:0008168	methyltransferase activity	5	4.05
MF: GO:0016740	transferase activity	3	2.11

Simple			Ancestry			SemSim		
F_1	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall
0.0	0.0	0.0	0.225	1.0	0.127	0.467	1.0	0.304
No terms overlap			GO:0032259 is ancestor of GO:0018022. GO:0008168 and GO:0016740 are ancestors of GO:0016279.			Prediction entails a significant part of the ground truth's information content		

Table 4: Example Protein × P24559: High-Information-Content-Terms Completely Missing in Prediction

Of seven GO terms in the ground truth the prediction was able to reproduce three exact matches. Thus, the "simple GO recall" is set to $3/7 \approx 0.429$ and the precision to 1.0.

Because some of the ground truth's terms have overlapping ancestries, the prediction's ancestry can cover more than half of it despite missing some important terms. Thus, the "ancestry GO recall" is set to ≈ 0.512.

The MF term "ATPase activity" (GO:0016887) which is missing in the prediction has only the root term in common with the predicted terms "nucleotide binding" (GO:0000166) and "ATP binding" (GO:0005524). So its information content is counted completely towards the false negatives. In light of their moderately sized ancestries the two BP terms "pilus retraction" (GO:0043108) and "type IV pilus-dependent motility" (GO:0043107) both have a comparatively high information content. Other than the root term they only overlap in one more term: "cellular process" (GO:0009987). Because for the BPO no prediction was made at all and the small overlap of the two ground truth terms most of their combined information content will be considered as false negative. The "SemSim GO recall" is set accordingly low with a value of ≈ 0.137.

Ground Truth:	sp|P24559|PILT_PSEAE	Ancestors:	IC:
CC:	GO:0005737 cytoplasm	4	1.62
CC:	GO:0044096 type IV pilus	5	12.17
BP:	GO:0043108 pilus retraction	7	13.92
BP:	GO:0043107 type IV pilus-dependent motility	6	11.35
MF:	GO:0000166 nucleotide binding	7	2.13
MF:	GO:0005524 ATP binding	10	2.50
MF:	GO:0016887 ATPase activity	8	3.89
Prediction:	sp|Q06581|PILT_NEIGO	Ancestors:	IC:
CC:	GO:0005737 cytoplasm	4	1.62
MF:	GO:0000166 nucleotide binding	7	2.13
MF:	GO:0005524 ATP binding	10	2.50

Simple			Ancestry			SemSim		
F_1	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall
0.6	1.0	0.429	0.677	1.0	0.512	0.24	1.0	0.137
Recalled three out of seven terms			The three recalled terms have just as many ancestors as the remaining four			GO:0016887 has only the root in common with GO:0000166 and GO:0005524. No predictions for Biological Process: Full penalization for the missing terms GO:0016887, GO:0043108 and GO:0043107		

Table 5: Example Protein ◇ Q557B8: Overlap of Only Root Terms
The only immediate intersection of ground truth and prediction are the root terms of the CCO and the MFO. The other two ground truth terms are missing and the "simple GO recall" is consequently set to ¾=0.5. Three of the five predicted terms are not present in the ground truth, so the "simple GO precision" turns out to $^2/_5 = 0.4$.

Because the ancestry of the three predicted CC terms overlap, the number of predicted terms considered for the ancestry-based scores, is only seven. The predicted root term of the BPO is an ancestor of "aggregation involved in sorocarp development" (GO:0031152) in the ground truth, thus the number of true positives is three terms. Thus, the "ancestry GO precision" is calculated as $^3/_7 \approx 0.429$. Counting 24 ancestral terms, the term "aggregation involved in sorocarp development" (GO:0031152) makes up the majority of the ground truth's ancestry. Because it is missing in the prediction, the "ancestry GO recall" turns out to $^3/_{27} = 0.11\overline{1}$.

The ground truth term "extracellular region" (GO:0005576) has only its root term in common with the predicted terms "membrane" (GO:0016020) and "integral component of membrane" (GO:0016021). But the root terms have no information content. Consequently, the "SemSim GO score" records a true positive semantic similarity of 0.0 and thus sets the precision as well as the recall to 0.0.

Ground Truth: sp|Q557B8|COMF_DICDI	Ancestors:	IC:
CC: GO:0005575 cellular_component	1	0.00
CC: GO:0005576 extracellular region	3	3.40
BP: GO:0031152 aggregation involved in sorocarp development	24	9.17
MF: GO:0003674 molecular_function	1	0.00
Prediction: sp|Q54LY1|Y6311_DICDI	**Ancestors:**	**IC:**
CC: GO:0005575 cellular_component	1	0.00
CC: GO:0016020 membrane	3	1.86
CC: GO:0016021 integral component of membrane	5	2.26
BP: GO:0008150 biological_process	1	0.00
MF: GO:0003674 molecular_function	1	0.00

Simple			Ancestry			SemSim		
F_1	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall
0.444	0.4	0.5	0.176	0.429	0.111	0.0	0.0	0.0
Overlap of two root terms			The two root terms become a smaller portion of the prediction and ground truth.			Ignoring the root terms eliminates all similarity of prediction and ground truth.		

Table 6: Example Protein ▽ Q9S851: Prediction With Many Ancestors but Low Information Content

If all predicted terms also appear in the ground truth, the "simple GO precision" is determined to be 1.0. But only three of the seven terms in the ground truth have been reproduced, so the "simple GO recall" is set to $3/7 \approx 0.429$.

Most of the ancestors in the ground truth are also found in the prediction, the "ancestry GO recall" thus turns out to be 0.75.

The recalled term "regulation of transcription, DNA-templated" (GO:0006355) with the highest number of ancestors in the ground truth has a comparatively low information content. In contrast "meristem initiation" (GO:0010014) and "organ boundary specification between lateral" (GO:0010199) — both ground truth terms missing in the prediction — have a high information content in regard to the sizes of their ancestries. Consequently, with a value of ≈ 0.233, the "SemSim GO recall" is set much lower.

Ground Truth:	sp|Q9S851|NAC31_ARATH	Ancestors:	IC:
CC: GO:0005634	nucleus	8	3.05
BP: GO:0006355	regulation of transcription, DNA-templated	19	3.53
BP: GO:0007275	multicellular organism development	5	5.32
BP: GO:0010014	meristem initiation	3	10.18
BP: GO:0010199	organ boundary specification between lateral organs and the meristem	6	11.72
MF: GO:0003677	DNA binding	6	3.00
MF: GO:0003700	DNA-binding transcription factor activity	3	4.30
Prediction:	tr|F5B9T7|F5B9T7_BRAOL	Ancestors:	IC:
CC: GO:0005634	nucleus	8	3.05
BP: GO:0006355	regulation of transcription, DNA-templated	19	3.53
MF: GO:0003677	DNA binding	6	3.00

Simple			Ancestry			SemSim		
F_1	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall
0.6	1.0	0.429	0.857	1.0	0.75	0.378	1.0	0.233
Three out of seven terms recalled			GO:0006355 has many ancestors			Despite its may ancestors GO:0006355 has a low information content. High information content terms GO:0007275, GO:0010014 and GO:0010199 are missing		

Table 7: Example Protein ☒ P20962: GO Term With Few Ancestors but High Information Content

The prediction contains just one term also found in the ground truth among two others. Consequently, the "simple GO recall" for this prediction is $1/3 = 0.33\overline{3}$.

The ancestry of the solely predicted term "immune system process" (GO:0002376) contains only the root of the BPO and itself. With this in mind, its information content is comparatively high with a value of 4.25. In contrast, the ground truth term "DNA replication" (GO:0006260) has far more ancestors (n=22), but with a value of 5.08, a comparatively small information content. Therefore, the "ancestry GO recall" is very low ($\frac{2}{8+2+21} \approx 0.065$) while the "SemSim GO recall" is relatively high ($\frac{4.25}{3.05+4.25+5.08} \approx 0.343$).

Ground Truth: sp|P20962|PTMS_HUMAN								Ancestors:	IC:
CC: GO:0005634 nucleus								8	3.05
BP: GO:0002376 immune system process								2	4.25
BP: GO:0006260 DNA replication								22	5.08
Prediction: sp|P08814|PTMS_BOVIN								Ancestors:	IC:
BP: GO:0002376 immune system process								2	4.25
Simple			Ancestry			SemSim			
F_1	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	
0.5	1.0	0.333	0.121	1.0	0.065	0.511	1.0	0.343	
Recalled one of three terms			GO:0005634 and GO:0006260 have many ancestral terms; GO:0002376 has only one.			About a third of the ground truth's information content is from GO:0002376.			

Table 8: Example Protein ✳ Q9P4R5: Prediction of a Crucial Term With Many Ancestors and High Information Content

Only one term was predicted. As it is also present in the ground truth, the "simple GO recall" turns out to ⅕=0.2.

Because the predicted term "carbon catabolite repression of transcription" (GO:0045013) has a high number of ancestors as well as a high information content, the "ancestry GO recall" and "SemSim GO recall" are both much higher than the "simple GO recall" ($\frac{53}{1+7+53+1+2} \approx 0.828$ and $\frac{9.86}{3.05+9.86+2.62} \approx 0.635$)

Ground Truth: sp|Q9P4R5|CREC_EMENI								Ancestors:	IC:
CC: GO:0005575 cellular_component								1	0.00
CC: GO:0005634 nucleus								8	3.05
BP: GO:0045013 carbon catabolite repression of transcription								53	9.86
MF: GO:0003674 molecular_function								1	0.00
MF: GO:0005515 protein binding								3	2.62
Prediction: tr|C5FCR9|C5FCR9_ARTOC								Ancestors:	IC:
BP: GO:0045013 carbon catabolite repression of transcription								53	9.86
Simple			Ancestry			SemSim			
F_1	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	
0.333	1.0	0.2	0.906	1.0	0.828	0.777	1.0	0.635	
One out of five terms recalled			GO:0045013 has many ancestors			GO:0045013 has a high information content			

Table 9: Example Protein ⌖ P08148: Prediction Recalls Most of the Ground Truth but Misses the Most Important Terms

The ground truth consists of 13 terms spread over all three ontologies. All but three terms (GO:0036523, GO:0052014 and GO:0052032) are reproduced by the prediction. Thus, the "simple GO score" is $^{10}/_{13} \approx 0.769$.

Each of the three missing terms has a high number of ancestors. Although these three ancestries overlap partially, they still make up the majority of the combined ancestry of the ground truth. As a consequence, the "ancestry GO recall" is set relatively low (≈ 0.254).

The three missing GO terms also have a high information content each. But compared to the number of ancestors the increase is not as pronounced and the aforementioned overlap of their ancestries lowers their combined information content further. Thus, the information content missing in the prediction is only about half of the information content of the ground truth. Therefore, with a value of ≈ 0.532, the "SemSim GO recall" is higher than the "ancestry GO score".

Ground Truth:	sp|P08148|GP63_LEIMA	Ancestors:	IC:
CC: GO:0005886	plasma membrane	4	2.56
CC: GO:0016020	membrane	2	1.86
CC: GO:0031225	anchored component of membrane	4	6.26
BP: GO:0006508	proteolysis	9	4.60
BP: GO:0007155	cell adhesion	3	5.43
BP: GO:0036523	positive regulation by symbiont of host cytokine secretion	69	13.92
BP: GO:0052014	catabolism by symbiont of host protein	33	12.82
BP: GO:0052032	modulation by symbiont of host inflammatory response	26	12.82
MF: GO:0004222	metalloendopeptidase activity	7	5.77
MF: GO:0008233	peptidase activity	5	3.98
MF: GO:0008237	metallopeptidase activity	7	5.11
MF: GO:0016787	hydrolase activity	3	2.17
MF: GO:0046872	metal ion binding	5	2.12
Prediction:	sp|P15706|GP63_LEICH	Ancestors:	IC:
CC: GO:0005886	plasma membrane	4	2.56
CC: GO:0016020	membrane	2	1.86
CC: GO:0031225	anchored component of membrane	4	6.26
BP: GO:0006508	proteolysis	9	4.60
BP: GO:0007155	cell adhesion	3	5.43
MF: GO:0004222	metalloendopeptidase activity	7	5.77
MF: GO:0008233	peptidase activity	5	3.98
MF: GO:0008237	metallopeptidase activity	7	5.11
MF: GO:0016787	hydrolase activity	3	2.17
MF: GO:0046872	metal ion binding	5	2.12

Simple			Ancestry			SemSim		
F_1	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall
0.87	1.0	0.769	0.405	1.0	0.254	0.695	1.0	0.532
Recalled 10 out of 13 terms			The three missing terms GO:0036523, GO:0052014 and GO:0052032 have large ancestries.			The three missing terms GO:0036523, GO:0052014 and GO:0052032 have a high information content.		

Table 10: Example Protein ⊕ D4A770: Root Terms in Prediction
The three root terms in the prediction are not present in the ground truth and are therefore considered false positives in the calculation of the "simple GO score". So the "simple GO precision" is set to ⅖=0.4.

The correctly predicted term "regulation of mast cell degranulation" (GO:0043304) has a large ancestry. This lowers the effect of the falsely predicted root terms and results in an "ancestry GO precision" of $\frac{4+28}{1+3+1+27+1} \approx 0.97$.

In the calculation of the "SemSim GO score" the zero information containing root terms are treated as such and are consequently not considered false positives. Accordingly, the "SemSim GO precision" is set to 1.0.

Ground Truth: sp|D4A770|CL004_RAT							Ancestors:	IC:
CC: GO:0005737 cytoplasm							4	1.62
BP: GO:0043304 regulation of mast cell degranulation							28	9.15
Prediction: sp|Q91YN0|CL004_MOUSE								
CC: GO:0005575 cellular_component							1	0.00
CC: GO:0005737 cytoplasm							4	1.62
BP: GO:0008150 biological_process							1	0.00
BP: GO:0043304 regulation of mast cell degranulation							28	9.15
MF: GO:0003674 molecular_function							1	0.00
Simple			Ancestry			SemSim		
F_1	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall
0.571	0.4	1.0	0.985	0.97	1.0	1.0	1.0	1.0
Only two of the five predicted terms are also in the ground truth			GO:0043304 has many ancestral terms			The "SemSim GO score" ignores root terms		

Table 11: Example Protein ⊞ Q9UBM4: Subontology Missing in Ground Truth

Out of the four GO terms in the prediction and the three GO terms in the ground truth only two overlap directly. Consequently, the "simple GO precision" is $^2/_4 = 0.5$ and the "simple GO recall" is $^2/_3 = 0.66\overline{6}$.

Because the false positive term "negative regulation of angiogenesis" (GO:0016525) has a comparatively high number of ancestors, the "ancestry GO precision" is lower ($\frac{3+1+1}{3+1+15+1} = 0.25$). The "ancestry GO recall", on the other hand, is slightly higher ($\frac{3+1+1}{3+1+3} \approx 0.714$).

If for an entire subontology (in this case the BPO) no prior knowledge is available, a prediction for the particular functional aspect cannot be known to be wrong either. For this reason the "SemSim GO score" does not consider the predicted term "negative regulation of angiogenesis" (GO:0016525) a false positive. The root term of the MFO with its 0.0-information-content also does not add to the false positives. Therefore, no information content was predicted that was not also part of the ground truth and the "SemSim GO precision" is thus set to 1.0. But because the complete ground truth information content in the MFO (7.69) was missing in the prediction, the "SemSim GO recall" is the lowest in comparison to the other two methods ($\frac{3.4+5.87}{3.4+5.87+7.69} \approx 0.547$).

Ground Truth:	sp|Q9UBM4|OPT_HUMAN	Ancestors:	IC:
CC:	GO:0005576 extracellular region	3	3.40
CC:	GO:0031012 extracellular matrix	4	5.87
MF:	GO:0005201 extracellular matrix structural constituent	3	7.69
Prediction:	**sp|Q920A0|OPT_MOUSE**	**Ancestors:**	**IC:**
CC:	GO:0005576 extracellular region	3	3.40
CC:	GO:0031012 extracellular matrix	4	5.87
BP:	GO:0016525 negative regulation of angiogenesis	15	8.07
MF:	GO:0003674 molecular_function	1	0.00

Simple			Ancestry			SemSim		
F_1	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall
0.571	0.5	0.667	0.37	0.25	0.714	0.707	1.0	0.547
Recalled two out of three terms. Two out of four predicted terms are false positives.			The false positive term GO:0016525 has many ancestors.			As long as there is no ground truth available for the biological process ontology, predictions for it are not considered false positives.		

11 Variation of GO Prediction Performance Based on Factors Other Than the Prediction Algorithm

To assess the magnitude of improvements provided by new GO annotation approaches (section 7.6.1), a baseline of mean F_1-score variation due to other factors had to be established. The two other major factors that influence the mean F_1-score in the presented test setup are the random partitioning of the ground truth proteins into training and test set (section 7.6) and the randomness inherent to AHRD's parameter optimization techniques (sections 6.3 and 7.4). Figure 5 shows boxplots of the mean F_1-scores based on these two influences. Once for 1000 random samples of 1000 proteins from all 10 splits and 10 times for 1000 repeated parameter optimizations of each of the 10 splits. Comparing the inter quantile range (IQR) of the training-based mean F_1-scores (< 0.0031) with the IQR of the split-based mean scores ($= 0.0096$) demonstrates a high difference of their spread. This is confirmed by Fligner-Killeen tests [94] (null hypothesis: variances in groups are the same) of the aforementioned training-based mean scores and the split-based mean scores (supplemental table 13).

It follows that in order to control for the variation between splits the scores from different prediction algorithms should be compared in a pairwise fashion on a by protein basis (sections 12 and 13).

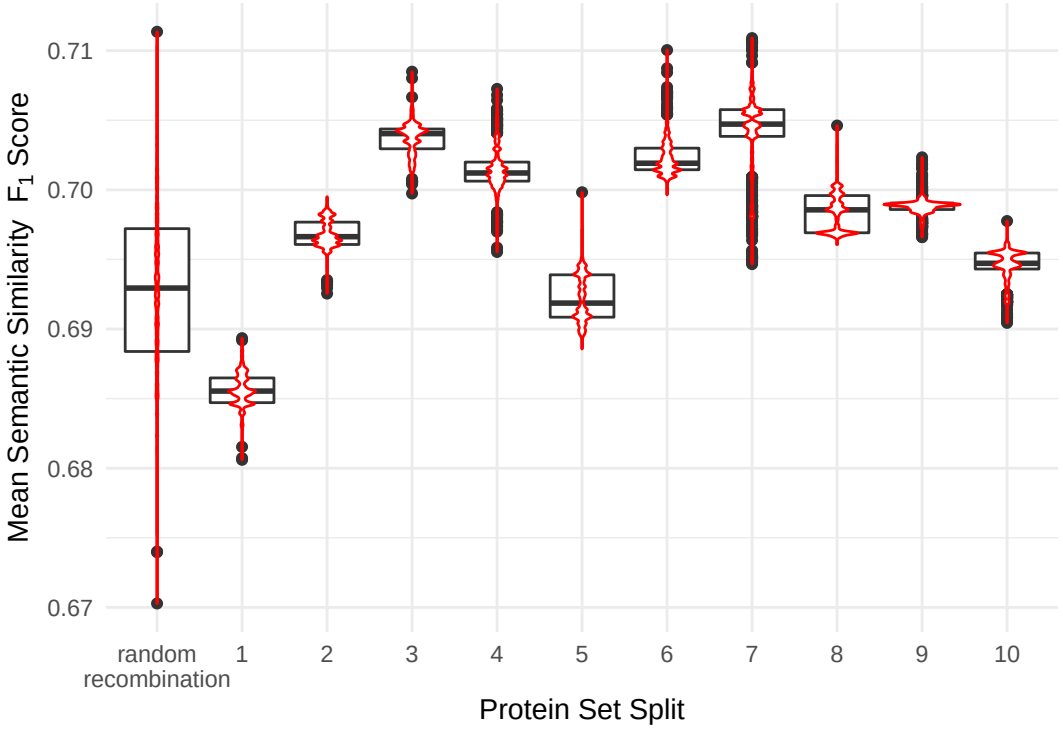

Figure 5: The Influence of Random Protein Set Partitioning and Randomness in Parameter Optimization

The mean F_1-score of 1000 GO predictions randomly drawn from all 10 splits were calculated 1000 times and are shown in the left-most boxplot. The aim was to simulate the splitting itself and to show the possible variation of the mean F_1-score based on this factor. For each split, AHRD's parameter optimization was performed 1000 times resulting in the other 10 boxplots. Each boxplot is overlain with a violin plot: A mirrored density plot with a standard deviation for the gaussian smoothing kernel of 0.0001.

While the random samples from the pool of all splits show a normal distribution of their mean F_1-scores, the same cannot be observed for the individual splits. This observation is confirmed by Shapiro-Wilk normality tests [95] (supplemental table 13).

In light of the non-normality of the mean F_1-scores obtained by repeated training the homogeneity of the variance between the random split samples and the retrained splits was tested with non-parametric Fligner-Killeen tests [94] (supplemental table 13). The tests confirm the visually intuitive observation that the variation introduced by the randomness built into AHRD's training is much smaller than the variation caused by the random splitting of the protein set.

12 Impact of the New Parameter "Informative Token Threshold" in Conventional Description-Based Prediction

12.1 The "Informative Token Threshold" Does Not Significantly Improve the Prediction of Human Readable Descriptions

We compared AHRD's description prediction performance using the new parameter "informative_token_threshold" (commit dab2244) with the conventional approach lacking it (commit 9b3a4de). As described in section 7.6 the test setup consisted of 10 random ground truth set splits à 1000 proteins. In figure 6 the mean description evaluation scores of both variants are plotted against each other on a per-split basis. One-sided pairwise t-tests on the scores of all 10 000 description annotation pairs show an increase of the mean precision at a significance level of 0.001 but a decrease of the mean recall at a significance level of 0.05. These effects cancel each other out partially and thus result in F_1-scores that are only increased at a significance threshold of 0.05. The genetic algorithm (section 7.4) set the new parameter to values between 0.7 and 0.98 (not shown here).

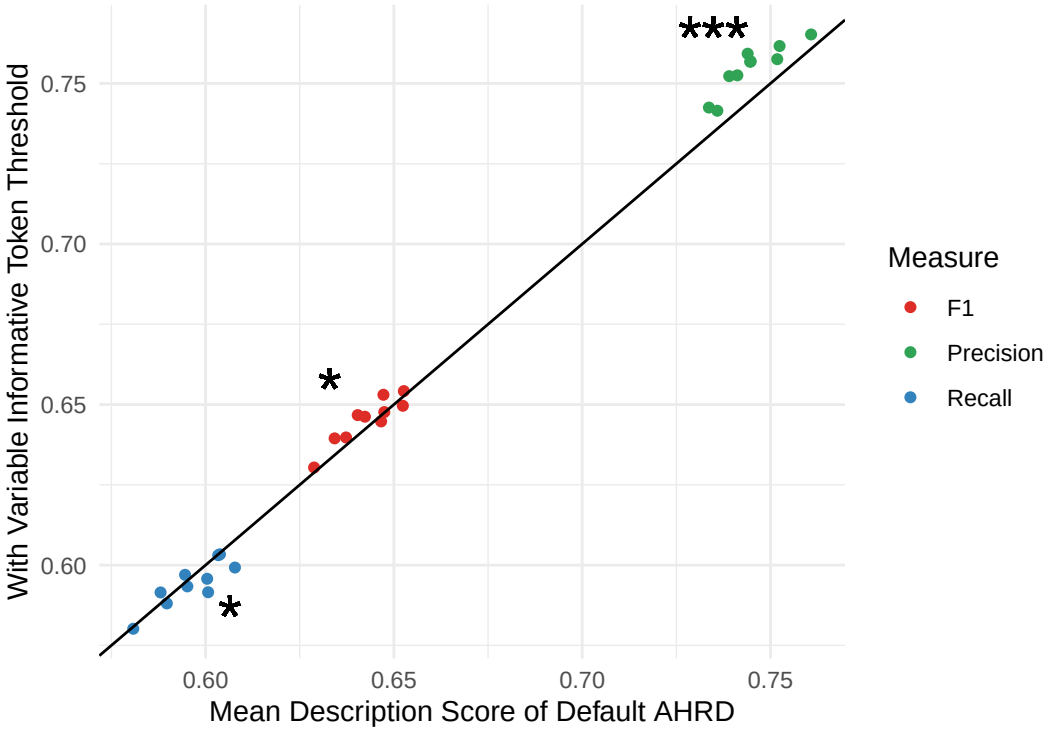

Figure 6: Impact of the New Parameter "Informative Token Threshold" on the Description Annotation Performance
Parameter optimization and parameter evaluation of 10 splits of our ground truth set (section 5) were performed with two versions of AHRD. One with the new parameter "informative token threshold" (commit dab2244) and one without (commit 9b3a4de). One-sided pairwise t-tests were performed on the description score pairs of all 10 000 protein annotations. The significance level of test results is indicated above the diagonal in cases of an increase to the scores and below the diagonal in cases of a significant test of the opposite hypothesis pair. The p-value thresholds for the "stars notation" are: `0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1`.

