Protein Function Prediction using Phylogenomics, Domain Architecture Analysis, Data Integration, and Lexical Scoring

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

vorgelegt von

Asis Hallab

aus
Köln

Bonn 2014

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 : 10. Februar 2015

Erscheinungsjahr : 2015

For instance, on the planet Earth, man had always assumed that he was more intelligent than dolphins because he had achieved so much — the wheel, New York, wars and so on — whilst all the dolphins had ever done was muck about in the water having a good time. But conversely, the dolphins had always believed that they were far more intelligent than man — for precisely the same reasons.

— Douglas Adams, *The Hitchhiker's Guide to the Galaxy*

Contents

List of Figures

List of Tables

Part I.

Introduction

1. Protein function prediction

The ever and increasingly rapidly growing amount of protein sequences requires fast and reliable annotation tools in order to enable the identification of gene products of interest. Because for example once such proteins of interest are identified educated experiments aimed at their further characterization can be designed. Also on a genomic scale, such protein function annotations, when available for the proteome of a whole organism, can be applied to identify more systematic properties such as function enrichments or losses, that in turn provide cues to the organism's adaptations, ecological role and evolutionary history. Historically these protein characterizations were first made by assigning the query proteins short descriptions, for example like those found in the various public protein databases. While these descriptions give the human reader a good and comprehensive summary, they are not suitable for computational analysis, because the same function can be described by very different textual descriptions (Hawkins and Kihara 2007) and hence impede reliable analyses e.g. aimed at the mentioned identification of function enrichment or loss. To solve this problem many protein characterising ontologies have been proposed that extend pure textual protein descriptions with unique terms for each distinct protein characteristic. Among these ontologies the most frequently used (Hawkins and Kihara 2007) are the Gene Ontology (GO) (Ashburner, Ball, Blake, et al. 2000), the Enzyme Commission (Webb 1992), Kyoto Encyclopedia of Genes and Genomes (KEGG) (Kanehisa and Goto 2000), and the MIPS Functional Catalogue (FunCat) (Ruepp, Zollner, Maier, et al. 2004). Of which according to Hawkins and Kihara the Gene Ontology is the one with most widespread application, because it is highly amenable for computational annotation methods and because, to the human curator, its hierarchical organization into a directed acyclic graph (GO DAG) enables finer protein characterizations with increasing distance from the DAG's root (GO level). Furthermore the Gene Ontology's usefulness is also founded in the separation of terms describing either a gene product's involvement in biological processes, or its cellular localization, or finally the gene product's molecular function.

Many methods have been proposed to predict protein function and assign functional descriptions from the above ontologies. All of them rely on existing knowledge of well studied proteins. Some methods apply a measure of similarity to transfer annotations from homologous proteins to the uncharacterized query while other methods aim to learn the most distinguishing characteristics of a group of proteins with similar function so that when a query matches those characteristics it can be annotated accordingly.

1.1. Use cases

Depending on the research context, two use cases of protein function prediction methods can be distinguished. The first focusses on a small group of proteins of interest for example candidate drug receptors in a pharmaceutical study or genes being highly likely to contribute to a phenotypic trait. Electronic characterization of such a set of query proteins can be done manually using for example various tools available on the web. Here method parameters and intermediary results can be carefully selected and steps easily repeated until accurate and confident results are obtained. Also computational resource limits hardly play a role, because the analyses are often executed on a remote server. The second type of use case, however, is set in a high throughput environment or on genomic scale, often

within the context of characterizing a newly sequenced genome and comparing it with other reference organisms. Here the computational characterization of large sets of query proteins is commonly carried out using locally installed electronic tools on available compute clusters. In this second context, manual inspection of intermediary results and selection of optimal parameters can only be done on a much more global level and thus the applied methods have to meet a higher demand on robustness. Furthermore computational resource requirements play a more significant role and have to be within the limits of the locally used computers.

The great variety of published procedures is not always suitable for both use cases, often due to inherent restrictions of their incorporated algorithms.

1.2. Function prediction based on sequence similarity

The first group of methods applies different measures of similarity, of which the historically oldest is sequence similarity computed as the score of pairwise sequence alignments. Because some protein characteristics are only associated with partial protein sequence like e.g. hydrophobic transmembrane domains the original algorithm from Needleman and Wunsch to produce global protein alignments (Needleman and Wunsch 1970) was modified to enable the generation of local sequence alignments (Smith and Waterman 1981; Waterman, Smith, and Beyer 1976). While this approach is reasonably resource demanding for two sequences it becomes time consuming when pairwisely aligning a query sequence with a large set of candidates with the goal of finding good matches. To enable such searches of large sequence databases two heuristic methods with widespread application "Basic Local Alignment Search Tool (BLAST)" (Altschul, Madden, Schaffer, et al. 1997; McGinnis and Madden 2004) and FASTA (Lipman and Pearson 1985; Pearson and Lipman 1988) were developed, both precede the optimal alignment step with faster but non optimal searches for promising candidate sequences. Subsequently the significance of such search results can be determined by the sequence alignment score, which is based on gap opening and extension costs, as well as the rates at which one amino acid changes into another over time. These rates are typically held in so called substitution matrices, that are computed using large sets of proteins and hence can not account for locally different rates of mutation or conservation respectively. Because conserved regions of short sequence motifs or patterns often encode conserved protein domains (Durbin 1998), and these domains in turn are often associated with specific protein functions, algorithms to reliably detect such sequence motifs have been developed. These algorithms depend on Position Specific Scoring Matrices (PSSMs) (Altschul, Madden, Schaffer, et al. 1997) or Hidden Markov Models (HMMs) (Zdobnov and Apweiler 2001) that enable specific scoring according to amino acid conservation rates at different alignment positions and thus the accurate detection of conserved protein domains in query sequences. The widely used InterProScan suite (Zdobnov and Apweiler 2001) unifies several algorithms to identify conserved sequence motifs (Apweiler, Attwood, Bairoch, et al. 2000) and includes mappings to terms from the Gene Ontology. According to Hawkins and Kihara "InterPro provides a powerful tool for protein sequence classification and function prediction.", and "has been used in many genome annotation projects, as well as by UniProt curators for individual protein sequence annotation."

The introduced tools are widely used to transfer annotations from the most similar sequences to queries and because they are both accurate and have relatively low computational resource demands they are applied on genomic scales or in high throughput environments (Pierri, Parisi, and Porcelli 2010; Messih, Chitale, Bajic, et al. 2012; Hawkins and Kihara 2007; Rentzsch and Orengo 2009). For instance using highly significant results from BLAST searches and output from InterProScan "the functions of 69% of the [arabidopsis] genes were classified according to sequence similarity to proteins of known function in all organisms" (Arabidopsis Genome Initiative 2000). Another genome

scale example is how enzymatic functions in the grape proteome were annotated with terms from the Enzyme Commission (Webb 1992). This annotation was achieved by matching predicted grape proteins (Jaillon, Aury, Noel, et al. 2007) to position specific scoring matrices of the PRIAM collection (Claudel-Renard, Chevalet, Faraut, and Kahn 2003) with the RPS-BLAST tool (Altschul, Madden, Schaffer, et al. 1997). A final example for protein function prediction on a genomic scale is the annotation of the rice proteome with terms from the Gene Ontology, for which conserved protein domains and GO terms associated with them, were identified by the InterProScan suite (Zdobnov and Apweiler 2001).

Because of the explained reasons, methods based on measuring sequence similarity are often *exclusively* used to characterize — especially large — sets of query proteins. In spite of this they have been known to make false annotations. For example "Top hit sequences (using BLAST) for open reading frames in *E. coli* fail to represent the closest phylogenetic neighbor 27.3% of the time." (Koski and Golding 2001). Hence refined methods were proposed that e.g. take into account the distribution of ontology terms in the set of significantly similar sequences (homologs). The OntoBlast tool (Zehetner 2003) for example assigns terms from the Gene Ontology to query proteins, weighting annotations found in the query's homologs by multiplying the scores of the BLAST results they appear in. This approach is extended by propagating the computed GO term specific weights to parent terms in the before mentioned GO DAG, as implemented in both the GOfigure (Khan, Situ, Decker, and Schmidt 2003) as well as the GOtcha (Martin, Berriman, and Barton 2004) tools. Extending this method the "Protein Function Prediction (PFP)" tool (Hawkins, Chitale, Luban, and Kihara 2009) computes scores for GO term annotations not only based on their respective frequency and alignment scores, but also takes into account terms that frequently are annotated together. The latter part of the score is assessed as the conditional probability of co-annotation, which is looked up in the precomputed Function Association Matrix (FAM), whose entries were inferred by counting co-annotations on a selected set of reference proteins. In spite of the fact that in their published evaluations these refined methods achieve good results and are, according to their authors, applicable on large sets of query proteins, the tools themselves are only available as web services. At the time of this writing only the PFP server (*kiharalab.org/web/pfp.php*) and the GOtcha server (*compbio.dundee.ac.uk/gotcha/gotcha.php*) were online and accepted a maximum of 10 and 1 protein sequences at a time, respectively, thus impeding the annotation of large query sets and contradicting the statement of genome scale applicability.

Another approach to increase accuracy and reliability of protein function predictions based on sequence similarity search results is to actually take into account a query's evolutionary history in the form of a phylogenetic tree. After the required phylogenetic reconstruction, annotations found in more and closer related homologs are assigned higher probabilities of being accurate annotations for the query, while those terms found more rarely and only in distant homologs are assigned lower probabilities. This approach aims to reflect that function mutation becomes more probable the more sequence mutation is accumulated, where sequence distance correlates with passed evolutionary time. To reflect that function mutation also becomes much more likely after duplication than after speciation events the Sifter method (Engelhardt, Jordan, Muratore, and Brenner 2005; Engelhardt, Jordan, Srouji, and Brenner 2011) increases the probability of GO term annotation loss after duplication events. In order to infer the evolutionary type of phylogenetic nodes — either speciation or duplication events — the tree is reconciled with a manually curated species tree using "a simple algorithm to infer gene duplication and speciation events on a gene tree" (Zmasek and Eddy 2001).

1.3. Function prediction based on protein structure

The function of a protein is strongly associated with its structure, which is generally more conserved than the protein's sequence (Wilson, Kreychman, and Gerstein 2000; Gille, Goede, Preissner, et al. 2000). While "Sequence alignments unambiguously distinguish between protein pairs of similar and non-similar structure when the pairwise sequence identity is high (>40% for long alignments). The signal gets blurred in the *twilight zone* of 20-35% sequence identity." (Rost 1999) For this reason methods to predict protein structure and subsequently annotate the associated functions have been proposed. Secondary structure can be predicted by various approaches: By homology, *ab initio*, and by threading (Hawkins and Kihara 2007). The first homology based approach identifies similar sequences as templates whose structure the query is aligned with. Subsequently the query's structure is modelled using information from the selected templates, and finally the models can be evaluated for example using their "Z score" a function measuring the goodness of fit between the query sequence and the proposed structure model (Fiser and Sali 2003). The Z score is computed on knowledge based mean fields (Sippl 1993a; Sippl 1993b) that can also be applied for threading methods, feeding the query protein through databases of known structures and returning the best fit. By applying methods based on molecular dynamics ab initio modelling can be executed, which is still very resource demanding and should only be applied on short protein sequences with less than 100 amino acid residues (Hardin, Pogorelov, and Luthey-Schulten 2002). Ab initio modelling however when it is applied in a combined approach together with homology searches and threading has been evaluated as effective and applicable on large query protein sets (Skolnick, Zhang, Arakaki, et al. 2003). Finally inherent sequence characteristics can be effectively employed to predict protein structure and transmembrane topology with machine learning methods such as neural networks (Jones 2007) or support vector machines (Nugent and Jones 2009). After the prediction of a query protein's structure the associated protein functions can be looked up in various public databases (Hawkins and Kihara 2007). Because some global folds are known to be associated with different functions and many methods are still relatively new and under development close manual inspection of results is recommended (Hawkins and Kihara 2007; Pierri, Parisi, and Porcelli 2010; Fiser and Sali 2003), thus somewhat impeding the application of these methods on large sets of query proteins.

1.4. Genomic Context based function predictions

Gene fusions are rare evolutionary events in which closely interacting genes are fused into a new single gene (Rentzsch and Orengo 2009) and hence their identification can successfully be exploited to predict protein function. Another approach regards the conservation of a gene's neighborhood which is often associated with the gene product's function (Rentzsch and Orengo 2009; Hawkins and Kihara 2007), i.e. because a conserved genomic context can point to an operon organisation. Hence methods to detect such conserved genomic contexts and — based on this conservation — predict a query's function have been proposed. To this end sets of neighboring genes are clustered and a query's function is predicted based on matching its cluster with similar ones whose functions are known. Another neighborhood method not only regards existing genes in a query's proximity but also missing ones. Here, phylogenetic profiles are constructed that record co-occurrences and co-absences of genes using boolean vectors, and subsequently function annotations can be transferred from similar profiles to the query profile. A limitation of genomic context based function predictions results from the sparse genomic data, which is not always available both for query proteins as well as for reference genes and reference functions. Also the genomic context is not always conserved for genes with a given function, which is why not all functions can be predicted with methods based on genomic context (Rentzsch

and Orengo 2009; Hawkins and Kihara 2007).

1.5. Co-expression and network based function prediction

Networks of interacting proteins can also be constructed from expression analyses i.e. using microarrays. Such networks of protein protein interactions can for example be used to transfer function annotations from close neighbors where increasing distance implies reduced probability of a shared function. Simply annotating the query with the most common function found among its neighbors (Schwikowski, Uetz, and Fields 2000) or transferring annotations from clusters or subgraphs that share common interaction partners (Samanta and Liang 2003; Brun, Chevenet, Martin, et al. 2004) have been successfully applied (Hawkins and Kihara 2007). Other clustering methods for protein protein interaction networks aim to identify highly interconnected subgraphs by counting neighboring edges (Rougemont and Hingamp 2003), or by weighting local network density (Bader and Hogue 2003), or finally by computing specific distances and interpreting closely positioned nodes as clusters (Brun, Herrmann, and Guenoche 2004). The popular Markov Clustering Algorithm has also been applied successfully on protein protein interaction networks to the end of characterizing unknown protein function (Asur, Ucar, and Parthasarathy 2007; Satuluri, Parthasarathy, and Ucar 2010). Here Markov Clustering mimics random walks through a graph with labelled edges, treating the matrix of vertices as a probabilistic matrix of a discrete Markov Process. After each iteration weights of edges walked often are amplified, while those of poorly visited edges are decreased. Only few iterations are normally required to achieve satisfying clustering even of very large graphs (Van Dongen 2008).

1.6. Mapping missing links in metabolic networks

A *specialized* efficient method to predict a gene product's involvement in biological processes is the identification of missing links in metabolic networks. For example after having annotated a newly sequenced proteome missing metabolic core functions can be identified and mapped on predicted proteins with so far unknown functions. This mapping can be achieved e.g. by sequence similarity searches with proteins that are known to have the missing function or motifs generated from a set of proteins fulfilling the missing function (Karp, Paley, and Romero 2002; Hawkins and Kihara 2007). Because protein function prediction based on missing links in metabolic networks requires existing function annotations to identify the missing links in the first place and then assigns so far unannotated proteins the missing function based on sequence similarity search results the method's usefulness on genomic scale is somewhat limited, while it also can not be applied outside genomic contexts, because in such environments missing links simply can not be identified.

1.7. Machine learned function prediction

The chemical and biological properties associated with protein function can be numerous and correlation varies depending on the function (Lee, Shin, Oh, et al. 2009). Machine learning approaches enable the selection of those features that distinguish candidate sequences best and thus are accurate and reliable (Lee, Shin, Oh, et al. 2009). Though because they typically make binary decisions and rely on being trained for each function separately they often are only applicable to make coarse function predictions (Rentzsch and Orengo 2009), while on the other hand evaluation proves them to be accurate and reliable predictors (Rentzsch and Orengo 2009; Cai, Han, Ji, and Chen 2004; Lee, Shin, Oh, et al. 2009; Guan, Myers, Hess, et al. 2008). Popular protein characteristics used as features are

amino acid composition, surface tension, hydrophobicity, normalized Van der Vaals Volume, protein length, molecular weight, number of atoms, periodicity, theoretical isoelectric point, secondary and tertiary structure among many others. Most commonly used machine learning techniques are support vector machines (SVM), neural networks, or k nearest neighbor (kNN). Among these SVMs aim to fit a hyperplane that separates two classes of training points in parameter space with maximum margin. If the training data points are not linearly separable SVMs use the "Kernel Trick", with which data points are separated by a hyperplane in a higher dimensional space in which distances can be computed with the kernel function without actually having to project the data points into the selected higher dimensional space. Neural networks typically consist of three layers of artificial neurons that can be switched off or on, and when in the latter state, stimulate other neurons they are connected with. Training determines thresholds in such a way that the output neuron is stimulated correctly when a data point belongs to a given training class. Finally the k nearest neighbor approach identifies central points for each class of training data and when applied measures distances between these centres and an input data point. Subsequently kNN reports back the k classes closest to the input. Parameter space dimensionality reduction is applied in order to speed up computation and overcome the "dimensionality curse", i.e. the trend by which in higher dimensions data points tend to be close to more and more trained class centres. This reduction of dimensionality can for example be achieved by Principal Component Analysis (PCA).

1.8. Ensemble machine learned classifiers

Another machine learning method is the construction and subsequent usage of decision trees, that at each node uses an attribute to classify the input. The final decision is made once a tip is reached. Several methods exist to construct decision trees. The popular C4.5 method (Quinlan 1986) uses at each node that attribute that splits the currently evaluated training set with the highest increase of information entropy (Shannon 1948). Analyzing pancreatic cancer proteomic data Ge and Wong found that combined binary machine learning classifiers always outperformed single decision tree based classifiers generated with the C4.5 method (Ge and Wong 2008). The authors evaluated different popular ensemble classifiers on a reduced subset of features. One — Bootstrap aggregating ("Bagging") — trains each classifier with a bootstrapped subset of training data and returns a classification by majority rule (Breiman 1996). By further introducing a random selection of features used to construct each classifier (decision tree) "random decision forests" are generated (Breiman 2001), which have been successfully applied in a modified form to predict protein protein interactions (Chen and Liu 2005). Another ensemble classifier — Adaptive Boosting ("AdaBoost") — weights each classifier during training, where in each iteration the weights for misclassified examples are increased at the cost of correctly classified ones (Freund and Schapire 1997).

1.9. Example combined methods pipeline

Pierri, Parisi, and Porcelli propose a bioinformatics pipeline of different tools to accurately and reliably characterise query proteins in a pharmaceutical context (Pierri, Parisi, and Porcelli 2010). In the process they extend the pure sequence similarity search with secondary structure prediction, followed by fold recognition methods and secondary structure alignment, subsequent three dimensional modelling based on the crystallized structure of close homologs, if none is available for the query itself, and finally binding pocket proposal based upon the predicted 3D model as well as mutagenesis data and literature mining. The authors suggest close manual inspection of intermediate results at various

steps of their proposed pipeline, which shows that it is intended to be used for a small set of candidate proteins a pharmaceutical study typically is focused upon.

1.10. Accuracy and Evaluation

The presented plethora of function prediction methods calls for an assessment of their accuracy. Typically, when publishing a new annotation method, the authors present their evaluation of it, in which the new tool has been compared to other competitors on a set of reference proteins. In this the applied measurements and reference sets vary and thus impede the comparison of different evaluations published separately. Engelhardt, Jordan, Srouji, and Brenner for example apply their latest version of "Sifter" on the Nudix protein family, among others, and measure the predictions' accuracy in terms of "the percentage of proteins for which the functional term with the highest rank is an exact match to one of the experimental annotations for that protein" (Engelhardt, Jordan, Srouji, and Brenner 2011). The authors also infer Sifter's accuracy by measurements of true and false positive rates, for which they only accept exact matches of the reference GO term and the predicted one. Hence they ignore the Gene Ontology's hierarchical structure. Another example is the accuracy assessment applied by Martin, Berriman, and Barton, who measure the performance of their "GOtcha" tool in terms of the selectivity, that is "the proportion of predictions by GOtcha that are correct" (Martin, Berriman, and Barton 2004). Because here the number of correct GO term annotations is counted on the *whole* set of reference proteins, this selectivity measurement is not to be confused with an assessment of the methods precision or "positive predictive value". This is because precision is computed *separately* for each reference protein as the fraction of true positives, that is the number of correct predictions, in the set of all predictions made for each particular protein. Furthermore the authors introduce their own "new accuracy measure [which] encompasses true positives, false positives and false negatives, so combining sensitivity and selectivity in one value." (Martin, Berriman, and Barton 2004). This new measure was introduced to compensate difficulties in the comparison of function prediction methods. One of which, as the authors point out, is that "One method may only annotate to relatively general terms, allowing for a better claimed specificity than a method that attempts to annotate to a more specific level." (Martin, Berriman, and Barton 2004) This was their motivation to conceive their new accuracy measure, the "Relative Error Quotient (REQ)", that corrects for the postulated bias. Finally the example function annotation pipeline presented by Pierri, Parisi, and Porcelli is only evaluated using case studies, because the proposed method requires manual inspection of intermediate results and educated selection of parameters used in the various pipelined methods, and hence is only applicable on a small set of query proteins, as mentioned before. The need for general comparability of function prediction methods lead to the conception of the "Critical Assessment of Function Annotation experiment (CAFA)" project (Radivojac, Clark, Oron, et al. 2013). In this experiment a world wide comparison of latest protein function prediction methods was carried out using the well established "F-measure", the harmonic mean of precision and recall, for assessing the accuracy of electronically made GO term annotations on a carefully selected set of reference proteins, the "gold standard". Here the gold standard was taken from the "Swissprot" database (Boeckmann, Bairoch, Apweiler, et al. 2003) of manually curated proteins and their annotations. Already carrying out a new experiment, the last CAFA terminated in January 2011 and compared the performance of predictions made by 54 different prediction methods on a gold standard of 866 reference proteins, taken from 11 organisms. In this only experimentally verified reference GO term annotations from the "molecular function" (MF) and "biological process" (BP) ontologies were used. Interestingly 38% of the reference proteins had only "protein binding" (GO:0005515) as a molecular function annotation, thus limiting predictions in these cases to a somewhat general prediction. The experiment showed that the widely used standard method

"best BLAST" or "top BLAST", which passes the annotation from the best BLAST (McGinnis and Madden 2004) hit to the query protein, "is largely ineffective at predicting functional terms related to the BP ontology." (Radivojac, Clark, Oron, et al. 2013) Also top BLAST was outperformed by most other competitors. Interestingly half of the best performing methods included additional data sources, like for example co-expression and protein-protein-interaction (PPI) networks. This additional data might of course not always be available for query proteins, especially in a high throughput environment. Another interesting result of the experiment is, that most best performers are machine learned methods, with the disadvantage that they "require experience in selecting classification models (for example, a support vector machine), learning parameters, features or the training data that would result in good performance." (Radivojac, Clark, Oron, et al. 2013) The authors also point out some shortcomings of the applied performance assessment based on the F-measure. So are all terms considered equally important, even though the distribution of (reference) proteins over predictable GO terms greatly varies. Also are all of these reference proteins equally considered, "that is a correct prediction on a protein annotated with a shallow term (and its ancestors) is considered as good as a correct prediction on a protein annotated with a deep term." And "finally, in some cases, it is not clear whether to consider a prediction correct or erroneous; with our current approach, we consider only the experimental annotation and more general predictions to be correct." (Radivojac, Clark, Oron, et al. 2013)

While the CAFA provides a solution to the much needed global performance assessment of different protein function prediction methods, it does not evaluate how resource demanding an electronic tool is. Also complexity of installation on a local computer as well as availability on the web are not taken into account, indeed the research groups maintaining the competitive annotation tools were asked to provide the predicted GO term annotations for the query proteins themselves. In my opinion installation complexity or availability on the web are important traits of protein function prediction tools. Depending on the use case (chapter 1.1, page 12), a web page, where one can submit a number of query sequences enables a user to predict protein functions with great ease, while in an high throughput environment such web front ends are not applicable to the task at hand, because they typically limit the number of query sequences. As mentioned before (section 1.2, page 14), the "PFP" tool (Hawkins, Chitale, Luban, and Kihara 2009) for example is only available through the provided web page (*kiharalab.org/web/pfp.php*) allowing a maximum of 10 query sequences to be submitted at a time, while the web front end of the "GOtcha" tool (*compbio.dundee.ac.uk/gotcha/gotcha.php*) even accepts only a single query. Another similar example of function prediction tools, that are only available for limited analyses is the "GOfigure" web page (*udgenome.ags.udel.edu/frm_go.html*), that at the time of writing could not be accessed. These considerations demonstrate why on a genomic scale it is important to the user to have the option of installing the function prediction software locally on their own computing environment. Two frequently used tools meeting this requirement are InterProScan (Zdobnov and Apweiler 2001) and Blast2GO (Conesa and Gotz 2008). If installed locally the computational resources such a tool demands become also important. Because short analyses can be done using provided web interfaces local computation should be applicable on large sets of query proteins within reasonable memory, processor, and time requirements.

As explained earlier, protein function is also annotated in the form of short descriptions like those found in public protein databases (chapter 1, page 12). Although some methods exist to annotate query proteins with such descriptions, no evaluation of the annotation accuracy has been carried out to our knowledge. Hence the need for a method to enable the comparison of electronic annotation tools carrying out this task.

2. Motivation

Many of the introduced methods are good classifiers for a number of protein functions (Pierri, Parisi, and Porcelli 2010; Hawkins and Kihara 2007; Rentzsch and Orengo 2009, and section 1.10, page 18). Although their applicability greatly depends on the use case (section 1.1, page 12). Where, as mentioned before, the first common use case, typically encountered in pharmaceutical research (Pierri, Parisi, and Porcelli 2010), is the characterization of a low number of query proteins, possibly just a single one, while predicting protein functions in a high throughput environment or on genomic scales, is the second also frequently encountered use case.

The reasons the presented methods are not widely applied on genomic scales or in high throughput environments are manifold and have partially been introduced. One such reason is that some methods simply are not available as programs, while others are only accessible through web forms that only accept a limited amount of query sequences at a time (Hawkins, Chitale, Luban, and Kihara 2009, and section 1.10, page 18). Some only have been evaluated on a very limited set of reference proteins and hence little is known about their accuracy and reliability (Hawkins and Kihara 2007). While those relying on inherent sequence characteristics often are binary classifiers like support vector machines or neural networks (Pierri, Parisi, and Porcelli 2010), that require training with carefully selected up to date data before application, and frequently come with the inherent restrictions to be computationally resource demanding, at least when applied to predict a fairly representative set of protein characteristics on genome scale or in high throughput environments. Finally using protein function prediction methods that rely on more input data than the pure query sequences themselves must of course be provided with it, e.g. the secondary or tertiary structure of the protein or its expression correlations with other reference gene products. While the latter approach requires additional experiments e.g. with expression arrays, protein structure on the other hand can be predicted, where the prediction itself introduces specific strengths and weaknesses (Rentzsch and Orengo 2009). Thus these methods often come with the recommendation to manually inspect their results (Pierri, Parisi, and Porcelli 2010; Hawkins and Kihara 2007). Hence — to our knowledge — protein function prediction methods widely applied on genome scale or in high throughput environments are those based on sequence similarity.

2.1. Research Objectives

Founded on these observations was our motivation to develop *three new methods* of protein function prediction based on the sequence similarity approach, but extending it to achieve higher levels of accuracy and reliability, while still being applicable on large sets of query proteins:

- Automated Assignment of Human Readable Descriptions (AHRD) (*github.com/groupschoof/AHRD*; Hallab, Klee, Srinivas, and Schoof 2014),
- AHRD on gene clusters, and
- Phylogenetic predictions of Gene Ontology (GO) terms with specific calibrations (PhyloFun).

Developing these tools we especially aimed at meeting the requirements of predicting protein functions for large sets of query sequences (section 1.10, page 18). Hence the new methods should not only perform well in the context of accuracy, but also be reasonable in their computational resource demands, and finally should be easily installed on a local compute environment. Because so far no method existed to asses the quality of electronically annotated short protein descriptions, e.g. as assigned by AHRD, we also needed to develop new procedures to measure the accuracy of such short textual descriptions.