12.2 The Performance of AHRD's Conventional Description-Based GO Prediction Is Improved by Lowering the "Informative Token Threshold"

AHRD's GO prediction performance using the conventional description-based algorithm was evaluated using a fixed (commit 24a82dc) and a variable (commit dab2244) threshold for the distinction of informative and uninformative tokens. The test setup — laid out in section 7.6 — consisted of 10 random ground truth set splits resulting in 10 test sets with 1000 proteins each. Figure 7 shows the mean scores (F_1, recall and precision) of both variants on a per-test-set-basis. Testing for pairwise differences in the scores of all 10 000 GO annotation pairs showed an increase of both precision and recall on a significance level of 0.001. The resulting F_1-scores were thus also observed to be increased on this significance level.

AHRD's parameter optimization procedure (section 7.4) determined the optimal value for the threshold to be very close to 0.0 (0.037 and smaller) with one exception out of the 10 repetitions where the value was set to 1.0.

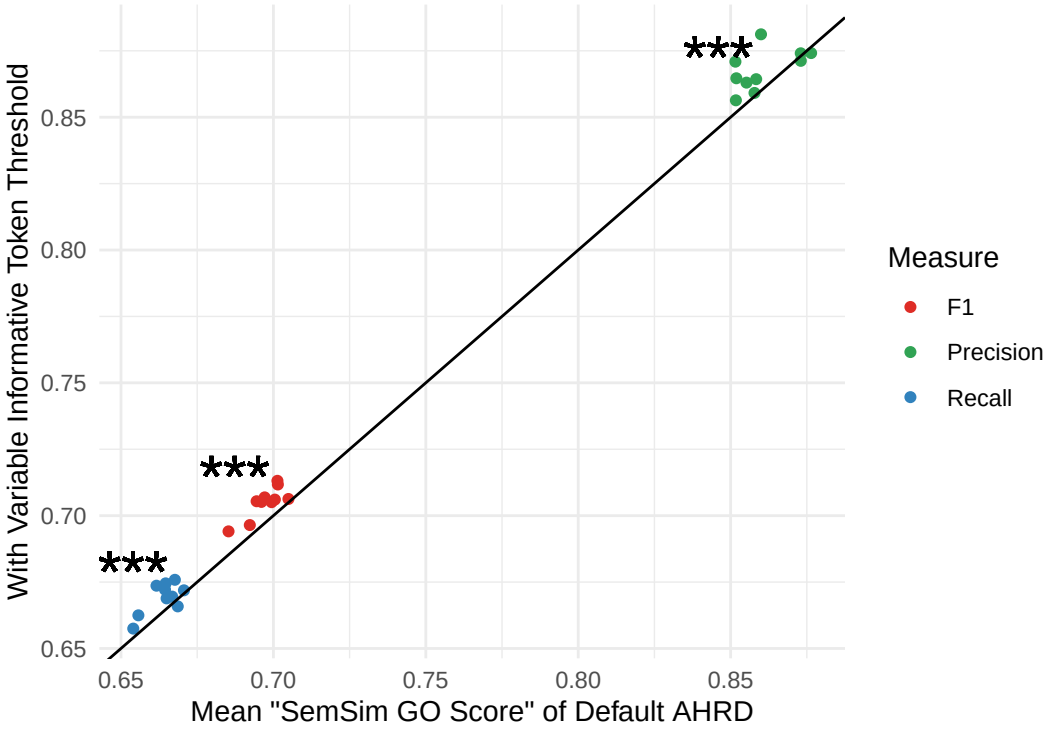

Figure 7: Impact of the New Parameter "Informative Token Threshold" on GO Annotation Performance in HRD-Bound GO Prediction
Parameter optimization and parameter evaluation of 10 splits of our ground truth set (section 5) was performed with two versions of AHRD. One with the new parameter "informative token threshold" (commit dab2244) and one without (commit 24a82dc). Here the mean semantic similarity GO evaluation scores over the 1000 proteins of each test set are shown using both methods. To assess the significance of the prediction-performance-differences between the two versions of AHRD, two-sided pairwise t-tests were performed on all pairs of the 10 000 test protein GO annotations. The "stars notation" (`0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1`) is used to indicate the significance level of a score increase (above the diagonal) or of a score decrease (below the diagonal).

13 Separate GO-Based Approaches to the GO Prediction Increase Performance Significantly

Section 7.6.1 shows the combinatorial abundance of different parameters leading to many GO-based prediction approaches that we explored. Here we compare all combinations with each other to determine the contribution of single aspects to the improvement of the prediction performance. In order to keep this from becoming convoluted, we opted for a matrix of pairwise comparisons instead of a separate plot for each pair. Each score is shown in a separate plot: The F_1-scores are compared in figure 8 (red), whereas figure 9 (blue) shows the comparison of the recall and the precision-based plot is in figure 10 (green).

The intersections of the first two columns (on the x-axis) with the first two lines (on the y-axis) repeat the comparisons shown in figure 7 in section 12.2: Making the threshold for the distinction of informative from uninformative tokens variable (`itt`) results in an increase of AHRD's GO prediction performance (F_1: $p < 0.001$) provided by both higher recall ($p < 0.001$) and precision ($p = 0.001$).

The GO-based prediction without the "`itt`" (`- gtas - -`) shows a significant performance regression compared to the description-based approaches (`- - - -` and `itt - - -`). The addition of the information content score (`- gtas ics -`) recoups some of the recall at a cost to the precision. Using the evidence code score (`- gtas - ecs`) instead achieves even higher scores. But both at the same time (`- gtas ics ecs`) are necessary to be better than the description-based approach using the informative token threshold (`itt - - -`) at a significance level of 0.01.

The variable informative token threshold (`itt`) is also a major influence on the GO prediction performance in all approaches using a GO-based algorithm (`gtas`) instead of a description-based one. The genetic algorithm found very low values to be optimal for it. The highest informative token threshold found in the here presented tests was ≈ 0.066. It provides a boost to the recall at the cost of precision. But the higher recall outweighs the lower precision. The addition of the information content score to the prediction algorithm (`itt gtas ics -`) slightly shifts the balance between recall and precision resulting in no net gain to F_1. However, the evidence code score (`itt gtas - ecs`) can increase both recall and precision by a little, resulting in an improvement to the mean F_1-score at a significance level of 0.01. The fully featured algorithm (`itt gtas ics ecs`) shows the highest mean F_1-score and recall. The F_1-score is significantly increased compared to all other prediction approaches baring the previous one.

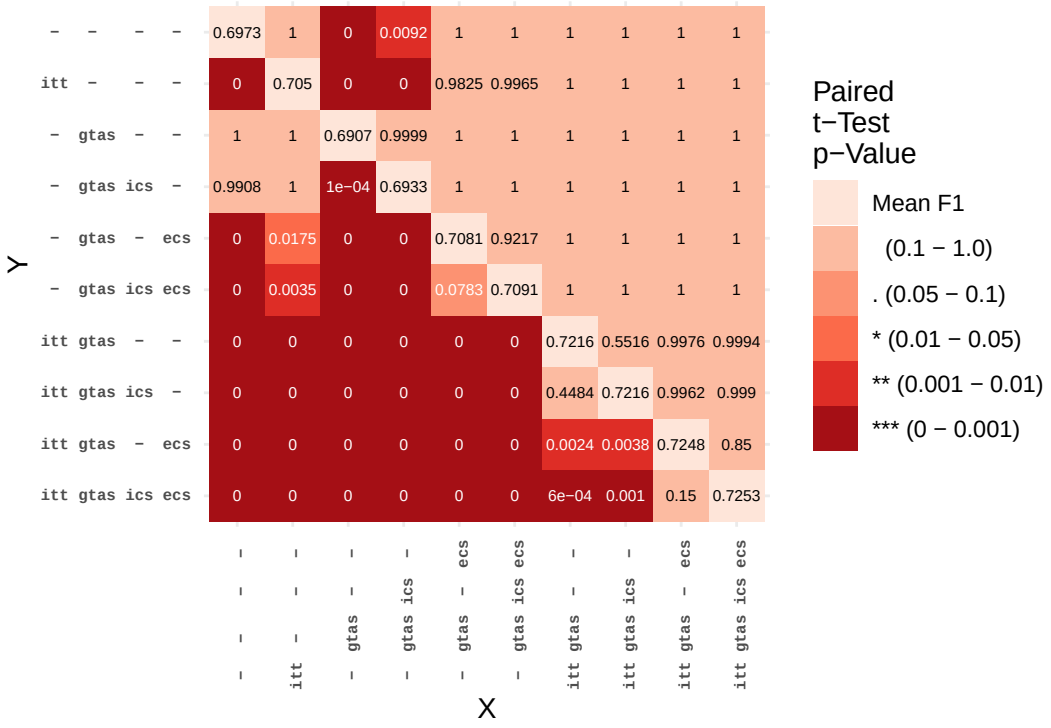

Figure 8: One-Sided Pairwise T-Tests of the F_1-Scores Obtained With Different GO Prediction Approaches

Each cell shows the p-value of a one-sided pairwise t-test of F_1-scores from 1000 proteins in 10 training/test-set splits each. P-values very close to 1.0 or 0.0 had to be rounded to the whole number due to readability considerations. The cells are color graded in accordance to commonly used p-value thresholds. The alternative hypothesis for each t-test below the diagonal is: The mean of the pairwise differences of the F_1-scores of the approach on the y-axis compared to the F_1-scores of the approach on the x-axis is greater than 0.0. While the alternative hypothesis for each t-test above the diagonal is: The mean of the pairwise differences is smaller than 0.0.

`itt:` `informative token threshold`
`gtas:` `go term abundancy score`
`ics:` `information content score`
`ecs:` `evidence code score`

A detailed explanation for the feature codes of the prediction algorithms is provided in section 7.6.1. The diagonal shows the mean F_1-score over all 10 000 proteins.

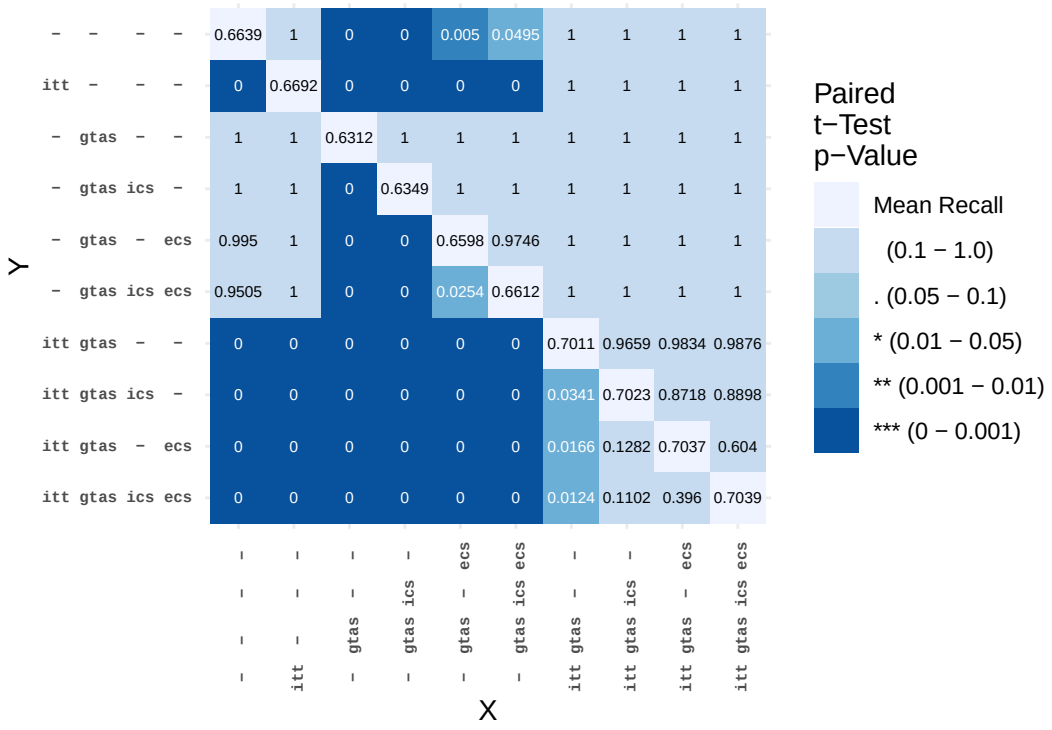

Figure 9: One-Sided Pairwise T-Tests of the Recall Obtained With Different GO Prediction Approaches

Each cell shows the p-value of a one-sided pairwise t-test of the recall from 1000 proteins in 10 training/test-set splits each. P-values very close to 1.0 or 0.0 had to be rounded to the whole number due to readability considerations. The cells are color graded in accordance to commonly used p-value thresholds. The alternative hypothesis for each t-test below the diagonal is: The mean of the pairwise differences of the recall of the approach on the y-axis compared to the recall of the approach on the x-axis is greater than 0.0. While the alternative hypothesis for each t-test above the diagonal is: The mean of the pairwise differences is smaller than 0.0.

`itt:` `informative token threshold`
`gtas:` `go term abundancy score`
`ics:` `information content score`
`ecs:` `evidence code score`

A detailed explanation for the feature codes of the prediction algorithms is provided in section 7.6.1. The diagonal shows the mean recall over all 10 000 proteins.

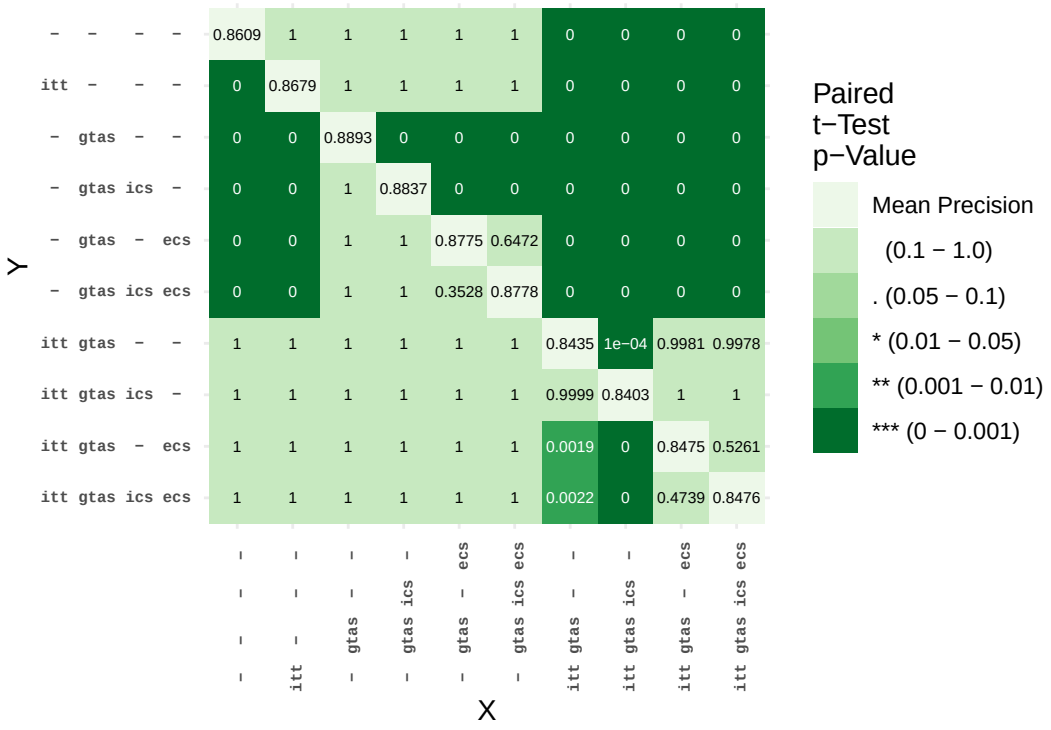

Figure 10: One-Sided Pairwise T-Tests of the Precision Obtained With Different GO Prediction Approaches

Each cell shows the p-value of a one-sided pairwise t-test of the precision from 1000 proteins in 10 training/test-set splits each. P-values very close to 1.0 or 0.0 had to be rounded to the whole number due to readability considerations. The cells are color graded in accordance to commonly used p-value thresholds. The alternative hypothesis for each t-test below the diagonal is: The mean of the pairwise differences of the precision of the approach on the y-axis compared to the precision of the approach on the x-axis is greater than 0.0. While the alternative hypothesis for each t-test above the diagonal is: The mean of the pairwise differences is smaller than 0.0.

`itt:` `informative token threshold`
`gtas:` `go term abundancy score`
`ics:` `information content score`
`ecs:` `evidence code score`

A detailed explanation for the feature codes of the prediction algorithms is provided in section 7.6.1. The diagonal shows the mean precision over all 10 000 proteins.

14 Direct GO Annotation Instead of Transfer From a Reference Protein Severely Lowers Performance

Usually AHRD ranks the candidate reference proteins found via sequence similarity and transfers the GO annotation along the description from the top protein to the query. But the rest of the reference proteins can contain valuable information too. Combining human readable descriptions would necessitate complex natural language processing — a large research area by itself. However, GO annotations from multiple references can be combined much more easily. To test this in AHRD, we used a version (commit abaff70) of our GO-based ranking algorithm (section 7.3.1) and switched it from transferring whole sets of GO terms from a single reference protein to annotating the query with all GO terms from the pool of candidates as long as they exceed a certain quality threshold (commit e33e711). This GO term score threshold is relative to the maximum GO term score (equation 25) of the GO term candidates and was optimized with the genetic algorithm-based trainer along AHRD's other parameters. Both GO annotation approaches were tested in accordance to the procedure described in section 7.6.1.

98.7% of the proteins received GO annotations regardless of the annotation method. On average, the classical approach annotated proteins with 12.5 GO terms, while the new method increased this number to 20.4. The GO annotation performance of this new version is plotted on the ordinate against its predecessor's performance on the abscissa in figure 11. Although the increased number of GO terms increases the recall significantly, this also leads to the precision suffering drastically, which in turn leads to a significant decrease of the F_1-score.

Figure 11: Annotation of Top Ranking GO Terms Instead of Annotation of GO Term Set From Top Ranking Protein

Parameter optimization and parameter evaluation of 10 splits of our ground truth set (section 5) was performed with two versions of AHRD.

In the conventional version of AHRD (commit abaff70, shown on the horizontal axis) GO term scores (equation 25 in section 7.3.1) are used to determine a GO annotation score (equation 33) for each candidate protein. These are used to rank the candidate proteins to transfer the set of GO terms from the top candidate to the query.

The alternative approach (commit e33e711, shown on the vertical axis) tested here uses the GO term scores to rank the GO terms of all candidate proteins directly instead. Then, the informative token threshold (parameter taking part in the optimization of AHRD) is used to determine a cutoff value relative to the highest GO term score. All GO terms making the cutoff are transferred to the query.

Here the mean semantic similarity GO evaluation score of each 1000-protein-test-set is shown using both methods. To assess the significance of the prediction-performance-differences between the two versions of AHRD, two-sided pairwise t-tests were performed on all pairs of the 10 000 test protein GO annotations. The "stars notation" (`0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1`) is used to indicate the significance level of a score increase (above the diagonal) or of a score decrease (below the diagonal).

15 AHRD's Prediction Performance Compared to Competitors

As described in section 7.6.2 we assessed the annotation performance of AHRD and its competitors on our low redundancy test set (section 5). Descriptions were evaluated according to words overlapping between prediction and ground truth, while GO annotations were scored according to their semantic similarity to the ground truth (section 7.2.3).

Annotation with human readable descriptions
Figure 12 shows that AHRD can annotate more proteins with descriptions, which perfectly fit the ground truth, than any of its competitors. It also shows that none of the other methods can annotate as many proteins as AHRD does. The granularity of the evaluation of descriptions is limited by the relatively small number of words present in typical protein descriptions. Figure 13 shows the differences between AHRD and its closest competitors (Blast2GO and BBsprot). AHRD has a better score in the majority of cases and there are some cases where only AHRD can annotate the protein. But there are no proteins that have only been annotated by a competitor.

GO annotation
As the smooth curves in figure 14 show, our semantic similarity-based approach allows a very fine-grained evaluation of GO annotations. Together with BBsprot, AHRD annotates the most proteins with a perfect set of GO terms. But when it comes to the number of proteins that cannot be annotated at all BBsprot loses to AHRD. NetGO can annotate even more proteins (i.e. higher coverage) but cannot provide many good annotations. Only AHRD provides both high quality and high coverage. The direct comparison of proteins that received different GO annotations by competitors in figure 15 shows more proteins with a better score for AHRD. Compared to Blast2GO, BBsprot, and EggNOGmapper AHRD also has more exclusively annotated proteins, i.e. it has a higher coverage. Only NetGO can exceed AHRD's coverage but does so by sacrificing quality.

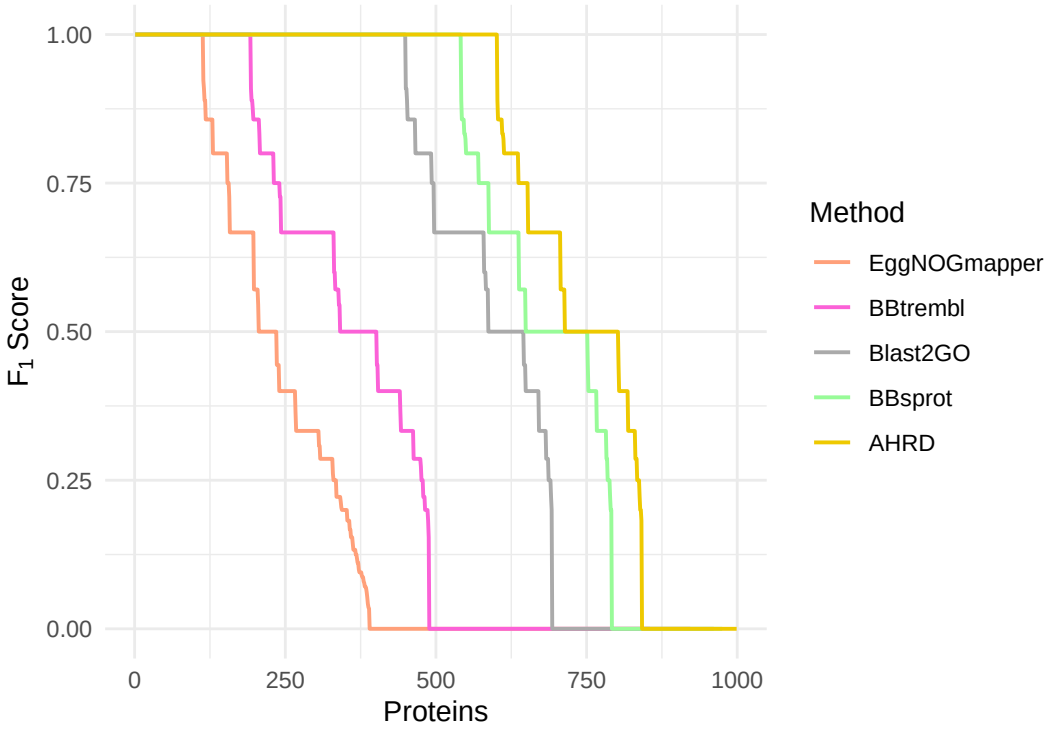

Figure 12: Description Annotation Performance of AHRD and Competitors

The same 1000 random proteins from the ground truth set were annotated by AHRD and competing methods. Here each method's scores are separately sorted in descending order.

With around 600 proteins, AHRD annotates more proteins with perfect descriptions than any of its competitors. The steepness of the lines shows that the number of proteins annotated with loosely fitting descriptions is similar between all competitors. AHRD annotates the highest amount of proteins (841) at a non-zero score.

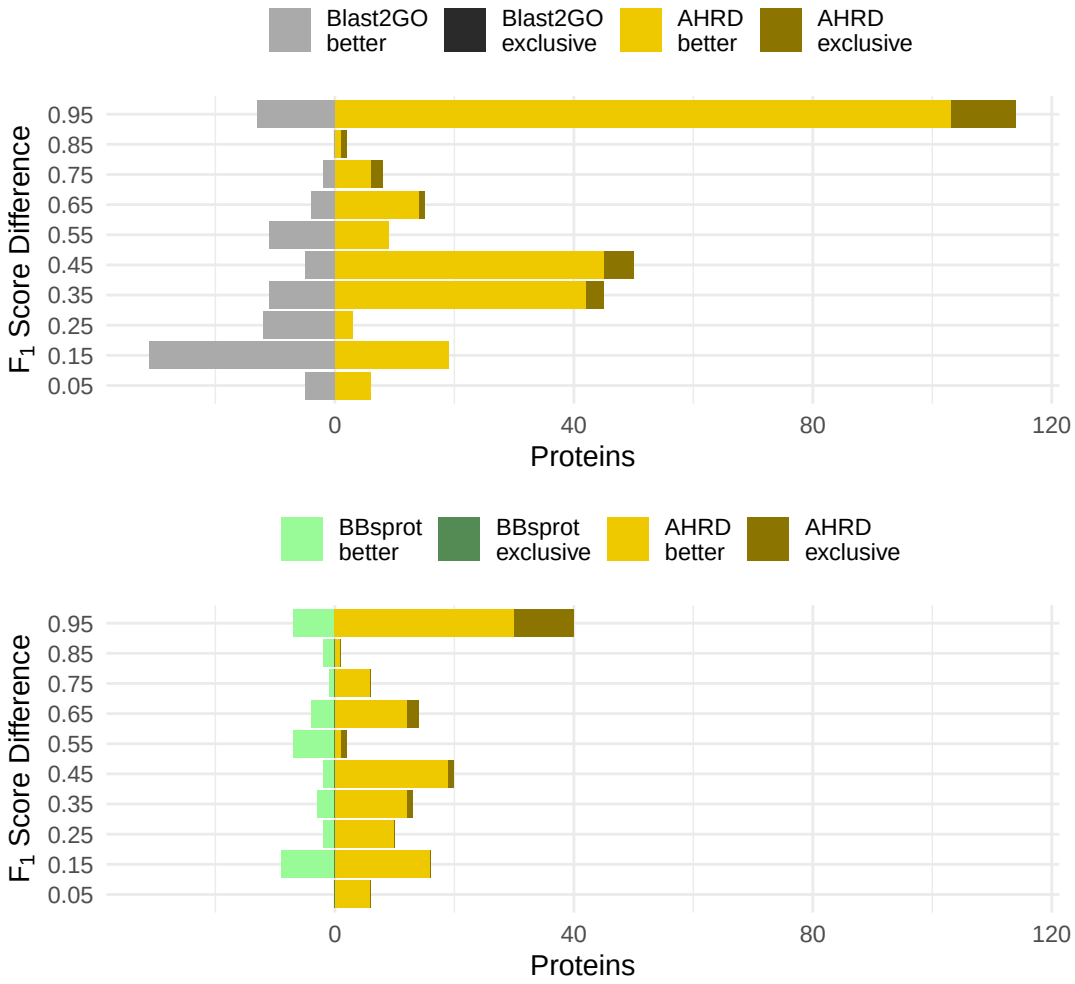

Figure 13: Difference in Description Annotation Performance Between AHRD, Blast2GO and BBsprot
In many cases the F_1-score of AHRD and the competitor is the same. Thus, difference values of zero had to be removed to attain proper scaling.

When AHRD's description differs from the one annotated by its competitors, most of the time AHRD achieves a better score. Additionally, some proteins that are not annotated by the competitors at all were able to be annotated by AHRD — often at a high quality. The proteins AHRD was unable to annotate were also not annotated by Blast2GO or BBsprot.

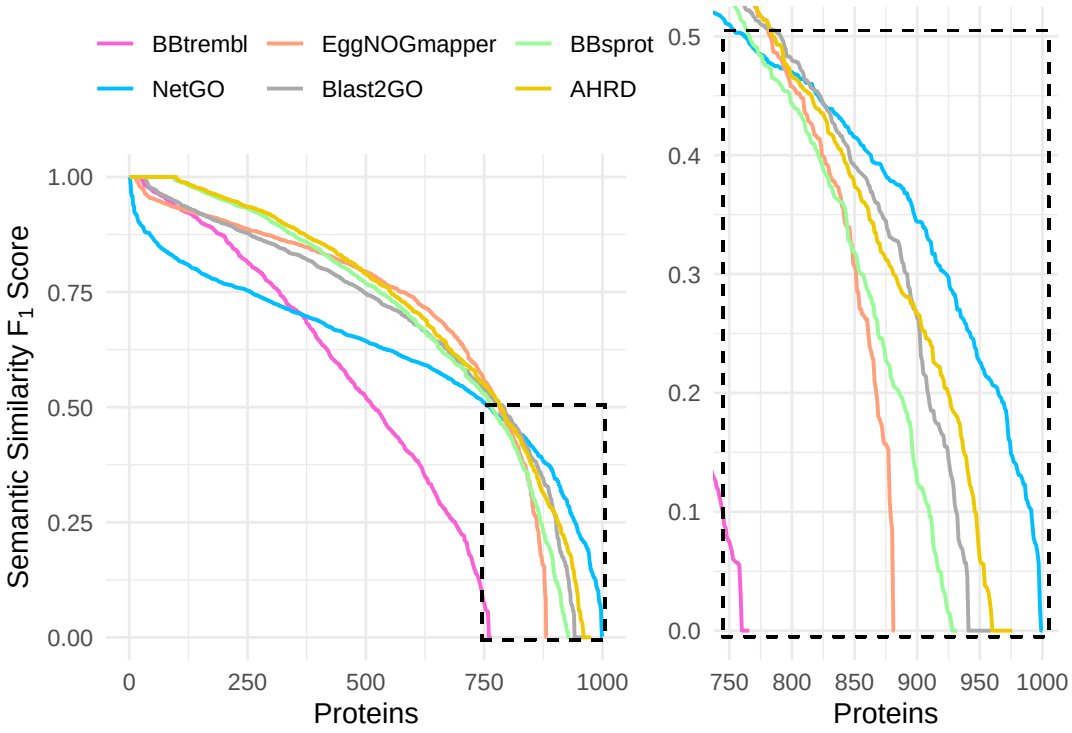

Figure 14: GO Annotation Performance of AHRD and Competitors
The same 1000 random proteins from the ground truth set were annotated by AHRD and competing methods. Each method's scores are separately sorted in descending order.

AHRD and BBsprot achieve a perfect F_1-score equal to 1.0 in 99 and 94 cases respectively. All other competitors only manage to do this for far fewer proteins. Following the curves down to lower F_1-scores, AHRD beats BBsprot in terms of coverage (959 instead of 927 proteins annotated). Only NetGO provides a higher coverage than AHRD but does so mostly with annotations scoring comparatively low. AHRD is the only method providing high quality and high coverage of annotations at the same time.

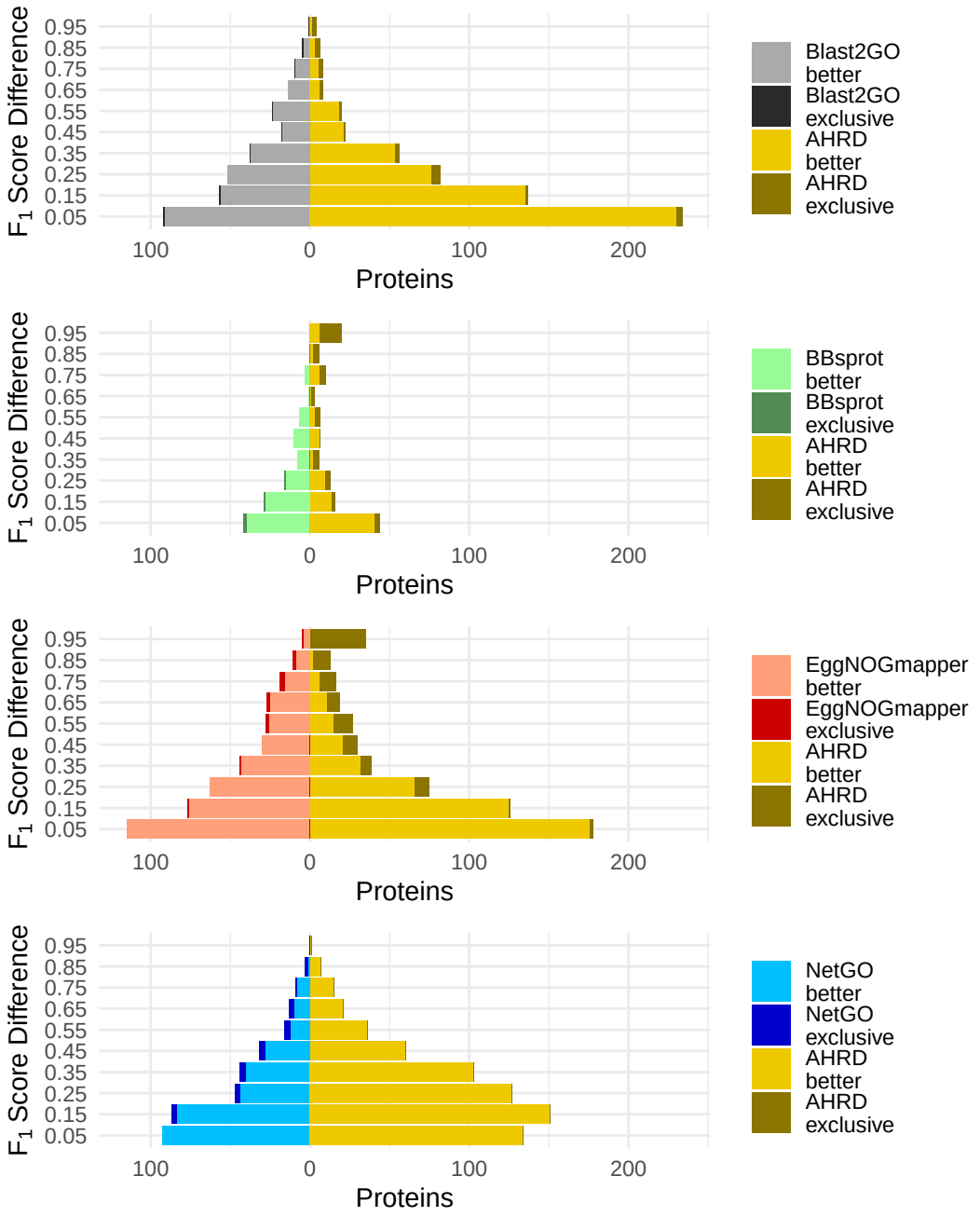

Figure 15: Difference in GO Annotation Performance of AHRD Versus Blast2GO, BBsprot, EggNOGmapper and NetGO
We removed difference values of zero to maintain proper scaling.

Compared to Blast2GO, AHRD annotates more proteins exclusively and annotates hundreds of proteins a little better. BBsprot's GO annotations are more similar to AHRD's, but here more proteins are only annotated by AHRD. And these AHRD-exclusive annotations tend to be good matches to the ground truth. EggNOGmapper has small advantages at a lot of proteins, but so does AHRD. In addition, AHRD can annotate a lot of proteins with good GO terms that EggNOGmapper cannot annotate at all. NetGO can annotate the proteins that AHRD cannot. But AHRD provides higher quality annotations for the majority of proteins that are annotated by both.

16 AHRD Can Increase the Annotation Coverage of Established Proteomes

To test the comprehensiveness of AHRD's annotations we compared its annotations for the proteomes of Barley and *Blumeria graminis* to prior knowledge by Mascher et al. [73] and annotations from the UniProtKB [4] respectively. Figure 16 shows the prior proteome coverage and the coverage AHRD, BBsprot and BBtrembl achieve. An additional 13% of the proteome is annotated with descriptions by AHRD, and an additional 43% of the proteome is annotated with GO terms. The "Best BLAST" results from Swiss-Prot can match and sometimes even exceed the prior annotation coverage but only AHRD annotates over 90% of the proteome consistently.

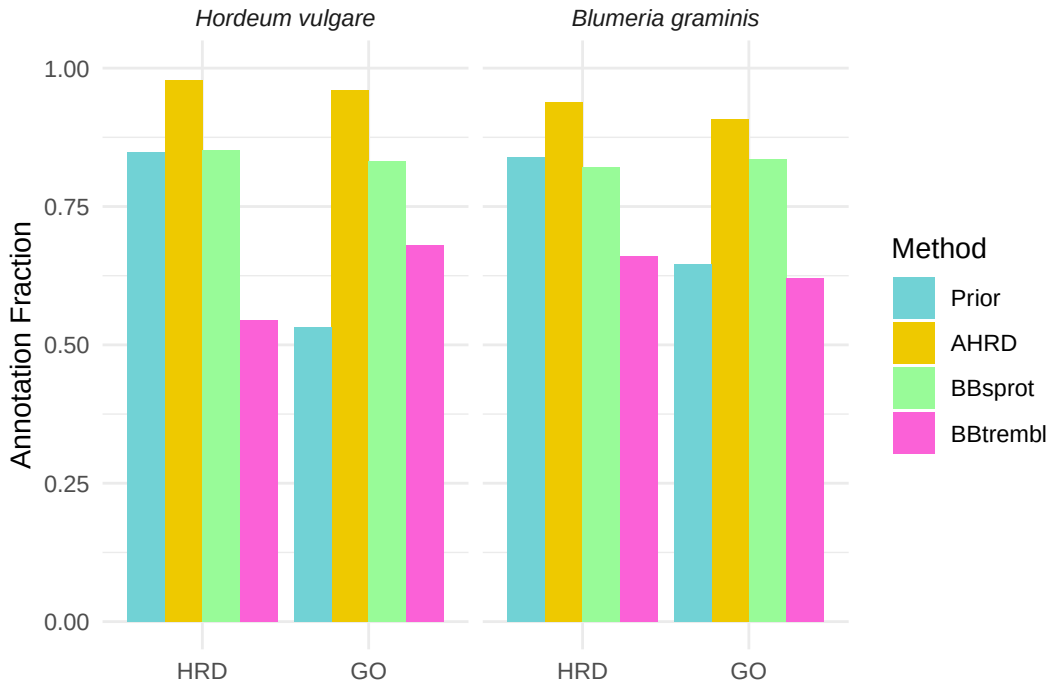

Figure 16: Coverage of Two Proteomes Prior to and After Annotation With AHRD

Prior annotations by Mascher et al. [73] were used for Barley and extracted from the UniProtKB for Blumeria. In both cases the proteins associated with any closely related subspecies were removed from AHRD's search databases to avoid self-matches. In the case of Barley AHRD was able to raise the fraction of annotated proteins by 15% (from 85% to 98%) for descriptions and by 80% (from 53% to 96%) for GO terms. AHRD was able to increase the portion of Blumeria proteins with descriptions by 12% (from 84% to 94%) and the portion of proteins with GO terms by 40% (from 65% to 91%). BBsprot and BBtrembl were sometimes able to match or even exceed the previous annotation coverage of the two example proteomes. But only AHRD was consistently able to provide a comprehensive annotation fraction of over 90%.

17 Parameter Optimization Improves AHRD's Annotation Performance

Figure 17 shows the effect that optimizing AHRD's parameters has on its performance when our low redundancy ground truth set is annotated (section 5). Parameters that we chose based on intuition did not improve AHRD's performance in a significant way in comparison to naive parameters that give equal weight to all annotation quality metrics AHRD uses. In contrast, the genetic trainer (section 7.4) was able to improve the average F_1-score by 7% for descriptions and 11% for GO terms.

AHRD currently supports two different methods for the optimization of its parameters: Simulated annealing (section 6.3) and a genetic algorithm (section 7.4). In figure 18 both methods are compared to the repeated evaluation of randomly drawn parameter sets. All three methods work with the same computational budget and therefore arrived at the shown scores after a similar runtime. The improvements the machine learning algorithms were able to achieve over the random method are small but significant. Furthermore, the genetic algorithm was able to show the best results by an even smaller but also significant margin.

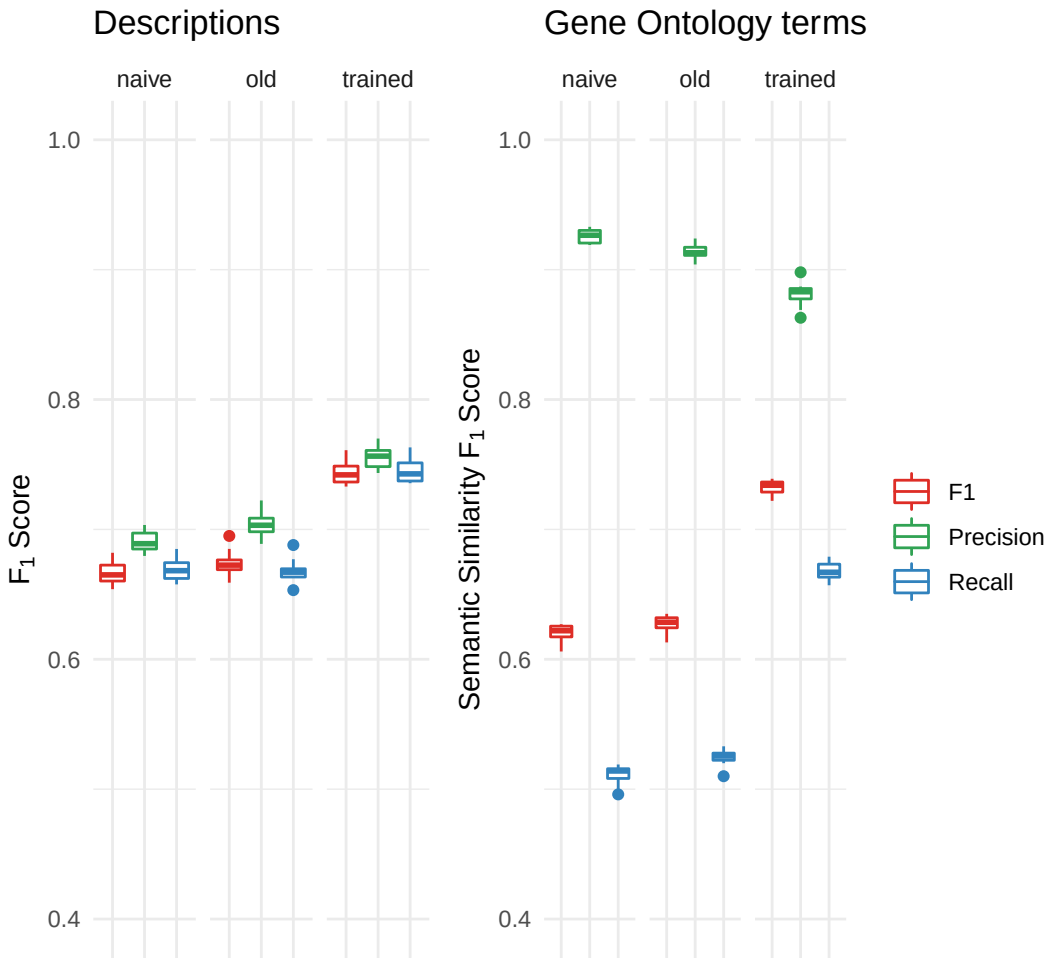

Figure 17: AHRD's Annotation Performance Before and After Optimization of Its Parameters
AHRD's annotation performance on our ground truth set (section 5) was assessed depending on different parameter sets. Naive parameters give equal weight to the various quality metrics and search databases that AHRD uses (section 6.1.2). Intuitive parameters were manually selected by us and the optimized parameters were obtained using machine learning (section 7.4).

While the intuitively chosen parameters can only increase the scores marginally, machine learning can give AHRD a boost of around 7% for descriptions by increasing both precision and recall. When predicting GO terms AHRD's optimization can even increase its performance by 11%. This is achieved by reducing the underprediction the naive and intuitive sets result in (shown by the high precision but low recall) and considerably raising the recall at only a small cost to the precision.

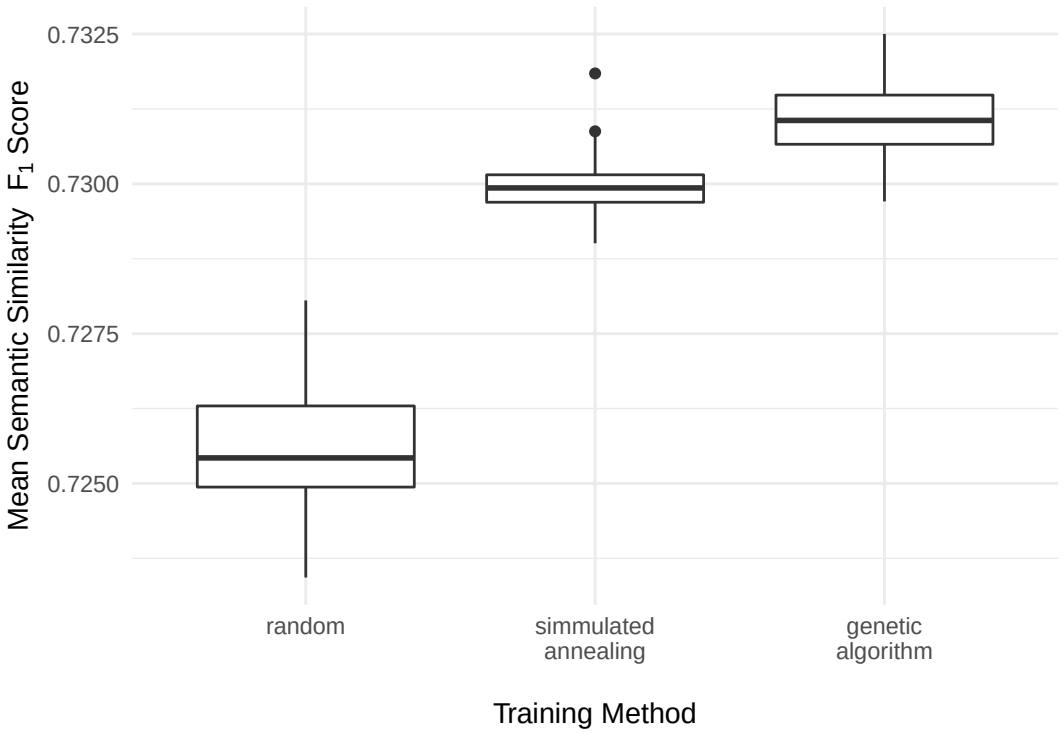

Figure 18: AHRD's Parameter Optimization Methods in Comparison
Here the training score (mean semantic similarity F_1-score over all training proteins) is plotted for three optimization methods of AHRD's parameters. We used our low redundancy ground truth set (section 5), gave each algorithm a budged of 5000 evaluation/optimization cycles and repeated the procedure 100 times. Simply picking the best of 5000 randomly drawn parameter sets shows the worst results and the greatest spread of score values. Comparing medians, the simulated annealing (section 6.3) improves the training score by only $\approx 0.6\%$. But the improvement is highly significant ($p < 2.2e^{-16}$). The improvement the genetic algorithm (section 7.4) can provide over the simulated annealing is even smaller but highly significant as well ($p < 2.2e^{-16}$).