2.2. Automated Assignment of Human Readable Descriptions (AHRD)

Often a Biologist's first contact with new proteins is through their description, for example when searching a database with a similar amino acid sequence. Hence a method to assign concise, trustworthy and human readable descriptions to proteins is needed. The two most commonly used methods as mentioned before have been for one passing the description of the most similar protein found in sequence similarity searches, while the other method is provided by the Blast2GO suite's "annot" function. We developed a new method that assigns human readable descriptions to query proteins based on a lexical analysis of the candidate descriptions (*github.com/groupschoof/AHRD*; Hallab, Klee, Srinivas, and Schoof 2014).

We further evaluated how taking into account similarity of domain architecture could improve AHRD's annotations (Bangalore 2013). To evaluate and subsequently score the similarity of a protein pairs' respective domain architectures we used the "cosine similarity measure" (Lee and Lee 2009), for which we modified the domain weight formula and computed the weights it is based on for all available Protein Domains in the public InterPro database (Apweiler, Attwood, Bairoch, et al. 2000).

2.3. AHRD on gene clusters

In many genome projects characterizing the newly generated query protein sequences involves putting them into their respective gene family context in order to elucidate their evolutionary relationships and history. This enables detection of organism specific genes, or function expansion or loss, respectively, where as mentioned before these expansions or losses in turn may help understanding the organism's own evolution. Furthermore the phylogenetic reconstruction of gene families enables estimation of the query's origin and, using for instance molecular clock approaches, also its age (Wang, Jiang, Kim, et al. 2011; Weir and Schluter 2008; Kimura 1968; Battistuzzi, Feijao, and Hedges 2004; Battistuzzi and Hedges 2009). Such phylogenies, when compared to manually curated species trees, can also be used to identify duplication and speciation events, respectively (Zmasek and Eddy 2001). Finally comparing these gene family phylogenies with reference trees (Shimodaira and Hasegawa 1999; Lerat, Daubin, and Moran 2003) or finding unexpected species compositions in them (Nelson, Clayton, Gill, et al. 1999) can support the identification of Horizontal Gene Transfer events.

The method applied to generate these gene families typically clusters query proteins and selected references by their pairwise similarity. For instance for the tomato proteome (Consortium 2012) we generated over 17000 gene families from the tomato query proteins and references obtained from the rice (Project 2005), grape (Jaillon, Aury, Noel, et al. 2007), and arabidopsis (Arabidopsis Genome Initiative 2000) genomes. This large number of families hindered their further investigation, because identification of families of interest was not straight forward due to the lack of short, concise, trustworthy and Human Readable Descriptions (HRD) that summarized the type of family for the expert Biologists. Hence we developed a new simple method "AHRD on gene clusters" to annotate these clusters with such HRDs. It identifies InterPro Families (Apweiler, Attwood, Bairoch, et al. 2000)

each cluster's genes are attributed with and then uses the *most frequently* annotated InterPro Family as the cluster's HRD, while the frequency itself serves as the annotation score. If no InterPro Family exceeds the annotation frequency threshold of 50% other types of InterPro annotations are utilized, for instance InterPro Domains.

2.4. GO term predictions based on Phylogenetic reconstruction (PhyloFun)

The Gene Ontology (GO) (Ashburner, Ball, Blake, et al. 2000) provides a standardized hierarchical vocabulary to describe the molecular function, involvement in biological process, and cellular localisation of gene products. As mentioned earlier (section 1, page 12) this widely used vocabulary enables the computational analyses of gene product characteristics both on the individual as well as on the systematic level. From the reasons explained before stemmed the motivation to develop a fast and accurate GO annotation method based on sequence similarity aiming to incorporate the accuracy of trained machine learning algorithms while still being applicable on large sets of query proteins. For each such query PhyloFun starts a sequence similarity search in a database of selected reference proteins, then reconstructs a phylogenetic tree (Felsenstein 2004) and subsequently uses it as input to an implementation (Højsgaard 2012; Engelhardt, Jordan, Muratore, and Brenner 2005; Engelhardt, Jordan, Srouji, and Brenner 2011) of the Belief Propagation algorithm (Pearl 1988) (an explanatory summary is given in chapter 3, page 23) to compute probabilities for each distinct GO annotation found in the homologs. In this, annotations found *only* in distant relatives receive a low probability while those found in *close* relatives receive higher ones, because the loss of a protein characteristic becomes more likely the more the respective homologs have diverged, i.e. the more evolutionary time has passed and allowed for the accumulation of non synonymous mutations.

PhyloFun was developed in two versions. The first was a pipeline (Jöcker 2009) constructed around the Sifter (v1.2) (Engelhardt, Jordan, Muratore, and Brenner 2005) annotation program, which when computing probability distributions for the respective set of candidate GO terms at any given phylogenetic node takes into account the branch length to the parent node, the evolutionary event that took place, which is either a speciation or duplication event, and finally the relatedness of the respective candidate GO terms in terms of their distance, measured as number of edges between them in the Gene Ontology directed acyclic graph (GO-DAG) (Ashburner, Ball, Blake, et al. 2000; Engelhardt, Jordan, Muratore, and Brenner 2005; Hawkins and Kihara 2007).

The second version of PhyloFun (v2.0) was implemented to function without depending on local and tedious to maintain databases, nor manually curated species trees. But most importantly its motivation was to base the computation of GO term annotation probabilities on empirical measurements of pairwise sequence distances rather than on a preconceived probability model like the one used in Sifter (Engelhardt, Jordan, Muratore, and Brenner 2005; Engelhardt, Jordan, Srouji, and Brenner 2011). Furthermore we aimed to avoid, by the exclusive usage of *trustworthy* sources, the propagation of annotation errors (Gilks, Audit, Angelis, et al. 2002). To this end all available proteins (Bairoch and Apweiler 2000; Boeckmann, Bairoch, Apweiler, et al. 2003) with *trustworthy* — experimentally verified, or curator made — GO annotations were used to empirically measure the probability distributions of a GO annotation getting lost depending on pairwise sequence distance, which was inferred as the expected amount of character change (typically measured as phylogenetic branch length). The usage of this individual calibration of mutation probability distributions *for each trustworthy annotated GO term* was motivated by the fact that some protein characteristics are lost after accumulating only a few mutations while others can resist greater amounts. These results could then directly be used to infer conditional GO term annotation probability distributions for any given

node in a phylogenetic tree.

2.5. Genome scale application

We applied both "Automated Assignment of Human Readable Descriptions" and "PhyloFun" on the recently published genomes of tomato and the leguminous plant *Medicago truncatula*. The resulting human readable descriptions were used by members of the respective genome consortia, the gene family experts, to further investigate the roles the respective gene products assume in both organisms. Subsequent over and under-representation-analysis for annotated GO terms revealed adaptations of the two plants.

3. Belief Propagation

Pearl first proposed to represent structured knowledge in a probabilistic Bayesian Network (Pearl 1988) defined as a directed acyclic graph $G = (V, E)$, in which each node represents a random variable $R_i \in V : \Omega_i \to \mathbb{R}$ and each directed edge $(R_i \to R_j) \in E$ indicates statistical dependency of R_j on R_i. The Bayesian Network is fully defined through its joint probability distribution

$$P(R_1, \ldots, R_n) = \prod_{i=1}^{n} P(R_i \mid pa(R_i)), \text{ where} \tag{3.1}$$

$pa(R_i)$ is the set of parental nodes R_i *directly* depends on.

From this joint distribution the marginal probability $P(R_i = r_i)$ can be inferred as

$$P(R_i = r_i) = \sum_{(\phi_1, \ldots, \phi_n) \in \Psi} P(r_{\phi_1}, \ldots, r_i, \ldots, r_{\phi_n}), \tag{3.2}$$

where each random variable's R_i set of outcomes has indices $\Phi_i = \{\phi : r_\phi = R_i(\omega), \omega \in \Omega_i\}$ and Ψ is the Cartesian product of these index sets $\Psi = \prod_{i=1}^{n} \Phi_i$.

The so structured probabilistic knowledge can be used to infer the most likely state of any random variable of interest — for example the function annotation of a given query protein. To do so some of the network's random variables are set to observed evidence $E = \{e\}$, which in our example would be the known protein functions of the query protein's found homologs. This initialization of evidential random variables actually sets their probability of observed evidential event to one, thus discarding other events at these evidential nodes. Furthermore the network structure enables not only the application of such observed "diagnostic", bottom up, evidence E^-, but also the application of "predictive", top down, evidence E^+, which in our example could be setting the root node's probability distribution such that each different protein function, found in the query's homologs, is initialized to its respective prior, the observed annotation frequency in a suitable reference set. Having fed diagnostic and predictive evidence into the Bayesian Network, Belief Propagation recursively spreads the current strength of the predictive and diagnostic support *independently* from each node to its ancestors and descendants. Subsequently the probability of events of interest at a node of interest can be inferred as the so called "belief under evidence". Finally this enables querying the network for joint or marginal distributions given the fed in evidence.

The Belief Propagation algorithm can spread the current evidential strength through the network recursively and independently for diagnostic and predictive evidence due to the their statistical *independence* and the fact that each node only depends on their *direct* ancestors. Hence in a simple example network $X \to Y$ the belief of a selected event x at its respective node given evidence $E = \{Y = e\}$ can be computed as follows:

$$BEL(x) = P(x|e) = \frac{P(e|x) \cdot P(x)}{P(e)} = \alpha \cdot P(e|x) \cdot P(x), \text{ where} \tag{3.3}$$
$$P(e|x) \text{ is defined by the conditional probability matrix } M_{Y|X}.$$

After inserting another node in between the diagnostic evidence and the node of interest $X \to Y \to Z, E = \{Z = e\}$ the belief $BEL(x) = P(x|e) = \alpha \cdot P(e|x) \cdot P(x)$ still can be computed, even though the likelihood of the diagnostic evidence $\lambda(x) = P(e|x)$ can no longer be obtained from conditional probability matrix $M_{Y|X}$ *directly*, because Y separates Z from X. But the diagnostic support can be spread *recursively*

$$\begin{aligned}\lambda(x) &= \sum_y P(e|y,x) \cdot P(y|x) \\ &= \sum_y P(e|y) \cdot P(y|x) \text{ because } Z \text{ is independent of } X \\ &= M_{Y|X} \bullet \lambda(y).\end{aligned} \tag{3.4}$$

Hence node X can still calculate it's likelihood vector $\lambda(x)$, if it gains access to the likelihood vector $\lambda(y)$ of it's successor.

Let us now consider how "predictive" evidence is spread through a simple Bayesian Network. First the example network is further extended with two nodes inserted above the former root: $e^+ \to T \to U \to X \to Y \to Z \to e^-$, where the predictive evidence e^+ is set as the expected event observable at the root node. Because the Evidence E can be separated into two statistically independent sets of predictive evidence $\{e^+\}$ and $\{e^-\}$ we compute their respective support at node X independently with

$$\pi(x) = P(x|e^+), \text{ and} \tag{3.5}$$
$$\lambda(x) = P(e^-|x). \tag{3.6}$$

Because of the independence of predictive and diagnostic evidence the just introduced method to compute a node's λ message still applies in the above example network, while the current strength of the *predictive* support can be inferred as follows:

$$\begin{aligned}\pi(x) =& P(x|e^+) \\ =& \sum_u P(x|e^+, u) \cdot P(u|e^+) \\ =& \sum_u P(x|u) \cdot P(u|e^+) \\ =& \sum_u P(x|u) \cdot \pi(u) \\ =& \pi(u) \bullet M_{X|U}\end{aligned} \tag{3.7}$$

How to compute $\lambda(x)$ for a node X with multiple descendants $Y_1, \ldots, Y_n$? Because of the *conditional independence* of the descending nodes from each other, their current strength of diagnostic evidence

for the parent node X can be inferred directly. Hence the likelihood of any Node X with n descendants Y_i, each with likelihood $\lambda_{Y_i} = P(e^-_{Y_i}|x)$, is inferred by

$$\lambda(x) = \prod_{Y_i} \lambda_{Y_i}(x). \tag{3.8}$$

In the example Bayesian Network of figure 3.1 (page 26) the edge leading from node U to node X splits the network into its upper part, containing the parental node U, and the networks lower part, that contains the descending node X. During Belief Propagation the current strength both of predictive and diagnostic evidence *of the upper network part* is gathered in the $\pi_X(u)$ message passed from node U to its descendent X. To compute this message the parental node U requires the λ messages of its two *other* descendants V and W, as well as the predictive support descending from the root of the network into node U.

$$\begin{aligned}
\pi_Y(x) =& P(x|e^+_Y) = P(x|e^+_X, e^+_Z) \\
=& \frac{P(x, e^+_X, e^+_Z)}{P(e^+_X, e^-_Z)} \\
=& \frac{P(e^+_X) \cdot P(x|e^+_X) \cdot P(e^-_Z|x, e^+_X)}{P(e^+_x) \cdot P(e^-_Z|e^+_X)} \\
=& \alpha \cdot P(x|e^+_X) \cdot P(e^-_Z|x, e^+_X) \\
=& \alpha \cdot P(x|e^+_X) \cdot P(e^-_Z|x) \\
=& \alpha \cdot \pi_X(x) \cdot \lambda_Z(x)
\end{aligned} \tag{3.9}$$

Hence for any given node Y_i with Parent X its π message can be computed as the ancestor's belief given the descendant's current strength of diagnostic support:

$$\pi_{Y_i}(x) = \frac{BEL(x)}{\lambda_{Y_i}(x)} = \alpha \cdot \pi(x) \cdot \prod_{k \neq i} \lambda_{Y_k}(x) \tag{3.10}$$

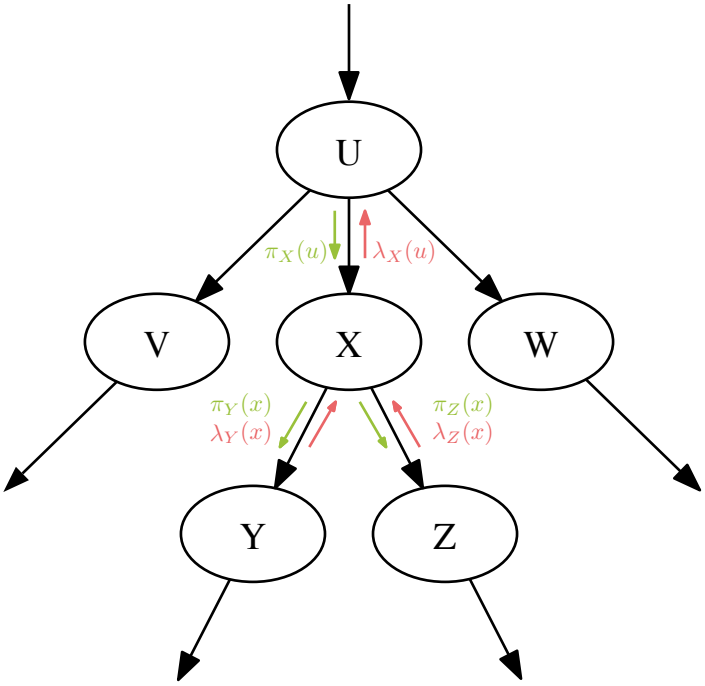

Figure 3.1.: Message Passing in a Bayesian Network — Messages passed to and from the Node X (Pearl 1988). The current strength of diagnostic support, the λ messages, are shown in red, while the current strength of predictive support, the π messages, are shown in green Belief propagation passes these messages recursively and independently. (Pearl 1988)

Figure 3.2 (page 27) shows how the recursive Belief Propagation algorithm works on a Bayesian Tree.

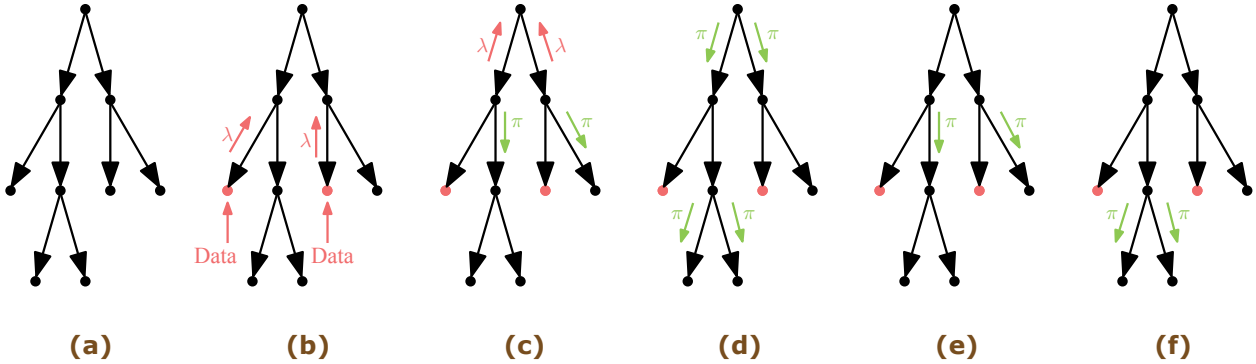

Figure 3.2.: Illustration of the recursively passed messages during Belief Propagation (Pearl 1988). (a) Structured knowledge is represented as a Bayesian Tree. (b) Two evidential nodes are initialized to found diagnostic data, which causes spreading of λ messages towards the root of the tree. (c) Nodes having received λ messages propagate their π messages to their descendants, excluding the evidential nodes (shown as red filled circles). (d–e) Once the root node has received its λ messages according π messages are propagated iteratively towards the tree's leaves, again leaving out the already initialized evidential nodes. — λ messages are shown in red, and π messages in green. Red filled circles indicate nodes initialized to diagnostic evidence data. (Pearl 1988)

So far Belief Propagation has been explained as applied on Bayesian Trees. But the method is also applicable on directed acyclic graphs, dubbed "Poly-Trees" by Pearl, in which inner nodes may have multiple ancestors. In this case the π message received by node X with n ancestors U_i can be computed as

$$\begin{aligned}
\pi(x) =& P(x|e_X^+) \\
=& P(x|e_{U_1X}^+, \ldots, e_{U_nX}^+) \\
=& P(x|e_{U_1X}^+) \cdot \ldots \cdot P(x|e_{U_nX}^+) = \prod_{i=1}^{n} \pi_{U_i}(x) \\
=& \sum_{u_1,\ldots,u_n} P(x|u_1,\ldots,u_n) \cdot P(u_1,\ldots,u_n|e_{U_1X}^+,\ldots,e_{U_nX}^+) \\
=& \sum_{u_1,\ldots,u_n} P(x|u_1,\ldots,u_n) \cdot P(u_1|e_{U_1X}^+) \cdot \ldots \cdot P(u_n|e_{U_nX}^+) \\
=& \sum_{u} P(x|u) \cdot \prod_{i=1}^{n} \pi_x(u_i). \qquad (3.11)
\end{aligned}$$

3.1. Maximum likelihood inference of Phylogenetic Trees — an application of Belief Propagation

The introduced method has numerous applications. One of the most commonly known in the context of Bioinformatics is Felsenstein's maximum likelihood inference of phylogenetic trees (Felsenstein 2004). This method in fact only applies the "bottom-up" part of Belief Propagation, i.e. the passing of λ messages from the leaves towards the root of the tree. In this the likelihood of node k to assume state $s \in \{A, C, G, T\}$ at site i given descendent branches l and m, in states x and y respectively, is

computed by

$$L_k^{(i)}(s) = (\sum_x P(x|s,t_l) \cdot L_l^{(i)}(x)) \cdot (\sum_y P(y|s,t_m) \cdot L_m^{(i)}(y)). \tag{3.12}$$

Formula 3.12 (Felsenstein 2004) is applied recursively until the resulting likelihood at the root node gives that of the whole tree, which subsequently enables the discovery of optimal topologies of maximum likelihood.

Part II.

Material and Methods

4. Public resources, software, and programming languages

4.1. Databases

Sequence similarity searches and those for conserved protein domains were executed in the following publicly available protein databases.

UniprotKB Swissprot and trEMBL (Boeckmann, Bairoch, Apweiler, et al. 2003; Bairoch and Apweiler 2000). Annotation of the Tomato and *Medicago truncatula* genomes were based on versions as available in January 2012 and July 2011, respectively. The evaluation and optimization of "Automated Assignment of Human Readable Descriptions (AHRD)" (*github.com/groupschoof/AHRD*; Hallab, Klee, Srinivas, and Schoof 2014) used the version from July 2011, while the version as available in December 2012 was used in the PhyloFun (v2.0) project.

TAIR The *Arabidopsis* information resource (Poole 2007) was obtained in January 2012 in order to annotate the Tomato genome and downloaded in July 2011 for the annotation of the *Medicago truncatula* genome, respectively. Both were version 9, while the optimization and evaluation of AHRD was executed using TAIR version 10.

InterPro The integrated documentation resource for protein families, domains and functional sites database (Apweiler, Attwood, Bairoch, et al. 2000) was obtained in January 2012 for the Tomato, and in July 2011 for the annotation of the *Medicago truncatula* genomes, respectively.

4.1.1. Reference Sets

The following protein databases were used for calibration or parameter optimization and in the accuracy assessment of competitive annotation methods' predictions. The first three reference sets were used within in the AHRD project, while the last two were applied in the PhyloFun (v2) development. All protein databases are available in the electronic supplement.

"Tomato" 1132 manually curated, expert annotated proteins from the recently published tomato genome (section 5.3, page 35). Supplemental file: `tomato.fasta`

"Swissprot" 1000 proteins, that had a creation date in July 2011, were randomly extracted from the UniprotKB/Swissprot (section 5.3, page 35). Supplemental file: `swissprot.fasta`.

"*B.graminis*" 1419 manually curated, expert annotated proteins from the recently completed Blumeria graminis fungal genome (section 5.3, page 35). Supplemental file: `b_graminis.fasta`.

"trust-UniKB" All available proteins in the public UniprotKB database that had GO term annotations with one of the following evidence codes (*geneontology.org/GO.evidence*): "`EXP`", "`IDA`", "`IPI`", "`IMP`", "`IGI`", "`IEP`", "`TAS`", and "`IC`" (section 7.2, page 43). Supplemental file of all included Uniprot accessions: `trust-UniKB_accessions.txt`.

"PF-test" 1000 randomly selected proteins from the trust-UniKB set (section 7.2.4, page 47). Supplemental file: `PF-test.fasta`.

4.2. Genomes

The following genomes were annotated or used as references for optimization and evaluation of the AHRD (*github.com/groupschoof/AHRD*; Hallab, Klee, Srinivas, and Schoof 2014) and PhyloFun projects.

Powdery Mildew *Blumeria graminis* (Spanu, Abbott, Amselem, et al. 2010)

Tomato *Solanum lycopersicum* (Consortium 2012)

Barrel Clover *Medicago truncatula* (Young, Debellé, Oldroyd, et al. 2011)

Thale Cress *Arabidopsis thaliana* (Arabidopsis Genome Initiative 2000)

Grapevine *Vitis vinifera* (Jaillon, Aury, Noel, et al. 2007)

Soybean *Glycine max* (Schmutz, Cannon, Schlueter, et al. 2010)

4.3. Software

The following software was used to annotate the Tomato and the *Medicago truncatula* genomes, or was incorporated into the AHRD (*github.com/groupschoof/AHRD*; Hallab, Klee, Srinivas, and Schoof 2014) or PhyloFun projects.

InterProScan Conserved protein domains were identified with InterProScan (Zdobnov and Apweiler 2001), where version 4.5 was used to annotate the Tomato and *Medicago truncatula* genomes, and also employed in the PhyloFun (v1.0) pipeline. AHRD and PhyloFun (v2.0) were applying release candidate 5, version 5 (RC5−5).

BLAST Sequence similarity searches were executed with the Basic Search and Alignment Tool (Altschul, Madden, Schaffer, et al. 1997; McGinnis and Madden 2004). Version 2.2.21 was used for the annotation of the Tomato and *Medicago truncatula* genomes, as well as inside the PhyloFun (v1.0) pipeline. Optimization and evaluation of AHRD and the PhyloFun (v2.0) pipeline used BLAST version 2.2.25.

HMMER Sequence similarity searches based on profile Hidden Markov Models used to produce input for the PhyloFun (v2.0) pipeline were executed with the HMMER suite version 3 (Eddy 2011).

Mafft Multiple protein sequence alignments based on the chemical properties of their respective amino acid residues were generated with MAFFT version v6.851b (Katoh, Misawa, Kuma, and Miyata 2002).

GBlocks Filtering of multiple sequence alignments (MSA) for regions of conserved positions was introduced into the PhyloFun (v2.0) pipeline in form of the GBlocks (v0.91b) program (Castresana 2000; Talavera and Castresana 2007).

FastTree Maximum likelihood phylogenetic trees with Shimodaira-Hasegawa local support values (Shimodaira and Hasegawa 1999) for usage in the PhyloFun (v2.0) pipeline were generated with the FastTree (v2.1.7) (Price, Dehal, and Arkin 2009) program in multi processor mode ("FastTreeMP").

BioNJ, BioML If the respective MSAs consisted of more than 10 sequences phylogenetic trees in the first PhyloFun (v1.0) pipeline were generated using the Bio-Neighbor-Joining algorithm (Gascuel 1997) implemented in the bionj program, while trees of less than 11 sequences were generated with the maximum likelihood algorithm (Felsenstein 2004) implemented in the PhyML program (Guindon, Dufayard, Lefort, et al. 2010).

Blast2GO Annotation of query proteins with Gene Ontology terms and descriptions were made with Blast2GO (Conesa and Gotz 2008), whose pipeline version (pipe v2.5) was used to evaluate the performance of AHRD, while both the pipeline as well as the graphical user interface (v2.6.4) were used in the evaluation of PhyloFun (v2.0). The pipeline version was used with a local database, which was set up in September 2011 for the AHRD project, and again in March 2013 for PhyloFun (v2.0).

Archaeopteryx / Forester Display and export of phylogenetic trees for this thesis (v0.968 beta BG) as well as inferring the evolutionary type of inner phylogenetic tree nodes in the PhyloFun (v1.0) pipeline was done with the Archaeopteryx / forester program (v0.957 beta) (Zmasek and Eddy 2001).

Sifter Sifter (v1.2) (Engelhardt, Jordan, Muratore, and Brenner 2005; Engelhardt, Jordan, Srouji, and Brenner 2011) was used in the PhyloFun (v1.0) pipeline to annotate query proteins with GO terms.

OrthoMCL Gene families as clusters of amino acid sequences with significant similarity were generated using the Markov Clustering algorithm implemented in OrthoMCL (v2.0.3) (Van Dongen 2008; Li, Stoeckert, and Roos 2003).

Apache Ant AHRD uses Apache Ant (v1.8.0) as build tool (*ant.apache.org*).

4.4. Programming languages

Programing languages used to implement the methods presented here were the following.

Java AHRD was implemented in Java (jdk v1.6), but is compatible for all versions $\geq$ 1.5.

Ruby "AHRD on gene clusters" was implemented in Ruby (v1.8.2).

Perl PhyloFun (v1.0) was implemented in Perl (v5) (Jöcker 2009).

R PhyloFun (v2.0) was written both in R (v2.15.2) (R Core Team 2012) as well as in

C++ Rcpp (v0.10.3) (Eddelbuettel and François 2011; Eddelbuettel 2013).

5. Automated Assignment of Human Readable Descriptions (AHRD)

5.1. Algorithm

AHRD's (*github.com/groupschoof/AHRD*; Hallab, Klee, Srinivas, and Schoof 2014) design aims to mimic a human expert curator. We have observed that curators do not only consider the most similar reference protein, but look for consistency among a number of similar proteins. To this end, AHRD uses as input three results computed for a protein sequence query: (1) BLASTP (Altschul, Madden, Schaffer, et al. 1997; McGinnis and Madden 2004) search results from several databases e.g. Swissprot, TAIR and trEMBL (Boeckmann, Bairoch, Apweiler, et al. 2003; Bairoch and Apweiler 2000; Huala, Dickerman, Garcia-Hernandez, et al. 2001; Poole 2007), (2) protein domain search results from InterProScan (Bairoch and Apweiler 2000; Zdobnov and Apweiler 2001) and (3) Gene Ontology (GO) annotations (Ashburner, Ball, Blake, et al. 2000), for example predicted by PhyloFun (Engelhardt, Jordan, Muratore, and Brenner 2005; Engelhardt, Jordan, Srouji, and Brenner 2011) or Interpro2GO (Zdobnov and Apweiler 2001).