18 AHRD's Placements in the CAFA3 Challenge

Table 12 lists the numbers and percentages of AHRD's best placements in the third installment of the CAFA challenge (section 8.1). AHRD's annotations resulted in more high placements when the evaluation was performed according to the F_{max} metric as opposed to the S_{min}. When the F_{max} is weighted by the information content of the GO terms (wF_{max}), AHRD's annotations were considered to be even better. The normalized S_{min} (nS_{min}) also resulted in a ranking that gave AHRD much more good placements.

The number of AHRD's placements combined for all metrics but with respect to four evaluation aspects (section 8.1) are shown in figure 19. AHRD's annotations for *E. coli* did fairly well, which also gave it many top 10 placements in the categories for all prokaryotes. Our annotations for *Arabidopsis* proteins also resulted in a high number of top 10 placements. Furthermore, AHRD received some top 10 placements for its annotation of human, mouse and *Drosophila* proteins. We were also sometimes in the top 10 of methods for all proteins. AHRD's placements were similarly divided among the three (sub-)ontologies of the GO and between "Limited Knowledge" and "No Knowledge" proteins. Most of our top 10 placements were achieved in the partial evaluation mode.

Table 12: AHRD's Best Placements in CAFA3
The CAFA3 challenge was evaluated in accordance to four different metrics and under various aspects (section 8.1). Each combination of the evaluation aspects is given as a separate evaluation category. Here we report the number of times AHRD was among the top 10 and in the upper half of the annotation method rankings of the different categories.

AHRD's annotations were evaluated more favorably by the F_{max} than the S_{min} metric. But AHRD was able to get into the top placements most often when the weighted version of F_{max} (wF_{max}) and the normalized version of S_{min} (nS_{min}) were used. When looking at the wF_{max} metric, for example, in 91% of the categories AHRD was better than half of the other annotation methods.

Metric	in the top 10	in the better half
F_{max}	25 (19.5%)	115 (89.8%)
S_{min}	5 (3.9%)	43 (33.6%)
wF_{max}	30 (23.4%)	117 (91.4%)
nS_{min}	33 (25.8%)	107 (83.6%)

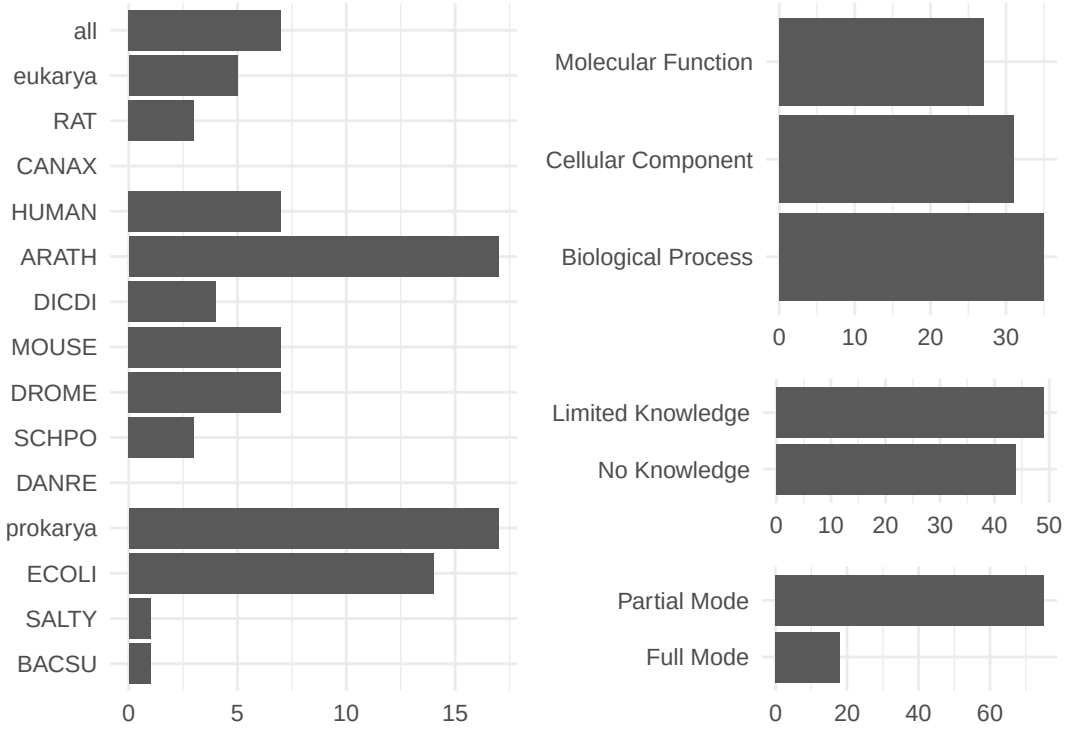

Figure 19: AHRD's Top 10 Placements in CAFA3
In the CAFA3 challenge the final evaluation was subdivided into multiple aspects (section 8.1). Each combination of these aspects is a separate evaluation category and has its own ranking for all methods that provided relevant annotations. Here we show the distribution of AHRD's top 10 placements under the various aspects and over all four metrics.

In all of the species in the evaluation, AHRD performed best in *Arabidopsis thaliana* (ARATH) and *Escherichia coli* K12 (ECOLI). Despite only one top 10 placement in *Salmonella typhimurium* (SALTY) and *Bacillus subtilis* (BACSU) AHRD got 17 top 10 placements in the combined evaluation for all three prokaryotes. AHRD achieved seven top 10 placements in human, mouse, *Drosophila melanogaster* (DROME) and in the combined evaluation over all species. No top 10 placements were achieved for *Candida albicans* (CANAX) and *Danio rerio* (DANRE).

AHRD's top 10 placements show a slight preference for "Biological Process" over the other two ontologies as well as a minor leaning towards proteins with "Limited Knowledge" as opposed to "No Knowledge" proteins.

AHRD's annotations were evaluated much more favorably in the "Partial Mode" in comparison to the "Full Mode".

Part IV
Discussion

19 A Non-Redundant Set of Ground Truth Proteins With Experimentally Verified GO Annotations Is Vital for Training and Testing AHRD

Question or hypothesis Is the ground truth set we created useful for the optimization and evaluation of protein function prediction methods?

Result summary In comparison to random expGOA proteins from Swiss-Prot the mean annotation performance and in some cases the annotation coverage are lowered significantly.

Critique of the method One step in the workflow to create the ground truth set (section 5) is to eliminate proteins with descriptions containing tokens in AHRD's blacklist. This could be seen as the introduction of a bias, making it easier for AHRD to annotate the remaining proteins. AHRD is in fact unable to reproduce such descriptions because it must not transfer descriptions containing words found in its blacklist. But as can be observed in figure 3 in section 9, the description annotation performance of AHRD and the "Best BLAST" methods are actually lower when the ground truth set is used compared to random proteins from expGOA proteins in Swiss-Prot. The same can be seen for the GO annotation performance. Admittedly the exact cause for the differences in the presented annotation scores is not elucidated here.

Contextualisation By lowering the number of proteins with similar descriptions and similar sequences we strive to minimize the effects of database biases on the tests the ground truth set is used in. These biases can be caused naturally, like the large number of copies of some genes in certain genomes (e.g. cytochrome P450 in *A. thaliana* [96]), or can have a technical reason because some protein families are just easier to experiment with than others.

A ground truth set, with a bias that is as low as possible, is important to enable unbiased testing, but also in order to generate a parameter set for AHRD that is not only useful for the application to certain proteins, but instead to all proteins found in typical proteomes. Even well-annotated proteomes of model organisms are far from free of these problems. This would also inevitably put AHRD's parameters into a taxonomic niche (the one which the model organism belongs to). In contrast, our ground truth set contains proteins of all branches of the tree of life and can thus facilitate generating parameter sets that are taxon agnostic.

Interpretation The lack of proteins that are easy to annotate leads to lower annotation scores of AHRD and "Best BLAST" methods for our ground truth set. We interpret this lack of "easy" proteins as a confirmation of the elimination of biases in the protein set. Using such an unbiased protein set for parameter

training facilitates a broad applicability of AHRD to whole proteomes of species from different taxa.

Conclusion The low redundancy in the ground truth set we generated is not only useful for the testing of protein annotation methods but also necessary for the training of an unbiased parameter set for AHRD. We perceive the performance scores derived from this set as a lower bound, most use cases will have less rare (and thereby difficult to annotate) proteins.

20 AHRD Performs the Most Nuanced and Consistent Evaluation When Facilitating the Semantic Similarity of Protein GO Annotations

Question or hypothesis What is the most appropriate method for the evaluation of protein GO predictions?

Result summary Of the three methods we implemented in this work the "SemSim GO score" showed the highest robustness to edge cases and the best adherence to the congruence of the biological functions in example proteins.

Critique of the method The rising complexity of the evaluation methods makes it increasingly difficult to follow how the computation arrives at the end result. This can be seen in many of the example proteins in section 10: The "simple GO score" is very straight forward, even in cases with many ground truth and prediction GO terms. The "ancestry GO score" first expands both ground truth and prediction with the GO terms ancestral GO terms. Ancestral terms that are found multiple times are counted only once. It is thus necessary to know the ancestries' overlap between ground truth terms or between multiple predictions terms to correctly arrive at the presented score values. The "SemSim GO score" makes it even harder because the maximum common information content of GO terms in the ground truth or the prediction is often needed to follow the calculation correctly.

Both the "ancestry GO score" and to a higher degree the "SemSim GO score" are subject to changes made in the Gene Ontology and in case of the "SemSim GO score" also the Swiss-Prot GO annotations. Scores for the same ground truth-prediction-pairs can very well be different after an update of AHRD's underlying GO database (built from freshly downloaded versions of the Gene Ontology and Swiss-Prot). In contrast, given a fixed pair of GO term sets the "simple GO score" will always yield the same result.

Contextualisation The reliance of the "SemSim GO score" on reference annotations to quantify the information content of a GO term makes it susceptible to biases in the database that is used [55]. As Swiss-Prot has been shown to

have biases in its GO annotations [49], this sometimes called "corpus bias" [97] is very likely to also be a relevant factor for the "SemSim GO score". Two affected GO terms are laid out in section 10.

Proteins can be multifunctional. These proteins have more than one catalytic function in a single amino acid chain [98]. Constance J. Jeffery coined the term "Moonlighting proteins" for these cases [99]. Many moonlighting proteins have been found, but so far, no common structural feature has been identified [100]. Consequently, a lot more are expected to be discovered in the future [101]. For this biological reason in particular and for plenty of other monetary and time constraint reasons in general, functional annotations of proteins can almost never be expected to be exhaustively complete. Hence GO annotations are considered subject to the "open world assumption" [51, 52]. The "open world assumption" implies an incomplete state of knowledge. So the absence of a GO annotation does not necessarily mean absence of the biological function in regard to the protein in question. This can lead to unfounded false positives when more comprehensive predictions are compared to a sparse ground truth of a test protein. So observed values for the precision are likely to be systematically underestimated and should be seen as lower bound. With the "SemSim GO score" we try to mitigate this effect, to an extent by excluding predicted GO terms from the evaluation if no knowledge is available about their respective ontology (i.e. BP, MF or CC) in the ground truth.

Proteins found in databases often have only very generic GO annotations close to the root of the ontology. This is known as the shallow annotation problem which has been shown to be handled incorrectly by some semantic similarity-based methods [97]. The "simple GO score" can have very inconsistent results when shallow annotations are compared. When ground truth and prediction share exactly matching shallow terms, the score will be close to 1.0 while slight mismatches will evaluate close to 0.0. Both the "ancestry" and "SemSim GO score" will give exact matches a very high score (close to 1.0) regardless of shallowness. But both metrics will regard close matches (e.g. two different children of the same term) as much more important when they happen deep in the ontology than when they are found close to the root. However, shallow annotations that merely share the root of the ontology are only recognized as completely dissimilar by the "SemSim GO score".

A factor that is relevant for all three GO annotation evaluation metrics: The balance between precision and recall in the calculation of the F-score is configurable with the beta-parameter. Values for beta smaller than 1.0 give more weight to the precision and less to recall. When the metric is used as objective function in the training of a GO prediction method this leads to sparser and shallower annotations in order to minimize the number of wrong annotations. Statistical analysis of whole proteome datasets (e.g. gene set enrichment analysis) requires low false positive rates. But there are also arguments conceivable for the opposite, a value for beta greater than 1.0. This leads to a greater weight for the recall and a smaller weight for the precision. Focusing on the recall favors the annotation of a greater number of terms per protein as well as terms that are less shallow. Predicting annotations as a direct source

of information for bench researchers who are interested in a single protein at a time means as many functional hints as possible are desired. AHRD pursues a balanced approach using a value for the beta parameter of 1.0. Thus, AHRD's annotations are useful from the level of a single protein up to a proteome-wide scale. This makes AHRD well-equipped for its main use case, the annotation of whole proteomes in genome projects.

Interpretation We recommend using the "SemSim GO score" because in our observation it shows the closest adherence to the similarity of the functions and the highest robustness to edge cases.

Conclusion Although the three examined GO evaluation methods often give very similar results when they differ, the "SemSim GO score" is the best approach as it shows a close adherence to functional congruence of GO terms, fine-grained score values and robustness to edge cases.

20.1 The "Simple GO Score" Is Only Useful for Rudimentary Evaluation

Question or hypothesis Can a direct GO ID overlap-based comparison of ground truth and prediction yield a useful annotation metric?

Result summary In 11 examples where the GO scoring methods disagreed, the "simple GO score" was completely wrong seven times (section 10).

Critique of the method Out of the three presented GO annotation scoring methods the "simple GO score" is by far the easiest to follow because only the presence and absence of exact GO ID matches are considered. The glaring issue with this approach is that it completely fails to account for the structure of the Gene Ontology. AHRD and other GO prediction tools can make more generalized predictions (i.e. predict GO terms closer to the root than the terms in the ground truth). With the "simple GO score" these will be considered false positives leading to an underestimation of the precision. Simultaneously the related ground truth terms will be treated as false negatives leading to an underestimation of the recall as well. An extreme example where the "simple GO score" completely fails to recognize any similarity between ground truth and recall is shown in table 3.

The comparatively small number of possible presence/absence occurrences leads the "simple GO score" to be often calculated from common small fractions. This leads to a quantization of the score values which can be observed as vertical bands in the scatter plots of score comparisons (panel A and B in figure 4).

Sets of GO terms found as annotations for database proteins often stem from previous transfer events in the process of high throughput electronic annotation. Because only exact matches are recognized by the "simple GO score", these kind of annotations are scored much higher than annotations close but not exactly the same as the ground truth. The further a fixed set of annotations

has automatically been propagated from database entry to database entry the less it might have in common with the true biological function of the protein. If used for the training of AHRD the "simple GO score" could lead to an undesired bias towards previously automatically transferred annotations thus limiting the performance of AHRD to that of previous tools.

The "simple GO score" has no mechanism to distinguish how specific GO terms are. In its extreme, this leads to edge cases where root terms (devoid of any information) in either ground truth or prediction cause completely inaccurate scores (see tables 1, 5 and 10).

Contextualization Sometimes no prior knowledge about a functional aspect of a protein is available. This means that the ground truth is empty for one the three subontologies. In this case, GO terms predicted for the particular subontology cannot be known to be incorrect ("open world assumption" [51, 52]). But, as can be seen in table 11, the "simple GO score" considers these predictions false positives, leading to an underestimation of the precision.

Interpretation In our view the "simple GO score" overlooks too much of the information encoded in the Gene Ontology. To make matters worse its susceptibility to edge cases can make the "simple GO score" inconsistent.

Conclusion The "simple GO score" is not a useful metric for the congruence of GO annotations.

20.2 The "Ancestry GO Score" Leverages the Topology of the Gene Ontology to Evaluate GO Annotations with Greater Nuance

Question or hypothesis Does taking the topology of the Gene Ontology into account (section 7.2.2) result in a better GO annotation performance metric?

Result summary In seven out of the 11 examples shown in section 10 the "ancestry GO score" results in misleading or completely wrong values.

Critique of the method Using the topology of the Gene Ontology inevitably necessitates additional computational effort that is associated with acquiring this information.

The calculation of "ancestry GO scores" is less intuitive to follow than the one of the "simple GO score", as some GO terms have large numbers of ancestral terms and the ancestries of GO Terms from the same ontology (BP, MF or CC) can overlap.

The number of false negatives (expansion of ground truth) and false positives (expansion of prediction) can increase as well as decrease but true positives can only increase. Thus, on average, the "ancestry GO F_1-scores" are higher in comparison to average "simple GO F_1-scores".

The higher numbers in the numerator and denominator of the fractions to calculate the recall and precision also result in more possible values for the metric. The lack of banding perpendicular to the plot axis for the "ancestry GO score" in panels A and C of figure 4 exemplifies this fact.

Because certain knowledge domains can be further developed than others, the level of detail in various parts of the Gene Ontology can vary drastically [102]. This can lead to uneven ancestry sizes for GO terms with similar specificity. So this can be a source of bias for the "ancestry GO score". GO terms can have a disproportionately large ancestry, giving them too much weight in comparison to other terms that might have the same importance to the function of a particular protein.

Root terms are treated equally to any other ancestral GO term. Their occurrence in the compared GO term sets can thus lead to score values drastically different from the perceived agreement between the annotations.

Contextualization A similar metric for the congruence of GO annotations coined simply "term overlap" has been shown to be a simple but effective alternative to methods using explicit information content calculations [97].

Interpretation Thanks to the recognition of GO term depth and more fine-grained values, in our view the "ancestry GO score" provides a clear improvement over the "simple GO score". But just like the "simple GO score" it also lacks consistency when edge cases are encountered.

Conclusion The "ancestry GO score" results in a better performance metric for GO predictions than the "simple GO score".

20.3 The "Semantic Similarity GO Score" Uses Topology in Conjunction With Annotation Frequency to Consistently Evaluate GO Annotations Without Susceptibility to Edge Cases

Question or hypothesis Does the addition of annotation frequency to the estimation of the information content of GO terms facilitate a better GO prediction performance evaluation?

Result summary In all of the 11 example comparisons in section 10, the "SemSim GO score" exhibits score values accurately reflecting the evident congruence of ground truth and prediction.

Critique of the method and contextualization The "SemSim GO score" is even harder to follow than the "ancestry GO score". The added complexity of calculating the information content of GO terms and then finding the maximum common information content between GO terms is at fault.

In addition to parsing the Gene Ontology, using "SemSim GO scores" also necessitates reading in the Swiss-Prot database. This adds computational effort that has to be done once in order to precompute the information content of each GO term (section 7.1).

GO evaluation methods based on information content derived from a set of reference annotations can exhibit a so-called "corpus bias" [55]. Shallow terms rarely encountered in the annotation corpus can exhibit an inflated information content. The opposite can be true for protein functions deep in the ontology but over-represented in the corpus.

The "SemSim GO score" handles edge cases more correctly. GO root terms have an information content of 0.0 and are thus barred entirely from influencing the score. This can be substantial in the case of very sparse annotations. If no prior knowledge is available in a particular ontology (BP, MF or CC), predictions for this ontology are not counted as false positives. This is a step into the right direction to account for the "open world assumption" GO annotations are subject to [51, 52].

Using the maximum common information content of nodes in a taxonomy to determine their semantic similarity was first proposed by Philip Resnik [103].

Interpretation Disagreement between the "ancestry GO score" and the "SemSim GO score" often cannot clearly be sorted out, but in our assessment the "SemSim GO score" shows a clear advantage over both other methods when edge cases are encountered.

Conclusion Among the here covered methods the "SemSim GO score" is the most accurate approach to the evaluation of protein GO annotations.

21 Parameter Optimization Is Necessary for Optimal Performance

Question or hypothesis Is the optimization of AHRD's parameters necessary and if so, which optimization method is the best?

Result summary Parameter optimization increases AHRD's annotation performance for human readable descriptions as well as GO annotation significantly. By a slight margin the best optimization results are achieved with the genetic algorithm.

Critique of the method Figure 17 in section 17 shows how much AHRD benefits from the optimization of its parameters. Naive parameters, which lead to an equal weighting of all indicators AHRD uses to rank the reference candidate proteins, already accomplish useful annotations. Surprisingly, the manual customization of the parameters to put emphasis on criteria we deemed as more important did not lead to a worthwhile improvement. Parameter optimization on the other hand, proved itself to be beneficial by providing a significant uplift

to the annotation performance for descriptions as well as GO terms. Because we used separate sets of proteins for training and testing we can be confident that the observed improvement is not due to overfitting.

Our comparison of the parameter optimization procedures that AHRD has to offer is shown in figure 18 of section 17. Considering the small difference that can be observed between the two algorithms' performance no clear winner can be identified.

Variation between training results based on the split of the protein ground truth into training and test set are shown in figure 5. Additionally, the size of the boxplots and the shape of the violinplots convey the distribution of the performance based on the variation caused by the randomness inherent to the genetic algorithm. The far greater variation between the splits is most likely caused by the uneven distribution of easy and hard to annotate proteins. This shows that the training is close to the limit of what can be achieved considering the given quality indicators of candidate annotations.

Contextualization In machine learning, parameters that are not optimized by the training but control the learning process itself are known as hyperparameters [104]. For example, both of AHRD's optimization algorithms are influenced by the mean and the standard deviation used for the mutation of parameters. The key factors for the simulated annealing approach are the stating temperature, the cool down in each step and the probability to accept worse parameter sets. The genetic algorithms most important hyperparameters are the size of the parameter set population and the number of generations to be evaluated. A complete list of configurable options for AHRD's trainers can be found in supplemental section 28.3.5. Both of AHRD's machine learning methods were run with intuitively chosen hyperparameters that are likely to provide additional performance if investigated further.

Regardless of the optimization approach, the most important hyperparameter is always the parameter space in which the optimization takes place. So probably the primary way to increase performance further is to add more factors to the prediction algorithm to facilitate additional differentiation of correct annotations from wrong ones. See sections 12 and 13 for results and sections 22 and 23 for a discussion about our effort to do exactly that. There it was vital to control for the variation present in our testing setup demonstrated in figure 5.

Interpretation We see the optimization of AHRD's parameters as an essential step to achieve the best possible annotation performance. When one does so, it does not matter which of the two methods we implemented is used.

Conclusion Optimization of AHRD's parameters is necessary and can be performed by simulated annealing just as well as a genetic algorithm.

22 The Informative Token Threshold Is Only Useful for the Annotation With Descriptions but not With GO Terms

Question or hypothesis Is the informative token threshold a useful mechanic to increase AHRD's performance?

Result summary Outside of what has been observed for description annotation, when optimizing for GO annotations, parameter optimization always resulted in very low threshold values.

Critique of the method AHRD splits the descriptions of the candidate annotations that are found in the reference databases into their tokens. Each token is assigned a score (equation 2 in section 6.1.2) to model how informative it is. This token score depends on the token's abundance in descriptions that are found in trusted databases and that have high bit scores as well as good alignment overlaps. Previously AHRD used 50% of the score that was assigned to the most informative token, to distinguish informative tokens from uninformative ones. When we made this threshold variable and included it in the parameter optimization (section 7.4) it was set to higher values (between 0.7 and 0.98) but increased the description annotation performance only insignificantly (section 12.1). When applied to the optimization of AHRD's GO annotation (section 12.2), a significant increase of the performance was achieved but it was set to either 1.0 or very low values (< 0.037). Both values have the same effect: Practically all tokens are treated as equally informative.

When the scoring of candidate reference proteins is based on their GO annotations instead of their descriptions, the new threshold is switched from differentiating tokens to differentiating informative from uninformative GO terms (equation 31 in section 7.3). Then, in every case the variable informative token threshold was combined with various other new parameters, it led to a significant advantage over all variants with a fixed threshold (figure 8 in section 13). This was always achieved by a significant increase to the recall at a cost to the precision (figures 9 and 10 in section 13). The highest value the informative token threshold was set to in all these different optimization tests was ≈ 0.066. So in effect, the training of the GO annotation always disabled the differentiation of informative from uninformative GO terms in order to increase the recall. But the distribution of GO term scores is unknown. Perhaps there is a significant number of GO terms that, even at the observed threshold values, still fall under it. So to confirm our findings for the informative token threshold in the GO term prediction, the tests can be repeated with fixed values of either 0.0 or 1.0.

Interpretation The differentiation of informative from uninformative tokens is useful for the annotation of descriptions. Optimizing it to other values than 50% yields only insignificant improvements.

The extreme threshold values obtained by optimizing for GO annotation performance show that the distinction of informative GO terms is more of a hindrance than an advantage.

Conclusion The informative token threshold is only necessary for the annotation of human readable descriptions.

23 The Separate GO Prediction Algorithm Improves AHRD's GO Annotation Performance

Question or hypothesis What is the best version of AHRD for the prediction of GO annotations? Different versions of AHRD lead to different rankings of the homologous candidate proteins. Is the same ranking equally useful for the annotation of descriptions and GO terms or should different rankings be used?

Result summary Using separate algorithms for descriptions and GO annotations permits a significant increase in annotation performance.

Critique of the method Although the improvements have been shown to be statistically significant (figure 8 to 10 in section 13), overall the increase in annotation performance is small.

Figure 3 in section 9 shows the theoretical maximum possible annotation performance AHRD could achieve if it always found the ideal ranking of candidate proteins. It is unclear whether information about the candidate proteins in the databases is still to be leveraged and could make a better ranking possible. But by the difference between AHRD's performance and the "Max Attainable" performance it is clear that even our best prediction algorithm still leaves performance on the table (the version used for figure 3 is functionally closest to `itt gtas ics -` shown in figures 8 – 10).

Using a separate GO term-based algorithm to predict GO annotations does not increase AHRD's performance on its own. The variable informative token threshold also plays a bigger role at first (see below). Only the addition of the information content score and especially the addition of the evidence code score enable the separate GO-based method to prevail. These two metrics are derived at a per-GO term basis and thus require the separate GO term-based method.

The information content score favors reference annotations with GO terms that have a high information content (equation 21). The GO term information content is central to the subsequent evaluation of the annotations with the "SemSim GO score" (section 7.2.3). So this can be interpreted as a "self fulfilling prophecy" situation. When predicted high information content GO terms are matched in the ground truth, the recall will be higher. But on the other hand mismatches will lead to a decline of the precision when the information content of the compared GO terms is higher. In figures 9 and 10 this exact

behavior can be observed every time the information content score is added to the prediction algorithm. So the actual usefulness of the information content score remains to be investigated further.

Contextualization As discussed in the previous section (22), making the informative token threshold variable and including it in the parameter set for optimization showed that the differentiation of informative from uninformative GO terms hindered the annotation in the first place. So this means the informative token threshold is not a useful parameter but at least gave us the evidence to exclude the step in the annotation algorithm in the future.

Interpretation The best version of AHRD to predict GO annotations is one that uses a separate algorithm for GO terms, has the informative token threshold set to either 0.0 or 1.0 and uses the information content score as well as the evidence code score. It results in a ranking for the candidate reference annotations that is better suited to GO term prediction. In our view the additional GO prediction performance justifies using this separate GO term-based annotation method.

Conclusion Using a newer version of AHRD with a separate GO term-based approach is the recommended GO annotation method.

24 Annotation With a New Set of GO Terms Mixed From Multiple Reference Proteins Is Not a Viable Strategy

Question or hypothesis Can AHRD's GO prediction performance be increased by creating new combinations of GO terms from the candidate reference proteins?

Result summary The direct annotation of GO terms significantly increases recall but at such a drastic cost to precision that the F_1-score is lowered severely.

Critique of the method The mean score per training/test-set split is plotted in figure 11. The spread of these means is slightly higher along the y-axis (for the new approach tested here) than along the x-axis (for the conventional approach). The higher variation of these mean scores hints at higher differences between the test protein scores within the sets. So the performance of the newly created GO term sets seems to be less stable (i.e. very good in some cases but equally as bad in just as many other cases).

The most important parameter responsible for the performance of the approach tested here is the informative token threshold. It has been reused once again and directly controls which GO terms are annotated and which are not depending on their GO term score. It is used relative to the score of the top

scoring GO term and in effect controls how lax or strict the prediction is made. Lower threshold values lead to more GO terms and higher values lead to fewer GO terms annotated. Annotating more GO terms generally entails a higher recall at a cost to the precision and vice versa. So when the genetic algorithm optimized the informative token threshold, it had to find the optimal value by balancing recall and precision much more directly than with any other parameter. This has to be kept in mind to understand that the strong increase and decrease of recall and precision depicted in figure 11 still shows the optimal performing parameters for this new annotation approach. This is a strong sign for the existence of typical patterns in the GO annotations of the UniProtKB. Instead of propagating these patterns further the approach tested here creates new GO term sets that just do not fit the patterns in the ground truth annotations.

Contextualization Another probable reason for the poor performance could be the combination of multiple distinct functions through the creation of these new annotations. A ground truth protein with multiple functions is unlikely to have all functions accurately annotated because these so-called "Moonlighting proteins" [99] are common but often undetected. But their sequence can show homology to multiple other, shorter proteins where this is the only function and has thus been accurately annotated. When the approach discussed here gathers the evidence for various GO terms found in the homology search results, it will combine these functions. But it will seemingly fail when these are compared to the incomplete annotation of the ground truth. So this problem is related to the "open world assumption" [51, 52] GO annotations are subject to. Protein functions not known to be absent from a protein should not be considered as such only because their corresponding GO terms have not been annotated (yet). In the future, ground truth sets with more complete annotations could make this new approach more viable.

Interpretation Deliberations about the reasons and possible solutions for the lower F_1-scores put aside, the recombination of GO terms from multiple candidate proteins is currently just not a viable alternative to the conventional GO annotation procedure of AHRD. This could change if data on functions shown not to occur in proteins become available (enabling the evaluation of true negatives).

Conclusion As it does not lead to increased performance, we do not recommend to use this annotation method.

25 AHRD Delivers Both Broad Coverage and High Quality When Annotating Functionally Diverse Proteins With Descriptions and GO Terms

Question or hypothesis Can AHRD offer better predictions for descriptions and GO terms and can AHRD cover a greater range of proteins than its competitors?

Result summary Compared to competitors, on a ground truth set, AHRD annotated more proteins with perfect scoring descriptions and overall more proteins with descriptions (figures 12 and 13). On the ground truth set AHRD also annotated many proteins with perfect scoring GO annotations while covering a big part of it too (figures 14 and 15). In a typical use case scenario AHRD increased the annotation coverage of two proteomes (figure 16).

Critique of the method We were able to reveal the differences between AHRD and its competitors because of our ground truth set (section 19). By removing redundant proteins from the comparison, a greater variety of proteins that can be difficult to annotate can be used for comparison.

A git commit of the "sem_sim_go_OWL" branch was used to record the annotation performance of AHRD. This version of AHRD parses the Gene Ontology in the OWL format to calculate term information content (section 7.1) and is thus able to train and evaluate AHRD in accordance to the semantic similarity of GO annotations (section 7.2.3). However, when annotating, it lacks the ability to rank the candidate reference proteins based on anything other than their descriptions. So the improvements (section 13) achieved by performing the GO term prediction based on the candidate protein's GO annotations instead (section 7.3.1), are not yet factored in here. Thus, for the GO annotation performance, there would likely be a bigger difference to the competitors if this comparison was repeated using an AHRD git commit from the "separate_go_prediction" (section 7.3) branch.

The line plots (figures 12 and 14) show the evaluation scores for the same 1000 proteins ordered by their numerical values. It is important to keep in mind that the order is consequently different for each competitor. If two competitors had completely complementary strengths and weaknesses, so that the incorrectly annotated proteins of the first were to be perfectly annotated by the other and visa versa, these plots would still show them as roughly equal. It is better to consider the line plots presented here as 90° rotated cumulative histograms. The amount of proteins in 10 bins of increasing score difference shown in figures 13 and 15 is better suited to convey differences between AHRD and the competing methods.

AHRD is meant to be used in an automated fashion as part of genome projects. Consequently, it is fast and easy to set up and can handle even the largest proteomes (e.g. *Hordeum vulgare*). But the same cannot be said about

many of AHRD's competitors. So we are aware that there are many more methods but we restricted ourselves to the selection presented here because many competitors only offer a web interface limited to a small number of proteins or the effort of installing and running them was forbidding. We used the proteomes of barley (*Hordeum vulgare*) and powdered mildew (*Blumeria graminis*) as typical use cases of AHRD covering two eukaryotic kingdoms (Plantae and Fungi). To demonstrate the universal applicability of AHRD, more different proteomes of eukaryotic and prokaryotic organisms should be tested.

The annotations publicly available for the two use case proteomes cannot be considered free of natural and technical biases. The available evidence for these annotations can be seen as roughly equivalent to the evidence available to AHRD. Consequently, they should not be used as a ground truth and thus we refrained from calculating and presenting F_1-scores.

AHRD's coverage of a typical proteome exceeds the "Best BLAST" methods as it integrates the information from both databases. A rather simple method using "Best BLAST TrEMBL" as fallback if "Best BLAST Swiss-Prot" cannot deliver will achieve the same coverage as AHRD. But the qualitative differences to "Best BLAST Swiss-Prot" demonstrated on the non-redundant ground truth set show that AHRD's annotations are generally better than such an approach.

Contextualization NetGO [76] is the successor to GOLabeler [105], the de facto winner of the third CAFA challenge [77]. It is the only method that achieves a higher coverage on our test set than AHRD. But it does so at great cost to the quality of annotations. Much fewer proteins are predicted with GO annotations that achieve scores greater than 0.75. This can also be seen by the majority of score differences that favor AHRD in figure 15. But there the few proteins that AHRD was not able to annotate at all, can be seen as exclusives at the NetGO side. In theory, in these cases AHRD's predictions could be complemented by the ones from NetGO to achieve an even better predictor than each on their own.

Interpretation Used on the ground truth set AHRD shows superiority when it comes to the annotation of descriptions. Not all GO terms AHRD predicts are always better than what can be obtained with competitors. But AHRD is often closer to the GO ground truth overall. Simultaneously AHRD can maintain a high annotation coverage on many different protein classes and whole proteomes.

Conclusion Based on our evaluation, using a "difficult" test set covering a wide range of functions, we find that AHRD is able to predict descriptions and GO annotations better and at a greater coverage than all competitors except NetGO. While NetGO achieves greater coverage, we find the reduced accuracy to be unacceptable.

26 AHRD Can Also Keep up With the Competition in a Very Different Evaluation, the CAFA3 Challenge

Question or hypothesis Can AHRD also succeed in a completely different test — the CAFA3 challenge [77]?

Result summary AHRD got into the top 10 of many evaluation categories (figure 19) and was often better than half of the other contestants (table 12).

Critique of the method Gauging AHRD's performance in the CAFA challenge by the number of its top 10 placements in various categories is problematic in the sense that the categories all have a different number of placements. This is due to the fact that not all methods participated in every category. The smallest number of participants was 46 and the highest 67. Of course a top 10 placement in the latter is a greater achievement than in the former. Unfortunately the rules of the challenge allow the participants to get detailed data only about their own method's performance. So we cannot compare AHRD's placements to other contestants directly.

AHRD did particularly well in *A. thaliana* (n=17) and *E. coli* (n=14). The parameters we used for AHRD to make the CAFA3 predictions were obtained by optimization on our non-redundant, species agnostic ground truth set. So we did not expect AHRD to show such a preference for particular species. But this apparent species bias might also be explained by an uneven addition of new experimental annotations. Research into particular protein families that are easy for AHRD to annotate correctly might be the source.

Of the three GO knowledge domains, AHRD had the most top 10 placements in "Biological Process" (n=35), closely followed by "Cellular Component" (n=31) and "Molecular Function" (n=27). So no clear preference for the knowledge domains can be discerned. There is also no clear difference between "Limited Knowledge" (n=49) and "No Knowledge" (n=44). Both of these findings can be seen as testaments to AHRD's broad applicability.

The number of top 10 placements in "Partial Mode" (n=75) was much higher than in "Full Mode" (n=18). Although AHRD generally offers a good coverage (figures 14 and 16), as figures 7, 9, 10 and 17 show, its precision is consistently higher than its recall. This is likely the reason for its strength in CAFA's "Partial Mode" and its shortcomings in "Full Mode".

When divided by evaluation metric (table 12), AHRD shows the most consistent placement in the upper half in the case of the weighted F_{max} (91% of categories). The weighted F_{max} takes the information content of the GO terms in consideration. We also use term information content in the calculation of the "SemSim GO score" (section 7.2.3), which was used as objective function for the optimization of AHRD's parameters prior to the CAFA predictions. So it makes sense for AHRD to excel here. But the S_{min} metric is also based on semantic similarity (minimization of semantic dissimilarity to be precise) and AHRD can

only get a placement in the upper half of the field in one third of the categories. The S_{min} metric is not weighted. So a few high information content GO terms that have been miss-predicted or are erroneously missing from the prediction can have a proportionally high effect on the score. And at the same time many low information content terms that have been predicted correctly can have a comparatively small effect. The nS_{min}, a normalized version of S_{min} brings the score to a 0 to 1 scale for every protein and thus avoids these problems. This, again, is similar to the way the "SemSim GO score" (section 7.2.3) used to optimize AHRD is calculated, which is likely the reason AHRD achieves the most of its top 10 placements measured with this metric and a sizable number of placements in the upper half of categories (84%). So AHRD is not focused on the annotation of a few high value targets but on correct annotation of a great variety of proteins.

Contextualization The motivation for AHRD's participation in the CAFA3 challenge was the comparison to our own test procedure (section 7.6.1). CAFA does not have some of the shortcomings of our evaluation method (section 7.6.1) that is based on our non-redundant ground truth set (section 5). One of these shortcomings is the potential occurrence of annotations that have been present in the UniProtKB in the past and have "percolated" [6] through the database as a fixed set of annotations to various related proteins. But the annotations that are evaluated in CAFA have their own biases. For one, the characterization of completely unknown proteins is rarely the focus of research efforts. It is more likely for existing computational annotations to be verified and then upgraded to the "experimental" status.

The top contender in CAFA3 [77], GOLabeler [105], is the predecessor of NetGO [76]. NetGO is included in our evaluation in figures 14 and 15 of section 15. Although there it showed just as good of a coverage as it did in the CAFA3 challenge, it was not able to predict many proteins with very high F_1-scores. This might be a sign that their approach profits from the above-discussed potential biases in the CAFA annotations. With GO term frequencies and sequence alignments NetGO uses some of same information AHRD also relies on. Furthermore, NetGO uses the frequency of amino acid trigrams, InterPro [11] features (domains and motifs), ProFET [106] features (biophysical properties) and the STRING database [43] (protein interactions). It is a conscious decision to not incorporate these or similar features into AHRD. For a great number of users that want to perform annotations on a genomic scale, the additional data that has to be retrieved and the computational time that has to be invested easily leads to an inability to use complex methods like these.

For the term-centered annotations in CAFA-π we were not able to receive any results for AHRD. In the case of the GO term "cilium or flagellum-dependent cell motility" (GO:0001539) in *Pseudomonas aeruginosa* some methods (especially the aforementioned GOLabler) were able to outperform naive BLAST-based methods, but only by a small margin. The CAFA organizers also included annotations based on data integrated from a previously published compendium of expression analysis [107]. This method, which is to be used as baseline if the

challenge is repeated, was able to outperform all regular contestant methods.

Interpretation AHRD is not a top performer in all categories because the CAFA challenge focuses more on single proteins and we decided to optimize AHRD for broad applicability. But we remain competitive in all areas despite the fact that AHRD was optimized with our generalized training set and the fact that it is comparatively easy to install and execute. So as other methods become easier to run, AHRD remains to be challenged as well.

Conclusion In the completely different evaluation method that is the CAFA challenge AHRD was able to show satisfactory results.

Part V
Appendix

27 List of Abbreviations

AHRD	Automated Assignment of Human Readable Descriptions
API	Application Programming Interface
BBsprot	"Best BLAST Swiss-Prot"
BBtrembl	"Best BLAST TrEMBL"
BLAST	Basic Local Alignment Search Tool [61]
BLOSUM	Blocks Substitution Matrix [21]
BPO	Biological Process Ontology (GO subontology)
CAFA	Critical Assessment of Functional Annotation [91, 92, 77]
CCO	Cellular Component Ontology (GO subontology)
CPU	Central Processing Unit
DAG	Directed Acyclic Graph
DB	Database
DNA	Deoxyribonucleic acid
ecs	evidence code score (equation 28)
expGOA	experimentally verified GOA
f.sp.	forma specialis
FASTA	FAST-All (Works with any alphabet rather than only proteins = FASTP or only nucleotides = FASTN) [22, 23]
GNU	"GNU's not Unix!"
GO	Gene Ontology [8, 71]
GOA	GO Annotation
gtas	go term abundancy score (equation 26)
HRD	Human Readable Description
HSP	High-scoring Segment Pair
ics	information content score (equation 27)
ID	Identifier
IDE	Integrated Development Environment
IQR	Interquantile Range
itt	informative token threshold (equations 3 – 5 and 29 – 31)
JRE	Java Runtime Environment
KEGG	Kyoto Encyclopedia of Genes and Genomes [13]
MFO	Molecular Function Ontology (GO subontology)
NaN	Not a Number
OBO	Open Biomedical Ontologies
OWL	Web Ontology Language
PAM	Point Accepted Mutation [20]
RNA	Ribonucleic acid
SemSim	Semantic Similarity
SHA-1	Secure Hash Algorithm 1
subsp.	subspecies
TAIR	The Arabidopsis Information Resource [70]
TSV	Tabulator Separated Values
UniProtKB	UniProt Knowledgebase [4]
YML (YAML)	"YAML Ain't Markup Language"

28 Supplement

28.1 Variation in GO Prediction Performance Due to Splitting and Training

Table 13: Variation in GO Prediction Performance Due to Splitting and Training
This table is meant to supplement figure 5 which is based on the same data. The ground truth set "nrSpotExpGOAv2" (section 5) was split into a training and test set 10 times. To bootstrap a distribution of the mean evaluation scores 1000 evaluations where performed for each sample. The "random" sample is based on sampling of 1000 proteins from a pool of predictions from all 10 splits. A Shapiro-Wilk test of normality [95] (null hypothesis: data is normal distributed) was performed on each split. A Fligner-Killeen test [94] (non-parametric test for homogeneity of variance that is robust against non-normality; null hypothesis: variance is homogeneous between groups) was performed comparing each split with the random sample.

Split	Min	First Quantile	Median	Mean	Third Quantile	Max	IQR	Standard Deviation	Shapiro p-value	Fligner p-value
random	0.670	0.688	0.693	0.693	0.697	0.711	0.00884	0.00675	0.157	NA
1	0.681	0.685	0.686	0.686	0.686	0.689	0.00178	0.00116	7.87E-11	2.51E-181
2	0.693	0.696	0.697	0.697	0.698	0.700	0.00161	0.00096	2.61E-19	8.81E-195
3	0.700	0.703	0.704	0.704	0.704	0.708	0.00142	0.00114	2.62E-17	2.98E-184
4	0.696	0.701	0.701	0.701	0.702	0.707	0.00137	0.00146	1.69E-18	3.25E-168
5	0.689	0.691	0.692	0.692	0.694	0.700	0.00305	0.00171	2.08E-18	2.85E-134
6	0.700	0.701	0.702	0.702	0.703	0.710	0.00155	0.00125	1.93E-26	1.43E-178
7	0.695	0.704	0.705	0.704	0.706	0.711	0.00192	0.00275	2.70E-28	2.70E-106
8	0.696	0.697	0.699	0.698	0.700	0.705	0.00267	0.00141	8.48E-23	5.70E-161
9	0.697	0.699	0.699	0.699	0.699	0.702	0.00037	0.00056	1.10E-31	4.40E-236
10	0.690	0.694	0.695	0.695	0.695	0.698	0.00115	0.00107	2.89E-25	5.49E-191

28.2 AHRD's Default Blacklists and Filter

28.2.1 Description Line Blacklist

Reference proteins with matching description lines are excluded from the annotation process. "`(?i)`" ensures case insensitive matching, "`^`" matches the start of the description line and "`\s+`" matches one or more white space characters.

```
(?i)^similar\s+to
(?i)^probable
(?i)^putative
(?i)^predicted
(?i)^uncharacterized
(?i)^unknown
(?i)^hypothetical
(?i)^unnamed
(?i)^whole\s+genome\s+shotgun\s+sequence
(?i)^clone
```

28.2.2 Description Line Filter

Elements of the description line matched with any of the regular expressions in the filter are replaced with whitespace. Elements that are filtered include sequence-specific information typical to UniProtKB entries such as organism name, organism identifier, gene name, protein existence and sequence version (for example: `OS=Oryza sativa OX=4530 GN=H0502B11.9 PE=4 SV=1`). Additionally, InterPro accessions [11], the word "Fragment" and many special characters are also filtered.

```
\sOS=.*$
(?i)OS.*[.].*protein
(?i)^H0.*protein
(?i)contains.*
IPR.*
\w{2,}\d{1,2}(g|G)\d+(\.\d)*\s+
\b\[.*
\b\S+\|\S+\|\S+
\(\s*Fragment\s*\)
^(\s|/|\(|\)|-|\+|\*|,|;|\.|\:|\||\d)+$
```

28.2.3 Token Blacklist

Tokens with a match in the token blacklist are excluded from the token scoring process and thus do not influence the decision process about which reference protein is used for the annotation transfer. Our evaluation procedure of human readable descriptions also ignores tokens with a match in the token blacklist. "`(?i)`" ensures case insensitive matching, "`\b`" matches a word boundary, "`\w?`" matches a word character zero or one times and `\d+` matches a digit one ore more times.

```
(?i)\bunknown\b
(?i)\bmember\b
(?i)\blike\b
(?i)\bassociated\b
(?i)\bcontaining\b
(?i)\bactivated\b
(?i)\bfamily\b
(?i)\bsubfamily\b
(?i)\binteracting\b
(?i)\bactivity\b
(?i)\bsimilar\b
(?i)\bproduct\b
(?i)\bexpressed\b
(?i)\bpredicted\b
(?i)\bputative\b
(?i)\buncharacterized\b
(?i)\bprobable\b
(?i)\bprotein\b
(?i)\bgene\b
```

```
(?i)\btair\b
(?i)\bfragment\b
(?i)\bhomolog\b
(?i)\bcontig\b
(?i)\brelated\b
(?i)\bremark\b
(?i)\b\w?orf(\w?|\d+)\b
```

28.3 AHRD's Settings

28.3.1 Parameters

`token_score_bit_score_weight`

floating-point	mandatory	Weight of the candidate protein's alignment bit score in the calculation of the token score (equation 2) for descriptions or the go term abundancy score (equation 26) for GO terms; Between 0.0 and 1.0; Must add up to 1.0 with the other token score weights

`token_score_database_score_weight`

floating-point	mandatory	Weight of the candidate protein's database score in the calculation of the token score (equation 2) for descriptions or the go term abundancy score (equation 26) for GO terms; Between 0.0 and 1.0; Must add up to 1.0 with the other token score weights

`token_score_overlap_score_weight`

floating-point	mandatory	Weight of the candidate protein's alignment overlap score in the calculation of the token score (equation 2) for descriptions or the go term abundancy score (equation 26) for GO terms; between 0.0 and 1.0; must add up to 1.0 with the other token score weights

`weight`

integer	mandatory	The database weight; Subordinate of entries in `blast_dbs`; Greater than 0