The descriptions of the top 200 BLAST hits (based on e-value) from each database search form the set of candidate descriptions d_i. In some databases, e.g. Swissprot, the description lines contain structured information, for example taxonomy given as "OS=Arabidopsis thaliana". This data is stripped from the description. Each candidate description has to pass a blacklist, which filters descriptions starting with e.g. "hypothetical" or "similarity to", as these are highly probable to be descriptions from an automated genome annotation. Using these descriptions could lead to propagation of errors (Gilks, Audit, Angelis, et al. 2002).

Next, descriptions are split into words, and each unique word occurring in d_i is assigned a score. This score is computed from the sum of scores taken from all descriptions that contain the word, and takes into account the expected quality of descriptions per database, the BLAST bit score and the overlap between query and hit (see below). Filters and corrections include a blacklist of uninformative words like "protein", which are not considered in the scoring, and a score for co-occurrence of words. Additionally, predicted GO terms are used to preferentially select standard terminology as found in GO term descriptions.

The highest scoring description is assigned to the query and the database accession of the hit protein added to enable evidence tracking. This results in a description being transferred from a high-scoring BLAST match which contains words occurring frequently in the descriptions of high scoring BLAST matches, does not contain meaningless "fill words" and preferentially contains words also occurring in GO terms assigned to the query protein.

If InterProScan results are available, the most informative domain names are extracted and appended to the description. As most informative in this context we select the children in parent-child relationships, discarding contained subdomains. Additionally each line contains a quality code for the assigned human readable description. The quality code is composed of four characters, each being one of "-" (criterion not fulfilled) or "*" (criterion fulfilled). The criteria are, in order: (1) e-value of the BLAST result is <e-10 and bit score is >50; (2) overlap of the BLAST result is >60%; (3) top token score from lexical analysis is >0.5; (4) annotated gene ontology terms share words with the ones in

the description. The available output formats within AHRD are FASTA (*wikipedia/FASTA_format*) and `tab` delimited tables.

5.2. Scoring

For a query protein p we run a BLAST (Altschul, Madden, Schaffer, et al. 1997; McGinnis and Madden 2004) search against k different databases and retrieve a set of n description candidates d_i. We express the trust in each database as database weight w_k which is configurable by the user. For each d_i we store the database weight w_i, the bit score b_i of the BLAST alignment and the overlap score o_i, which is the fraction of the protein sequence length, averaged over query and hit, covered by the significant local alignment:

$$o_i = \frac{(QueryEnd - QueryStart + 1) + (SubjectEnd - SubjectStart + 1)}{QueryLength + SubjectLength}, \tag{5.1}$$

where $Query$ is the query protein's amino acid sequence,

$Subject$ the found hit protein's sequence,

$Start$ and End refer to the respective sequence position in the BLAST alignment,

$Length$ is the respective sequence length.

Each description d_i is split into a set of words or "tokens", where the set of tokens T contains all distinct words t_m found in all d_i. After passing blacklists words are scored using a linear combination of sequence similarity b_i, database weight w_i and overlap score o_i. For normalization, we compute the sum of scores over description candidates d_k containing the word divided by the total sum over all i descriptions:

$$ts(t) = \beta\frac{\sum_k b_k}{\sum_i b_i} + \omega\frac{\sum_k w_k}{\sum_i w_i} + \sigma\frac{\sum_k o_k}{\sum_i o_i}, \tag{5.2}$$

where $t \in T$,

$k \in K$ the set of the indices of description candidates that contain t,

β, ω, σ are configurable weights

We further penalize low-scoring words in order to more clearly distinguish high (T_{ifr}) from low (T_{non}) scoring words. Half the maximum token score is taken as threshold, and this value is subtracted from all token scores below the threshold to compute the adjusted token score:

Let $t_l \in d_i$

$$T_{ifr} = \left\{ t_l \mid ts(t_l) \geq \frac{\max_m ts(m)}{2} \right\} \tag{5.3}$$

$$T_{non} = \left\{ t_l \mid ts(t_l) < \frac{\max_m ts(m)}{2} \right\} \tag{5.4}$$

$$ts_{adjusted}(t_l) = \begin{cases} ts(t_l), & t_l \in T_{ifr} \\ ts(t_l) - \dfrac{\max_m ts(m)}{2}, & t_l \in T_{non} \end{cases} \tag{5.5}$$

where m indexes all words in T

In order to score a description candidate, all words t_l occurring in a description line d_i are considered. The token scores are summed and, in order to counteract bias towards longer or shorter descriptions, corrected by the proportion of informative/high-scoring tokens:

$$ls(d_i) = \frac{\mid T_{ifr} \mid}{\mid T_{ifr} \mid + \mid T_{non} \mid} \frac{\sum_{t_l \in d_i} ts_{adjusted}(t_l)}{\max\limits_{m} ts(t_m)} + gs(d_i) \tag{5.6}$$

$gs(d_i)$ is the gene ontology (GO) score which is based on GO term annotations (Ashburner, Ball, Blake, et al. 2000) of the query protein p, which we index by g. Each such term has its description GO_g and comes with a confidence probability cp_g. The gene ontology score is the sum of all those confidence probabilities, where the GO description shares a word with the description candidate d_i:

$$gs(d_i) = \sum_{t_l} \sum_{g:t_l \in GO_g} cp_g, \quad t_l \in d_i \tag{5.7}$$

Using the lexical score a final description score is calculated, combining the lexical score, blast score and a bonus if the exact same description (combination of words, considering order) occurs frequently among description candidates:

$$ds(d_i) = ls(d_i) + \delta \frac{b_i}{\max\limits_{n} b_n} + \alpha \; ps(d_i) \tag{5.8}$$

where:

- $ps(d_i)$ is the number of description candidates identical to d_i divided by the number of occurrences of the most frequent description candidate,
- δ, α are configurable weights,
- n is an index into all description candidates, like i, and
- b_i is the BLAST score for d_i (see above).

In the end the highest scoring description is assigned to the query protein.

5.3. Implementation, Evaluation and Optimization

AHRD 2.0 has been written in Java version 1.5 (*java.com*) and requires Apache Ant (*ant.apache.org*) for compilation. We designed it using the "test driven development" approach with the framework JUnit (*junit.org*). AHRD is configured using YML files (*yaml.org*) which allow adaptation of parameters and inclusion of an arbitrary number of reference databases. The blacklists are configurable and given as lists of regular expressions. The three different sets of proteins used to evaluate and optimize AHRD were obtained as follows: 1419 manually curated, expert annotated proteins from the recently completed *Bluemeria graminis* fungal genome (Spanu, Abbott, Amselem, et al. 2010) were selected as the "B. graminis" test set. To generate the "swissprot" test set 1000 proteins, that had a creation date in July 2011, were randomly extracted from the UniprotKB/Swissprot (Boeckmann, Bairoch, Apweiler, et al. 2003; Bairoch and Apweiler 2000) database version July 2011. Finally the "tomato" test set contains 1132 manually curated, expert annotated proteins from the recently published tomato genome. Genes from the tomato set mainly are proteins involved in pathogen resistance.

5.3.1. Reference sets' characteristics

To infer the description diversity found in a given reference protein set, I divided the number of distinct descriptions by number of contained proteins. Furthermore the frequency of each distinct protein description was assessed, after they had been blacklisted and filtered using the described procedure (section 5.1, page 33). Subsequently the number of most common descriptions was computed as the minimum number of descriptions that accounted for a fourth of the proteins in the reference set. Afterwords, those most common descriptions, found to account for the first quarter of proteins in a given reference set, were ignored, and the next common ones, accounting for the second, third, and finally fourth quarter of the references were measured iteratively. These measures were assessed with the goal to answer the question whether more diverse reference sets preferred different optimal parameters than less diverse do.

Subsequently was inferred, whether the proteins of the three reference sets were drawn from often annotated and frequently studied proteins or had a broader spectrum of functions, for example as found in complete eukaryotic angiosperm proteomes. In this context, sequence similarity searches were carried out, separately for each reference set, and in the three public protein databases UniprotKB/Swissprot, UniprotKB/trEMBL, and TAIR10. The following comparison of the results revealed which reference set had more hits of high sequence similarity in each of the public databases. UniprotKB/Swissprot entries undergo a manual revision by expert curators before they are added to the public database (Boeckmann, Bairoch, Apweiler, et al. 2003). Because the confidence of an expert curator in a candidate protein annotation is surely increased, the more reference proteins of high sequence similarity share the candidate function, a tendency can be expected to find more Swissprot entries of frequently annotated functions and currently favoured research interests. Thus a reference set with results showing such a tendency was interpreted to be drawn from frequently annotated proteins and less to resemble the protein function spectrum expected from a random selection of proteins from a complete proteome like for example that of *A.thaliana*. Finally, in order to obtain a measure of how alike, according to their function, any two proteins in the *B.graminis* reference set are, pairwise sequence identities were measured using BLAST (McGinnis and Madden 2004) with an E-Value cutoff of 10.0. After self matches had been excluded the distribution of these pairwise sequence identities was examined.

5.3.2. Reference set curation

The sequence similarity searches for proteins in the above three reference sets, *B.graminis*, Swissprot, and Tomato, were done with "blastp" (version 2.2.21) (Altschul, Madden, Schaffer, et al. 1997; McGinnis and Madden 2004) with an e-value threshold of 0.0001. For each query protein in these test sets we searched three different protein databases for similar sequences: UniprotKB/Swissprot (version July 2011) (Boeckmann, Bairoch, Apweiler, et al. 2003; Bairoch and Apweiler 2000), UniprotKB/trEMBL (version July 2011) (Boeckmann, Bairoch, Apweiler, et al. 2003; Bairoch and Apweiler 2000) and TAIR10 (Huala, Dickerman, Garcia-Hernandez, et al. 2001; Poole 2007). From these, to avoid self matches, we removed all proteins belonging to species *Solanum lycopersicum* and all proteins from the swissprot test set. Because the *Blumeria graminis* genome had not yet been published, none of its proteins were contained in the three searched protein databases. Gene ontology term annotations (Ashburner, Ball, Blake, et al. 2000) were obtained from matching InterProScan (version 4.5) (Apweiler, Attwood, Bairoch, et al. 2000; Zdobnov and Apweiler 2001) results to the InterPro2GO mappings (file version March 2nd 2011) (Apweiler, Attwood, Bairoch, et al. 2000; Zdobnov and Apweiler 2001) and using our in house pipeline PhyloFun (version 1.0) based on Sifter, version 1.2 (Engelhardt, Jordan, Muratore, and Brenner 2005; Engelhardt, Jordan, Srouji, and Brenner 2011).

5.3.3. Competitors and Quality-Assessment

AHRD's annotations were compared to two competitive methods. The first was the Blast2GO-Suite "b2g4pipe" version 2.5.0 (Conesa and Gotz 2008; Conesa, Gotz, García-Gómez, et al. 2005), which enables execution on the command line. The required Blast2GO database was downloaded and set up according to the provided documentation with the latest data available in July 2011. The second competitive method took protein descriptions from the best BLAST hits (Altschul, Madden, Schaffer, et al. 1997; McGinnis and Madden 2004) from the above three independent sequence similarity searches. We assessed AHRD's performance averaging the F2-score (Rijsbergen 1979) on every Human Readable Description (HRD) assigned by our program. The F2-score is calculated as the weighted harmonic mean of the two statistics precision and recall, both of which are based on counting shared words, ignoring case, in both the reference (REF) and the assigned HRDs. Treating the reference and the assigned description as mathematical sets of words, precision and recall can be calculated as follows:

$$precision = \frac{|REF \cap HRD|}{|HRD|}, \tag{5.9}$$

$$recall = \frac{|REF \cap HRD|}{|REF|}, \tag{5.10}$$

where $|\cdot|$ is the set cardinality.

AHRD's performance, measured as the average F2-score, was compared with the two other competitive annotation methods explained above.

5.3.4. Parameter optimization

Using this mean F2-score as the objective function, we were able to optimize AHRD's parameters and asses its robustness. To achieve this we implemented a simulated annealing approach (Kirkpatrick, Gelatt, and Vecchi 1983), and ran it on the three above test sets. In order to avoid overfitting, the resulting parameters found to be optimal for one test set were cross validated on the other two, respectively. In detail during each iteration of the optimization the mean F2-score was calculated for the currently used parameters, which then were accepted, if the score had improved compared to the currently accepted parameters. Worse performing parameters could also be accepted with the probability p_{acpt} depending on the current temperature T_c and the constant scaling factor k:

$$p_{acpt} = e^{-(F_2(a)-F_2(c))\cdot\frac{k}{T_c}}, \tag{5.11}$$

where $F_2(\cdot)$ is the mean F2-Score,
a is the accepted parameter set,
c is the currently evaluated parameter set,
T_c is the current temperature, and
k is the scale parameter.

Each iteration of this simulated annealing implementation cooled down its temperature by 1 degree, after which a neighboring set of the accepted parameters was generated and evaluated in the next iteration. This neighbor generation was achieved by slightly mutating a randomly selected parameter

by a value M based on two custom parameters c_1, c_2 and a Gaussian distributed random value r with mean 0 and standard deviation 1:

$$M = (r \cdot c_1) + c_2, \tag{5.12}$$
where c_1, c_2 are configurable weights.

We executed eight separate optimization runs with different start temperatures, number of starting parameter sets, and different values for c_1, c_2, and k as well as some differences in implementation. Of these optimization runs the first six still used a different formula to compute AHRD's "old overlap score". This formula computes the coverage on the query sequence *only* while the "new overlap score" (formula 5.1, page 34) takes into account the coverage on both the query and the hit (subject) sequences.

$$o_i = \frac{QueryEnd - QueryStart + 1}{QueryLength}, \tag{5.13}$$
where $Query$ is the query protein's amino acid sequence,
$QueryStart$ and $QueryEnd$ refer to the query sequence position in the BLAST alignment,
$QueryLength$ is the query sequence length.

We also increased the likelihood of the simulated annealing approach following the mean F2-Score slope uphill, which we achieved by introducing a probability of mutating that parameter again, that lead to an increase of mean F2-Score during the last iteration. This probability p_h was termed "hill climbing probability" and computed as follows:

$$p_h = \frac{e^{-(1-d)} + s}{e^0 + s}, \tag{5.14}$$
where
d is the increase in mean F2-Score achieved in the last iteration,
s scaling factor set to 0.7.

Its distribution is plotted in figure 5.1 (page 39).

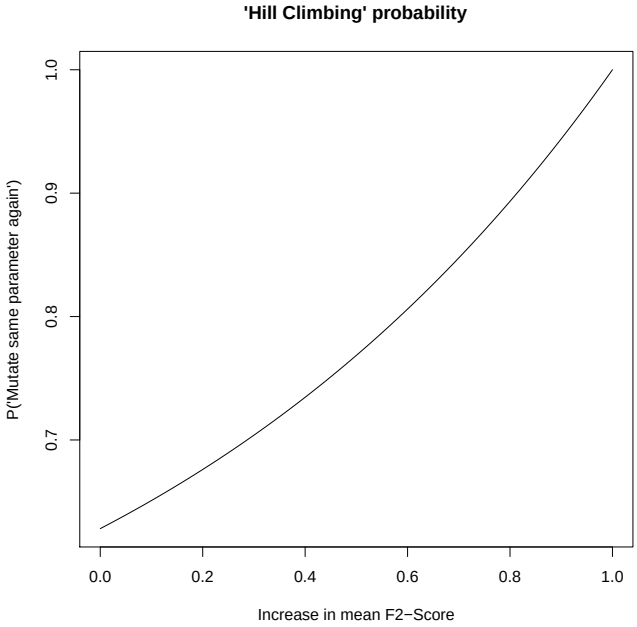

Figure 5.1.: Simulated annealing "Hill climbing probability" distribution

The parameters and method used in each simulated annealing run are summarized in table 5.1.

Table 5.1.: Simulated annealing parameters

Run	No. start points	Start Temperature	c_1	c_2	k	Use "Hill climbing"
One	1000	10000	1.5	1.5	3500000	No
Two	1000	30000	1.5	1.5	3500000	No
Three	1000	30000	0.25	0.25	15000000	Yes
Four	1000	30000	0.25	0.25	9250000	Yes
Five	10000	10000	0.25	0.25	7000000	No
Six	1000	10000	0.25	0.25	7000000	Yes
Seven	709571	1	0	0	0	No
Eight	6	50000	0.25	0.25	7000000	No

In order to estimate the proportion of the parameter space that had been evaluated we first approximated its size by limiting its axis to the intervals from zero to one for all weight parameters, or zero to a hundred for the database weights, respectively. Then we discretized each axis into a hundred distinct values and computed the size of the parameter space as 100^{n_a} where n_a denotes the number of different parameters subjected to optimization, and hence evaluates to 10. Furthermore each set of parameters generated and evaluated during simulated annealing was compared with all others in order to measure how many pairwise distinct parameter sets had been tested. To further estimate coverage of the parameter space and performance of the optimization itself, the fractions of parameter mutations that yielded an increase, decrease, or no change in F2-Score, respectively, was assessed, as well as the euclidean distances walked in parameter space by each simulated annealing

run. Finally we assessed the influence of the temperature on the used implementation of simulated annealing, specifically the current rates of accepting, or rejecting mutated parameter sets were measured on intervals of 1000 degrees and plotted together with the F2-Scores of currently accepted and all evaluated parameter sets.

Optimization runs seven and eight were done after switching to the introduced overlap scoring and discontinuing the usage of the "old overlap-score" (see chapter 5.3.4, page 38), which made new optimization necessary. After assessing six high scoring parameter sets in run seven we submitted those to further optimization in run eight.

5.4. Scoring Domain Architecture Similarity

We evaluated if AHRD's annotation quality could be improved by taking into account the similarity between the domain architectures of the query and that of each candidate description's protein (Bangalore 2013). This was achieved by scoring those descriptions better that come from candidates with a higher similarity measure, which was inferred by first constructing a vector space model of the respective domain architectures, then assigning each domain architecture a vector in this space and finally computing the cosine architecture similarity measure as proposed by Lee and Lee (Lee and Lee 2009). This similarity measure is based on precomputed weights for each Protein Domain which we calculated for all available Domains (Bangalore 2013) in the public InterPro database (Apweiler, Attwood, Bairoch, et al. 2000).

To find optimal parameters for the so extended AHRD (Dom-Arch-Sim-AHRD) we then used the above optimization approach (5, page 33), particularly searching for an optimal domain architecture similarity weight. Finally the quality of the protein descriptions generated by Dom-Arch-Sim-AHRD was evaluated and compared to the standard AHRD annotations in order to answer the initial question, if this approach could improve AHRD's annotation quality (Bangalore 2013).

6. AHRD on gene clusters

6.1. Algorithm

"AHRD on gene clusters" (*github.com/groupschoof/AHRD_on_gene_clusters*) was implemented to assign short, concise, trustworthy Human Readable Descriptions to gene families. These families are given as sets of amino acid sequences of significant similarity. To generate this method's input first a database of proteins is created and then a BLAST sequence similarity search (Altschul, Madden, Schaffer, et al. 1997; McGinnis and Madden 2004) in this database itself is conducted for every single contained protein. The bit scores of each resulting protein pair is then fed into a Markov Clustering algorithm (Van Dongen 2008) implemented in the program OrthoMCL (Li, Stoeckert, and Roos 2003). Further required input is the InterPro annotations of these proteins, as well as the latest InterPro database file itself (Apweiler, Attwood, Bairoch, et al. 2000).

"AHRD on gene clusters" then iterates over all input gene clusters, first filtering all InterPro annotations found in the current cluster's gene products in order to only retain the most informative InterPro entities, which are the *children* in the hierarchical parent child relations the InterPro database (Apweiler, Attwood, Bairoch, et al. 2000). The method then continues with measuring the frequency

of the retained InterPro annotations in the current cluster. And finally — as briefly mentioned before (section 2.3, page 21) — the cluster is assigned a description based on the most frequently annotated InterPro Family, if the Family's measured frequency meets the threshold 0.5, otherwise any type of InterPro annotation is utilized as a source of the description, for instance InterPro domains. In the latter case all available InterPro annotations are utilized. Finally if no type of InterPro annotation can be found attributed to a cluster's genes no annotation is generated. The resulting cluster descriptions are composed of four separate parts: The first part is the measured frequency of the selected InterPro annotation and is intended to give an estimate of the description's quality. The rest is composed of the InterPro identifier, followed by the type of the selected InterPro annotation, and finally terminates with the full name of the InterPro entity (see tables 9.2–9.4, page 68).

6.2. Human Readable Descriptions for Tomato gene families

We used "AHRD on gene clusters" to annotate the 17487 gene families with members found in the Tomato genome. These gene clusters had been generated by Manuel Spannagl (Consortium 2012) using the method described earlier (section 6, page 40), where the reference proteins were taken from *Arabidopsis thaliana*, *Vitis vinifera*, and *Oryza sativa*, while the InterPro (Apweiler, Attwood, Bairoch, et al. 2000) annotations for both the reference and the Tomato proteins were generated using the "Similarity Matrix of Proteins" (SIMAP) (Arnold, Rattei, Tischler, et al. 2005; Rattei, Arnold, Tischler, et al. 2006) in house pipeline on the InterPro database as available in December 2010. For this purpose the SIMAP pipeline used InterProScan, version 4.5, (Zdobnov and Apweiler 2001). The gene families consist on average of approximately *five* genes. A more detailed summary of the distribution of gene cluster sizes is given in the following table:

Table 6.1.: Summary of the distribution of Tomato gene family sizes in number of member genes.

Minimum	1st Quartile	Median	Mean	3rd Quartile	Maximum
2	3	4	5.06	5	177

7. PhyloFun

PhyloFun was implemented as a pipeline — and in two versions — to annotate query proteins with terms from the Gene Ontology (GO) (Ashburner, Ball, Blake, et al. 2000). Its first step is to find homologous proteins in a database of selected reference proteins, which then are used to reconstruct each query's evolutionary history as a phylogenetic tree. In the final step this tree is then interpreted as a Bayesian network and an implementation (Engelhardt, Jordan, Muratore, and Brenner 2005; Højsgaard 2012) of the message passing algorithm (Pearl 1988) is used to compute probabilities for each distinct GO term annotation found in the tree, that is in the query's homologs. These annotations receive a low probability of being accurate characterization of the query protein if they are found *only*

in distant homologs, and vice versa receive higher probabilities if they are found in close homologs and with a higher annotation frequency (see part I, page 12).

7.1. Version 1.0 (v1.0)

The first implementation of the PhyloFun pipeline (version 1.0) was done in Perl (version 5) by my colleague Jöcker in the context of her PhD thesis (Jöcker 2009). It starts, as mentioned above, for each query with searching a database of reference proteins , that contains only proteins from 42 fully sequenced genomes (table 7.1, page 43). This sequence similarity search is executed in two stringent steps in order to obtain highly similar results while extending the set of homologs by searching for inparalogous sequences for the ones found in the first round (Jöcker 2009). In the next step the resulting sets of homologous proteins are submitted to phylogenetic reconstruction, which first generates a multiple sequence alignment (MSA) based on the chemical properties of the amino acid residues, as implemented in the program "MAFFT" (Katoh, Misawa, Kuma, and Miyata 2002). The resulting MSA is filtered for highly conserved positions, discarding all positions that have a gap in more than 60% of the aligned sequences. Next the PhyloFun pipeline reconstructs the phylogenetic tree for each query protein either using Neighbor Joining (Saitou and Nei 1987) as implemented in the program "BioNJ" (Gascuel 1997) or the Maximum Likelihood method (Felsenstein 2004) implemented in the program "PhyML" (Guindon, Dufayard, Lefort, et al. 2010), where the Maximum Likelihood method is chosen only if the MSA contains no more than 10 homologs. Next the resulting phylogenetic tree is reconciled with a manually curated species tree in order to identify the inner nodes of the tree as evolutionary speciation or duplication events, for which an implementation of the SDI algorithm is used (Zmasek and Eddy 2001). In the final step this tree is interpreted as a Bayesian Network and submitted to an implementation of the message passing algorithm (Pearl 1988) in order to compute, for the query protein, annotation probabilities of each distinct GO annotation found in the tree, which is executed by the program Sifter (version 1.2) (Engelhardt, Jordan, Muratore, and Brenner 2005; Engelhardt, Jordan, Srouji, and Brenner 2011). Sifter computes at each node of the tree the conditional probability of one GO term mutating into another depending on the type of evolutionary event —speciation or duplication—, the branch length to the parent node, and the number of edges between the two considered GO terms in the hierarchical directed acyclic graph (GO-DAG), the Gene Ontology is organized in (Ashburner, Ball, Blake, et al. 2000). This results in a higher GO term mutation probability on longer branches, after duplication events, and for GO terms separated by few edges in the GO-DAG.

Together with Jens Warfsmann and Haili Song we fixed some errors in PhyloFun's code and updated the relational database this pipeline stores sequence, species and annotation data in, in order to prepare the prediction of protein functions for the *Medicago truncatula*, and *Solanum lycopersicum* proteomes.

Table 7.1.: PhyloFun (v1.0) — Species of the reference proteomes

Scientific names of the reference proteins' species included in the PhyloFun (v1.0) protein database.
Agrobacterium tumefaciens, *Anaplasma phagocytophilum*, *Arabidopsis thaliana*, *Bacillus anthracis str*, *Bacillus anthracis*, *Bos taurus*, *Caenorhabditis elegans*, *Campylobacter jejuni*, *Candida albicans*, *Carboxydothermus hydrogenoformans*, *Clostridium perfringens*, *Colwellia psychrerythraea*, *Coxiella burnetii*, *Danio rerio*, *Dehalococcoides ethenogenes*, *Dictyostelium discoideum*, *Drosophila melanogaster*, *Ehrlichia chaffeensis*, *Escherichia coli*, *Gallus gallus*, *Geobacter sulfurreducens*, *Homo sapiens*, *Hyphomonas neptunium*, *Leishmania major strain Friedlin*, *Leishmania major*, *Listeria monocytogenes serotype 4b*, *Magnaporthe grisea*, *Methylococcus capsulatus*, *Mus musculus*, *Neorickettsia sennetsu*, *Oryza sativa subsp*, *Oryza sativa*, *Plasmodium falciparum*, *Pseudomonas aeruginosa*, *Pseudomonas fluorescens*, *Pseudomonas syringae pv*, *Rattus norvegicus*, *Saccharomyces cerevisiae*, *Schizosaccharomyces pombe*, *Shewanella oneidensis*, *Silicibacter pomeroyi*, *Vibrio cholerae*

As mentioned before the Gene Ontology is organized as a hierarchical directed acyclic graph (DAG) whose nodes, representing the GO terms, with increasing depth are more specific or "fine" in terms of the gene product characterization itself (Ashburner, Ball, Blake, et al. 2000). Hence the mean distance to the GO DAG's root node (*GO level*) of an GO term is a measure of the description's fineness of protein molecular function, involvement in biological processes, and cellular localization. We measured these levels of GO term annotations, both in the Tomato and *Medicago truncatula* genomes, respectively.

To get a measure of GO annotation sensitivity in terms of which terms a given method was able to annotate, the number of pairwise distinct annotated GO terms was assessed and compared with that of other methods.

7.2. Version 2 (v2.0)

To overcome certain shortcomings of PhyloFun (v1.0) (see chapter 2.4, page 22, and chapter 13.1, page 95) we decided on a complete reimplementation of the pipeline in R (version 2.15.2). Version 2 does not require a local database, because it uses the publicly available Gene Ontology (GO) relational database (*European Bioinformatics Institute (EBI) mirror of the Gene Ontology MySQL database*) and the Uniprot Web-Service (*ebi.ac.uk/Tools/webservices/services/dbfetch_rest*) to obtain amino acid sequences and functional protein annotations, where both ensure the usage of up-to-date data. Furthermore the preconceived model (Engelhardt, Jordan, Muratore, and Brenner 2005; Engelhardt, Jordan, Srouji, and Brenner 2011) used to compute the conditional probability distributions of GO term annotations at each phylogenetic node for the message passing algorithm (Pearl 1988) was replaced by our new empirical model, which uses lookup tables of pre computed GO annotation mutation probabilities. These probabilities of a given GO annotation getting lost over certain evolutionary distances (phylogenetic branch lengths) were measured individually for each GO term using all available proteins in the public UniprotKB database (Bairoch and Apweiler 2000; Boeckmann, Bairoch, Apweiler, et al. 2003) with *trustworthy*, i.e. experimentally or curator made, GO annotations (*trust-UniKB*). Here the accepted evidence codes were: "`EXP`", "`IDA`", "`IPI`", "`IMP`", "`IGI`", "`IEP`", "`TAS`", and "`IC`" (*geneontology.org/GO.evidence*).

7.2.1. Measurement of Gene Ontology term mutation probabilities

This set of proteins with trustworthy GO annotations (trust-UniKB) (see previous chapter 7.2, page 43) was first submitted to an "all versus all" BLAST (McGinnis and Madden 2004; Altschul, Madden, Schaffer, et al. 1997) search, in which for each of the set's proteins others of significant sequence similarity from the same database were identified. This search was executed with an expectation value (E-Value) threshold of 1. The so obtained homologous protein pairs were then used to measure GO term mutation probabilities individually for each *trustworthy* GO term annotation found in the trust-UniKB protein set. This calibration was computed for each distinct GO term as follows. First all trust-UniKB proteins annotated with it, and those homologous to such an annotated protein, were identified and used as the calibration set for the currently processed GO term. Then for each homologous pair of proteins in the calibration set their pairwise sequence distances were measured as the expected amount of character change, which was computed by first globally aligning the protein pair with the "Smith-Waterman-Algorithm" (Waterman, Smith, and Beyer 1976) using the functions provided with the "Biostrings" R package (Pages, Aboyoun, Gentleman, and DebRoy 2013). Then, in the next step, the sequence distance was fitted on this global pairwise alignment using the maximum likelihood approach described by Yang (Yang 2000) which is implemented in the R package "phangorn" (Schliep 2011). Here "phangorn" bases the computation of a branch length's likelihood on a modified version (Kosiol and Goldman 2005) of the "Dayhoff Point Accepted Mutation Matrices" (Dayhoff and Schwartz 1978).

The so obtained pairwise sequence distances for all pairs in a GO terms calibration set were then ordered and subsequently used to count at each given distance the number of pairs *sharing* or *not sharing* the current GO term annotation, respectively. These counts were then used to define the GO term's mutation probability for a given sequence distance as the number of pairs with equal or less sequence distance sharing the GO term annotation divided by the total number of pairs with equal or less sequence distance. These mutation probabilities were then computed for each sequence distance found in the ordered list of homologous protein pairs in an ascending manner with applying the constraint, that the GO term's mutation probability could never decline, even if for a greater sequence distance more pairs were found that shared the GO term (figure 7.1, page 45). To compute these probabilities the random event space of GO term mutation g^{mut} and preservation g^{pres} was used to formally denote the probability of mutation for a given GO term g on a sequence distance d as

$$P(\, g^{mut} \,|\, d \,)\,. \tag{7.1}$$

Data:

- g the GO term to compute the mutation probability lookup table for
- $\mathcal{H}$ the *ordered* set of homologous protein pairs p_i, where
  at least one member has annotation g,
  with $\mathcal{G}_x$ the set of GO term annotations of protein x:
  $p_i = \{\, p_i^1, p_i^2 \mid \exists\ k \in \{1,2\} : g \in \mathcal{G}_{p_i^k} \,\}$, and
  each pair p_i has a sequence distance d_i such that: $d_{i-1} \leq d_i$

Result:

- M ordered set of mutation probabilities $P(\, g^{mut} \,|\, d_k \,)$, with
  $m_{k-1} \leq m_k$ for $d_{k-1} \leq d_k$.

$n_{not\cdot sharing} \leftarrow 0$
$n_{all} \leftarrow 0$
$m_{candidate} \leftarrow 0$
$m_{current} \leftarrow 0$
foreach $p_i \in P,\ 1 \leq i \leq |\,\mathcal{H}\,|$ **do**
  increase n_{all} by 1
  if $\exists\ k \in \{1,2\} : g \notin \mathcal{G}_{p_i^k}$ **then**
    increase $n_{not\cdot sharing}$ by 1
  end
  $m_{candidate} \leftarrow \frac{n_{not\cdot sharing}}{n_{all}}$
  if $m_{candidate} > m_{current}$ **then**
    $m_{current} \leftarrow m_{candidate}$
    append $m_{current}$ to M
  end
end

Figure 7.1.: Computation of the mutation probability lookup table for a GO term g, where "$x \leftarrow \xi$" means the variable x is assigned the value ξ, and $|\cdot|$ is the set cardinality.

The so generated GO term mutation probability lookup tables were stored in the dataset `p_mutation_tables_R_image` in the R package "PhyloFun" (*github.com/groupschoof/PhyloFun*) and are used within PhyloFun (v2.0) to compute for each phylogenetic node the conditional GO term annotation probability tables required as input for the message passing algorithm (Pearl 1988) as implemented in the R package "gRain" (Højsgaard 2012). The generation of these lookup tables could only be done for GO terms that had at least a single *trustworthy* annotation and at least a single protein pair in trust-UniKB, which was true for 18610 GO terms.

To estimate the spread of sequence distances mapped to the mutation probabilities of the *trustworthy* GO terms annotated in trust-UniKB we first binned the mutation probabilities into five intervals always including their upper bound, that were: $0.2, 0.4, \ldots, 1.0$. We then plotted the respective

maximum sequence distance that was found to be mapped to mutation probabilities in these intervals, where available, or set them to "NA" otherwise. Next the so generated pairs of maximum mutation probabilities and maximum sequence distances were used to generate box-plots, which we did not only for all GO terms we had generated lookup tables for, but also those of the following GO levels (see chapter 7.1, page 43): 2, 3, and $\geq$4.

7.2.2. The pipeline

PhyloFun (v2.0) was implemented as a phylogenetic pipeline that assigns each query protein a set of Gene Ontology (GO) terms, which is achieved by first generating a phylogenetic tree of the query and its found sequence homologs and then creating a Bayesian Network from the nodes of the tree and using a Message Passing Algorithm (Pearl 1988) to compute GO term annotation probability distributions for each node in the network. Finally the set of GO terms receiving highest probabilities at the query node is used as the pipeline's result.

The pipeline requires as input a file of query protein sequences in FASTA format (*wikipedia/-FASTA_format*), and the tabular result of a sequence similarity search for the queries homologs, which are required to have Uniprot (Bairoch and Apweiler 2000; Boeckmann, Bairoch, Apweiler, et al. 2003) accessions. These sequence similarity searches were carried out by two tools PHMMER (v3.0) (Eddy 2011) and BLAST (Altschul, Madden, Schaffer, et al. 1997; McGinnis and Madden 2004), where we applied the following three E-Value thresholds to searches carried out with both tools: 10, 10^{-3}, and 10^{-6}. All sequence similarity searches were executed in the trust-UniKB database, described earlier (see chapter 7.2, page 43).

Each so generated pair of input files, the query sequences in FASTA format (*wikipedia/-FASTA_format*) and their found sequence homologs in tabular format, was then fed into the PhyloFun (v2.0) pipeline. It first downloads the homologs' amino acid sequences using the Uniprot web service (*ebi.ac.uk/Tools/webservices/services/dbfetch_rest*), then generates from the query and its homologs a multiple sequence alignment (MSA) based on the chemical properties of the respective amino acid residues, which is implemented in the program MAFFT (v6.851b) (Katoh, Misawa, Kuma, and Miyata 2002). In the next step the pipeline filters this MSA for conserved positions with the program GBlocks (v0.91b), set to allow maximum half of the positions within a conserved block to be gaps (`-b5=h` command line parameter) (Castresana 2000; Talavera and Castresana 2007). And finally, before executing the GO term annotation itself, the pipeline reconstructs a Maximum Likelihood phylogenetic tree (Felsenstein 2004) using the program FastTree (v2.1.7) (Price, Dehal, and Arkin 2009). The parameters passed to the pipeline's programs were as follows:

Table 7.2.: Command line arguments for tools used in the PhyloFun (v2.0) pipeline

Tool	Version	Command line argument (s)
MAFFT	v6.851b	`--auto`
GBlocks	v0.91b	`-b5=h -t=p -p=n`
FastTree	v2.1.7	`no parameters used`

7.2.3. Query Protein Annotation

After having reconstructed the phylogenetic tree of the query protein and its found homologs, PhyloFun (v2.0) translates the tree into a Bayesian Network and uses an implementation (Højsgaard 2012) of the Message Passing Algorithm (Pearl 1988) to compute GO term annotation probability distributions

for candidate sets of GO term annotations found in the query's homologs. These candidate sets are used as a whole, that is, as a compound GO annotation $\mathcal{G}_i$ (figure 7.1, page 45), and are composed of each homologs' h_i atomic GO annotations, which as mentioned before are all those annotations that are experimentally verified or curator made. Hence all unique compound trustworthy GO annotations form the event space Ω from which PhyloFun selects the most likely as the query protein's annotation. This is achieved by applying the message passing algorithm once for each of the three different GO term ontologies: "biological process" (BP), "cellular component" (CC), and "molecular function" (MF) separately, whith the result of assigning each compound GO annotation $\mathcal{G}_i \in \Omega$ a probability of being the adequate characterization for the query protein. In this the event space Ω is reduced to its intersection with the currently processed GO ontology, such that at each such iteration the compound GO annotations G_i only contain atomic GO terms of the respective ontology.

To infer these GO annotation probability distributions PhyloFun (v2.0) needs to generate conditional probability tables $cpt(X)$ for each node X of the network, where the length d of the branch leading from the parental node Q to the node X is the expected amount of character change, or 0 for the root of the tree. These conditional probability tables are quadratic matrices $cpt(X)_{i,j}$ which hold for each parental compound GO annotation $\mathcal{G}_i$ the probability to evolve to another $\mathcal{G}_j$ along the phylogenetic branch of length d. In this the probability of the compound GO annotation getting lost on this branch, is set to the maximum mutation probability $P(\, g_k^{mut} \,|\, d\,)$ (notation 7.1, page 44) found for any of its contained atomic GO terms $g_k \in \mathcal{G}_i$, such that

$$P(X \neq \mathcal{G}_i \,|\, d, Q = \mathcal{G}_i) = \max_{g_k \in \mathcal{G}_i} P(\, g_k^{mut} \,|\, d\,)\,. \tag{7.2}$$

Subsequently the actual probability of the compound GO annotation $\mathcal{G}_i$ mutating to any other given compound GO annotation $\mathcal{G}_j \in \Omega, i \neq j$ is considered equally likely. Hence this probability can be computed as the fraction

$$P(X = \mathcal{G}_j \,|\, d, Q = \mathcal{G}_i) = \frac{P(X \neq \mathcal{G}_i \,|\, d, Q = \mathcal{G}_i)}{|\, \Omega \setminus \mathcal{G}_i \,|}, \quad i \neq j \tag{7.3}$$

Using 7.2 and 7.3 the conditional probability table for any node of the Bayesian network can be computed as

$$cpt(X)_{i,j} = \begin{cases} 1 - P(X \neq \mathcal{G}_i \,|\, d, Q = \mathcal{G}_i), & i = j \\ \\ P(X = \mathcal{G}_j \,|\, d, Q = \mathcal{G}_i), & i \neq j \end{cases} \tag{7.4}$$

PhyloFun can be run in two modes producing more or less *restrictive* results, respectively. The first one only uses the three compound annotations with highest probabilities — one for each GO ontology BP, CC, and MF —, while the other mode considers all GO annotations that received a probability greater than equal distribution, and thus is *less* restrictive, while the first "restrictive" mode *only* assigns composite GO annotations as they are found *as a whole* in at least a single homolog.

7.2.4. Evaluation

We evaluated the quality of GO term annotations made by PhyloFun and the competitors Blast2GO (Conesa and Gotz 2008) and InterProScan (Zdobnov and Apweiler 2001) on a set of 1000 randomly selected proteins (PF-test) from the trust-UniKB set, described earlier (section 7.2, page 43). Here the required sequence similarity searches were done in the protein database derived from trust-UniKB

by excluding the 1000 randomly selected query proteins (PF-search = trust-UniKB \ PF-test). As quality measure we computed the mean F2-Score, the weighted harmonic mean of the statistical quality measures precision and recall (Rijsbergen 1979). Both of these measures were based on identifying true and false positives, as well as true and false negatives. These were estimated by first identifying reference annotations as the *trustworthy* GO annotations available for the query proteins in PF-test. Trustworthy as explained earlier in this context means experimentally verified or curator made GO term annotations. Then all predicted GO annotations that were equal to reference annotations were considered *true positives*, while only those predictions that did not equal and were not — direct or indirect — ancestors of reference annotations were considered *false positives*. The latter definition was used, because for example the predicted GO term "binding" is not false if the reference is a "DNA binding" protein. Finally, in order to enable a more detailed interpretation of the competitive methods annotations we also computed the mean specificity and recall rates.

We compared the earlier described setups of PhyloFun (v2.0) with results from running different setups and versions of Blast2GO and the latest available version of InterProScan (table 7.3, page 49).

Blast2GO was used in two versions, one provided with a graphical user interface (B2G_gui, version 2.6.4) and another intended to be used via the command line (B2G_pipe, version 2.5). The GUI version was run with default settings and used to generate its own BLAST (Altschul, Madden, Schaffer, et al. 1997; McGinnis and Madden 2004) results from searches in the Uniprot Swissprot protein database (Bairoch and Apweiler 2000; Boeckmann, Bairoch, Apweiler, et al. 2003), which we did to most accurately mimic the usage a Biologist might apply on his set of query proteins. In contrast the latter command line version of B2G_pipe (v2.5) was used to apply Blast2GO on the same sequence similarity search results we fed into the different runs of PhyloFun. This was done in order to better enable the comparison of PhyloFun and Blast2GO, which unfortunately was not able to interpret the BLAST result files in XML format by default, but could be made to correctly read these files when the respective contents of `<Hit_def>` tags were modified as in the following example (figure 7.2.4, page 48):

The introduction of additional pipe separated parts to the accession

`Q8WW12` → `sp|Q8WW12|Batman|fake|In reality Bat-Sheep does all the work for Batman`

can be achieved with the following `sed` (*GNU sed (stream editor)*) command

```
sed -e 's/\(<Hit_def>\)\(\S\+\)\(<\/Hit_def>\)/\1sp|\2|Batman|fake|In reality
Bat-Sheep does all the work for Batman \3/g' original_blast_results.xml >
processed_blast_results.xml
```

Figure 7.2.: Blast2GO BLAST results XML pre parser. First line shows an example accession into which four pipe "|" symbols are introduced which is required by Blast2GO pipe (v2.5). This is achieved by calling the `sed` command as shown.

B2G_pipe (v2.5) was executed on two BLAST results, generated by searches in the PF-search protein database (see chapter 7.2.4, page 47) using the E-Value thresholds of 10^{-3} and 10^{-6}, and with different "Evidence Code" (Ashburner, Ball, Blake, et al. 2000) weights. These weights control which GO term annotations Blast2GO accepts as source for its own annotations, so that a user can allow any type of GO term annotation, which is the default setup, or use only *trustworthy* annotations by setting all weights except those for experimental verification and manual curator annotation to zero.

Both setups were used and their results compared with those of the other annotation methods.

InterProScan (v5-RC5) was also applied to generate GO term annotations for the PF-test query proteins, which was done with default settings.

Table 7.3.: PhyloFun (v2.0) and competitor methods and their setups.

Abbreviation	Tool Version	Input, Setup
PhyloFun_PH	PhyloFun (v2.0)	on PHMMER E≤10
PhyloFun_PH_high_scr	PhyloFun (v2.0)	on PHMMER E≤10, all annos
PhyloFun_E-6	PhyloFun (v2.0)	on Blast E≤10^{-6}
PhyloFun_E-6_high_scr	PhyloFun (v2.0)	on Blast E≤ 10^{-6}, all annos
PhyloFun_E-3	PhyloFun (v2.0)	on Blast E≤10^{-3}
PhyloFun_E-3_high_scr	PhyloFun (v2.0)	on Blast E≤ 10^{-3}, all annos
B2G_gui	Blast2GO GUI v2.6.4	on Swissprot, default settings
B2G_pipe	Blast2GO Pipe v2.5	on Blast E≤10^{-3}
B2G_pipe_trusted	Blast2GO Pipe v2.5	on Blast E≤ 10^{-3}, only trusted evidence codes
InterProScan	InterProScan 5-RC5	PF-test in FASTA format

"E≤x" stands for an E-Value threshold of x, "all annos" stands for the less restrictive PhyloFun mode in which all predicted GO annotations that received a probability higher than equal distribution were selected (see chapter 7.2.3, page 47), and "only trusted evidence codes" means that the configurable evidence code weights for Blast2GO had been set to zero for all evidence codes not indicating experimentally verified or curator made GO term annotations.

To elucidate each method's *sensitivity* for annotating specific GO terms we first generated sets of pairwise distinct GO terms from the annotations made by each method, than these sets were intersected in order to infer which GO terms were annotated by several methods and which could only be predicted by some or only a single method. In this intersections regarding ancestral terms were also considered. Such that in case a term predicted by one method was found to be ancestral to another term predicted by the compared method, those two GO terms were interpreted to be part of an "ancestral intersection".

As a measure of each methods *fineness* we computed the mean GO level (see chapter 7.1, page 43) of the above sets of pairwise distinct GO terms, respectively.

Part III.

Results

8. Automated Assignment of Human Readable Descriptions (AHRD)

8.1. Example

In this example I show how AHRD (*github.com/groupschoof/AHRD*; Hallab, Klee, Srinivas, and Schoof 2014) works and where its strengths are compared to the best BLAST hit method (Altschul, Madden, Schaffer, et al. 1997; McGinnis and Madden 2004) and BLAST2GO (Conesa, Gotz, García-Gómez, et al. 2005; Conesa and Gotz 2008). The example protein sequence "bgh04634_polypeptide" from *Blumeria graminis* (Spanu, Abbott, Amselem, et al. 2010) is annotated as an 'aminoglycoside phosphotransferase' by manual expert annotation. AHRD selects the correct annotation, while both the descriptions taken from the best BLAST hit and Blast2GO are wrong (see Table 8.1, page 52). All hits from the Swissprot (Bairoch and Apweiler 2000; Boeckmann, Bairoch, Apweiler, et al. 2003) database are proteins with at least two conserved domains; one is a phosphotransferase, the other a dehydrogenase domain. The example query sequence aligns only to the phosphotransferase domain of the hit proteins, not the dehydrogenase domain. Thus, it should not be annotated as dehydrogenase, as it does not contain this domain. But most tools assign this description since hit proteins are annotated as dehydrogenase and this description is transferred. Both Swissprot and TAIR (Huala, Dickerman, Garcia-Hernandez, et al. 2001; Poole 2007) databases with their more limited scope seem not to contain a functional homolog to the query protein. The more comprehensive trEMBL (Bairoch and Apweiler 2000; Boeckmann, Bairoch, Apweiler, et al. 2003) database does contain hits with a similar domain composition, but the three highest scoring hits are labeled as "uncharacterized protein" (see Figure 8.1, page 53). In Figure 8.1, only the first 20 hits from trEMBL are shown. AHRD predicts the right description as it considers all hits, not only the best hit. A hit protein that, like the query, only contains the phosphotransferase domain and is annotated as such is selected for annotation transfer. Swissprot hits annotated as dehydrogenase have lower BLAST scores, which in the end lowers their token scores. The word "phosphotransferase" instead occurs frequently and passes the threshold for high-scoring token, as does "aminoglycoside". Token scores as computed from blast, database and overlap score are shown in Table 8.2 (page 52) for the three tokens with highest scores. The bit score (shown in bold green) and overlap score (shown in bold orange) influence the token score in such a way that "phosphotransferase" and "aminoglycoside" receive much higher scores than "dehydrogenase", which still has a good database score due to the hits in Swissprot. However, the token "dehydrogenase" does not pass the threshold for high-scoring tokens, as it did not reach half of the maximum token score, and becomes an uninformative token with a penalized score of -0.155. Additionally the frequency of the descriptions influences the selection of the correct description. The description "aminoglycoside phosphotransferase" occurs with a frequency of 77, while "Acyl-CoA dehydrogenase" occurs only 14 times in all BLAST results. We conclude that the high scoring hit proteins in trEMBL annotated as Acyl-CoA dehydrogenase, e.g. B2WD91 and E4V074 are annotation errors, as we could not detect the dehydrogenase domain in these proteins. As demonstrated here, AHRD is robust against such annotation errors, even if they are the best BLAST hit, if they are rare in the total BLAST results. AHRD selects the 18th position of the trEMBL BLAST hits for description transfer, resulting in the same description chosen by a human curator.

Table 8.1.: AHRD example — Comparison with other competitors. Descriptions assigned by the different tools to the example protein "bgh04634_polypeptide" from *Blumeria graminis* (Hallab, Klee, Srinivas, and Schoof 2014).

Tool	Assigned Description
Manual expert annotation	aminoglycoside phosphotransferase
AHRD annotation	Aminoglycoside phosphotransferase
BLAST2GO annotation	acyl-dehydrogenase family member 10
Best BLAST hit (Swissprot)	Acyl-CoA dehydrogenase family member 10
Best BLAST hit (TAIR)	IBR3 (IBA-RESPONSE 3); acyl-CoA dehydrogenase/oxidoreductase
Best BLAST hit (trEMBL)	Putative uncharacterized protein

Table 8.2.: Token scoring for the example protein (Hallab, Klee, Srinivas, and Schoof 2014).

Token	Token-Score Calculation	Token-Score
phosphotransferase	$0.5 * 0.799 + 0.3 * 0.535 + 0.2 * 0.843$	0.729
aminoglycoside	$0.5 * 0.497 + 0.3 * 0.385 + 0.2 * 0.603$	0.484
dehydrogenase	$0.5 * 0.128 + 0.3 * 0.42 + 0.2 * 0.097$	0.210

Figure 8.1.: BLAST results with AHRD scoring for the example protein. Here the work flow of AHRD for one example protein is shown, with the list of used BLAST hits from the three different databases, their filtering steps, assigned internal scores, description frequencies and the final result (Hallab, Klee, Srinivas, and Schoof 2014).

8.2. Application to a whole genome

We applied AHRD within the whole genome annotation pipeline for tomato (Consortium 2012). 87% of 34727 predicted proteins could be annotated with protein domains using InterProScan (Zdobnov and Apweiler 2001; Apweiler, Attwood, Bairoch, et al. 2000) and 30% with Gene Ontology terms (Ashburner, Ball, Blake, et al. 2000) using PhyloFun (Engelhardt, Jordan, Muratore, and Brenner 2005; Engelhardt, Jordan, Srouji, and Brenner 2011). 80% of all predicted proteins could be assigned a human readable description by AHRD while 20% were annotated as "unknown proteins". 63% of all proteins fulfilled the criteria of having a BLAST bit score >50, a BLAST e-value <e-10 and an overlap in the BLAST alignment >60% (see Figure 8.2, page 54) (Altschul, Madden, Schaffer, et al. 1997; McGinnis and Madden 2004). Of 10580 proteins with Gene Ontology term annotation 7339 proteins share words with the assigned human readable description.

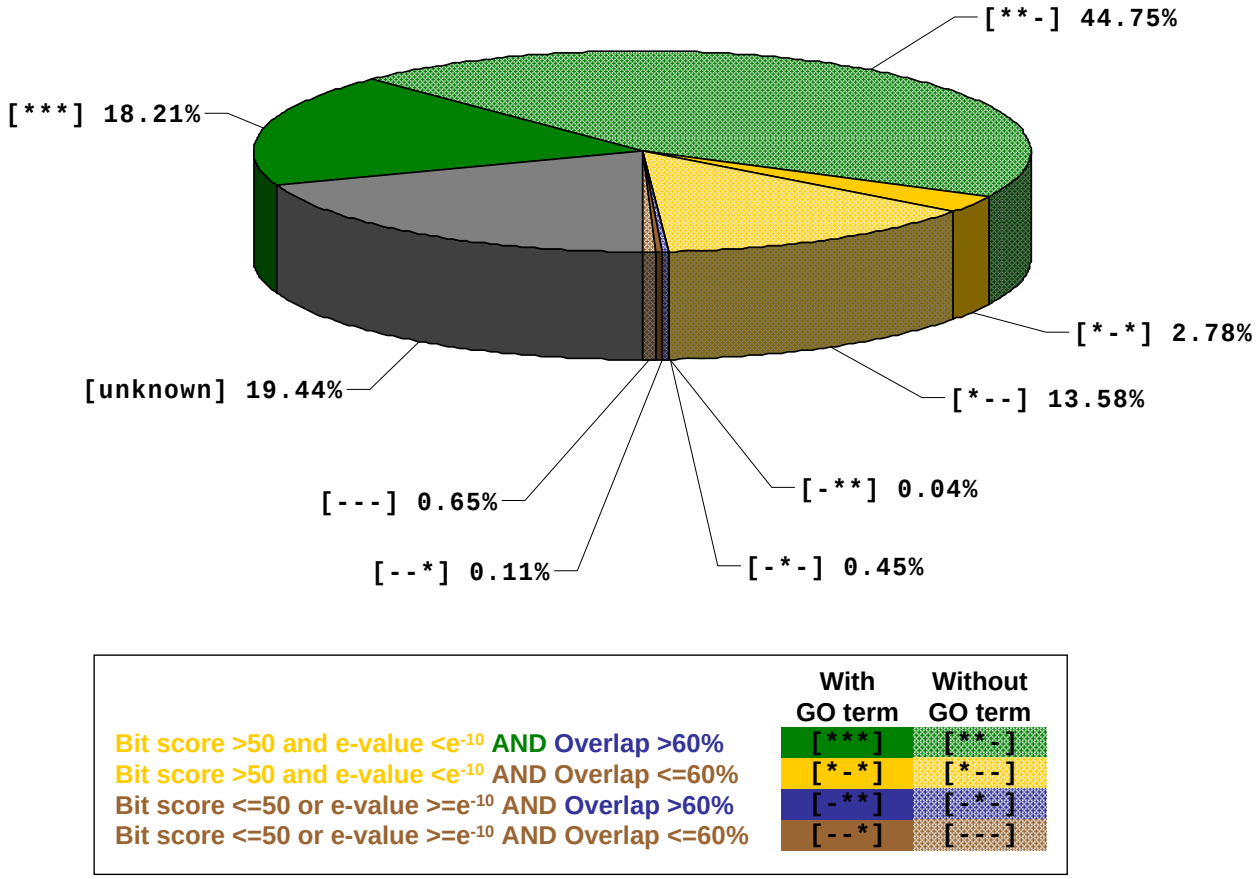

Figure 8.2.: AHRD quality code distribution for the tomato genome annotation. Human readable descriptions with shared words in GO terms are fully colored, while descriptions not matching any GO terms have patterned color. For clarity, only three of the for quality code criteria are shown. These are the e-value and bit score criterion, the overlap criterion and the GO term criterion (Hallab, Klee, Srinivas, and Schoof 2014).

8.3. Runtime

Given pre calculated calculated input data, AHRD annotated a batch of 100 sequences in approximately one minute on a single processor core.