`description_score_bit_score_weight`

floating-point	mandatory	Weight of the candidate protein's relative bit score in the calculation of the description score (equation 7); Subordinate of entries in `blast_dbs`); between 0.0 and 1.0

`informative_token_threshold`
floating-point optional Threshold for the differentiation of informative from uninformative tokens or GO terms (equations 3 – 5 and 29 – 31); Introduced in AHRD commit a5bccd1; Between 0.0 and 1.0; Default = 0.5

`go_term_score_information_content_weight`
floating-point optional Weight for the information content score (equation 27) in the calculation of the GO term score (equation 25); Introduced in AHRD commit 0a5fe1b; Between 0.0 and 1.0; Default = 0.5

`go_term_score_evidence_code_weight`
floating-point optional Weight for the evidence code score (equation 28) in the calculation of the GO term score (equation 25); Introduced in AHRD commit 175ce62; Between 0.0 and 1.0; Default = 0.5

28.3.2 General Input Settings

`proteins_fasta`
string mandatory Path to FASTA file with the amino acid sequences of the query proteins

`proteins_fasta_regex`
string optional Regular expression to extract protein accessions from proteins FASTA description lines

`blast_dbs`
string(s) mandatory Names of the databases that sequence similarity search results are to be used from

`file`
string mandatory Path to file with sequence similarity search results; Subordinate of entries in `blast_dbs`

`database`
string mandatory Path to FASTA file with the amino acid sequences of the database; Subordinate of entries in `blast_dbs`

`fasta_header_regex`
string optional Regular expression to extract the protein accession and description from the sequence description lines of the `database`-file; Subordinate of entries in `blast_dbs`

`blacklist`
string optional Path to file with regular expressions for description line blacklisting; Facultative subordinate of entries in `blast_dbs`; See supplement 28.2.1

`filter`
string optional Path to file with regular expression to filtering uninformative contents from description lines; Facultative subordinate of entries in `blast_dbs`; See supplement 28.2.2

`token_blacklist`
string optional Path to file with regular expressions to match tokens ignored during candidate description scoring; Facultative subordinate of entries in `blast_dbs`; See supplement 28.2.3

`gene_ontology_reference`
string optional Path to file with reference GO annotations; Necessary for GO term annotation; Subordinate of entries in `blast_dbs`

`gene_ontology_reference_regex`
string optional Regular expression to extract the protein accession and GO term accession from lines in the `gene_ontology_reference`-File; Subordinate of entries in `blast_dbs`

`short_accession_regex`
string optional Regular expression to extract the significant part of protein accessions at various points in the execution of AHRD

`prefer_reference_with_go_annos`
Boolean optional If, in addition to descriptions, GO terms are to be annotated, candidate proteins without GO annotations will not be considered; Default = `true`

`go_slim`
string optional Path to an OBO-File with a set of GO slim terms which will be annotated and written to the output; Section 7.3.2

28.3.3 General Output Settings

`output`
string mandatory Path to TSV file that AHRD writes its output to

`output_fasta`
Boolean optional Switches AHRD's output from the TSV to the FASTA format; default = `false`

28.3.4 Evaluation Settings

`ground_truth_fasta`
string mandatory Path to amino acid FASTA file with the same accessions as `proteins_fasta` and the ground truth descriptions in the sequence description lines

`ground_truth_fasta_regex`
string optional Regular expression to extract the protein accessions and descriptions from the sequence description lines in the `ground_truth_fasta`

`ground_truth_go_annotations`
string optional Path to TSV file with the ground truth GO annotations in two columns (protein accession and GO term ID)

`f_measure_beta_parameter`
floating-point optional Weighting parameter between precision and recall; Greater that 0.0; Default = 1.0

`write_best_blast_hits_to_output`
Boolean optional Evaluate the sequence similarity search result with the highest bit score for each database and write them to the output file; Default = `false`

`find_highest_possible_evaluation_score`
Boolean optional Find the candidate protein resulting in the highest evaluation score and add its description with the corresponding score to the output; Default = `false`

`evaluate_only_valid_tokens`
Boolean optional Ignore ground truth tokens that have a match in the token blacklist; Default = `true`

`simple_GO_f1_scores`
Boolean optional If GO terms are to be annotated calculate set overlap-based GO scores (section 7.2.1) and add them to the evaluation output; Default = `true`

`ancestry_GO_f1_scores`
Boolean optional If GO terms are to be annotated calculate GO term ancestry-based GO scores (section 7.2.2) and add them to the evaluation output; Default = `false`

`semsim_GO_f1_scores`
Boolean optional If GO terms are to be annotated calculate semantic similarity-based GO scores (section 7.2.3) and add them to the evaluation output; Default = `false`

`competitors`
string(s) optional Names of competitors to evaluate alongside AHRD

`descriptions`
string optional Path to a TSV-file with protein accessions and descriptions; Subordinate of entries in `competitors`

`go_annotations`
string optional Path to a TSV-file with protein accessions and GO terms Subordinate of entries in `competitors`

`find_highest_possible_go_score`
Boolean optional Find the candidate protein resulting in the highest GO F_1-score and add the corresponding score to the output; Default = `false`

`find_highest_possible_precision`
Boolean optional Find the candidate protein resulting in the highest GO precision and add the corresponding value to the output; Default = `false`

`find_highest_possible_recall`
Boolean optional Find the candidate protein resulting in the highest GO recall and add the corresponding value to the output; Default = `false`

`write_evaluation_summary`
Boolean optional Where applicable (numerical columns), write the mean of all non-NaN values and fraction of non-NaN values at the end of the evaluation table; Default = `false`

28.3.5 Parameter Optimization Settings

`path_log`
string optional Path to file that AHRD writes intermediary results to during parameter optimization

`temperature`
integer optional Start temperature for the simulated annealing-based parameter optimization (section 6.3); Greater than 0

`cool_down_by`
integer optional Temperature reduction after each iteration of the simulated annealing parameter optimization (section 6.3); Greater than 0

`optimization_acceptance_probability_scaling_factor`
floating-point optional Scaling factor for the probability to accept worse parameter sets (equation 13) in the simulated annealing-based parameter optimization (section 6.3); Default = 2 500 000 000.0

`mutator_mean`
floating-point optional Mean value for the Gaussian distribution (equation 12) in the mutation of the parameter values in the course of parameter optimization (sections 6.3 and 7.4); Default = 0.25

`mutator_deviation`
floating-point optional Standard deviation for the Gaussian distribution (equation 12) in the mutation of the parameter values in the course of parameter optimization (sections 6.3 and 7.4); Default = 0.15

`remember_simulated_annealing_path`
Boolean optional The parameter sets evaluated in the simulated annealing-based parameter optimization (section 6.3) will be saved together with their evaluation scores and do not need to be evaluated if encountered again; Default = `false`

`p_mutate_same_parameter_scale`

floating-point	optional	Scaling factor for the calculation of the probability (equation 11) to modify the same parameter again after a mutation led to a positive change in the simulated annealing-based parameter optimization (section 6.3); Default = 0.7

`number_of_generations`

integer	optional	The number of generations to be evaluated by the genetic algorithm-based parameter optimization (section 7.4); Greater than 0; Default = 100

`population_size`

integer	optional	The number of parameter sets to be evaluated per generation by the genetic algorithm-based parameter optimization (section 7.4); Greater than 0; Default = 200

28.4 Evidence Code Weights Used for the GO Term Evidence Code Score

Experimental Evidence Codes:

EXP	1.0	Inferred from Experiment
IDA	1.0	Inferred from Direct Assay
IPI	1.0	Inferred from Physical Interaction
IMP	1.0	Inferred from Mutant Phenotype
IGI	1.0	Inferred from Genetic Interaction
IEP	1.0	Inferred from Expression Pattern

High Throughput Experimental Evidence Codes:

HTP	1.0	Inferred from High Throughput Experiment
HDA	1.0	Inferred from High Throughput Direct Assay
HMP	1.0	Inferred from High Throughput Mutant Phenotype
HGI	1.0	Inferred from High Throughput Genetic Interaction
HEP	1.0	Inferred from High Throughput Expression Pattern

Computational Analysis Evidence Codes:

ISS	0.4	Inferred from Sequence or structural Similarity
ISO	0.4	Inferred from Sequence Orthology
ISA	0.4	Inferred from Sequence Alignment
ISM	0.4	Inferred from Sequence Model
IGC	0.4	Inferred from Genomic Context
IBA	0.4	Inferred from Biological aspect of Ancestor
IBD	0.4	Inferred from Biological aspect of Descendant
IKR	0.4	Inferred from Key Residues
IRD	0.4	Inferred from Rapid Divergence
RCA	0.4	Reviewed Computational Analysis

Author Statement Evidence Codes:

TAS	0.4	Traceable Author Statement
NAS	0.2	Non-traceable Author Statement

Curatorial Statement Codes:

IC	0.4	Inferred by Curator
ND	0.1	No Biological Data Available

Automatically-Assigned Evidence Codes:

IEA	0.2	Inferred from Electronic Annotation

Obsolete Evidence Codes:

NR	0.0	

28.5 AHRD Has Been Used in Many Genome Annotation Projects and for Annotation Databases

- Barley [72]
- Wheat [108]
- Tomato [109]
- Melon [110]
- Spinach [111]
- Pineapple [112]
- Rye [113]
- Zucchini [114]
- Spirodela [115]
- *Brassica oleracea* [116]
- Bottle gourd [117]
- Kiwi [118]
- Protomyces [119]
- Petunia [120]
- *Arachis duranensis* [121]
- *Arachis ipaensis* [122]
- *Cardamine hirsuta* [123]
- *Eutrema heterophyllum* and *Eutrema yunnanense* [124]
- *Rhizophagus irregularis* [125]
- *Jaltomata sinuosa* [126]
- *Penium margaritaceum* [127]
- A new thraustochytrid strain [128]
- Legume information system (LegumeInfo.org) [129]
- Cucurbit Genomics Database (CuGenDB) [130]
- PlantsDB [131]
- ORCAE-AOCC [132]
- TodoFirGene [133]

29 Publications

- Buerger H, Boecker F, Packeisen J, Agelopoulos K, Poos K, Nadler W, Korsching E. "Analyzing the basic principles of tissue microarray data measuring the cooperative phenomena of marker proteins in invasive breast cancer." Open Access Bioinformatics. 2013;5:1-21. doi:10.2147/OAB.S36565

- Boecker F, Buerger H, Mallela NV, Korsching E. "TMAinspiration: Decode Interdependencies in Multifactorial Tissue Microarray Data." Cancer informatics vol. 15 143-9. 29 Jun. 2016, doi:10.4137/CIN.S39112

- Zhou, N, Jiang, Y, Bergquist, TR et al. "The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens". Genome Biol 20, 244 (2019). doi:10.1186/s13059-019-1835-8

In Preparation

- Florian Boecker, Heiko Schoof et al. "AHRD: Automatically Annotate Proteins with Human Readable Descriptions and Gene Ontology Terms"

30 Presentations

Oral Presentation

- "AHRD: Using lexical analysis for the CAFA3 challenge"
  at the 25th Conference on Intelligent Systems for Molecular Biology and the 16th European Conference on Computational Biology — ISMB/ECCB 2017
  July 2017 in Prague, Czechia

Posters

- "AHRD: Automated assignment of Human Readable Descriptions and Gene Ontology terms"
  at the Plant and Animal Genome Conference — PAG XXIV
  January 2016 in San Diego, CA United States

- "AHRD — Function Transfer for Gene Ontology and Human Readable Description Annotation"
  at the 26th Conference on Intelligent Systems for Molecular Biology — ISMB 2018
  July 2018 in Chicago, IL United States

31 Acknowledgments

First and foremost I would like to express my immense gratitude towards my supervisor Prof. Dr. Heiko Schoof. After I already had the pleasure to do my master thesis under his wings, he provided the topic, a whole host of ideas, an exceptionally welcoming working environment and even the computing power to complete the endeavor that culminated in writing this dissertation. Over and over again he invested his time and patience with invaluable input to my project and even gave me the opportunity to present our work at conferences in Prague and Chicago.

I am also very thankful for the time the second reviewer Prof. Dr. Martin Hofmann-Apitius invested examining my thesis.

In 1675 Isaac Newton wrote, "If I have seen further it is by standing on the shoulders of giants." In a similar manner, I too must thank Prof. Dr. Heiko Schoof for coming up with the idea for AHRD, Dr. Girish Srinivas for its first implementation and Dr. Asis Hallab as well as Kathrin Klee for refining it to the state in which I was allowed to start improving it.

I am also grateful to the students Dennis Pohl, Richard Lange, Cornelia Mechlen, Samaneh Jozashoori, Robin Will and Fiona Stahl who provided additional insights into AHRD during their projects in the Crop Bioinformatics group. My fellow PhD students in our group Lena Altrogge, Arif Saeed, Lucia Vedder, Tyll Stöcker and Carolin Uebermuth provided a friendly, fun and productive environment to do bioinformatics science in. I also need to mention the always kind and helpful administrative staff of our work group: Karin Olschewski, Michael Plümer and Ellen Laurenzen.

I have no doubt that my nurturing home and family, where curiosity and knowledge about the natural world is part of everyday life, is what set me on the path to pursue a career in science. My mother Karla Boecker is, and my late father Dr. Maximilian Boecker was, a role model biologist, educator and parent. These are qualities I draw from every day. I cannot imagine to have arrived in life where I am today without them also providing the financial foundation that allowed me to start a second education: Becoming a biological technical assistant, and then Bachelor of Science in Recklinghausen and Master of Science in Bonn.

Last but far from least, my biggest thanks go to my girlfriend Ellen Dlaske. I am bound to her by eternal gratitude (and love, of course): Not only did me working on this thesis test her patience countless times, but she also proofread it from top to bottom (including this very sentence).

References

[1] Kishore R. Kumar, Mark J. Cowley, and Ryan L. Davis. Next-generation sequencing and emerging technologies. *Seminars in Thrombosis and Hemostasis*, 45(07):661–673, May 2019. doi:10.1055/s-0039-1688446.

[2] Amanda Raine, Ulrika Liljedahl, and Jessica Nordlund. Data quality of whole genome bisulfite sequencing on illumina platforms. *PLOS ONE*, 13 (4):e0195972, April 2018. doi:10.1371/journal.pone.0195972.

[3] Guy Cochrane, Ilene Karsch-Mizrachi, Toshihisa Takagi, and International Nucleotide Sequence Database Collaboration. The international nucleotide sequence database collaboration. *Nucleic acids research*, 44:D48–D50, January 2016. ISSN 1362-4962. doi:10.1093/nar/gkv1323.

[4] UniProt Consortium. Uniprot: the universal protein knowledgebase in 2021. *Nucleic acids research*, November 2020. ISSN 1362-4962. doi:10.1093/nar/gkaa1100.

[5] Seung Yon Rhee and Marek Mutwil. Towards revealing the functions of all genes in plants. *Trends in Plant Science*, 19(4):212–221, April 2014. doi:10.1016/j.tplants.2013.10.006.

[6] W. R. Gilks, B. Audit, D. De Angelis, S. Tsoka, and C. A. Ouzounis. Modeling the percolation of annotation errors in a database of protein sequences. *Bioinformatics*, 18(12):1641–1649, December 2002. doi:10.1093/bioinformatics/18.12.1641.

[7] Alexandra M. Schnoes, Shoshana D. Brown, Igor Dodevski, and Patricia C. Babbitt. Annotation error in public databases: Misannotation of molecular function in enzyme superfamilies. *PLoS Computational Biology*, 5(12):e1000605, December 2009. doi:10.1371/journal.pcbi.1000605.

[8] Michael Ashburner, Catherine A. Ball, Judith A. Blake, David Botstein, Heather Butler, J. Michael Cherry, Allan P. Davis, Kara Dolinski, Selina S. Dwight, Janan T. Eppig, Midori A. Harris, David P. Hill, Laurie Issel-Tarver, Andrew Kasarskis, Suzanna Lewis, John C. Matese, Joel E. Richardson, Martin Ringwald, Gerald M. Rubin, and Gavin Sherlock. Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nat Genet*, 25(1):25–29, May 2000. doi:10.1038/75556.

[9] Andreas Ruepp, Alfred Zollner, Dieter Maier, Kaj Albermann, Jean Hani, Martin Mokrejs, Igor Tetko, Ulrich Güldener, Gertrud Mannhaupt, Martin Münsterkötter, and H. Werner Mewes. The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucleic acids research*, 32:5539–5545, 2004. ISSN 1362-4962. doi:10.1093/nar/gkh894.

[10] Rainer Schwacke, Gabriel Y. Ponce-Soto, Kirsten Krause, Anthony M. Bolger, Borjana Arsova, Asis Hallab, Kristina Gruden,

Mark Stitt, Marie E. Bolger, and Björn Usadel. MapMan4: A refined protein classification and annotation framework applicable to multi-omics data analysis. *Molecular Plant*, 12(6):879–892, June 2019. doi:10.1016/j.molp.2019.01.003.

[11] Matthias Blum, Hsin-Yu Chang, Sara Chuguransky, Tiago Grego, Swaathi Kandasaamy, Alex Mitchell, Gift Nuka, Typhaine Paysan-Lafosse, Matloob Qureshi, Shriya Raj, Lorna Richardson, Gustavo A Salazar, Lowri Williams, Peer Bork, Alan Bridge, Julian Gough, Daniel H Haft, Ivica Letunic, Aron Marchler-Bauer, Huaiyu Mi, Darren A Natale, Marco Necci, Christine A Orengo, Arun P Pandurangan, Catherine Rivoire, Christian J A Sigrist, Ian Sillitoe, Narmada Thanki, Paul D Thomas, Silvio C E Tosatto, Cathy H Wu, Alex Bateman, and Robert D Finn. The InterPro protein families and domains database: 20 years on. *Nucleic Acids Research*, 49(D1):D344–D354, November 2020. doi:10.1093/nar/gkaa977.

[12] A. Bairoch. The ENZYME database in 2000. *Nucleic Acids Research*, 28 (1):304–305, January 2000. doi:10.1093/nar/28.1.304.

[13] M. Kanehisa and S. Goto. KEGG: kyoto encyclopedia of genes and genomes. *Nucleic acids research*, 28:27–30, January 2000. ISSN 0305-1048. doi:10.1093/nar/28.1.27.

[14] Burkhard Rost. Twilight zone of protein sequence alignments. *Protein Engineering, Design and Selection*, 12(2):85–94, February 1999. doi:10.1093/protein/12.2.85.

[15] Burkhard Rost. Enzyme function less conserved than anticipated. *Journal of Molecular Biology*, 318(2):595–608, April 2002. doi:10.1016/s0022-2836(02)00016-5.

[16] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48:443–453, March 1970. ISSN 0022-2836. doi:10.1016/0022-2836(70)90057-4.

[17] J. Janin and S. J. Wodak. Structural domains in proteins and their role in the dynamics of protein function. *Progress in biophysics and molecular biology*, 42:21–78, 1983. ISSN 0079-6107. doi:10.1016/0079-6107(83)90003-2.

[18] Cyrus Chothia, Julian Gough, Christine Vogel, and Sarah A. Teichmann. Evolution of the protein repertoire. *Science (New York, N.Y.)*, 300: 1701–1703, June 2003. ISSN 1095-9203. doi:10.1126/science.1085371.

[19] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of molecular biology*, 147:195–197, March 1981. ISSN 0022-2836. doi:10.1016/0022-2836(81)90087-5.

[20] M. O. Dayhoff and R. M. Schwartz. Chapter 22: A model of evolutionary change in proteins. In *in Atlas of Protein Sequence and Structure*, 1978.

URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.145.4315.

[21] S. Henikoff and J. G. Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences of the United States of America*, 89:10915–10919, November 1992. ISSN 0027-8424. doi:10.1073/pnas.89.22.10915.

[22] D. J. Lipman and W. R. Pearson. Rapid and sensitive protein similarity searches. *Science (New York, N.Y.)*, 227:1435–1441, March 1985. ISSN 0036-8075. doi:10.1126/science.2983426.

[23] W. R. Pearson and D. J. Lipman. Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences*, 85(8): 2444–2448, April 1988. doi:10.1073/pnas.85.8.2444.

[24] Scott McGinnis and Thomas L. Madden. BLAST: at the core of a powerful and diverse set of sequence analysis tools. *Nucleic Acids Res*, 32(Web Server issue):W20–W25, July 2004. doi:10.1093/nar/gkh435.

[25] Benjamin Buchfink, Chao Xie, and Daniel H Huson. Fast and sensitive protein alignment using DIAMOND. *Nature Methods*, 12(1):59–60, November 2014. doi:10.1038/nmeth.3176.

[26] C. A. Wilson, J. Kreychman, and M. Gerstein. Assessing annotation transfer for genomics: quantifying the relations between protein sequence, structure and function through traditional and probabilistic scores. *Journal of molecular biology*, 297:233–249, March 2000. ISSN 0022-2836. doi:10.1006/jmbi.2000.3550.

[27] Troy Hawkins and Daisuke Kihara. Function prediction of uncharacterized proteins. *Journal of bioinformatics and computational biology*, 5:1–30, February 2007. ISSN 0219-7200. doi:10.1142/s0219720007002503.

[28] A. M. Lesk and C. Chothia. How different amino acid sequences determine similar protein structures: the structure and evolutionary dynamics of the globins. *Journal of molecular biology*, 136:225–270, January 1980. ISSN 0022-2836. doi:10.1016/0022-2836(80)90373-3.

[29] wwPDB consortium. Protein data bank: the single global archive for 3d macromolecular structure data. *Nucleic acids research*, 47:D520–D528, January 2019. ISSN 1362-4962. doi:10.1093/nar/gky949.

[30] M. A. Martí-Renom, A. C. Stuart, A. Fiser, R. Sánchez, F. Melo, and A. Sali. Comparative protein structure modeling of genes and genomes. *Annual review of biophysics and biomolecular structure*, 29:291–325, 2000. ISSN 1056-8700. doi:10.1146/annurev.biophys.29.1.291.

[31] J. U. Bowie, R. Lüthy, and D. Eisenberg. A method to identify protein sequences that fold into a known three-dimensional structure. *Science (New York, N.Y.)*, 253:164–170, July 1991. ISSN 0036-8075. doi:10.1126/science.1853201.

[32] D. T. Jones, W. R. Taylor, and J. M. Thornton. A new approach to protein fold recognition. *Nature*, 358:86–89, July 1992. ISSN 0028-0836. doi:10.1038/358086a0.

[33] Jooyoung Lee, Peter L. Freddolino, and Yang Zhang. Ab initio protein structure prediction. In *From Protein Structure to Function with Bioinformatics*, pages 3–35. Springer Netherlands, 2017. doi:10.1007/978-94-024-1069-3_1.

[34] Longxing Cao, Inna Goreshnik, Brian Coventry, James Brett Case, Lauren Miller, Lisa Kozodoy, Rita E. Chen, Lauren Carter, Alexandra C. Walls, Young-Jun Park, Eva-Maria Strauch, Lance Stewart, Michael S. Diamond, David Veesler, and David Baker. De novo design of picomolar SARS-CoV-2 miniprotein inhibitors. *Science*, 370(6515):426–431, September 2020. doi:10.1126/science.abd9909.

[35] Johannes Söding and Andrei N. Lupas. More than the sum of their parts: On the evolution of proteins from peptides. *BioEssays*, 25(9):837–846, August 2003. doi:10.1002/bies.10321.

[36] Sara El-Gebali, Jaina Mistry, Alex Bateman, Sean R. Eddy, Aurélien Luciani, Simon C. Potter, Matloob Qureshi, Lorna J. Richardson, Gustavo A. Salazar, Alfredo Smart, Erik L. L. Sonnhammer, Layla Hirsh, Lisanna Paladin, Damiano Piovesan, Silvio C. E. Tosatto, and Robert D. Finn. The pfam protein families database in 2019. *Nucleic acids research*, 47: D427–D432, January 2019. ISSN 1362-4962. doi:10.1093/nar/gky995.