8.4. Evaluation

We assessed the performance of AHRD on approximately 1400 expert annotated reference proteins of the *Blumeria graminis* fungal genome (Spanu, Abbott, Amselem, et al. 2010) and compared it to two competitive methods (Hallab, Klee, Srinivas, and Schoof 2014). For each predicted description we computed the commonly used F2-score (Rijsbergen 1979). This is calculated as the weighted harmonic mean of precision and recall based on the number of words shared between the reference and the predicted description. We compared our algorithm's mean F2-scores with those of BLAST2GO (Conesa and Gotz 2008; Conesa, Gotz, García-Gómez, et al. 2005) and of the best BLAST hits (Altschul, Madden, Schaffer, et al. 1997; McGinnis and Madden 2004) in the Swissprot, trEMBL and TAIR databases (Boeckmann, Bairoch, Apweiler, et al. 2003; Bairoch and Apweiler 2000; Huala, Dickerman, Garcia-Hernandez, et al. 2001; Poole 2007) (Table 8.3, page 57, and Figure 8.5, page 57). AHRD clearly outperformes all of its competitors, even so when selecting the highest scoring competitor separately for each single reference protein in the gold standard (Table 8.3). Furthermore, AHRD assigns the exact same description as the reference to 366 (26%) proteins (F2-score of 1), while its competitors only achieve about half this number (12–14%). 175 (12%) of AHRD descriptions have no term in common with the reference description (F2-score of 0), a number similar to that inferred for the competitors (Figure 8.5, page 58).

8.5. Parameter Optimization

The original set of parameters was selected by intuition. In order to find optimal parameters for the algorithm's scoring we both tested a set of systematically varied parameters, and applied a simulated annealing approach (Kirkpatrick, Gelatt, and Vecchi 1983), evaluating in an iterative manner the performance of slightly changed (mutated) parameter sets (section 5.3, page 35). In this performance was calculated, as described above, using F2-scores (table 8.7, page 62). Eight rounds of optimizations were carried out. Their results are provided in the supplementary files `ahrd_sim_anneal_1.txt` – `ahrd_sim_anneal_8.txt`. Subsequently, to avoid over-fitting and to assess the versatility of a given set of parameters on different datasets, we cross evaluated AHRD by optimizing parameters on one dataset and then evaluating performance on two other curated sets (table 8.7, page 62). Beside the *Blumeria graminis* dataset, we used a set of 1000 randomly selected Swissprot proteins (Bairoch and Apweiler 2000; Boeckmann, Bairoch, Apweiler, et al. 2003), that had been added to the public database in July 2011, and a set of approx. 1100 proteins from the *S. lycopersicum* genome (Consortium 2012) that had been manually annotated by experts (section 5.3, page 35). Many of the latter are resistance genes: 235 (21 %) are annotated as "receptor-like kinase". From this observation arose the question, if the protein description diversity of each of the three reference sets yielded different optimal parameters. In order to answer it, the number of distinct protein descriptions, after blacklisting and filtering (section 5.1, page 33), was inferred, and based on these description diversities were computed for each reference set and additionally the arabidopsis proteome, as well as the complete Swissprot database (Boeckmann, Bairoch, Apweiler, et al. 2003), separately (section 5.3.1, page 36). Furthermore, the relative frequencies of these distinct protein descriptions were inferred and their distributions assessed. In the resulting measurements (table 8.4, page 59) several striking observations are made. First, of

the five evaluated protein databases the *B.graminis* reference set is the most diverse (table 8.4, page 59), and of the three reference sets has the largest number of distinct protein descriptions. Secondly, the Tomato reference set is the smallest and has a corresponding low description diversity. Finally the Swissprot reference set is of intermediate size and also shows an intermediate diversity measure. Interestingly, this random selection of 1,000 proteins, added to the Uniprot/Swissprot official database in July 2011 (section 4.1.1, page 30), shows a higher diversity than the *full* Swissprot database.

Hence the question was asked, if the proteins of the three reference sets were of the subset of often annotated and frequently studied proteins (section 5.3.1, page 36). To that end, sequence similarity searches were carried out, separately for each reference set, and in the three public protein databases UniprotKB/Swissprot, UniprotKB/trEMBL, and TAIR10 (figure 8.6, page 60). Subsequent comparison of the results revealed which reference set had more high scoring bit scores in each of the public databases. Here, the two UniprotKB databases were reduced to contain only sequences added *before* July 2011. Interestingly, the Swissprot references had many more high scoring hits in the public Swissprot database than any of the two other reference sets. This was observed in spite of the fact that, for one, self matches had been removed from the public *full* Swissprot database, as well as those proteins added at the same time or after the Swissprot references, that is from July 2011 onwards. This indicates, that the Swissprot references were *not* proteins of previously undescribed functions, but resembled more of the curated public Uniprot/Swissprot proteins than did those of the other two reference sets.

Furthermore, in terms of the number of most common descriptions that account for a quarter of the whole respective set, the resulting two most diverse sets are the *B.graminis* references and the *A.thaliana* proteome (table 8.4 and figure 8.5, page 59). In contrasts both the *full* Swissprot database as well as its reference subset Swissprot clearly show a bias for commonly annotated proteins. Finally this bias is even more extreme in the Tomato references, where very few distinct descriptions account for the majority of its proteins (figure 8.5, page 59). As mentioned before, most of them are kinases involved in resistance.

Finally, in terms of distinct descriptions, the *B.graminis* reference set is approximately a tenth of the size of the arabidopsis proteome, while its mean description frequency and those of its first three quartiles are approximately ten fold higher than the corresponding values found for the arabidopsis proteome. Hence, these ratios observed in the *B.graminis* set show the values expected of a random selection of a tenth of the arabidopsis references.

The parameter set returned as optimal for the *B.graminis* test set had a mean F2-Score 4 points higher than the original intuitive set (table 8.7, page 62), while the whole optimization (run seven and eight on *B.graminis*) never assumed mean F2-Scores below 0.5475 or above 0.6777. Here the distribution of values taken from the 4th quartile of high performing parameter sets (mean F2-Scores > 0.6454) approximately covered the whole value interval of the respective parameters, implying the presence of multiple local optima in parameter space. Finally the theoretical maximum mean F2-Score of 1.0 was not achievable due to the fact that not for all references there were optimal candidate descriptions, such that the highest achievable mean F2-Score was 0.8856.

The values given in table 8.7 (page 62) are the results from optimization runs seven and eight, that were executed using the "new overlap score" (formula 5.1, page 34), which computes the coverage of both the query and hit sequences in the alignment generated by BLAST (Altschul, Madden, Schaffer, et al. 1997; McGinnis and Madden 2004). The simulated annealing runs executed before the introduction of this "new overlap score", used the "old overlap-score" (formula 5.13, page 38), which computes the coverage of the query sequence only (chapter 5.3.4, page 38). The first six optimization runs had similar results as the runs seven and eight, that is after introduction of the new overlap-score. In detail their minimum mean F2-Score never was lower than 0.4, while the best performing parameters sets had

mean F2-Scores ranging from 0.6592 to 0.6645, approximately equal to the results from runs seven and eight. While only 13% of changes to any parameter caused an improvement of performance, 74% did not yield a change in the mean F2-Score (table 8.9, page 63), which is reflected in the distribution of the stepwise absolute differences of the mean F2-Scores shown in table 8.10 (page 63). I estimated the proportion of the parameter space evaluated by all simulated annealing runs measuring the number of pairwise distinct parameter sets tested, and inferring the cumulative stepwise euclidean distance that each separate process covered during each simulated annealing run. Furthermore the distribution of tested values for each parameter was assessed to answer the question of how much parameter space was evaluated. Simulated annealing tested 10010001 pairwise distinct parameter sets in the first run, where only every approximately 10000th parameter set was evaluated twice. While the walked cumulative euclidean distances in parameter space often exceed the maximum distance from the space's origin to the upper far corner (173.2253) (table 8.11, page 64), assuming it positioned at "maximum values" of 1 for all parameters except the database weights, for which a maximum coordinate of 100 is assumed. Finally the spread of parameter values tried during optimization approximately covers the whole range available for the respective parameters. (table 8.12, page 64).

Table 8.3.: Mean F2-scores of descriptions assigned by AHRD and competing methods.

Tool	Mean F2-Score
AHRD	0.63
BestBLAST (Swissprot)	0.47
BestBLAST (TAIR)	0.25
BestBLAST (trEMBL)	0.28
BLAST2GO	0.41
Best competitor	0.59

"Best competitor" means that for every single query protein the best performing competitor method was selected (Hallab, Klee, Srinivas, and Schoof 2014).

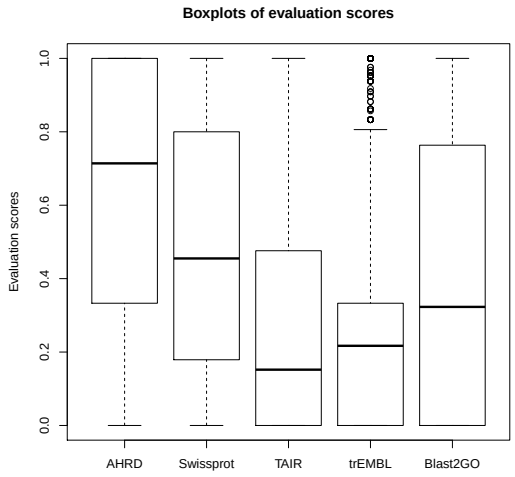

Figure 8.3.: Comparison of the distribution of evaluation scores (F2-Scores) from different methods (AHRD and competitors) (Hallab, Klee, Srinivas, and Schoof 2014).

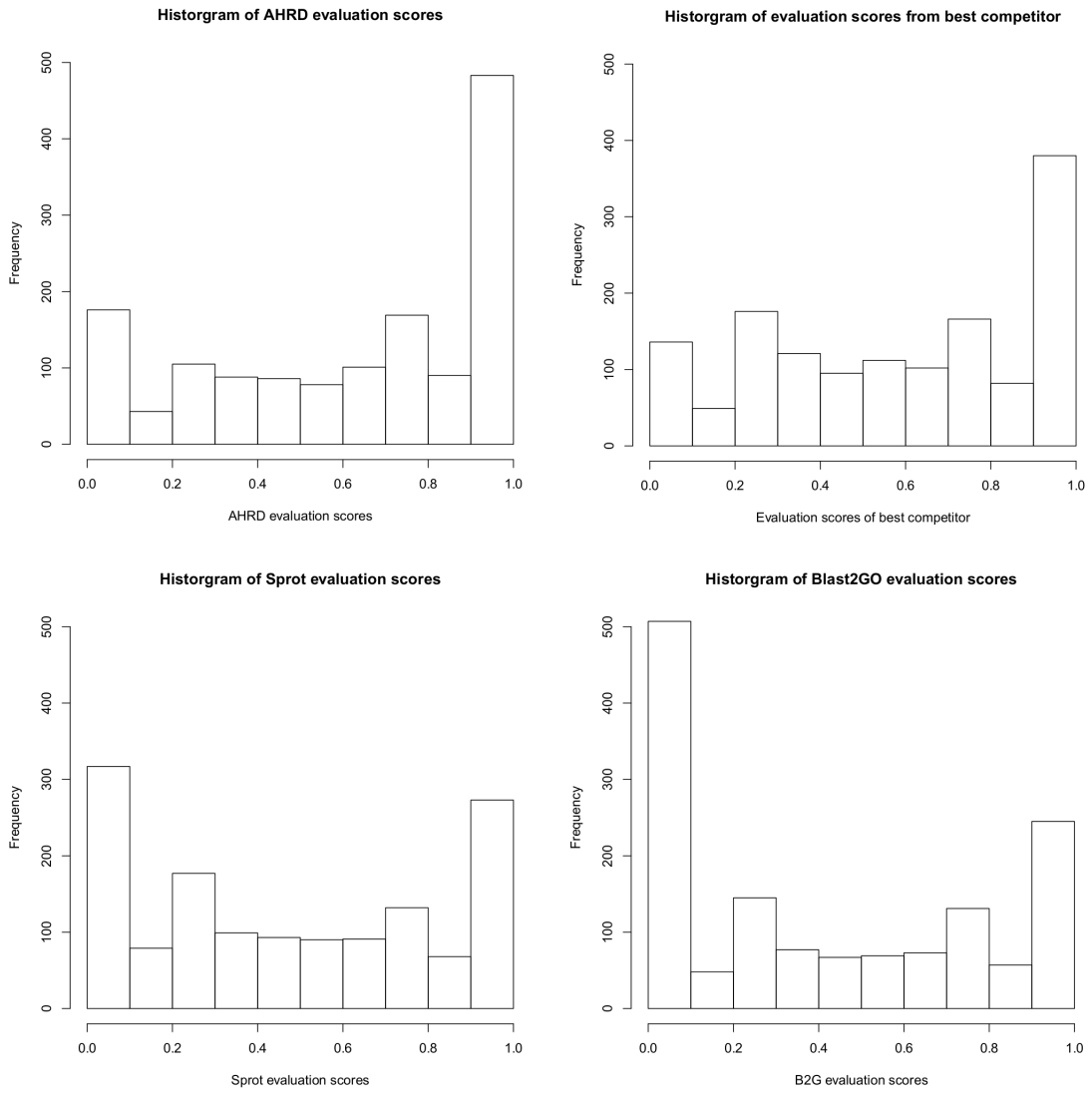

Figure 8.4.: Plotted F2-scores from AHRD, the best competitor (of Swissprot, TAIR, trEMBL and BLAST2GO), Swissprot and BLAST2GO in comparison (Hallab, Klee, Srinivas, and Schoof 2014).

Table 8.4.: Diversity of protein descriptions in the reference sets assessed as the distribution of description frequencies.

Protein-Set	Min.	1st Quartile	Median	Mean	3rd Quartile	Max.	Size	Div.	$\lvert\frac{1}{4}\rvert$
B.graminis	7.7e-04	7.7e-04	7.7e-04	8.2e-04	7.7e-04	5.4e-03	1213	0.85	244
Tomato	8.9e-04	8.9e-04	8.9e-04	7.5e-03	8.9e-04	2.1e-01	134	0.12	2
Swissprot	1.1e-03	1.1e-03	1.1e-03	1.5e-03	1.1e-03	1.4e-02	684	0.68	65
A.thaliana	3.7e-05	3.7e-05	3.7e-05	8.0e-05	7.4e-05	1.8e-02	12455	0.37	111
full Swissprot	2.0e-06	2.0e-06	2.0e-06	1.9e-05	6.1e-06	5.0e-02	53882	0.10	52

For each distinct protein description its frequency was measured. Above values summarize the distributions of these description frequencies for each distinct set of proteins, where column "Size" shows the number of distinct protein descriptions, column "Div." holds each sets' description diversity (section 5.3.1, page 36), and column "$\lvert\frac{1}{4}\rvert$" holds the minimum number of descriptions required to cover 25% of the proteins in the reference set (figure 8.5 and section 5.3.1, page 36). The first three rows refer to the reference sets used during evaluation and optimization, while the last two "*A.thaliana*" and "*full* Swissprot" rows show the values for the arabidopsis proteome and the complete Swissprot database, respectively. Numerical values are encoded such that "7.7e-04" means $7.7 \cdot 10^{-4}$.

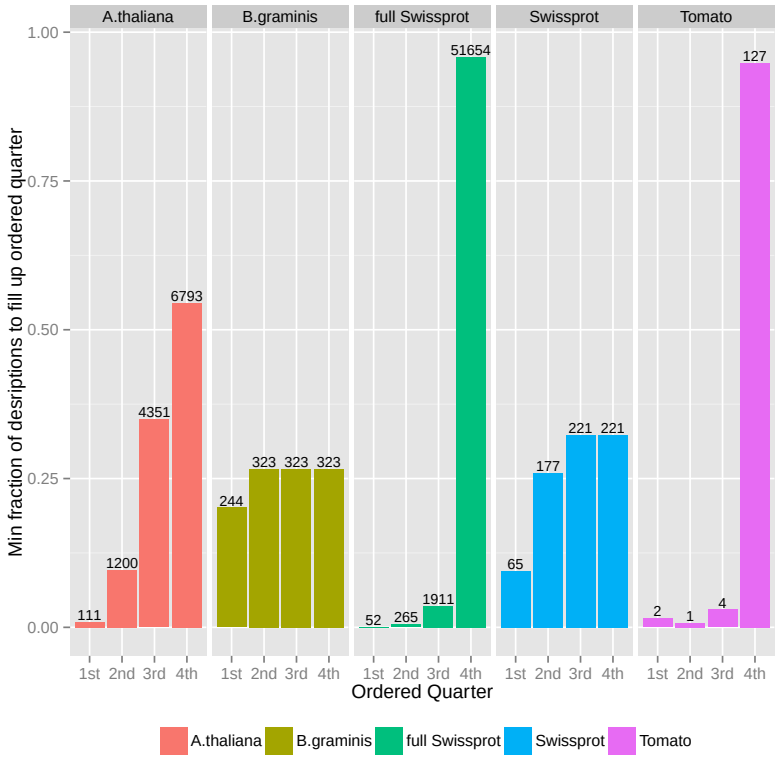

Figure 8.5.: Number of distinct descriptions covering each quartile of the reference sets. For each quarter of protein descriptions in the reference sets, the minimum number of distinct descriptions to account for the proteins contained in it was counted. This was done iteratively for the 1st, 2nd, 3rd, and 4th quarter of proteins in each reference set respectively (section 5.3.1, page 36). Shown above each bar is the absolute number of distinct descriptions accounting for the respective quarter.

Table 8.5.: Distribution of pairwise sequence identities for *B.graminis* proteins pairs. Using BLAST (McGinnis and Madden 2004) the pairwise sequence identities for all pairs of proteins were measured, excluding self matches (section 5.3.1, page 36).

Minimum	1st Quartile	Median	Mean	3rd Quartile	Maximum
16.31	26.23	30.43	32.47	36.36	95.39

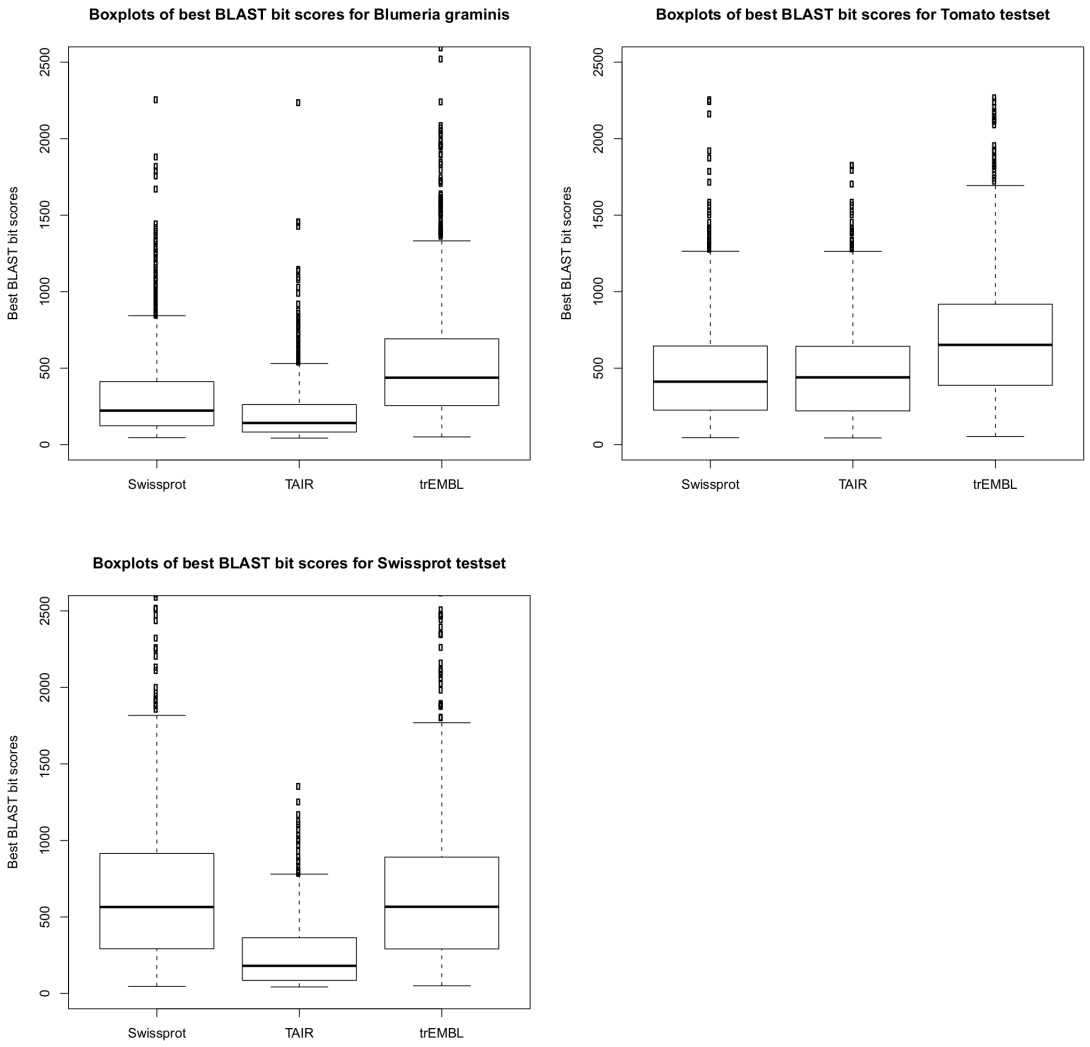

Figure 8.6.: AHRD — Comparison of the bit score distributions of the best blast hits to elucidate the different frequencies of high scoring BLAST hits in different databases (Hallab, Klee, Srinivas, and Schoof 2014).

8.5.1. Optimal Parameters

Table 8.6 (page 62) shows the different optimal parameters found by simulated annealing on the three used reference sets. The general parameter β, that controls the importance of a meaningful word depending on the associated BLAST Bit-Score (section 5.2, page 34), shows a clear preference in the *B.graminis* results. Its optimal value for the *B.graminis* test set is more than twice as high as its optimal value for the Swissprot test set. In contrast, the parameters specific to the Swissprot database show a clear preference for Swissprot itself when optimized on the very Swissprot test set. Here the value for the ω parameter, controlling the relative importance of descriptions obtained from the Swissprot database, is at least 3.5 times higher than the values obtained for the two other reference sets. Also the Sprot-w weight, reflecting the trust put into descriptions found in the Swissprot database, is 4 times higher than the weight found optimal for the *B.graminis* test set. Finally, the importance of BLAST Bit-Scores obtained from results found in the Swissprot database, which is expressed as the Sprot-δ parameter, is found to be optimal for the Swissprot test set when it is at least 7.5 times higher than the values obtained for the two other test sets. These biases clearly show that human readable descriptions obtained from the Swissprot database are optimal for annotating query proteins from the very same Swissprot database. Hence descriptions from the Swissprot database show a self preference, which is reflected by the fact that many proteins in the Swissprot database indeed share identical descriptions — probably due to expert manual revision of Swissprot protein descriptions (Boeckmann, Bairoch, Apweiler, et al. 2003). A final preference for descriptions obtained from well aligning proteins in the trEMBL database is expressed in the simulated annealing results for the trEMBL-δ parameter value. This is found to be optimal for the Swissprot test set approximately 3.7 times higher than it is for the *B.graminis* test set. Finally, the intuitive original value 0.6 for the parameter α, which controls the importance of a candidate description's frequency (section 5.2, page 34), is strongly reduced in the found to be optimal parameter values. Indeed it is set to a sixth of its original value for the annotation of the *B.graminis*, as well as the Tomato references, while it is found to be optimal at a third of its original value for the Swissprot reference set (table 8.6, page 62).

Cross validation of found to be optimal parameter sets was carried out in order to avoid over-fitting and also to elucidate AHRD's robustness to changes of its parameters. Here the results from optimizing parameter values for the three test sets *B.graminis*, Tomato, and Swissprot are compared by computing each optimal parameter set's performance when used on either test set it was *not* optimized on. Table 8.6 (page 62) shows the mean F2-scores achieved by each parameter set, when applied on each of the three reference protein databases separately. Here the cross validation of the Tomato parameter set clearly shows that simulated annealing failed to find the desired optimal parameter values for the Tomato references, because the resulting Tomato parameter set is outperformed by any of the two other parameter sets. Even though simulated annealing on Tomato failed, optimal parameters still outperform original ones in cross validation. Also parameters optimal for the B.graminis references perform better than the original intuitive settings, that is when they are applied on B.graminis and Tomato. On the other hand, in comparison with the original intuitive parameters, the Swissprot parameters decrease AHRD's performance on the other two reference sets.

Table 8.6.: Comparison of optimized parameter sets

Parameter	Set			
	Initial intuitive parameter set	*B.graminis*	Tomato	Swissprot
α	0.60	0.10	0.10	0.20
β	0.50	0.70	0.45	0.34
ω	0.30	0.10	0.16	0.57
σ	0.20	0.20	0.39	0.10
Sprot-w	100	30	90	130
Sprot-δ	0.20	0.30	0.60	4.50
trEMBL-w	10	10	200	150
trEMBL-δ	0.40	0.60	0.00	2.20
TAIR-w	50	50	50	110
TAIR-δ	0.40	0.90	0.10	0.60

Parameters found to be optimal by simulated annealing on the three different sets of reference proteins: *Blumeria graminis*, Tomato, and Swissprot. The shown parameter values were taken from simulated annealing run seven and eight, that is after the "new overlap score" had been introduced (section 5.3.4, page 38). The "Initial intuitive" set of parameters is also shown.

Table 8.7.: Mean F2-scores of different parameter sets on three test sets (Hallab, Klee, Srinivas, and Schoof 2014).

AHRD Setup	Dataset		
	B.graminis	Tomato	Swissprot
Maximal attainable mean F2-score	0.89	0.86	0.88
Initial intuitive parameter set	0.63	0.53	0.67
Parameter set from sim. anneal. on Blumeria	0.68	0.57	0.59
Parameter set from sim. anneal. on Tomato	0.65	0.48	0.68
Parameter set from sim. anneal. on Sprot	0.62	0.48	0.82
Overlap and database weight = 0	0.68	0.50	0.75

Mean F2-scores where obtained using parameters found to be optimal by simulated annealing on the three different sets of reference proteins: *Blumeria graminis*, Tomato, and Swissprot. (table 8.6, page 62) The shown values were taken from simulated annealing run seven and eight, that is after the "new overlap score" had been introduced (section 5.3.4, page 38).

Table 8.8.: Distribution of values tested during simulated annealing in 4th quartile of high scoring parameter sets (mean F2-Scores > 0.6454).

Parameter	Minimum	1st Quartile	Median	Mean	3rd Quartile	Maximum	SD
α	0.1000	0.2000	0.4000	0.4692	0.7000	1.0000	0.2669
β	0.0476	0.3704	0.4545	0.4622	0.5556	0.8333	0.1411
ω	0.0476	0.1250	0.2105	0.2324	0.3200	0.8333	0.1339
σ	0.0476	0.1875	0.3077	0.3054	0.4118	0.8333	0.1470
Sprot-w	10.0000	20.0000	30.0000	41.0200	60.0000	100.0000	27.9738
Sprot-δ	0.1000	0.3000	0.5000	0.5365	0.8000	1.000	0.2863
trEMBL-w	10.0000	50.0000	70.0000	67.1000	90.0000	100.0000	25.5704
trEMBL-δ	0.1000	0.5000	0.7000	0.6823	0.9000	4.5830	0.2461
TAIR-w	10.0000	30.0000	50.0000	53.7900	80.0000	100.0000	28.5853
TAIR-δ	0.1000	0.3000	0.5000	0.5392	0.8000	1.0000	0.2861

(Hallab, Klee, Srinivas, and Schoof 2014) Parameters are explained in chapter 5.2 (page 34). "Sprot", "trEMBL", and "TAIR" refer to the respective protein databases (Bairoch and Apweiler 2000; Boeckmann, Bairoch, Apweiler, et al. 2003; Huala, Dickerman, Garcia-Hernandez, et al. 2001). "SD" is the standard deviation of the respective measurements. Values are from simulated annealing runs seven and eight.

Table 8.9.: Rates of accepting or rejecting mutated parameter sets during simulated annealing.

Acceptance "Better"	Acceptance "Equal"	Acceptance "Worse"	Rejection "Worse"
0.13	0.74	0.09	0.05

Each column shows the rate of accepting or rejecting parameter sets after having mutated a randomly selected parameter. "Better" denotes a set that had an increased mean F2-Score compared to the currently accepted set, and "Equal", or "Worse" stand for unchanged or decreased mean F2-Scores, respectively. Given values were measured on the second simulated annealing run.

Table 8.10.: Distribution of stepwise absolute differences in mean F2-Scores during simulated annealing. Values were estimated on the second simulated annealing run.

Minimum	1st Quartile	Median	Mean	3rd Quartile	Maximum
0.0	0.0	0.0	0.0001958	0.0000259	0.017120

Table 8.11.: Euclidean distances in parameter space walked during simulated annealing. Values have been measured on the first simulated annealing run.

Mean	Maximum	Standard Deviation
210.30	997.10	133.80

Table 8.12.: Distribution of parameter values tried during simulated annealing. Values are based on the fifth run.

Parameter	Minimum	1st Quartile	Median	Mean	3rd Quartile	Maximum	SD
α	0.0	0.1766	0.4000	0.4592	0.6996	8.3180	0.3362
β	0.0	0.1603	0.2806	0.3030	0.4209	0.9996	0.1784
ω	0.0	0.2209	0.3522	0.3586	0.4846	0.9991	0.1826
σ	0.0	0.1977	0.3255	0.3387	0.4629	0.9998	0.1824
Sprot-δ	0.0	0.700	1.431	1.785	2.782	14.620	1.3354
Sprot-w	1.0	70.0	145.0	390.8	573.0	4603.0	484.218
trEMBL-δ	0.0	0.8639	1.8180	1.8370	2.5870	12.9300	1.1720
trEMBL-w	1.0	53.0	140.0	385.3	573.0	4559.0	489.0097
TAIR-δ	0.0	0.5923	1.0440	1.5510	2.3190	14.1200	1.3317
TAIR-w	1.0	60.0	143.0	386.5	569.0	5590.0	484.6021

For each parameter the distribution of values tried during optimization are summarized. Here column "SD" hold the standard deviation.

8.5.2. Evaluation of Simulated Annealing

Two example plots of simulated annealing processes obtained from run five are given here to elucidate acceptance and rejection rates of slightly mutated parameter sets and furthermore evaluate the range and smoothness of the mean F2-Scores the tested parameter sets assumed during these optimization processes. In the first example process (figure 8.7, page 65) the simulated annealing approach reached a local optimum in the F2-Score landscape, after crossing a minor "valley" of parameter sets with decreased mean F2-Scores, while during the entire computation most parameter mutations (68%) did not yield a change in mean F2-Score, which when changed never sank below 0.63 nor surpassed 0.653. The second example process (figure 8.8, page 66) had comparable acceptance and rejection rates, and also a comparable range of assumed mean F2-Scores, while it did not find a local optimum but only managed to recover an approximately similar performing parameter set compared to the initial start parameters.

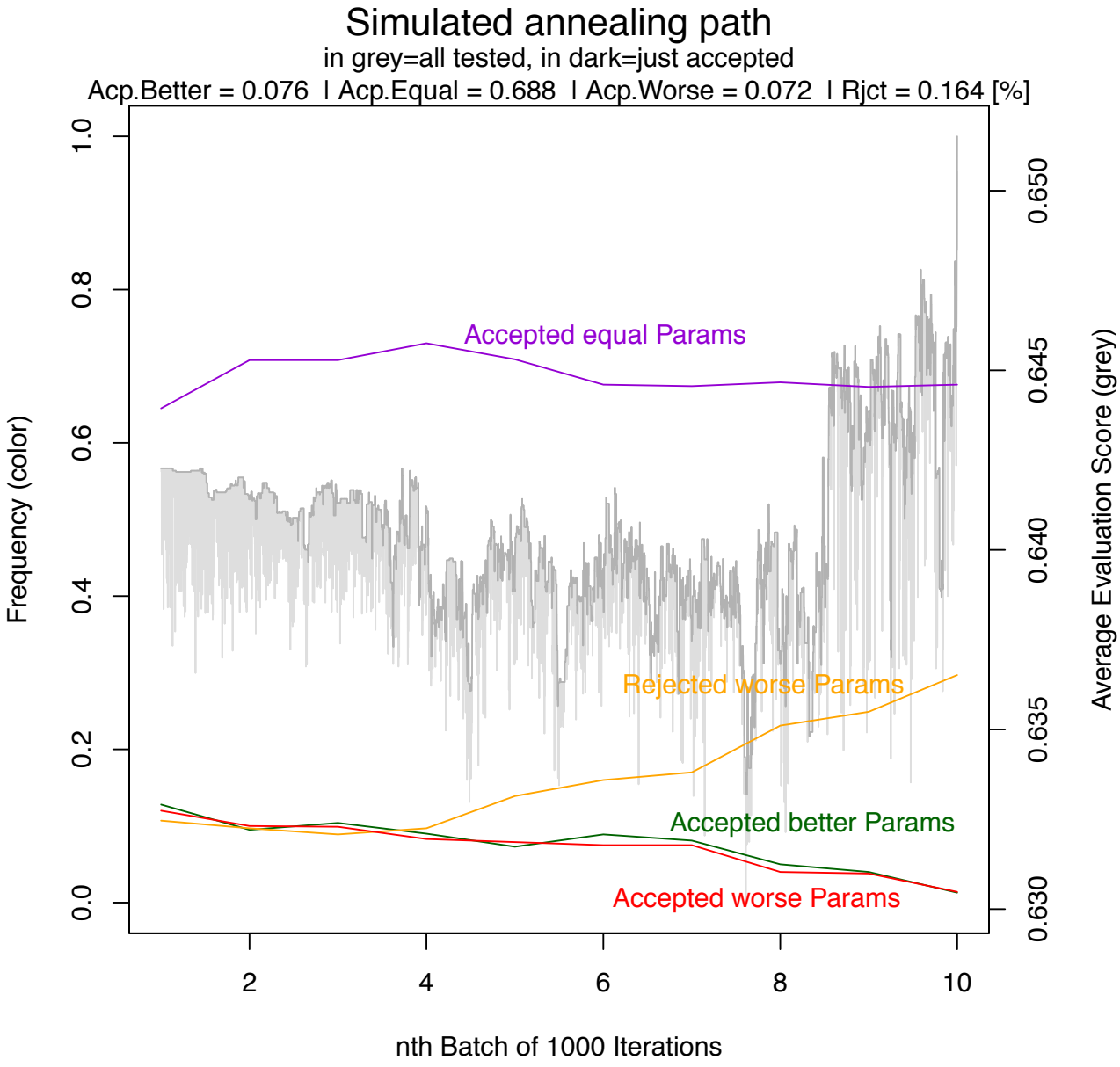

Figure 8.7.: Plot of simulated annealing optimization 1. In light grey the mean F2-Score (right axis) of all evaluated parameter sets is plotted, while dark grey shows the mean F2-Scores only of accepted parameter sets. Colored lines show rates of accepting or rejecting "equally", "better", or "worse" performing parameter sets measured on subsequent intervals of 1000 degrees. Overall rates are printed below the headline.

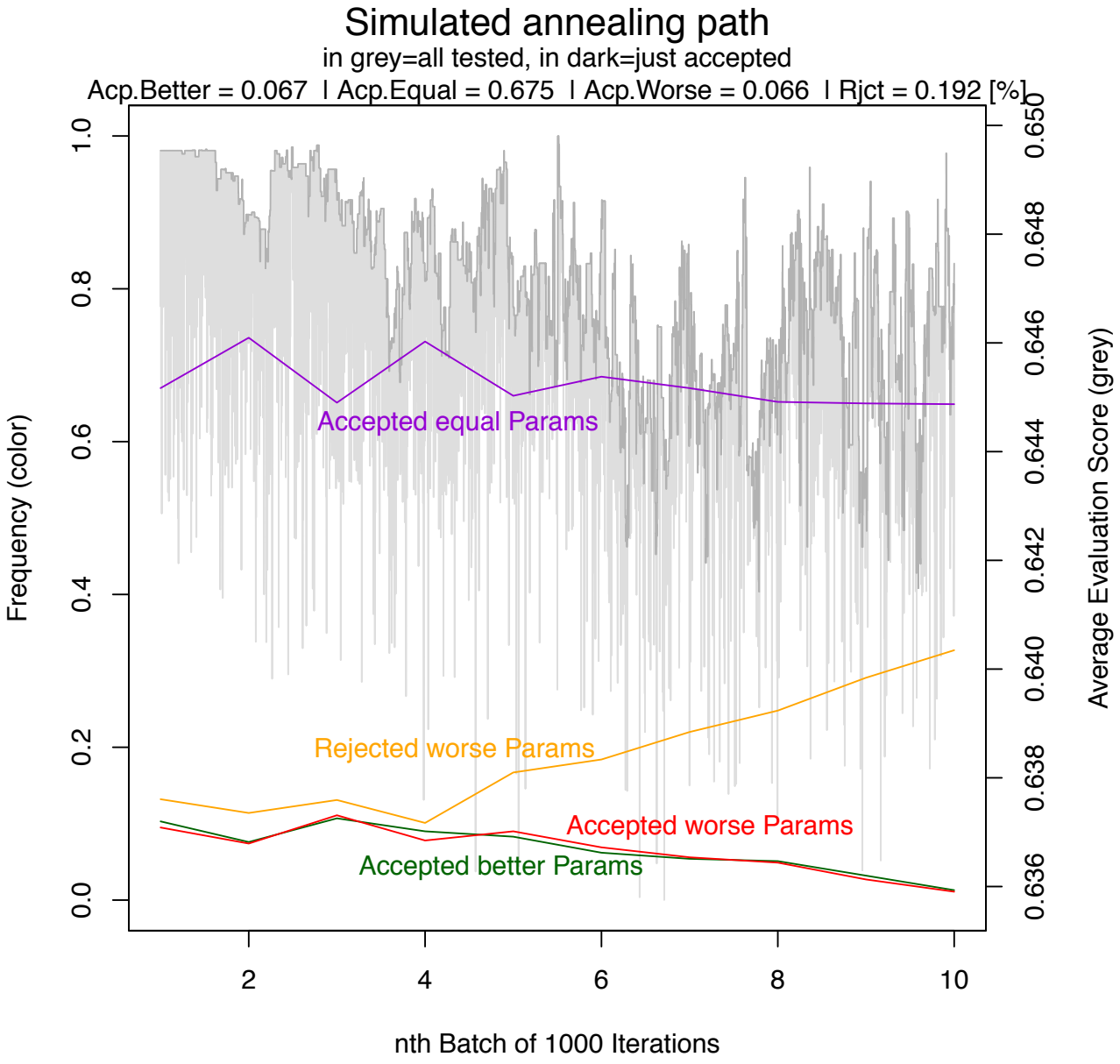

Figure 8.8.: Plot of simulated annealing optimization 2. In light grey the mean F2-Score (right axis) of all evaluated parameter sets is plotted, while dark grey shows the mean F2-Scores only of accepted parameter sets. Colored lines show rates of accepting or rejecting "equally", "better", or "worse" performing parameter sets measured on subsequent intervals of 1000 degrees. Overall rates are printed below the headline.

8.6. Scoring Domain Architecture Similarity

We extended AHRD to consider and score protein domain architecture similarity between the Query and each candidate (Dom-Arch-Sim-AHRD), to answer the question if such an approach could improve the quality of the method's annotations. To this end F2-Scores were measured with increasing importance of above domain architecture similarity, controlled by the appropriate weight parameter (Bangalore 2013).

The results show no significant change in the overall annotation quality. Indeed, depending on the tried parameters, only 6 to 28, less than 3% of the 1419 *B.graminis* references (section 5.3, page 35) received annotations that had different scores, comparing AHRD with Dom-Arch-Sim-AHRD results (Bangalore 2013).

9. Human Readable Descriptions for Tomato gene families

AHRD on gene clusters successfully assigned descriptions to 13678 (approximately 78%) of the 17487 gene families with members in the Tomato proteome (supplementary file `ahrd_on_gene_clusters_tomato.tar`). These descriptions had a median score of 0.71, that is half the clusters' descriptions were based on InterPro annotations that had at least an annotation frequency of 71%. This as well as the more detailed estimate of the overall quality of the assigned descriptions is shown in table 9.1 (page 67) and the histogram in figure 9 (page 68), in which the distribution of annotation frequencies is shown.

Table 9.1.: Distribution of description scores for the Tomato gene families.

Minimum	1st Quartile	Median	Mean	3rd Quartile	Maximum
0.0100	0.5000	0.7100	0.6808	0.8000	1.0000

This tables shows a summary of the distribution of InterPro annotation frequencies of those InterPro annotations selected by AHRD.

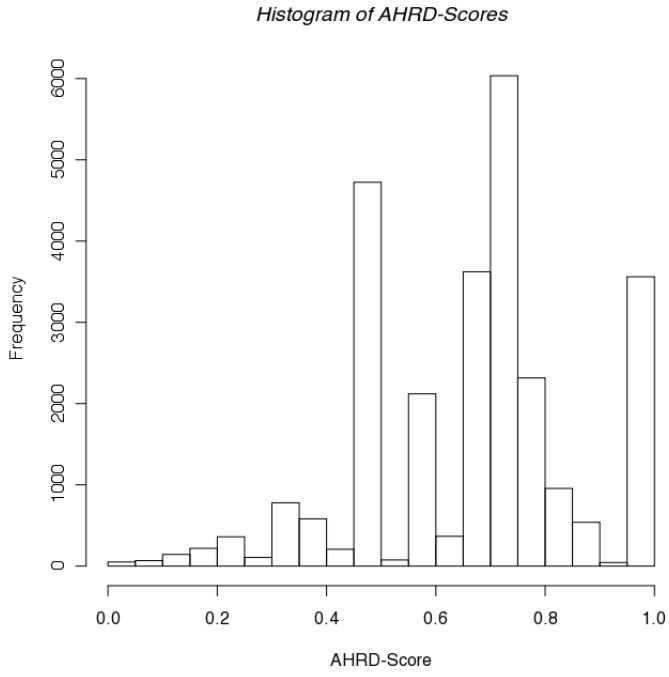

Figure 9.1.: Histogram of Tomato gene family description quality scores

In table 9.2 an example is given of how AHRD on gene clusters correctly annotates a Serine Peptidase gene family as such.

Table 9.2.: Human Readable Descriptions for Tomato gene families — Example 1 shows how the gene family was correctly described as a group of serine carboxypeptidases.

ORTHOMCL10233 (4 genes)
(AHRD-Score 1.00) IPR001563 Peptidase S10, serine carboxypeptidase

The example in table 9.3 shows a gene family in which half of the members have a kinase and the other have a more specialised Choline/ethanolamine kinase domain. No InterPro Family had been annotated to any gene of this cluster.

Table 9.3.: Human Readable Descriptions for Tomato gene families — Example 2 shows a gene cluster where no InterPro Family met the threshold and two description lines based on InterPro Domains were generated.

ORTHOMCL10004 (4 genes)
(AHRD-Score 0.50) IPR011009 Domain Protein kinase-like domain (AHRD-Score 0.50) IPR002573 Domain Choline/ethanolamine kinase

The largest gene family with Tomato members (table 9.4) has been annotated as being comprised

of at least 61 Peptidases that have a deubiquitinating function.

Table 9.4.: Human Readable Descriptions for Tomato gene families — Example 3 shows the description assigned to the largest gene family with members from the Tomato proteome. This description also consists of *two* separate lines.

ORTHOMCL0 (177 genes)
(AHRD-Score 0.35) IPR003653 Family Peptidase C48, SUMO/Sentrin/Ubl1 (AHRD-Score 0.15) IPR015410 Domain Domain of unknown function DUF1985

10. PhyloFun

10.1. Version 1 (v1.0) applied on the Tomato genome

The predicted proteins in the Tomato genomes were functionally annotated with the PhyloFun (v1.0) pipeline, as well as with the "InterProScan" tool (version 4.5).

The PhyloFun pipeline annotated almost a third of the Tomato proteome with over 1500 distinct Gene Ontology (GO) terms, while InterProScan (v4.5) could only annotate less than half as many distinct GO terms, while it covered on the other hand 20.14% more proteins of the Tomato proteome than PhyloFun (v1.0) did (table 10.1, page 69). Meanwhile PhyloFun was able to assign finer GO term annotations, which we measured as the mean GO levels of the respective tools' GO term annotations (table 10.2, page 70). Finally the number of pairwise distinct annotated GO terms show that each annotation method has its distinct domain of sensitivity, as the intersection of their pairwise distinct GO term annotations was small (table 10.3, page 70). In contrast, the intersection of proteins annotated by each method separately was large, because the unification of the two result sets only increased the annotation coverage of the proteome by 6% (table 10.1, page 69). Meaning, that the proteins each method was able to annotate with GO terms were largely the same. Hence InterProScan (v4.5) and PhyloFun (v1.0) largely annotated the *same* proteins but with *different* GO terms.

Table 10.1.: GO term annotations of the Tomato proteome made by InterProScan (v4.5) and PhyloFun (v1.0).

	PhyloFun (v1.0)	InterProScan (v4.5)	union of both
% of the Tomato proteome annotated	30.47	50.61	56.62

Table 10.2.: Distribution of levels of Tomato proteome GO term annotations made by PhyloFun (v1.0) and InterProScan (v4.5).

Tool	Minimum	1st Quartile	Median	Mean	3rd Quartile	Maximum
PhyloFun (v1.0)	2.000	5.000	6.000	6.093	7.000	11.000
InterProScan (v4.5)	2.000	4.000	5.000	5.542	7.000	10.000

Table 10.3.: Unique GO terms annotated for the Tomato proteome

Set	Size
PhyloFun (v1.0)	1560
InterProScan (v4.5)	654
Intersection of both	106
Intersection regarding ancestor-descendant-relations	162
Union of both	2108

Any two GO terms that have a ancestor-descendant-relationship are in the "Intersection regarding ancestor-descendant-relations".

We plotted the level two GO term annotation frequencies as a measure of reliability of the overall GO annotations (figure 10.1, page 71). The result shows a distribution comparable to those of other plant genomes (Consortium 2012), where most of the GO term annotations were made electronically (Jaillon, Aury, Noel, et al. 2007; Schmutz, Cannon, Schlueter, et al. 2010), while there are clear differences to the GO term annotation profile of *Arabidopsis thaliana* (Huala, Dickerman, Garcia-Hernandez, et al. 2001).

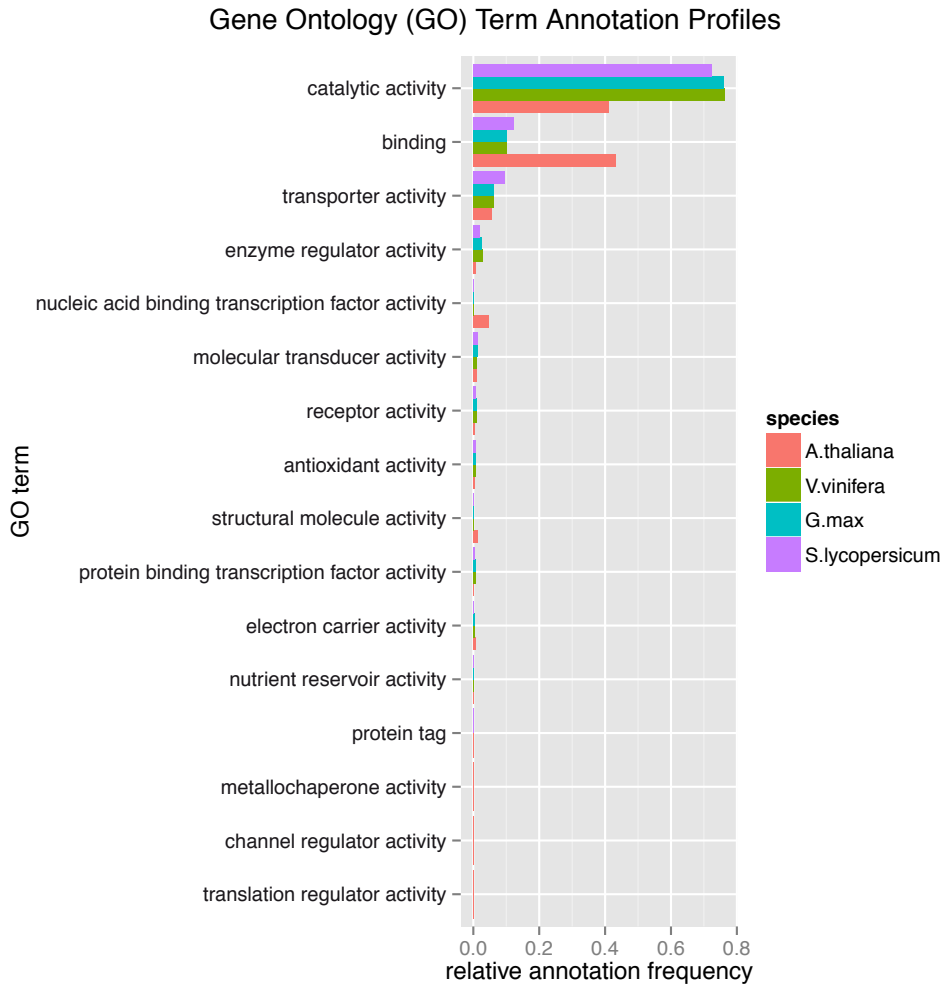

Figure 10.1.: Distribution of Tomato level two Gene Ontology (GO) term annotations made with PhyloFun (v1.0) and InterProScan (v4.5) in comparison with those obtained by published GO term annotations for *Arabidopsis thaliana*, *Glycine max*, and *Vitis vinifera* (Huala, Dickerman, Garcia-Hernandez, et al. 2001; Jaillon, Aury, Noel, et al. 2007; Schmutz, Cannon, Schlueter, et al. 2010).

10.2. Version 1 (v1.0) applied on the *Mediacgo truncatula* genome

The GO annotations made by PhyloFun (v1.0) and InterProScan (v4.5) for the *M truncatula* proteome have as a whole similar characteristics as those made for the Tomato proteome. That is InterProScan (v4.5) was able to annotate more proteins than PhyloFun (v1.0), while annotating less distinct GO terms of a mean reduced GO level (section 7.1, page 43). In more detail InterProScan (v4.5) annotated 8.73% more of the *M truncatula* proteome (table 10.4, page 72), while it assigned approximately three times less pairwise distinct GO terms (table 10.6, page 72), that had a mean GO level 0.5 points lower than that of the PhyloFun (v1.0) annotations (table 10.5, page 72). Again both methods proved to have their own domain of GO terms it could assign, because the intersection of the sets of pairwise distinct annotated GO terms was small. In contrast to these domains of annotated GO terms, both methods annotated almost the same proteins. A result also observed for the Tomato proteome. Here

as well as in the case of the *M.truncatula* proteome the intersection of protein sets, that received annotations from each method, was large, because the union of both only increased the coverage of proteins with function predictions by 6.7% (table 10.4, page 72).

Table 10.4.: Coverage of GO term annotations made for the *Medicago truncatula* proteome

	PhyloFun (v1.0)	InterProScan (v4.5)	union of both
annotated proteome proportion	24.67	33.40	40.30

Table 10.5.: Distribution of GO levels of *Medicago truncatula* proteome GO term annotations made by PhyloFun (v1.0) and InterProScan (v4.5).

Tool	Minimum	1st Quartile	Median	Mean	3rd Quartile	Maximum
PhyloFun (v1.0)	2.000	5.000	6.000	6.109	7.000	11.000
InterProScan (v4.5)	1.000	4.000	6.000	5.603	7.000	10.000

Table 10.6.: Unique GO terms annotated for the *Medicago truncatula* proteome

Set	Size
PhyloFun (v1.0)	1659
InterProScan (v4.5)	585
Intersection of both	119
Intersection regarding parent-child-relations	198
Union of both	2125

Any two GO terms that have a ancestor-descendant-relationship are in the "Intersection regarding ancestor-descendant-relations".

The level two GO term annotation frequencies observed for *M truncatula* are most alike to those observed for *Arabidopsis thaliana*, while they differ from those obtained from GO term annotations of *Vitis vinifera* and *M truncatula*'s close relative *Glycine max*, for reasons not investigated further.

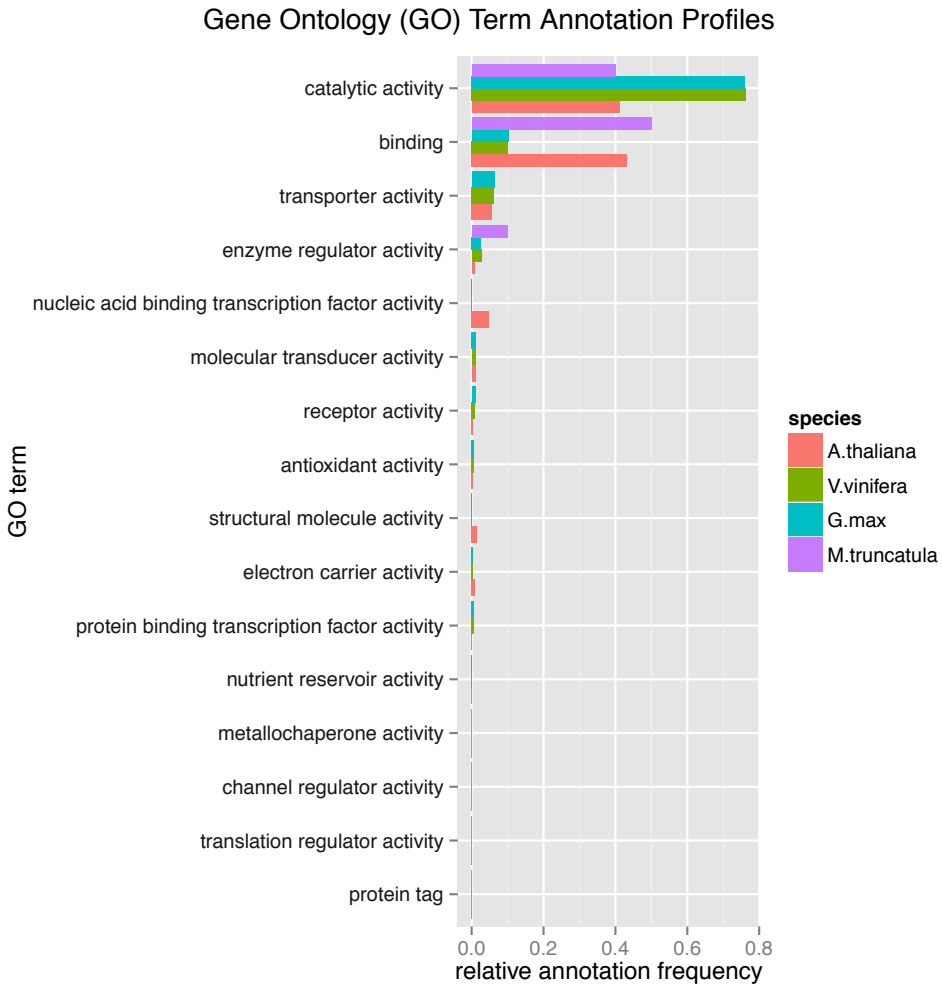

Figure 10.2.: Distribution of *M truncatula* level two Gene Ontology (GO) term annotations made with PhyloFun (v1.0) and InterProScan (v4.5) in comparison with those obtained from published GO term annotations made for *Arabidopsis thaliana*, *Glycine max*, and *Vitis vinifera*.

10.3. Version 2 (v2.0)

10.3.1. Measurement of Gene Ontology term mutation probabilities

We computed the mutation probability lookup tables for those 18610 Gene Ontology (GO) terms that were annotated as experimentally verified or curator made and that were found in the trust-UniKB set of proteins described earlier (section 7.2, page 43). The spread of maximum sequence distances found to be mapped to five different upper boundaries of mutation probabilities were visualized in the following box-plots (see figures 10.3 – 10.9, pages 76 – 79), which were generated not only for all GO terms, but also the subsets of them given by their respective GO level (see chapter 7.2.1, page 45).

Such lookup tables were generated for 18610 unique GO terms, of which 1707 belong to the "biological process", 12264 to the "cellular component", and 4624 to the "molecular function" ontologies, respectively. These GO terms split up by their respective GO levels yield 90 terms of GO level 2, 419

of level 3, and finally 18076 of level 4 or higher. The 18610 GO terms that could be assigned mutation probability lookup tables comprise only half of the GO terms in the public Gene Ontology database (Ashburner, Ball, Blake, et al. 2000), indicating only half of all GO terms have experimentally verified or curator made annotations in the UniprotKB dataset (Bairoch and Apweiler 2000; Boeckmann, Bairoch, Apweiler, et al. 2003).