[37] François Jacob, David Perrin, Carmen Sánchez, and Jacques Monod. L'opéron : groupe de gènes à expression coordonnée par un opérateur [c. r. acad. sci. paris 250 (1960) 1727–1729]. *Comptes Rendus Biologies*, 328(6):514–520, June 2005. doi:10.1016/j.crvi.2005.04.005.

[38] E. M. Marcotte, M. Pellegrini, H. L. Ng, D. W. Rice, T. O. Yeates, and D. Eisenberg. Detecting protein function and protein-protein interactions from genome sequences. *Science (New York, N.Y.)*, 285:751–753, July 1999. ISSN 0036-8075. doi:10.1126/science.285.5428.751.

[39] M. Pellegrini, E. M. Marcotte, M. J. Thompson, D. Eisenberg, and T. O. Yeates. Assigning protein functions by comparative genome analysis: Protein phylogenetic profiles. *Proceedings of the National Academy of Sciences*, 96(8):4285–4288, April 1999. doi:10.1073/pnas.96.8.4285.

[40] C. von Mering, E. M. Zdobnov, S. Tsoka, F. D. Ciccarelli, J. B. Pereira-Leal, C. A. Ouzounis, and P. Bork. Genome evolution reveals biochemical networks and functional modules. *Proceedings of the*

National Academy of Sciences, 100(26):15428–15433, December 2003. doi:10.1073/pnas.2136809100.

[41] Benno Schwikowski, Peter Uetz, and Stanley Fields. A network of protein–protein interactions in yeast. *Nature Biotechnology*, 18(12): 1257–1261, December 2000. doi:10.1038/82360.

[42] Roded Sharan, Igor Ulitsky, and Ron Shamir. Network-based prediction of protein function. *Molecular Systems Biology*, 3(1):88, January 2007. doi:10.1038/msb4100129.

[43] Damian Szklarczyk, Annika L. Gable, David Lyon, Alexander Junge, Stefan Wyder, Jaime Huerta-Cepas, Milan Simonovic, Nadezhda T. Doncheva, John H. Morris, Peer Bork, Lars J. Jensen, and Christian von Mering. String v11: protein-protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic acids research*, 47:D607–D613, January 2019. ISSN 1362-4962. doi:10.1093/nar/gky1131.

[44] Damiano Piovesan, Manuel Giollo, Carlo Ferrari, and Silvio C. E. Tosatto. Protein function prediction using guilty by association from interaction networks. *Amino Acids*, 47(12):2583–2592, July 2015. doi:10.1007/s00726-015-2049-3.

[45] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown. Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science (New York, N.Y.)*, 270:467–470, October 1995. ISSN 0036-8075. doi:10.1126/science.270.5235.467.

[46] Zhong Wang, Mark Gerstein, and Michael Snyder. Rna-seq: a revolutionary tool for transcriptomics. *Nature reviews. Genetics*, 10:57–63, January 2009. ISSN 1471-0064. doi:10.1038/nrg2484.

[47] Arun J. Singh, Stephen A. Ramsey, Theresa M. Filtz, and Chrissa Kioussi. Differential gene regulatory networks in development and disease. *Cellular and molecular life sciences : CMLS*, 75:1013–1025, March 2018. ISSN 1420-9071. doi:10.1007/s00018-017-2679-6.

[48] Omer Basha, Rotem Shpringer, Chanan M. Argov, and Esti Yeger-Lotem. The differentialnet database of differential protein-protein interactions in human tissues. *Nucleic acids research*, 46:D522–D526, January 2018. ISSN 1362-4962. doi:10.1093/nar/gkx981.

[49] Alexandra M. Schnoes, David C. Ream, Alexander W. Thorman, Patricia C. Babbitt, and Iddo Friedberg. Biases in the experimental annotations of protein function and their effect on our understanding of protein function space. *PLoS Computational Biology*, 9(5):e1003063, May 2013. doi:10.1371/journal.pcbi.1003063.

[50] Stavros Makrodimitris, Roeland C. H. J. van Ham, and Marcel J. T. Reinders. Automatic gene function prediction in the 2020's. *Genes*, 11 (11):1264, October 2020. doi:10.3390/genes11111264.

[51] Paul D. Thomas, Valerie Wood, Christopher J. Mungall, Suzanna E. Lewis, and Judith A. Blake on behalf of the Gene Ontology Consortium. On the use of gene ontology annotations to assess functional similarity among orthologs and paralogs: A short report. *PLoS Computational Biology*, 8(2):e1002386, February 2012. doi:10.1371/journal.pcbi.1002386.

[52] Christophe Dessimoz, Nives Škunca, and Paul D. Thomas. CAFA and the open world of protein function predictions. *Trends in Genetics*, 29(11): 609–610, November 2013. doi:10.1016/j.tig.2013.09.005.

[53] R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(1):17–30, 1989. doi:10.1109/21.24528.

[54] Peisen Zhang, Jinghui Zhang, Huitao Sheng, James J. Russo, Brian Osborne, and Kenneth Buetow. Gene functional similarity search tool (gfsst). *BMC bioinformatics*, 7:135, March 2006. ISSN 1471-2105. doi:10.1186/1471-2105-7-135.

[55] J. Z. Wang, Z. Du, R. Payattakool, P. S. Yu, and C.-F. Chen. A new method to measure the semantic similarity of GO terms. *Bioinformatics*, 23(10):1274–1281, March 2007. doi:10.1093/bioinformatics/btm087.

[56] Gaston K. Mazandu and Nicola J. Mulder. A topology-based metric for measuring term similarity in the gene ontology. *Advances in Bioinformatics*, 2012:1–17, May 2012. doi:10.1155/2012/975783.

[57] P. Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130, July 1999. doi:10.1613/jair.514.

[58] P. W. Lord, R. D. Stevens, A. Brass, and C. A. Goble. Investigating semantic similarity measures across the gene ontology: the relationship between sequence and annotation. *Bioinformatics*, 19(10):1275–1283, July 2003. doi:10.1093/bioinformatics/btg153.

[59] Gaston K. Mazandu, Emile R. Chimusa, and Nicola J. Mulder. Gene ontology semantic similarity tools: survey on features and challenges for biological knowledge discovery. *Briefings in Bioinformatics*, 18(5):886–901, July 2016. doi:10.1093/bib/bbw067.

[60] Girish Srinivas. Automatic assignment of human readable descriptions (AHRD) to uncharacterized protein sequences. Master's thesis, University of Bonn, December 2009.

[61] Stephen F. Altschul, Thomas L. Madden, Alejandro A. Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*, 25(17):3389–3402, September 1997. doi:10.1093/nar/25.17.3389.

[62] Larry Wall, Tom Christiansen, and Jon Orwant. *Programming perl*. " O'Reilly Media, Inc.", 2000. ISBN 9780596000271.

[63] Barbara E. Engelhardt, Michael I. Jordan, Kathryn E. Muratore, and Steven E. Brenner. Protein molecular function prediction by bayesian phylogenomics. *PLoS computational biology*, 1:e45, October 2005. ISSN 1553-7358. doi:10.1371/journal.pcbi.0010045.

[64] Anika Jöcker, Fabian Hoffmann, Andreas Groscurth, and Heiko Schoof. Protein function prediction and annotation in an integrated environment powered by web services (afawe). *Bioinformatics (Oxford, England)*, 24:2393–2394, October 2008. ISSN 1367-4811. doi:10.1093/bioinformatics/btn394.

[65] E. M. Zdobnov and R. Apweiler. InterProScan - an integration platform for the signature-recognition methods in InterPro. *Bioinformatics*, 17(9): 847–848, September 2001. doi:10.1093/bioinformatics/17.9.847.

[66] Ken Arnold, James Gosling, and David Holmes. *The Java programming language*. Addison Wesley Professional, 2005. ISBN 978-0-321-34980-4.

[67] Scott Chacon and Ben Straub. *Pro git*. Apress, 2014. ISBN 978-1-4842-0076-6. URL `https://git-scm.com/book/en/v2`.

[68] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019. URL `https://www.R-project.org/`.

[69] Asis Hallab. *Protein Function Prediction using Phylogenomics, Domain Architecture Analysis, Data Integration, and Lexical Scoring*. PhD thesis, University of Bonn, 2014. URL `http://hdl.handle.net/20.500.11811/6420`.

[70] E. Huala, A. W. Dickerman, M. Garcia-Hernandez, D. Weems, L. Reiser, F. LaFond, D. Hanley, D. Kiphart, M. Zhuang, W. Huang, L. A. Mueller, D. Bhattacharyya, D. Bhaya, B. W. Sobral, W. Beavis, D. W. Meinke, C. D. Town, C. Somerville, and S. Y. Rhee. The arabidopsis information resource (tair): a comprehensive database and web-based information retrieval, analysis, and visualization system for a model plant. *Nucleic acids research*, 29:102–105, January 2001. ISSN 1362-4962. doi:10.1093/nar/29.1.102.

[71] The Gene Ontology Consortium. The gene ontology resource: 20 years and still GOing strong. *Nucleic Acids Research*, 47(D1):D330–D338, November 2018. doi:10.1093/nar/gky1055.

[72] International Barley Genome Sequencing Consortium. A physical, genetic and functional sequence assembly of the barley genome. *Nature*, 491(7426): 711–716, October 2012. doi:10.1038/nature11543.

[73] Martin Mascher, Heidrun Gundlach, Axel Himmelbach, Sebastian Beier, Sven O. Twardziok, Thomas Wicker, Volodymyr Radchuk, Christoph Dockter, Pete E. Hedley, Joanne Russell, Micha Bayer, Luke Ramsay, Hui Liu, Georg Haberer, Xiao-Qi Zhang, Qisen Zhang, Roberto A. Barrero, Lin Li, Stefan Taudien, Marco Groth, Marius Felder, Alex Hastie, Hana Šimková, Helena Staňková, Jan Vrána, Saki Chan, María Muñoz-Amatriaín, Rachid Ounit, Steve Wanamaker, Daniel Bolser, Christian Colmsee, Thomas Schmutzer, Lala Aliyeva-Schnorr, Stefano Grasso, Jaakko Tanskanen, Anna Chailyan, Dharanya Sampath, Darren Heavens, Leah Clissold, Sujie Cao, Brett Chapman, Fei Dai, Yong Han, Hua Li, Xuan Li, Chongyun Lin, John K. McCooke, Cong Tan, Penghao Wang, Songbo Wang, Shuya Yin, Gaofeng Zhou, Jesse A. Poland, Matthew I. Bellgard, Ljudmilla Borisjuk, Andreas Houben, Jaroslav Doležel, Sarah Ayling, Stefano Lonardi, Paul Kersey, Peter Langridge, Gary J. Muehlbauer, Matthew D. Clark, Mario Caccamo, Alan H. Schulman, Klaus F. X. Mayer, Matthias Platzer, Timothy J. Close, Uwe Scholz, Mats Hansson, Guoping Zhang, Ilka Braumann, Manuel Spannagl, Chengdao Li, Robbie Waugh, and Nils Stein. A chromosome conformation capture ordered sequence of the barley genome. *Nature*, 544(7651):427–433, April 2017. doi:10.1038/nature22043.

[74] Pietro D. Spanu, James C. Abbott, Joelle Amselem, Timothy A. Burgis, Darren M. Soanes, Kurt Stüber, Emiel Ver Loren van Themaat, James K. M. Brown, Sarah A. Butcher, Sarah J. Gurr, Marc-Henri Lebrun, Christopher J. Ridout, Paul Schulze-Lefert, Nicholas J. Talbot, Nahal Ahmadinejad, Christian Ametz, Geraint R. Barton, Mariam Benjdia, Przemyslaw Bidzinski, Laurence V. Bindschedler, Maike Both, Marin T. Brewer, Lance Cadle-Davidson, Molly M. Cadle-Davidson, Jerome Collemare, Rainer Cramer, Omer Frenkel, Dale Godfrey, James Harriman, Claire Hoede, Brian C. King, Sven Klages, Jochen Kleemann, Daniela Knoll, Prasanna S. Koti, Jonathan Kreplak, Francisco J. López-Ruiz, Xunli Lu, Takaki Maekawa, Siraprapa Mahanil, Cristina Micali, Michael G. Milgroom, Giovanni Montana, Sandra Noir, Richard J. O'Connell, Simone Oberhaensli, Francis Parlange, Carsten Pedersen, Hadi Quesneville, Richard Reinhardt, Matthias Rott, Soledad Sacristán, Sarah M. Schmidt, Moritz Schön, Pari Skamnioti, Hans Sommer, Amber Stephens, Hiroyuki Takahara, Hans Thordal-Christensen, Marielle Vigouroux, Ralf Weßling, Thomas Wicker, and Ralph Panstruga. Genome expansion and gene loss in powdery mildew fungi reveal tradeoffs in extreme parasitism. *Science*, 330(6010):1543–1546, December 2010. doi:10.1126/science.1194573.

[75] S. Götz, J. M. García-Gómez, J. Terol, T. D. Williams, S. H. Nagaraj, M. J. Nueda, M. Robles, M. Talon, J. Dopazo, and A. Conesa.

High-throughput functional annotation and data mining with the blast2go suite. *Nucleic Acids Research*, 36(10):3420–3435, apr 2008. doi:10.1093/nar/gkn176.

[76] Ronghui You, Shuwei Yao, Yi Xiong, Xiaodi Huang, Fengzhu Sun, Hiroshi Mamitsuka, and Shanfeng Zhu. NetGO: improving large-scale protein function prediction with massive network information. *Nucleic Acids Research*, 47(W1):W379–W387, May 2019. doi:10.1093/nar/gkz388.

[77] Naihui Zhou, Yuxiang Jiang, Timothy R. Bergquist, Alexandra J. Lee, Balint Z. Kacsoh, Alex W. Crocker, Kimberley A. Lewis, George Georghiou, Huy N. Nguyen, Md Nafiz Hamid, Larry Davis, Tunca Dogan, Volkan Atalay, Ahmet S. Rifaioglu, Alperen Dalkıran, Rengul Cetin Atalay, Chengxin Zhang, Rebecca L. Hurto, Peter L. Freddolino, Yang Zhang, Prajwal Bhat, Fran Supek, José M. Fernández, Branislava Gemovic, Vladimir R. Perovic, Radoslav S. Davidović, Neven Sumonja, Nevena Veljkovic, Ehsaneddin Asgari, Mohammad R.K. Mofrad, Giuseppe Profiti, Castrense Savojardo, Pier Luigi Martelli, Rita Casadio, Florian Boecker, Heiko Schoof, Indika Kahanda, Natalie Thurlby, Alice C. McHardy, Alexandre Renaux, Rabie Saidi, Julian Gough, Alex A. Freitas, Magdalena Antczak, Fabio Fabris, Mark N. Wass, Jie Hou, Jianlin Cheng, Zheng Wang, Alfonso E. Romero, Alberto Paccanaro, Haixuan Yang, Tatyana Goldberg, Chenguang Zhao, Liisa Holm, Petri Törönen, Alan J. Medlar, Elaine Zosa, Itamar Borukhov, Ilya Novikov, Angela Wilkins, Olivier Lichtarge, Po-Han Chi, Wei-Cheng Tseng, Michal Linial, Peter W. Rose, Christophe Dessimoz, Vedrana Vidulin, Saso Dzeroski, Ian Sillitoe, Sayoni Das, Jonathan Gill Lees, David T. Jones, Cen Wan, Domenico Cozzetto, Rui Fa, Mateo Torres, Alex Warwick Vesztrocy, Jose Manuel Rodriguez, Michael L. Tress, Marco Frasca, Marco Notaro, Giuliano Grossi, Alessandro Petrini, Matteo Re, Giorgio Valentini, Marco Mesiti, Daniel B. Roche, Jonas Reeb, David W. Ritchie, Sabeur Aridhi, Seyed Ziaeddin Alborzi, Marie-Dominique Devignes, Da Chen Emily Koo, Richard Bonneau, Vladimir Gligorijević, Meet Barot, Hai Fang, Stefano Toppo, Enrico Lavezzo, Marco Falda, Michele Berselli, Silvio C.E. Tosatto, Marco Carraro, Damiano Piovesan, Hafeez Ur Rehman, Qizhong Mao, Shanshan Zhang, Slobodan Vucetic, Gage S. Black, Dane Jo, Erica Suh, Jonathan B. Dayton, Dallas J. Larsen, Ashton R. Omdahl, Liam J. McGuffin, Danielle A. Brackenridge, Patricia C. Babbitt, Jeffrey M. Yunes, Paolo Fontana, Feng Zhang, Shanfeng Zhu, Ronghui You, Zihan Zhang, Suyang Dai, Shuwei Yao, Weidong Tian, Renzhi Cao, Caleb Chandler, Miguel Amezola, Devon Johnson, Jia-Ming Chang, Wen-Hung Liao, Yi-Wei Liu, Stefano Pascarelli, Yotam Frank, Robert Hoehndorf, Maxat Kulmanov, Imane Boudellioua, Gianfranco Politano, Stefano Di Carlo, Alfredo Benso, Kai Hakala, Filip Ginter, Farrokh Mehryary, Suwisa Kaewphan, Jari Björne, Hans Moen, Martti E.E. Tolvanen, Tapio Salakoski, Daisuke Kihara, Aashish Jain, Tomislav Šmuc, Adrian Altenhoff, Asa Ben-Hur, Burkhard Rost, Steven E. Brenner, Christine A. Orengo, Con-

stance J. Jeffery, Giovanni Bosco, Deborah A. Hogan, Maria J. Martin, Claire O'Donovan, Sean D. Mooney, Casey S. Greene, Predrag Radivojac, and Iddo Friedberg. The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome Biology*, 20(1), November 2019. doi:10.1186/s13059-019-1835-8.

[78] Jaime Huerta-Cepas, Kristoffer Forslund, Luis Pedro Coelho, Damian Szklarczyk, Lars Juhl Jensen, Christian von Mering, and Peer Bork. Fast genome-wide functional annotation through orthology assignment by eggNOG-Mapper. *Molecular biology and evolution*, 34:2115–2122, August 2017. ISSN 1537-1719. doi:10.1093/molbev/msx148.

[79] JUnit4. `https://junit.org/junit4/`. Accessed: 2020-07-10.

[80] YamlBeans 1.06. `https://github.com/EsotericSoftware/yamlbeans/releases/tag/1.06`. Accessed: 2020-07-10.

[81] OWL API 4.5.4. `https://github.com/owlcs/owlapi/releases/tag/owlapi-parent-4.5.4`. Accessed: 2020-07-10.

[82] Apache Ant. `https://ant.apache.org/`. Accessed: 2020-07-10.

[83] Eclipse IDE. `https://www.eclipse.org/ide/`. Accessed: 2020-07-10.

[84] RStudio Team. *RStudio: Integrated Development Environment for R*. RStudio, PBC., Boston, MA, 2020. URL `http://www.rstudio.com/`.

[85] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. ISBN 978-3-319-24277-4. URL `https://ggplot2.tidyverse.org`.

[86] Baptiste Auguie. *gridExtra: Miscellaneous Functions for "Grid" Graphics*, 2017. URL `https://CRAN.R-project.org/package=gridExtra`. R package version 2.3.

[87] W. Li, L. Jaroszewski, and A. Godzik. Clustering of highly homologous sequences to reduce the size of large protein databases. *Bioinformatics*, 17(3):282–283, March 2001. doi:10.1093/bioinformatics/17.3.282.

[88] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, May 1983. doi:10.1126/science.220.4598.671.

[89] John H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975. ISBN 9780262581110. second edition, 1992.

[90] Thomas Lumley, Paula Diehr, Scott Emerson, and Lu Chen. The importance of the normality assumption in large public health data sets. *Annual Review of Public Health*, 23(1):151–169, May 2002. doi:10.1146/annurev.publhealth.23.100901.140546.

[91] Predrag Radivojac, Wyatt T Clark, Tal Ronnen Oron, Alexandra M Schnoes, Tobias Wittkop, Artem Sokolov, Kiley Graim, Christopher Funk, Karin Verspoor, Asa Ben-Hur, Gaurav Pandey, Jeffrey M Yunes, Ameet S Talwalkar, Susanna Repo, Michael L Souza, Damiano Piovesan, Rita Casadio, Zheng Wang, Jianlin Cheng, Hai Fang, Julian Gough, Patrik Koskinen, Petri Törönen, Jussi Nokso-Koivisto, Liisa Holm, Domenico Cozzetto, Daniel W A Buchan, Kevin Bryson, David T Jones, Bhakti Limaye, Harshal Inamdar, Avik Datta, Sunitha K Manjari, Rajendra Joshi, Meghana Chitale, Daisuke Kihara, Andreas M Lisewski, Serkan Erdin, Eric Venner, Olivier Lichtarge, Robert Rentzsch, Haixuan Yang, Alfonso E Romero, Prajwal Bhat, Alberto Paccanaro, Tobias Hamp, Rebecca Kaßner, Stefan Seemayer, Esmeralda Vicedo, Christian Schaefer, Dominik Achten, Florian Auer, Ariane Boehm, Tatjana Braun, Maximilian Hecht, Mark Heron, Peter Hönigschmid, Thomas A Hopf, Stefanie Kaufmann, Michael Kiening, Denis Krompass, Cedric Landerer, Yannick Mahlich, Manfred Roos, Jari Björne, Tapio Salakoski, Andrew Wong, Hagit Shatkay, Fanny Gatzmann, Ingolf Sommer, Mark N Wass, Michael J E Sternberg, Nives Škunca, Fran Supek, Matko Bošnjak, Panče Panov, Sašo Džeroski, Tomislav Šmuc, Yiannis A I Kourmpetis, Aalt D J van Dijk, Cajo J F ter Braak, Yuanpeng Zhou, Qingtian Gong, Xinran Dong, Weidong Tian, Marco Falda, Paolo Fontana, Enrico Lavezzo, Barbara Di Camillo, Stefano Toppo, Liang Lan, Nemanja Djuric, Yuhong Guo, Slobodan Vucetic, Amos Bairoch, Michal Linial, Patricia C Babbitt, Steven E Brenner, Christine Orengo, Burkhard Rost, Sean D Mooney, and Iddo Friedberg. A large-scale evaluation of computational protein function prediction. *Nature Methods*, 10(3):221–227, January 2013. doi:10.1038/nmeth.2340.