Example mutation probability lookup table for GO term "GO:0000009" (see table 10.7, page 74) shows that this GO term mutates with 100% probability on any given sequence distance, due to the fact, that the first protein pair, that is the one with minimal sequence distance, which was used to calibrate this mutation probability, did not share the GO annotation. 7593 of the generated lookup tables were of this type, that is their respective GO terms are estimated to mutate on any given sequence distance with a probability of 1.0. Also because of such mutation probabilities not all lookup tables had values for every bin of probabilities used to generate the following box-plots, in fact "NA" values were applied to approximately 50% of the probability intervals.

Table 10.8 shows a more detailed mutation probability distribution, which maps 20 increasing sequence distances to respective GO term mutation probabilities.

Figures 10.3–10.6 (pages 76–77) show the distribution of sequence distances in different bins of GO term mutation probabilities. As expected, higher mutation probabilities tend to be associated with higher sequence distances (table 10.9, page 75); note however that the quartile ranges of the distributions overlap in all bins except the highest one and that there are a lot of outliers in all distributions. The mean and range of the distributions are similar for the "biological process" (figure 10.4, page 76) and "cellular component" (figure 10.5, page 77) ontologies, while the ontology "molecular function" (figure 10.6, page 77) has a clearly increased mean and range in bins with mutation probability < 1.

Table 10.7.: Mutation probability lookup table for "GO:0000009" (alpha-1,6-mannosyltransferase activity).

Maximum Sequence Distance	Mutation Probability
1.73	1.00

The protein pair with minimal sequence distance used to calibrate this GO term's mutation probability distribution already did not share this GO term.

Table 10.8.: Mutation probability lookup table for "GO:0080039" (xyloglucan endotransglucosylase activity)

Maximum Sequence Distance	Mutation Probability
0.10	0.00
0.39	0.33
0.40	0.50
0.42	0.60
0.43	0.64
0.44	0.67
0.47	0.71
0.48	0.73
1.17	0.75
1.20	0.76
1.21	0.77
1.23	0.78
1.24	0.79
1.26	0.80
1.29	0.81
1.33	0.82
1.38	0.83
1.42	0.84
1.48	0.85
1.71	0.86

Table 10.9.: Approximate mean maximum sequence distances for binned GO term mutation probabilities of *all* ontologies

Maximum mutation probability	Mean maximum Sequence Distance
0.2	0.15
0.4	0.25
0.6	0.35
0.8	0.75
1.0	1.90

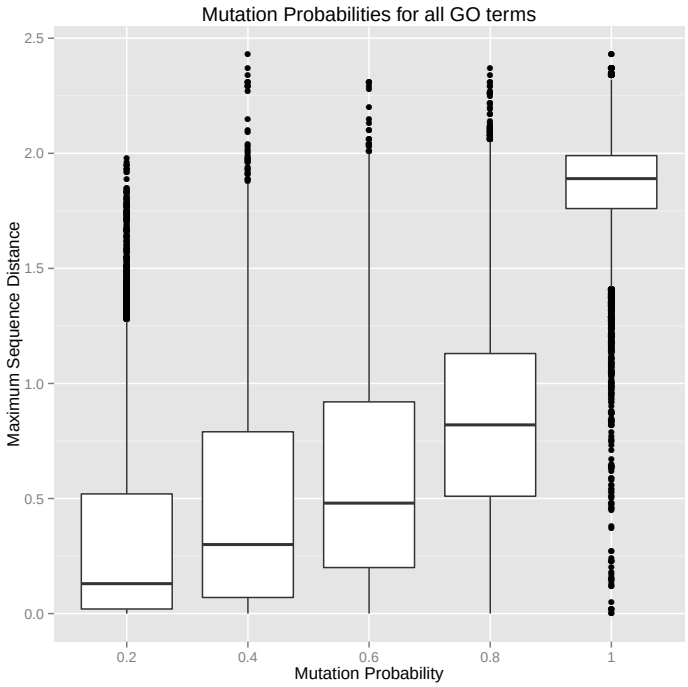

Figure 10.3.: Spread of maximum sequence distances in binned mutation probabilities for all GO terms

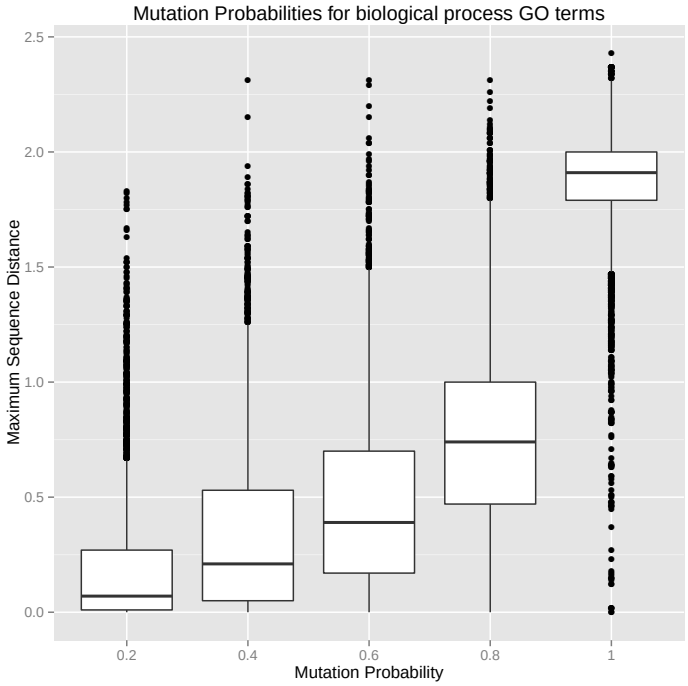

Figure 10.4.: Spread of maximum sequence distances in binned mutation probabilities for the 12264 GO terms of ontology "biological process"

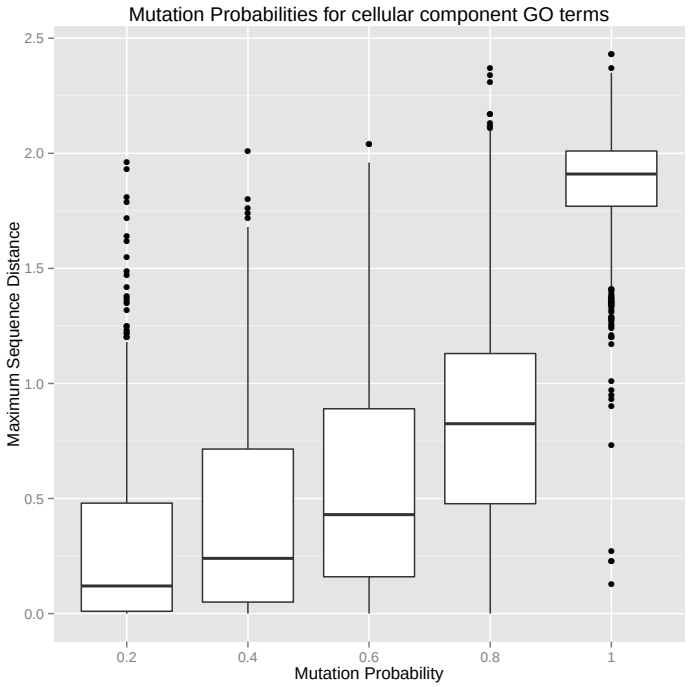

Figure 10.5.: Spread of maximum sequence distances in binned mutation probabilities for the 1707 GO terms of ontology "cellular component"

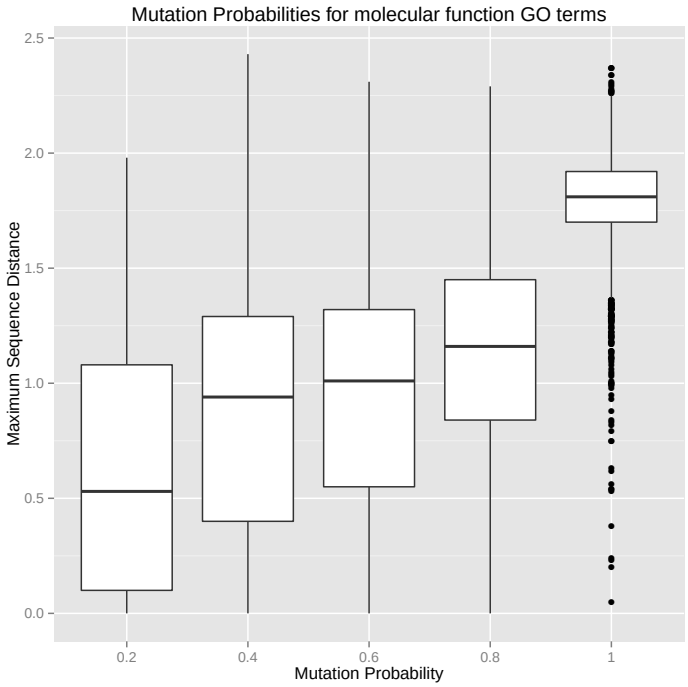

Figure 10.6.: Spread of maximum sequence distances in binned mutation probabilities for the 4624 GO terms of ontology "molecular function"

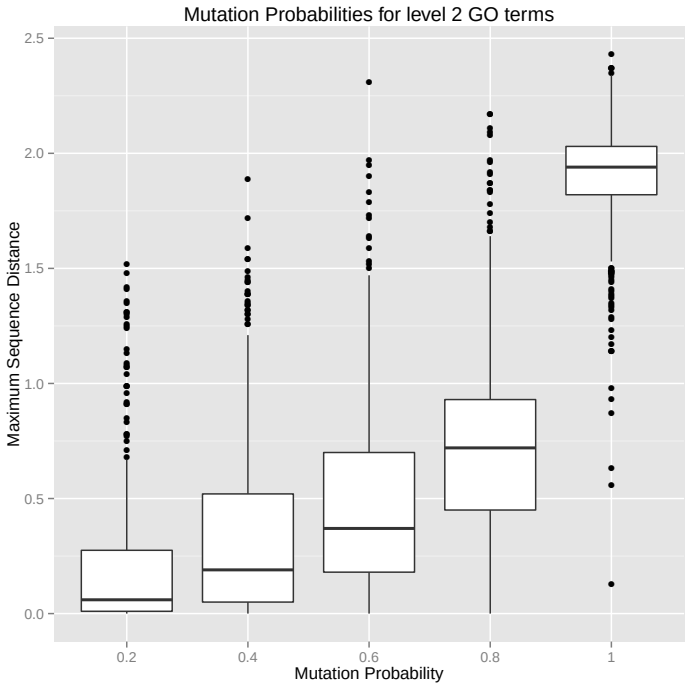

Figure 10.7.: Spread of maximum sequence distances in binned mutation probabilities for 90 GO terms of level 2

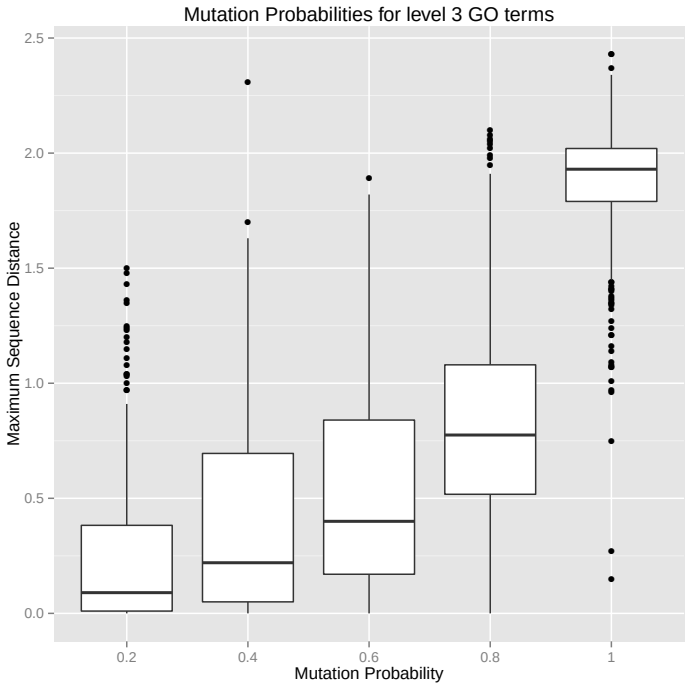

Figure 10.8.: Spread of maximum sequence distances in binned mutation probabilities for 419 GO terms of level 3

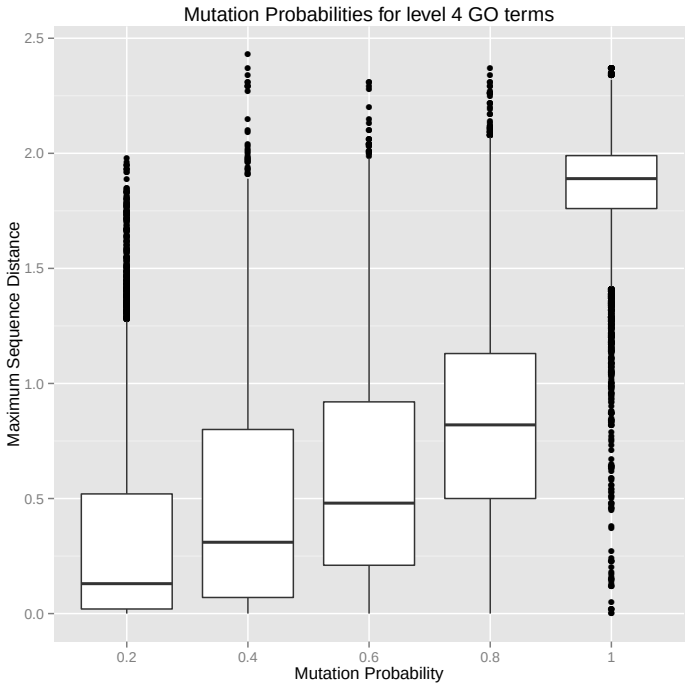

Figure 10.9.: Spread of maximum sequence distances in binned mutation probabilities for 18076 GO terms of level 4 and deeper

10.3.2. Examples

The result of applying PhyloFun (v2.0) on the query "Query_B7YZE7" (*uniprot.org/uniprot/-B7YZE7*) is given in figure 10.10 (page 80). Here the PhyloFun pipeline found 94 homologous proteins in the trust-UniKB database meeting the E-Value threshold of 10^{-6}, and subsequently identified 58 distinct candidate composite GO annotations for the "biological process" (BP) ontology, while there were 30 distinct candidates for the "cellular component" (CC), and finally 25 for the "molecular function" (MF) ontologies, respectively. The resulting BP composite annotation was only found in 9 homologs that clustered together with the query and had zero pairwise sequence distances, while the CC annotation was found only in three more distant homologs, and finally the MF annotation was selected from just two far homologs. Subsequent comparison with the reference (*uniprot.org/uniprot/-B7YZE7*) yielded a F2-Score of 0.8, and showed PhyloFun had annotated the query correctly with almost all reference GO term annotations of the BP ontology, only omitting the "transport" related terms. For the other two ontologies there were no *trustworthy* reference annotations, the only available were electronic annotations, which agreed both with experimentally inferred biological process characterizations as well as with the ones assigned by PhyloFun. The latter were in fact a finer annotation for the CC ontology, as PhyloFun localized the query in a "voltage-gated potassium channel complex", while the reference electronic annotation was a less specific "integral to membrane". While PhyloFun's MF annotation "cGMP-dependent protein kinase activity" does not agree with the available electronically produced reference annotation "voltage-gated potassium channel activity". But when looking into the annotations PhyloFun (v2.0) made for this query when used in its less restrictive mode (see chapter 7.2.3, page 47), the molecular function "voltage-gated potassium channel" is assigned among others.

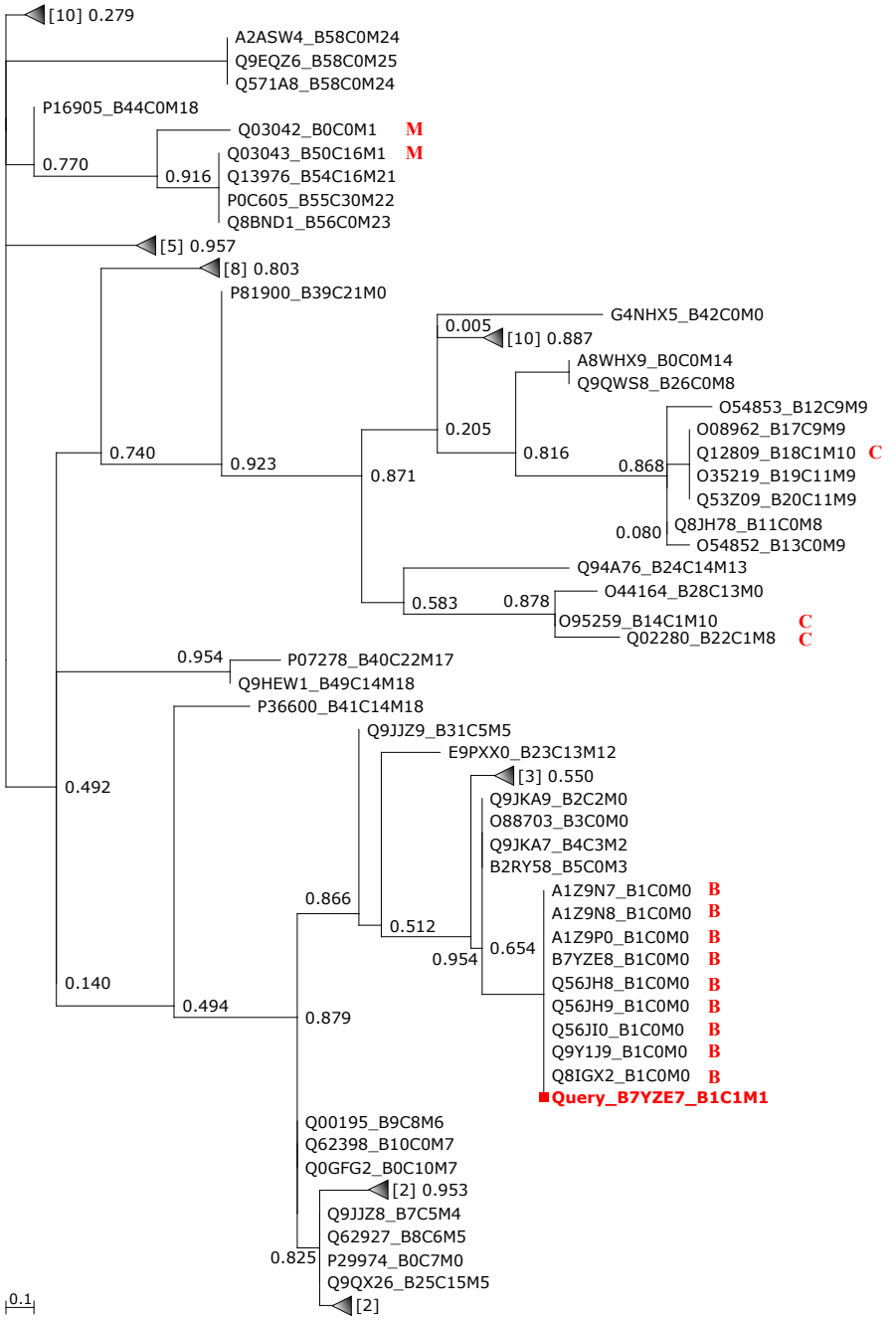

Figure 10.10.: Phylogenetic tree computed by the PhyloFun (v2.0) pipeline for the query "Query_B7YZE7" (*uniprot.org/uniprot/B7YZE7*), node labels are Shimodaira-Hasegawa local support values (Shimodaira and Hasegawa 1999). The query protein is highlighted in red as are those that were the source for PhyloFun's resulting annotations, where "B" denotes biological process, "C" cellular component, and "M" molecular function. Each protein accession is followed by an abbreviation of its composite GO annotations, one for each GO ontology, where the query was annotated with "B1C1M1": GO:0007637 (proboscis extension reflex), GO:0008340 (determination of adult lifespan), GO:0014059 (regulation of dopamine secretion), GO:0045475 (locomotor rhythm), GO:0045938 (positive regulation of circadian sleep/wake cycle, sleep), GO:0050802 (circadian sleep/wake cycle, sleep), "cellular component": GO:0008076 (voltage-gated potassium channel complex), and "molecular function": GO:0004692 (cGMP-dependent protein kinase activity). Explanation of other abbreviations is omitted. Collapsed subtrees are marked with shaded triangles, where the number in square brackets indicates the contained number of tips. Each of "B0", "C0", and "M0" means *no annotation* in the respective GO ontology.

In the following second example (figure 10.11, page 82) PhyloFun (v2.0) was applied on the query "Query_P38857" (*uniprot.org/uniprot/P38857*). Here the set of homologs consists of a smaller set of proteins obtained from a sequence similarity search with E-Value threshold 10^{-3}. Of these homologs the PhyloFun (v2.0) pipeline generated a phylogenetic tree with 18 tips and 12 internal nodes, in which the query forms a subtree together with the single homolog "P53311", that shares the reference annotations for the GO ontologies "biological process", and "molecular function", which PhyloFun correctly annotated. While the reference "cellular component" annotation, that PhyloFun also correctly assigned the query with is only found in the far homolog "P53157". This CC annotation had a much lower mutation probability than the other CC annotations found in the homologs (table 10.10, page 81). Altogether there were 3 different composite GO annotations of the "biological process" ontology, 6 of the "cellular component" ontology, and finally just a single "molecular function" annotation to be found in the set of homologs.

Table 10.10.: PhyloFun (v2.0) cellular component annotations mutation probabilities for "Query_P38857".

Composite CC annotation	Abbrev.	Mutation probability for branch length 0.5
GO:0031305 (integral to mitochondrial inner membrane)	C1	0.38
GO:0031966 (mitochondrial membrane)	C2	0.71
GO:0005886 (plasma membrane)	C3	0.67
GO:0005739 (mitochondrion)	C4	0.64
GO:0005739 (mitochondrion), and GO:0005774 (vacuolar membrane), and GO:0005886 (plasma membrane)	C5	0.92
GO:0005743 (mitochondrial inner membrane)	C6	0.6

"Abbrev" denotes the abbreviations appended to the protein accessions in figure 10.11 (page 82) to indicate the respective protein's annotation.

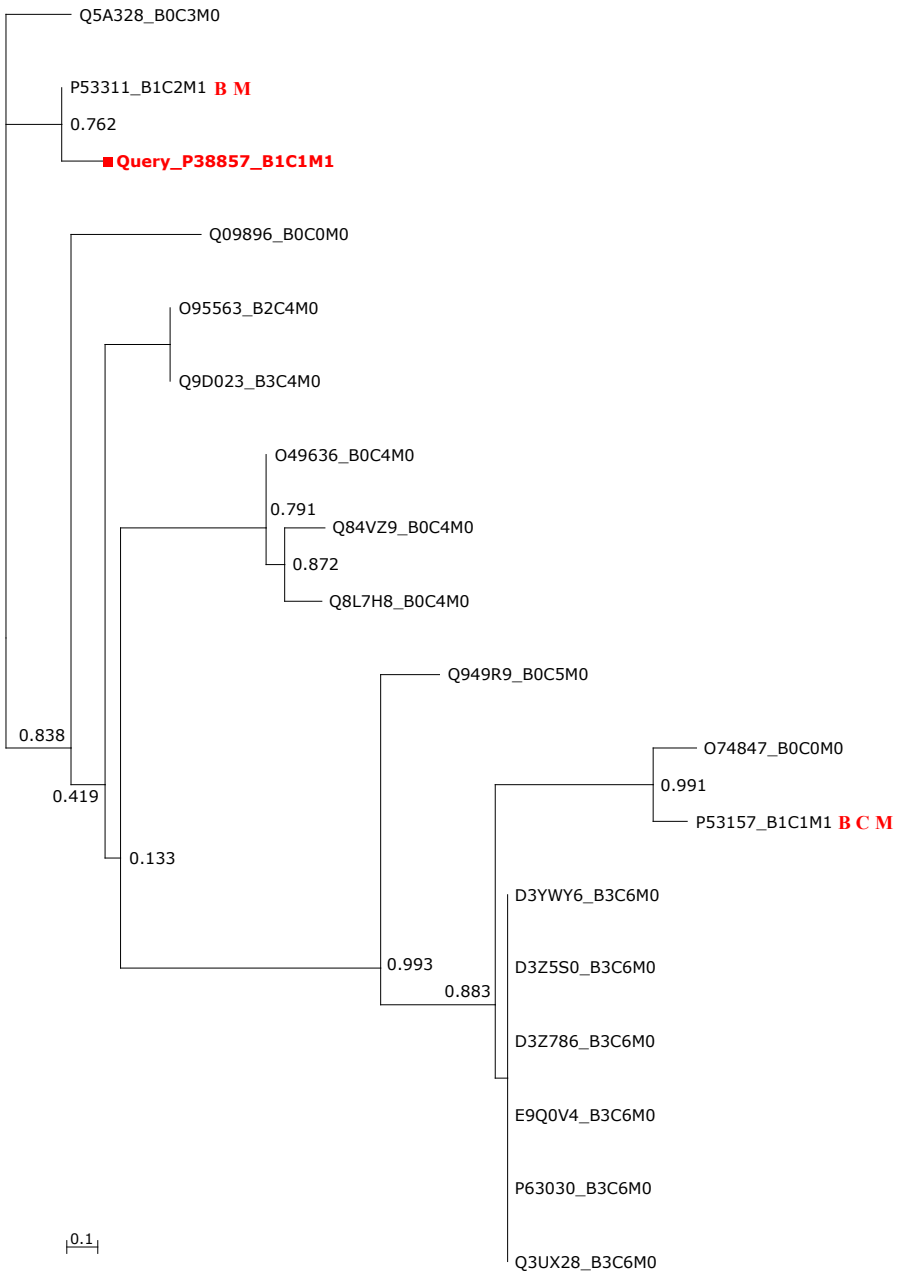

Figure 10.11.: Phylogenetic tree computed by the PhyloFun (v2.0) pipeline for the query "Query_P38857" (*uniprot.org/uniprot/P38857*), node labels are Shimodaira-Hasegawa local support values (Shimodaira and Hasegawa 1999). The query protein is highlighted in red, while the homologs that were the source of the resulting GO annotations are marked with red letters, where "B" denotes those that were source for "biological process", "C" for "cellular component", and "M" for molecular function annotations, respectively. These abbreviations are also used to annotate a homologs three composite GO annotations, in which "0" stands for "unknown". The Query was annotated with the following three composite GO annotations, one for each GO ontology: "B": GO:0006850 (mitochondrial pyruvate transport), "C": GO:0031305 (integral to mitochondrial inner membrane), and "M": GO:0050833 (pyruvate transmembrane transporter activity). Explanation of other abbreviations is omitted. Each of "B0", "C0", and "M0" means *no annotation* in the respective GO ontology.

In the final example (figure 10.12, page 84) the phylogenetic tree generated by PhyloFun (v2.0) for "Query_Q792F9" (*uniprot.org/uniprot/Q792F9*) consists of 63 internal nodes and 79 homologs, which were obtained from using the E-Value threshold of 10^{-3} on the respective sequence similarity search. These homologs had 55 different "biological process" composite GO annotations, 33 different "cellular component" annotations, and finally 8 different "molecular function" annotations. In this example PhyloFun (v2.0) was able to reproduce the correct reference annotation for GO ontology "cellular function" which is found in those two homologs with whom the query forms a branch of zero pairwise distances, as well as in some more distant homologs. While the prediction of the query's "molecular function" was incorrect and the original reference annotation "fibronectin binding" was even not found in any of the trees homologs. Also PhyloFun was not able to correctly reproduce the query's "biological process": heterophilic cell-cell adhesion, leukocyte cell-cell adhesion, blood vessel remodeling, heart development, cell migration, face development, chorio-allantoic fusion, and integrin-mediated pathway. PhyloFun failed to annotate these correct GO biological processes, even though the correct reference annotation can be found in the query's close homolog "Q8BQ25" to which the query has zero distance in the displayed tree. For the branch length 0 the *false* annotation "GPI anchor release" had *0.0* mutation probability, while in contrast the *correct* reference annotation had one of 0.52. Finally examining PhyloFun's result from its less restrictive mode (chapter 7.2.3, page 47) shows that the correct "biological process" annotation is assigned together with others.