[92] Yuxiang Jiang, Tal Ronnen Oron, Wyatt T. Clark, Asma R. Bankapur, Daniel D'Andrea, Rosalba Lepore, Christopher S. Funk, Indika Kahanda, Karin M. Verspoor, Asa Ben-Hur, Da Chen Emily Koo, Duncan Penfold-Brown, Dennis Shasha, Noah Youngs, Richard Bonneau, Alexandra Lin, Sayed M. E. Sahraeian, Pier Luigi Martelli, Giuseppe Profiti, Rita Casadio, Renzhi Cao, Zhaolong Zhong, Jianlin Cheng, Adrian Altenhoff, Nives Skunca, Christophe Dessimoz, Tunca Dogan, Kai Hakala, Suwisa Kaewphan, Farrokh Mehryary, Tapio Salakoski, Filip Ginter, Hai Fang, Ben Smithers, Matt Oates, Julian Gough, Petri Törönen, Patrik Koskinen, Liisa Holm, Ching-Tai Chen, Wen-Lian Hsu, Kevin Bryson, Domenico Cozzetto, Federico Minneci, David T. Jones, Samuel Chapman, Dukka BKC, Ishita K. Khan, Daisuke Kihara, Dan Ofer, Nadav Rappoport, Amos Stern, Elena Cibrian-Uhalte, Paul Denny, Rebecca E. Foulger, Reija Hieta, Duncan Legge, Ruth C. Lovering, Michele Magrane, Anna N. Melidoni, Prudence Mutowo-Meullenet, Klemens Pichler, Aleksandra Shypitsyna, Biao Li, Pooya Zakeri, Sarah ElShal, Léon-Charles Tranchevent, Sayoni Das, Natalie L. Dawson, David Lee, Jonathan G. Lees, Ian Sillitoe, Prajwal Bhat, Tamás Nepusz, Alfonso E. Romero, Rajkumar Sasidharan, Haixuan Yang, Alberto Paccanaro, Jesse Gillis, Adriana E. Sedeño-Cortés, Paul Pavlidis, Shou Feng, Juan M. Cejuela, Tatyana

Goldberg, Tobias Hamp, Lothar Richter, Asaf Salamov, Toni Gabaldon, Marina Marcet-Houben, Fran Supek, Qingtian Gong, Wei Ning, Yuanpeng Zhou, Weidong Tian, Marco Falda, Paolo Fontana, Enrico Lavezzo, Stefano Toppo, Carlo Ferrari, Manuel Giollo, Damiano Piovesan, Silvio C.E. Tosatto, Angela del Pozo, José M. Fernández, Paolo Maietta, Alfonso Valencia, Michael L. Tress, Alfredo Benso, Stefano Di Carlo, Gianfranco Politano, Alessandro Savino, Hafeez Ur Rehman, Matteo Re, Marco Mesiti, Giorgio Valentini, Joachim W. Bargsten, Aalt D. J. van Dijk, Branislava Gemovic, Sanja Glisic, Vladmir Perovic, Veljko Veljkovic, Nevena Veljkovic, Danillo C. Almeida e Silva, Ricardo Z. N. Vencio, Malvika Sharan, Jörg Vogel, Lakesh Kansakar, Shanshan Zhang, Slobodan Vucetic, Zheng Wang, Michael J. E. Sternberg, Mark N. Wass, Rachael P. Huntley, Maria J. Martin, Claire O'Donovan, Peter N. Robinson, Yves Moreau, Anna Tramontano, Patricia C. Babbitt, Steven E. Brenner, Michal Linial, Christine A. Orengo, Burkhard Rost, Casey S. Greene, Sean D. Mooney, Iddo Friedberg, and Predrag Radivojac. An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome Biology*, 17(1), September 2016. doi:10.1186/s13059-016-1037-6.

[93] Wyatt T. Clark and Predrag Radivojac. Information-theoretic evaluation of predicted ontological annotations. *Bioinformatics*, 29(13):i53–i61, June 2013. doi:10.1093/bioinformatics/btt228.

[94] Michael A. Fligner and Timothy J. Killeen. Distribution-free two-sample tests for scale. *Journal of the American Statistical Association*, 71(353): 210–213, March 1976. doi:10.1080/01621459.1976.10481517.

[95] S.S. Shapiro and M.B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3-4):591–611, December 1965. doi:10.1093/biomet/52.3-4.591.

[96] Søren Bak, Fred Beisson, Gerard Bishop, Björn Hamberger, René Höfer, Suzanne Paquette, and Danièle Werck-Reichhart. Cytochromes P450. *The Arabidopsis Book*, 9:e0144, January 2011. doi:10.1199/tab.0144.

[97] Meeta Mistry and Paul Pavlidis. Gene ontology term overlap as a measure of gene functional similarity. *BMC Bioinformatics*, 9(1), August 2008. doi:10.1186/1471-2105-9-327.

[98] K Kirschner and H Bisswanger. Multifunctional proteins. *Annual Review of Biochemistry*, 45(1):143–166, June 1976. doi:10.1146/annurev.bi.45.070176.001043.

[99] Constance J. Jeffery. Moonlighting proteins. *Trends in Biochemical Sciences*, 24(1):8–11, January 1999. doi:10.1016/s0968-0004(98)01335-8.

[100] Constance J. Jeffery. Moonlighting proteins—an update. *Molecular BioSystems*, 5(4):345, 2009. doi:10.1039/b900658n.

[101] Constance J. Jeffery. Protein moonlighting: what is it, and why is it important? *Philosophical Transactions of the Royal Society B: Biological Sciences*, 373(1738):20160523, December 2017. doi:10.1098/rstb.2016.0523.

[102] Pascale Gaudet and Christophe Dessimoz. Gene ontology: Pitfalls, biases, and remedies. In *Methods in Molecular Biology*, pages 189–205. Springer New York, November 2016. doi:10.1007/978-1-4939-3743-1_14.

[103] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, pages 448–453, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1558603638. URL `https://arxiv.org/pdf/cmp-lg/9511007.pdf`.

[104] Davide Chicco. Ten quick tips for machine learning in computational biology. *BioData mining*, 10:35, 2017. ISSN 1756-0381. doi:10.1186/s13040-017-0155-3.

[105] Ronghui You, Zihan Zhang, Yi Xiong, Fengzhu Sun, Hiroshi Mamitsuka, and Shanfeng Zhu. GOLabeler: improving sequence-based large-scale protein function prediction by learning to rank. *Bioinformatics*, 34(14): 2465–2473, March 2018. doi:10.1093/bioinformatics/bty130.

[106] Dan Ofer and Michal Linial. ProFET: Feature engineering captures high-level protein functions. *Bioinformatics*, 31(21):3429–3436, June 2015. doi:10.1093/bioinformatics/btv345.

[107] Jie Tan, Georgia Doing, Kimberley A. Lewis, Courtney E. Price, Kathleen M. Chen, Kyle C. Cady, Barret Perchuk, Michael T. Laub, Deborah A. Hogan, and Casey S. Greene. Unsupervised extraction of stable expression signatures from public compendia with an ensemble of neural networks. *Cell Systems*, 5(1):63–71.e6, July 2017. doi:10.1016/j.cels.2017.06.003.

[108] K. F. X. Mayer, J. Rogers, J. Dole el, C. Pozniak, K. Eversole, C. Feuillet, B. Gill, B. Friebe, A. J. Lukaszewski, P. Sourdille, T. R. Endo, M. Kubalakova, J. ihalikova, Z. Dubska, J. Vrana, R. perkova, H. imkova, M. Febrer, L. Clissold, K. McLay, K. Singh, P. Chhuneja, N. K. Singh, J. Khurana, E. Akhunov, F. Choulet, A. Alberti, V. Barbe, P. Wincker, H. Kanamori, F. Kobayashi, T. Itoh, T. Matsumoto, H. Sakai, T. Tanaka, J. Wu, Y. Ogihara, H. Handa, P. R. Maclachlan, A. Sharpe, D. Klassen, D. Edwards, J. Batley, O.-A. Olsen, S. R. Sandve, S. Lien, B. Steuernagel, B. Wulff, M. Caccamo, S. Ayling, R. H. Ramirez-Gonzalez, B. J. Clavijo, J. Wright, M. Pfeifer, M. Spannagl, M. M. Martis, M. Mascher, J. Chapman, J. A. Poland, U. Scholz, K. Barry, R. Waugh, D. S. Rokhsar, G. J. Muehlbauer, N. Stein, H. Gundlach, M. Zytnicki, V. Jamilloux, H. Quesneville, T. Wicker, P. Faccioli, M. Colaiacovo, A. M. Stanca, H. Budak, L. Cattivelli, N. Glover, L. Pingault, E. Paux, S. Sharma, R. Appels, M. Bellgard, B. Chapman, T. Nussbaumer, K. C. Bader, H. Rimbert, S. Wang, R. Knox,

A. Kilian, M. Alaux, F. Alfama, L. Couderc, N. Guilhot, C. Viseux, M. Loaec, B. Keller, and S. Praud. A chromosome-based draft sequence of the hexaploid bread wheat (triticum aestivum) genome. *Science*, 345 (6194):1251788–1251788, July 2014. doi:10.1126/science.1251788.

[109] Tomato Genome Consortium. The tomato genome sequence provides insights into fleshy fruit evolution. *Nature*, 485(7400):635–641, May 2012. doi:10.1038/nature11119.

[110] Valentino Ruggieri, Konstantinos G. Alexiou, Jordi Morata, Jason Argyris, Marta Pujol, Ryoichi Yano, Satoko Nonaka, Hiroshi Ezura, David Latrasse, Adnane Boualem, Moussa Benhamed, Abdelhafid Bendahmane, Riccardo Aiese Cigliano, Walter Sanseverino, Pere Puigdomènech, Josep M. Casacuberta, and Jordi Garcia-Mas. An improved assembly and annotation of the melon (cucumis melo l.) reference genome. *Scientific Reports*, 8(1), May 2018. doi:10.1038/s41598-018-26416-2.

[111] Chenxi Xu, Chen Jiao, Honghe Sun, Xiaofeng Cai, Xiaoli Wang, Chenhui Ge, Yi Zheng, Wenli Liu, Xuepeng Sun, Yimin Xu, Jie Deng, Zhonghua Zhang, Sanwen Huang, Shaojun Dai, Beiquan Mou, Quanxi Wang, Zhangjun Fei, and Quanhua Wang. Draft genome of spinach and transcriptome diversity of 120 spinacia accessions. *Nature Communications*, 8(1), May 2017. doi:10.1038/ncomms15275.

[112] Ray Ming, Robert VanBuren, Ching Man Wai, Haibao Tang, Michael C Schatz, John E Bowers, Eric Lyons, Ming-Li Wang, Jung Chen, Eric Biggers, Jisen Zhang, Lixian Huang, Lingmao Zhang, Wenjing Miao, Jian Zhang, Zhangyao Ye, Chenyong Miao, Zhicong Lin, Hao Wang, Hongye Zhou, Won C Yim, Henry D Priest, Chunfang Zheng, Margaret Woodhouse, Patrick P Edger, Romain Guyot, Hao-Bo Guo, Hong Guo, Guangyong Zheng, Ratnesh Singh, Anupma Sharma, Xiangjia Min, Yun Zheng, Hayan Lee, James Gurtowski, Fritz J Sedlazeck, Alex Harkess, Michael R McKain, Zhenyang Liao, Jingping Fang, Juan Liu, Xiaodan Zhang, Qing Zhang, Weichang Hu, Yuan Qin, Kai Wang, Li-Yu Chen, Neil Shirley, Yann-Rong Lin, Li-Yu Liu, Alvaro G Hernandez, Chris L Wright, Vincent Bulone, Gerald A Tuskan, Katy Heath, Francis Zee, Paul H Moore, Ramanjulu Sunkar, James H Leebens-Mack, Todd Mockler, Jeffrey L Bennetzen, Michael Freeling, David Sankoff, Andrew H Paterson, Xinguang Zhu, Xiaohan Yang, J Andrew C Smith, John C Cushman, Robert E Paull, and Qingyi Yu. The pineapple genome and the evolution of CAM photosynthesis. *Nature Genetics*, 47(12):1435–1442, November 2015. doi:10.1038/ng.3435.

[113] Eva Bauer, Thomas Schmutzer, Ivan Barilar, Martin Mascher, Heidrun Gundlach, Mihaela M. Martis, Sven O. Twardziok, Bernd Hackauf, Andres Gordillo, Peer Wilde, Malthe Schmidt, Viktor Korzun, Klaus F.X. Mayer, Karl Schmid, Chris-Carolin Schön, and Uwe Scholz. Towards a whole-genome sequence for rye (secale cerealeL.). *The Plant Journal*, 89 (5):853–869, February 2017. doi:10.1111/tpj.13436.

[114] Javier Montero-Pau, José Blanca, Aureliano Bombarely, Peio Ziarsolo, Cristina Esteras, Carlos Martí-Gómez, María Ferriol, Pedro Gómez, Manuel Jamilena, Lukas Mueller, Belén Picó, and Joaquín Cañizares. De novoassembly of the zucchini genome reveals a whole-genome duplication associated with the origin of theCucurbitagenus. *Plant Biotechnology Journal*, 16(6):1161–1171, December 2017. doi:10.1111/pbi.12860.

[115] W. Wang, G. Haberer, H. Gundlach, C. Gläßer, T. Nussbaumer, M.C. Luo, A. Lomsadze, M. Borodovsky, R.A. Kerstetter, J. Shanklin, D.W. Byrant, T.C. Mockler, K.J. Appenroth, J. Grimwood, J. Jenkins, J. Chow, C. Choi, C. Adam, X.-H. Cao, J. Fuchs, I. Schubert, D. Rokhsar, J. Schmutz, T.P. Michael, K.F.X. Mayer, and J Messing. The spirodela polyrhiza genome reveals insights into its neotenous reduction fast growth and aquatic lifestyle. *Nature Communications*, 5(1), February 2014. doi:10.1038/ncomms4311.

[116] Isobel AP Parkin, Chushin Koh, Haibao Tang, Stephen J Robinson, Sateesh Kagale, Wayne E Clarke, Chris D Town, John Nixon, Vivek Krishnakumar, Shelby L Bidwell, France Denoeud, Harry Belcram, Matthew G Links, Jérémy Just, Carling Clarke, Tricia Bender, Terry Huebert, Annaliese S Mason, J Pires, Guy Barker, Jonathan Moore, Peter G Walley, Sahana Manoli, Jacqueline Batley, David Edwards, Matthew N Nelson, Xiyin Wang, Andrew H Paterson, Graham King, Ian Bancroft, Boulos Chalhoub, and Andrew G Sharpe. Transcriptome and methylome profiling reveals relics of genome dominance in the mesopolyploid brassica oleracea. *Genome Biology*, 15(6):R77, 2014. doi:10.1186/gb-2014-15-6-r77.

[117] Shan Wu, Md Shamimuzzaman, Honghe Sun, Jerome Salse, Xuelian Sui, Alan Wilder, Zujian Wu, Amnon Levi, Yong Xu, Kai-Shu Ling, and Zhangjun Fei. The bottle gourd genome provides insights into cucurbitaceae evolution and facilitates mapping of aPapaya ring-spot virus-resistance locus. *The Plant Journal*, 92(5):963–975, November 2017. doi:10.1111/tpj.13722.

[118] Shengxiong Huang, Jian Ding, Dejing Deng, Wei Tang, Honghe Sun, Dongyuan Liu, Lei Zhang, Xiangli Niu, Xia Zhang, Meng Meng, Jinde Yu, Jia Liu, Yi Han, Wei Shi, Danfeng Zhang, Shuqing Cao, Zhaojun Wei, Yongliang Cui, Yanhua Xia, Huaping Zeng, Kan Bao, Lin Lin, Ya Min, Hua Zhang, Min Miao, Xiaofeng Tang, Yunye Zhu, Yuan Sui, Guangwei Li, Hanju Sun, Junyang Yue, Jiaqi Sun, Fangfang Liu, Liangqiang Zhou, Lin Lei, Xiaoqin Zheng, Ming Liu, Long Huang, Jun Song, Chunhua Xu, Jiewei Li, Kaiyu Ye, Silin Zhong, Bao-Rong Lu, Guanghua He, Fangming Xiao, Hui-Li Wang, Hongkun Zheng, Zhangjun Fei, and Yongsheng Liu. Draft genome of the kiwifruit actinidia chinensis. *Nature Communications*, 4(1), October 2013. doi:10.1038/ncomms3640.

[119] Kai Wang, Timo Sipilä, Sitaram Rajaraman, Omid Safronov, Pia Laine, Agate Auzane, Alfredo Mari, Petri Auvinen, Lars Paulin, Eric Kemen,

Jarkko Salojärvi, and Kirk Overmyer. A novel phyllosphere resident protomyces species that interacts with the arabidopsis immune system. *bioRxiv*, March 2019. doi:10.1101/594028.

[120] Aureliano Bombarely, Michel Moser, Avichai Amrad, Manzhu Bao, Laure Bapaume, Cornelius S. Barry, Mattijs Bliek, Maaike R. Boersma, Lorenzo Borghi, Rémy Bruggmann, Marcel Bucher, Nunzio D'Agostino, Kevin Davies, Uwe Druege, Natalia Dudareva, Marcos Egea-Cortines, Massimo Delledonne, Noe Fernandez-Pozo, Philipp Franken, Laurie Grandont, J. S. Heslop-Harrison, Jennifer Hintzsche, Mitrick Johns, Ronald Koes, Xiaodan Lv, Eric Lyons, Diwa Malla, Enrico Martinoia, Neil S. Mattson, Patrice Morel, Lukas A. Mueller, Joëlle Muhlemann, Eva Nouri, Valentina Passeri, Mario Pezzotti, Qinzhou Qi, Didier Reinhardt, Melanie Rich, Katja R. Richert-Pöggeler, Tim P. Robbins, Michael C. Schatz, M. Eric Schranz, Robert C. Schuurink, Trude Schwarzacher, Kees Spelt, Haibao Tang, Susan L. Urbanus, Michiel Vandenbussche, Kitty Vijverberg, Gonzalo H. Villarino, Ryan M. Warner, Julia Weiss, Zhen Yue, Jan Zethof, Francesca Quattrocchio, Thomas L. Sims, and Cris Kuhlemeier. Insight into the evolution of the solanaceae from the parental genomes of petunia hybrida. *Nature Plants*, 2(6), May 2016. doi:10.1038/nplants.2016.74.

[121] Hui Song, Juan Sun, and Guofeng Yang. Old and young duplicate genes reveal different responses to environmental changes in arachis duranensis. *Molecular Genetics and Genomics*, 294(5):1199–1209, May 2019. doi:10.1007/s00438-019-01574-8.

[122] David John Bertioli, Steven B Cannon, Lutz Froenicke, Guodong Huang, Andrew D Farmer, Ethalinda K S Cannon, Xin Liu, Dongying Gao, Josh Clevenger, Sudhansu Dash, Longhui Ren, Márcio C Moretzsohn, Kenta Shirasawa, Wei Huang, Bruna Vidigal, Brian Abernathy, Ye Chu, Chad E Niederhuth, Pooja Umale, Ana Cláudia G Araújo, Alexander Kozik, Kyung Do Kim, Mark D Burow, Rajeev K Varshney, Xingjun Wang, Xinyou Zhang, Noelle Barkley, Patrícia M Guimarães, Sachiko Isobe, Baozhu Guo, Boshou Liao, H Thomas Stalker, Robert J Schmitz, Brian E Scheffler, Soraya C M Leal-Bertioli, Xu Xun, Scott A Jackson, Richard Michelmore, and Peggy Ozias-Akins. The genome sequences of arachis duranensis and arachis ipaensis, the diploid ancestors of cultivated peanut. *Nature Genetics*, 48(4):438–446, February 2016. doi:10.1038/ng.3517.

[123] Xiangchao Gan, Angela Hay, Michiel Kwantes, Georg Haberer, Asis Hallab, Raffaele Dello Ioio, Hugo Hofhuis, Bjorn Pieper, Maria Cartolano, Ulla Neumann, Lachezar A. Nikolov, Baoxing Song, Mohsen Hajheidari, Roman Briskine, Evangelia Kougioumoutzi, Daniela Vlad, Suvi Broholm, Jotun Hein, Khalid Meksem, David Lightfoot, Kentaro K. Shimizu, Rie Shimizu-Inatsugi, Martha Imprialou, David Kudrna, Rod Wing, Shusei Sato, Peter Huijser, Dmitry Filatov, Klaus F. X. Mayer, Richard Mott, and Miltos Tsiantis. The cardamine hirsuta genome offers insight into the

evolution of morphological diversity. *Nature Plants*, 2(11), October 2016. doi:10.1038/nplants.2016.167.

[124] Xinyi Guo, Quanjun Hu, Guoqian Hao, Xiaojuan Wang, Dan Zhang, Tao Ma, and Jianquan Liu. The genomes of two eutrema species provide insight into plant adaptation to high altitudes. *DNA Research*, 25(3): 307–315, January 2018. doi:10.1093/dnares/dsy003.

[125] Kui Lin, Erik Limpens, Zhonghua Zhang, Sergey Ivanov, Diane G. O. Saunders, Desheng Mu, Erli Pang, Huifen Cao, Hwangho Cha, Tao Lin, Qian Zhou, Yi Shang, Ying Li, Trupti Sharma, Robin van Velzen, Norbert de Ruijter, Duur K. Aanen, Joe Win, Sophien Kamoun, Ton Bisseling, René Geurts, and Sanwen Huang. Single nucleus genome sequencing reveals high similarity among nuclei of an endomycorrhizal fungus. *PLoS Genetics*, 10(1):e1004078, January 2014. doi:10.1371/journal.pgen.1004078.

[126] Meng Wu, Jamie L Kostyun, and Leonie C Moyle. Genome sequence of jaltomata addresses rapid reproductive trait evolution and enhances comparative genomics in the hyper-diverse solanaceae. *Genome Biology and Evolution*, 11(2):335–349, January 2019. doi:10.1093/gbe/evy274.

[127] Chen Jiao, Iben Sørensen, Xuepeng Sun, Honghe Sun, Hila Behar, Saleh Alseekh, Glenn Philippe, Kattia Palacio Lopez, Li Sun, Reagan Reed, Susan Jeon, Reiko Kiyonami, Sheng Zhang, Alisdair R. Fernie, Harry Brumer, David S. Domozych, Zhangjun Fei, and Jocelyn K. C. Rose. The genome of the charophyte alga penium margaritaceum bears footprints of the evolutionary origins of land plants. *bioRxiv*, November 2019. doi:10.1101/835561.

[128] Khawla Seddiki, François Godart, Riccardo Aiese Cigliano, Walter Sanseverino, Mohamed Barakat, Philippe Ortet, Fabrice Rébeillé, Eric Maréchal, Olivier Cagnac, and Alberto Amato. Sequencing, de novo assembly, and annotation of the complete genome of a new thraustochytrid species, strain CCAP_4062/3. *Genome Announcements*, 6(11), March 2018. doi:10.1128/genomea.01335-17.

[129] Sudhansu Dash, Jacqueline D. Campbell, Ethalinda K.S. Cannon, Alan M. Cleary, Wei Huang, Scott R. Kalberer, Vijay Karingula, Alex G. Rice, Jugpreet Singh, Pooja E. Umale, Nathan T. Weeks, Andrew P. Wilkey, Andrew D. Farmer, and Steven B. Cannon. Legume information system (LegumeInfo.org): a key component of a set of federated data resources for the legume family. *Nucleic Acids Research*, 44(D1):D1181–D1188, November 2015. doi:10.1093/nar/gkv1159.

[130] Yi Zheng, Shan Wu, Yang Bai, Honghe Sun, Chen Jiao, Shaogui Guo, Kun Zhao, Jose Blanca, Zhonghua Zhang, Sanwen Huang, Yong Xu, Yiqun Weng, Michael Mazourek, Umesh K. Reddy, Kaori Ando, James D McCreight, Arthur A Schaffer, Joseph Burger, Yaakov Tadmor, Nurit Katzir, Xuemei Tang, Yang Liu, James J Giovannoni, Kai-Shu Ling, W Patrick

Wechter, Amnon Levi, Jordi Garcia-Mas, Rebecca Grumet, and Zhangjun Fei. Cucurbit genomics database (CuGenDB): a central portal for comparative and functional genomics of cucurbit crops. *Nucleic Acids Research*, 47(D1):D1128–D1136, October 2018. doi:10.1093/nar/gky944.

[131] Manuel Spannagl, Thomas Nussbaumer, Kai C. Bader, Mihaela M. Martis, Michael Seidel, Karl G. Kugler, Heidrun Gundlach, and Klaus F.X. Mayer. PGSB PlantsDB: updates to the database framework for comparative plant genome research. *Nucleic Acids Research*, 44(D1):D1141–D1147, November 2015. doi:10.1093/nar/gkv1130.

[132] Anna E. J. Yssel, Shu-Min Kao, Yves Van de Peer, and Lieven Sterck. ORCAE-AOCC: A centralized portal for the annotation of african orphan crop genomes. *Genes*, 10(12):950, November 2019. doi:10.3390/genes10120950.

[133] Saneyoshi Ueno, Yukino Nakamura, Masaaki Kobayashi, Shin Terashima, Wataru Ishizuka, Kentaro Uchiyama, Yoshihiko Tsumura, Kentaro Yano, and Susumu Goto. TodoFirGene: Developing transcriptome resources for genetic analysis of abies sachalinensis. *Plant and Cell Physiology*, 59(6): 1276–1284, March 2018. doi:10.1093/pcp/pcy058.