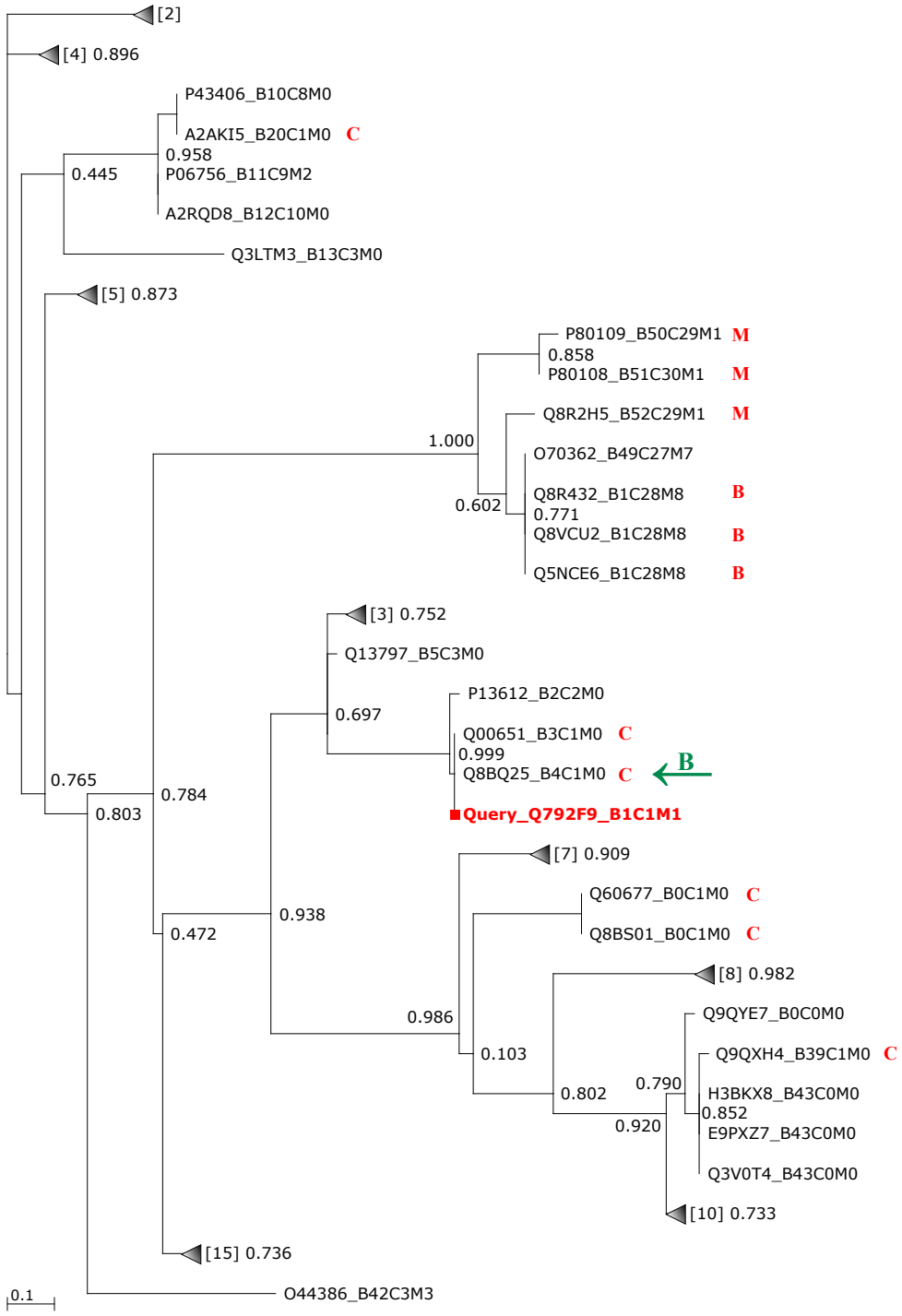

Figure 10.12.: Phylogenetic tree generated by PhyloFun (v2.0) for "Query_Q792F9", node labels are Shimodaira-Hasegawa local support values (Shimodaira and Hasegawa 1999). The query protein is highlighted in red, while the homologs that were the source of the resulting GO annotations are marked with red letters, where "B" denotes those that were source for "biological process", "C" for "cellular component", and "M" for molecular function annotations, respectively. These abbreviations are also used to annotate a homologs three composite GO annotations, in which "0" stands for "unknown". The Query was annotated with the following three composite GO annotations, one for each GO ontology: "B": GO:0006507 (GPI anchor release), "C": GO:0009897 (external side of plasma membrane), and "M": GO:0004621 (glycosylphosphatidylinositol phospholipase D activity). Explanation of other abbreviations is omitted. The green arrow marked with "B" indicates the homolog which has the correct reference "biological process" annotation. Collapsed subtrees are marked with shaded triangles, where the number in square brackets indicates the contained number of tips. Each of "B0", "C0", and "M0" means *no annotation* in the respective GO ontology.

10.3.3. Evaluation

We compared the performance, measured as the mean F2-Score (Rijsbergen 1979), of PhyloFun (v2.0) and its competitors Blast2GO (Conesa and Gotz 2008), and InterProScan (Zdobnov and Apweiler 2001). In this the mean F2-Score was computed from GO term annotations made for a set of 1000 randomly selected proteins (PF-test) from the earlier described trust-UniKB database. The sequence similarity searches, where required as input, were executed in the PF-search protein database, which contains all proteins from the trust-UniKB set excluding the 1000 randomly selected query proteins themselves (chapter 7.2.4, page 47). Of all compared methods PhyloFun (v2.0), used in its less restrictive mode, where it selects all GO term annotations whose probability exceeds that of equal distribution (PF_high_scr), and applied on the BLAST (Altschul, Madden, Schaffer, et al. 1997; McGinnis and Madden 2004) results that met the E-Value threshold of 10^{-6} outperformed all other competitors (table 10.11, page 86). Even PhyloFun (v2.0) applied on the same BLAST results but used in its more restrictive mode that only selects those GO term annotations with highest probability (PhyloFun_E-6), performed approximately equally well as Blast2GO's pipeline version with default settings (B2G_pipe) and applied on the same BLAST results, while the more comparable run of Blast2GO that only considered *trustworthy* GO annotations (chapter 7.2, page 43) as candidates (B2G_pipe_trusted) was still outperformed by the latter more restrictive PhyloFun annotations (PhyloFun_E-6). InterProScan (Zdobnov and Apweiler 2001) annotations were outperformed by all other compared methods, with the exception of them being more accurate "molecular function" predictions than those made by Blast2GO's GUI version (B2G_gui). These results from the measurements of mean annotation F2-Scores agree with both measurements of mean annotation recall (table 10.12, page 86) and mean annotation specificity (table 10.13, page 87) rates, with the exception that PhyloFun_E-6 had a lower mean recall rate than both B2G_pipe and B2G_pipe_trusted results, which was compensated by its mean specificity rate that was higher than those of both of the mentioned Blast2GO pipeline results. Again PhyloFun_E-6_high_scr's recall and specificityrates outperformed those of any other competitor.

As another measure of the competitors' sensitivity the number of pairwise distinct GO terms in each result as a whole were inferred, together with their mean GO levels (table 10.14, page 87), which is an indicator of each method's fineness. Here PhyloFun in its more restrictive mode annotated only approximately a quarter of the number of GO terms both Blast2GO pipeline setups, B2G_pipe and B2G_pipe_trusted, were able to annotate, while all PhyloFun setups achieved higher mean GO levels than the competitors outperforming them by approximately half a GO level. InterProScan however only annotated a small set of 428 GO terms with a mean GO level of 5.54.

Finally the intersections of each methods set of pairwise distinct annotated GO terms (table 10.15, page 88) show that PhyloFun in any setup approximately agrees in two thirds of its annotations with those made by the three different Blast2GO setups, while only 30–50% of the InterProScan annotated GO terms were also found in the different PhyloFun results.

Table 10.11.: Mean F2-Scores of GO term annotations made by PhyloFun (v2.0), Blast2GO, and InterProScan, for proteins in PF-test.

Tool	No. annotated Queries	FS-all	FS-BP	FS-CC	FS-MF
PhyloFun_PH	988	0.0621	0.0709	0.1203	0.1681
PhyloFun_PH_high_scr	987	0.1472	0.1613	0.2495	0.3091
PhyloFun_E-3	878	0.1155	0.1292	0.1967	0.3047
PhyloFun_E-3_high_scr	868	0.1475	0.1615	0.2507	0.3091
PhyloFun_E-6	883	0.1253	0.1408	0.2181	0.3029
PhyloFun_E-6_high_scr	874	0.1590	0.1799	0.2615	0.3259
B2G_gui	238	0.0578	0.0581	0.0794	0.0884
B2G_pipe	847	0.1391	0.1521	0.2098	0.2699
B2G_pipe_trusted	845	0.1252	0.1260	0.1777	0.2489
InterProScan	805	0.0544	0.0243	0.0359	0.1336

FS stands for mean F2-Score of all GO term annotations made the respective Tool and computed for the subset of GO terms specified by their ontology, which is one of "biological process" (BP), "cellular component" (CC), and "molecular function" (MF). "No. annotated queries" denotes the number of query proteins that received GO term annotations by the respective tool. The abbreviations used in column "Tool" are explained in table 7.3 (page 49)

Table 10.12.: Mean Recall rates of each methods GO term annotations.

PhyloFun run	mean Recall-Rate
PhyloFun_PH	0.0965
PhyloFun_PH_high_scr	0.3999
PhyloFun_E-3	0.1564
PhyloFun_E-3_high_scr	0.4024
PhyloFun_E-6	0.1729
PhyloFun_E-6_high_scr	0.4128
B2G_gui	0.1211
B2G_pipe	0.3996
B2G_pipe_trusted	0.3085
InterProScan	0.0400

The abbreviations used to denote the methods in column "Tool" are explained in table 7.3 (page 49).

Table 10.13.: Mean Specificity rates of each methods GO term annotations.

Tool	mean Specificity-Rate
PhyloFun_PH	0.9998
PhyloFun_PH_high_scr	0.9980
PhyloFun_E-3	0.9998
PhyloFun_E-3_high_scr	0.9980
PhyloFun_E-6	0.9998
PhyloFun_E-6_high_scr	0.9982
B2G_gui	0.9998
B2G_pipe	0.9991
B2G_pipe_trusted	0.9991
InterProScan	0.9999

The abbreviations used to denote the methods in column "Tool" are explained in table 7.3 (page 49).

Table 10.14.: Pairwise distinct GO terms computed from the annotations made by the competitors and their mean GO levels.

Tool	Number of GO terms	mean GO level
PhyloFun_PH	1508	5.939
PhyloFun_PH_high_scr	5203	6.129
PhyloFun_E-3	1456	5.942
PhyloFun_E-3_high_scr	5205	6.129
PhyloFun_E-6	1488	5.905
PhyloFun_E-6_high_scr	4986	6.113
B2G_gui	2390	5.901
B2G_pipe	6536	6.07
B2G_pipe_trusted	5758	6.035
InterProScan	428	5.54

Each methods pairwise distinct GO term annotations made for the query proteins in PF-test are shown. The abbreviations in column "Tool" are explained in table 7.3 (page 49).

Table 10.15.: Intersections of each methods pairwise distinct GO terms annotations.

Intersection	Number of GO terms
PhyloFun_PH ∩ InterProScan	169
PhyloFun_PH ∩ B2G_gui	429
PhyloFun_PH ∩ B2G_pipe	888
PhyloFun_PH ∩ B2G_pipe_trusted	820
PhyloFun_PH_high_scr ∩ InterProScan	290
PhyloFun_PH_high_scr ∩ B2G_gui	1424
PhyloFun_PH_high_scr ∩ B2G_pipe	3291
PhyloFun_PH_high_scr ∩ B2G_pipe_trusted	3009
PhyloFun_E-3 ∩ InterProScan	180
PhyloFun_E-3 ∩ B2G_gui	513
PhyloFun_E-3 ∩ B2G_pipe	992
PhyloFun_E-3 ∩ B2G_pipe_trusted	930
PhyloFun_E-3_high_scr ∩ InterProScan	290
PhyloFun_E-3_high_scr ∩ B2G_gui	1424
PhyloFun_E-3_high_scr ∩ B2G_pipe	3292
PhyloFun_E-3_high_scr ∩ B2G_pipe_trusted	3010
PhyloFun_E-6 ∩ InterProScan	191
PhyloFun_E-6 ∩ B2G_gui	546
PhyloFun_E-6 ∩ B2G_pipe	1062
PhyloFun_E-6 ∩ B2G_pipe_trusted	998
PhyloFun_E-6_high_scr ∩ InterProScan	283
PhyloFun_E-6_high_scr ∩ B2G_gui	1405
PhyloFun_E-6_high_scr ∩ B2G_pipe	3240
PhyloFun_E-6_high_scr ∩ B2G_pipe_trusted	2966

Abbreviations used in column “Intersection” are explained in table 7.3 (page 49).

10.3.4. Runtime

We assessed separately the runtime PhyloFun required to annotate each of the 1000 GO terms in the PF-test protein set, for each of which we allowed PhyloFun to occupy 10 cores in parallel. In 75% of these cases the whole PhyloFun pipeline terminated in less than 2 minutes (table 10.16, page 88), where maximum likelihood reconstruction of the phylogenetic tree and generation of the Bayesian Network required most resources.

Table 10.16.: Distribution of PhyloFun’s runtimes in minutes measured separately while annotating the 1000 query proteins in PF-test set with GO terms and using 10 cores in parallel.

Minimum	1st Quartile	Median	Mean	3rd Quartile	Maximum
0.0593	0.1617	0.4015	11.1017	1.8183	481.1667

Part IV.

Discussion

11. Automated Assignment of Human Readable Descriptions (AHRD)

11.1. Performance evaluation

AHRD (*github.com/groupschoof/AHRD*; Hallab, Klee, Srinivas, and Schoof 2014) was designed to computationally reproduce the decision making process of human expert curators and outperforms the competitive methods available to us for assigning human readable descriptions to new genome annotation datasets (table 8.3, page 57). AHRD produces a higher proportion of predictions that are identical to the reference description (26% on the *B. graminis* test set (Spanu, Abbott, Amselem, et al. 2010)), indicating success in reproducing the decisions of curators. I show by example that AHRD can overcome difficulties caused by multi-domain proteins with only partial homology to the query, as well as those caused by wrong annotations in databases (figure 8.1, page 53). In cases where high-quality databases like Swissprot (Bairoch and Apweiler 2000; Boeckmann, Bairoch, Apweiler, et al. 2003) do not contain relevant hits, descriptions from comprehensive databases like trEMBL (Bairoch and Apweiler 2000; Boeckmann, Bairoch, Apweiler, et al. 2003) are utilized automatically. The scoring and its parameters were initially developed intuitively, but then evaluated and optimized. This was done using three independent sets of reference proteins: A random subset of the *B.graminis* proteome, a set of Tomato resistance proteins, and a selection of the Swissprot database (section 5.3, page 35).

The characteristics of these three reference protein sets were assessed and compared with the *A.thaliana* proteome and the *full* Swissprot database (section 5.3.1, page 36) in order to answer the question whether certain statistical characteristics of the reference protein sets favored distinct optimal parameters. Here, three results strongly suggest that the *B.graminis* test set is best suited both for performance evaluation as well as parameter optimization. First, the *B.graminis* reference set contains the highest number of distinct descriptions, and hence, of the three reference sets, covers most protein functions (table 8.4, page 59). While we cannot exclude that this observation is due to the use of distinct descriptions of the same protein functions, this counter argument appears to be unlikely true, first because the descriptions were assigned manually by expert curators, who claimed to have aimed at conciseness, and also because of three quarters of these *B.graminis* proteins show a pairwise sequence identity of less than 37% (table 8.5, page 60), which strongly suggest them being functionally different (section 1.3, page 15). The second argument for the *B.graminis* references being best suited for our needs, is that, of the three, it has the highest protein description diversity, while the other two sets show an over-representation of frequently annotated descriptions (table 8.4 and figure 8.5, page 59). Additionally to this observed bias towards few but often annotated descriptions the Swissprot set has many more high scoring BLAST hits in the *full* Swissprot database than have the other two reference sets (figure 8.6, page 60). Thus the conclusion is supported, that this bias is probably a result from the manual expert selection and revision of UniprotKB/Swissprot entries (Boeckmann, Bairoch, Apweiler, et al. 2003) which appears to propagate descriptions from references to new proteins if the curator is convinced of their functional identity (section 5.3.1, page 36). Given this, the aim was not reached to form the Swissprot reference set from new and functionally diverse proteins, that are not of the subset of frequently studied and commonly annotated proteins (section 5.3, page 35). The third evidence is a reciprocal 10 fold relationship between size and distribution measures, observed between

the *B.graminis* references and the arabidopsis proteome (section 8.5, page 56). Meaning that, while the *B.graminis* set only contains ten times less distinct descriptions than does the arabidopsis proteome the description frequencies in the first three quartiles of the *B.graminis* set are ten times higher than those of the arabidopsis proteome (table 8.4, page 59). Hence the *B.graminis* set approximately shows the characteristics of a tenth size random selection from the arabidopsis proteome. And thus, of the three reference sets used for evaluation and optimization, the *B.graminis* set is best suited to find optimal parameters for the application of AHRD on whole genomes, where it is important to annotate also novel, not yet intensively studied or rare protein classes as well as perform well on a wide variety of functions (section 1.1, page 12).

Meanwhile, changing the parameters of AHRD scoring did *not* dramatically affect the performance on the *B. graminis* dataset (table 8.7, page 62). Parameters optimized for *B. graminis* worked well on the other datasets (table 8.7, page 62), with somewhat lower performance on the tomato dataset (Consortium 2012) and, in some cases, higher performance on the Swissprot dataset (Bairoch and Apweiler 2000; Boeckmann, Bairoch, Apweiler, et al. 2003). Optimizing parameters on the tomato and Swissprot datasets led to a somewhat decreased but comparable performance on the *B. graminis* dataset. I conclude that AHRD scoring is robust and not dependent on precise adjustment of the parameters. Setting the overlap score and all database related weights to zero improved performance on the *B. graminis* dataset slightly, but significantly on the Swissprot dataset, while reducing the performance on the tomato dataset (table 8.7, page 62). This parameter set makes AHRD more similar to best BLAST hit methods (Altschul, Madden, Schaffer, et al. 1997; McGinnis and Madden 2004), which perform very well on the Swissprot dataset. While this optimizes the identity criterion used by the evaluation score, manual inspection of results seems to indicate that sometimes, more concise descriptions are possible (Bangalore 2013), and we would thus favor the AHRD results that optimize for those. In my opinion, this observed bias is a consequence of the high curation standards at Swissprot, which result in highly consistent descriptions, and of the composition of the test set. The Swissprot test set contains proteins which all have highly significant matches in the Swissprot database (see figure 8.6, page 60). While we selected proteins for the test set based on recent addition to Swissprot, the bias towards proteins with highly significant matches to older Swissprot entries probably has to do with the way proteins are selected for annotation by the Swissprot curators. In contrast, in genome wide datasets from higher eukaryotes, for example the tomato genome, about approximately 30% of the proteins have no highly significant hit in the Swissprot database. In these cases, AHRD can improve over best BLAST hit methods. As for now AHRD does not make use of the annotated conserved protein domains from the InterPro database (Apweiler, Attwood, Bairoch, et al. 2000), but adds these annotations to the descriptions.

11.1.1. Accuracy of textual descriptions

The research goal to develop a new procedure to systematically measure the accuracy of textual protein descriptions was successfully reached. The F-measure is a widely used evaluation method (Rijsbergen 1979) that has also been applied in the global "Critical Assessment of Function Annotation experiment (CAFA)" (Radivojac, Clark, Oron, et al. 2013) comparing the accuracy of electronic tools assigning GO terms to query proteins (section 1.10, page 18). Defining a true positive as a case insensitive match between a word found in the predicted and a word contained in the reference protein description enabled the direct computation of the mentioned F2-scores for electronically assigned human readable descriptions (HRDs). Hence this new procedure to assess the accuracy of assigned HRDs could be applied to compare AHRD's performance with that of other competitive tools, and as well served as an objective function during parameter optimization.

However, when comparing HRDs in the high scoring segment, where the mean minimum F2-score

equalled 0.74 and the theoretically best achievable mean F2-score was 0.89 (table 8.7, page 62), some concerns about the applied F-measure's ability to reflect semantic differences are raised, at least in certain cases. Bangalore gives 7 examples where two competitive HRDs received different scores, because one competitor contained the additional and semantically uninformative word "putative" (Bangalore 2013). In 7 other examples the additional words "family", domain "containing", "protein", "like", and "2" caused differences in the resulting F2-scores while the compared descriptions had no or very little semantic differences (chapter 8 "Appendix" in Bangalore 2013). These case studies suggest that in the high scoring segment of competitive HRDs, differences in the F2-score do not necessarily reflect *true* semantic differences. Especially, one design goal in AHRD was to avoid descriptions that contain fill words that bloat descriptions without adding information, thus prioritising concise descriptions. However, comparing a concise but semantically identical description to a longer one leads to a less than perfect F2-score, even though based on our design criteria we would favor this description over the identical, longer one. Hence further refinements of this accuracy measure are required. To this end the effect of a simple filter excluding uninformative words like "protein" from the evaluation could be assessed. Also a dictionary of synonyms e.g. for enzymatic functions would be very useful.

11.1.2. Parameter optimization with Simulated Annealing

The simulated annealing approach used to find locally optimal parameter sets for the *B.graminis* references yielded an increase of 4 of the objective function (chapter 8.5, page 56). Furthermore the optimal parameters found for the Tomato references also increase AHRD's performance in comparison with the original intuitive settings, in spite of the fact, that the simulated annealing approach failed on the Tomato references (section 8.5.1, page 61). In contrast optimizing on Swissprot references caused a strong bias in AHRD's procedure to preferably annotate with Swissprot protein descriptions. In short "Swissprot reference proteins like to be annotated with descriptions already present in the Swissprot database" (section 8.5.1, page 61). This observation suggests two important conclusions. First AHRD does indeed manage to mimic the decision process of a human curator, because when optimized on Swissprot references, AHRD preferably annotates Swissprot descriptions. This preference is visible in the fact, that many protein descriptions in the Swissprot database are identical, hence the human curators involved in revising every single database entry (Boeckmann, Bairoch, Apweiler, et al. 2003) clearly aim to apply a standard nomenclature and pass protein descriptions from highly similar sequences to new database entries, when convinced of functional equality. The process of expert revision also enriches proteins that are in the context of popular research topics, because for the functions of these proteins more experimental verifications exist. This over-representation of proteins belonging to well studied research topics yields the announced second important conclusion. Namely should parameters found to be optimal for Swissprot references *not* be applied when annotating proteins on a genomic scale, because a genome contains also many proteins belonging to poorly studied groups, for which no, very few, or at least poorly similar curated homologs can be found in the well trusted Swissprot database. Considering the Tomato reference set, also used in this optimization, a strong over-representation of resistance genes is found (section 8.5, page 55). Because resistance genes are well studied, to no surprise, most of these reference descriptions resemble entries found in the Swissprot database. Hence optimal parameters obtained for the Tomato references are also not suitable when applying AHRD to annotate whole proteomes. On the other hand the B.graminis references are randomly selected from an expert annotated fungal proteome (section 5.3, page 35) and hence the parameters found optimal for this gold standard are much more recommendable for the task of annotating a whole query proteome.

In conclusion these considerations of reference sets, our design criteria applied to AHRD diverges

from the manual annotation process of Swissprot curators; while consistency in descriptions is desirable, it is more important to us to annotate a wide variety of protein classes and to provide concise descriptions that are readable in short formats such as BLAST hit tables. These preferences are not reflected in the F2-Score when the reference set does not implement these preferences, and thus Swissprot and tomato references, in our view, do not represent good optimization targets.

In the cross validation of found to be optimal parameter sets the mean F2-scores range between 0.62 and 0.67, with the best achievable evaluation score of 0.89 when annotating *B.graminis* queries (table 8.7, page 62). Because here also the distribution of parameter values in the upper quartile of high scoring parameter sets covered approximately their whole value-intervals (table 8.8, page 63), we concluded the before mentioned robustness. This conclusion is also supported by the quite narrow range (0.5475–0.6777) of mean F2-Scores assumed by any tested parameter set during optimization (chapter 8.5, page 56). Hence the research goal to develop a reliable and robust tool to annotate query proteins with human readable descriptions was reached.

Optimization by simulated annealing apparently walked through a quite smooth parameter-score-landscape, as most slight parameter changes yielded no change in the objective function (tables 8.9–8.10, page 63). While somewhat in contrast to this, those parameter modifications that *did* result in a changed mean F2-Score revealed localized "microscopic roughnesses" of this landscape. "Microscopic" because firstly in these cases the mean change of the objective function was as low as 10^{-4} while even the maximum absolute difference was not higher than 0.017 (table 8.10, page 63), and "roughness" secondly because changing the same parameter again almost never resulted in an repeated improvement or worsening of performance, respectively. This "microscopic roughness" impedes the application of a pure hill climbing optimization approach that expects a smooth increase or decrease in the objective function while walking the parameter space in any given direction — at least along an axis of the parameter space.

Altogether the above robustness and large spread of parameter values found to be optimal by simulated annealing and subsequent cross-validation supports the satisfying conclusion, that a user of AHRD does not need to infer optimal parameter values for the task at hand, and hence does not need to retrain AHRD for every new dataset. Furthermore, as mentioned, in the high scoring segment of competitive protein descriptions, differences in the evaluation score do not necessarily reflect true semantic differences (section 11.1.1, page 91). From these three observations can be concluded that optimizing AHRD towards the maximum achievable evaluation score is not only impeded to some degree, but also does not lead to concise and varied protein descriptions for a wide range of functions, as specified by our design criteria, while of course different design criteria may lead to other optimal parameters. In any case, AHRD clearly outperformed its competitors, whose mean evaluation scores did *not* range in the high scoring segment (table 8.3, page 57) and also had large fractions of bad performing annotations with F2-scores ≤ 0.1 (figure 8.5, page 58). Hence it can be concluded that these competitive annotations not only performed worse than those assigned by AHRD, but indeed in most cases failed to even approximately describe the query proteins, simply because they shared too few words — often none — with the reference descriptions.

11.2. Scoring Domain Architecture Similarity

When extending AHRD to take into account the similarity between a query's and a candidate protein's domain architecture the overall quality of the resulting descriptions did not change significantly. In fact only a very small number of Queries received a description that had different F2-Scores using this extended method (Dom-Sim-Arch-AHRD) (chapter 8.6, page 67). The lack of improvement might not necessarily point to Dom-Sim-Arch-AHRD not being useful, because considering similarity in protein

domain architecture has already been shown to improve protein characterization (Messih, Chitale, Bajic, et al. 2012). Bangalore also points out, that for a large number of proteins in the UniprotKB databases (Boeckmann, Bairoch, Apweiler, et al. 2003; Bairoch and Apweiler 2000) there simply is no available protein architecture information thus impeding the comparison of domain architectures and the evaluation of this method extension (Bangalore 2013). Furthermore some descriptions had a decreased F2-Score while on manual inspection they were no worse annotations, either due to uninformative fill-words or a faulty reference (Bangalore 2013). Bangalore shows by example how taking into account similarity of domain architecture can help overcome possible propagations of faulty protein characterizations that have been shown to occur frequently when basing ones predictions solely on sequence similarity (Gilks, Audit, Angelis, et al. 2002). Finally the value of the *Blumeria graminis* reference set in the context of evaluating the performance of the extended Dom-Sim-Arch-AHRD method has to be questioned, because the expert curators used manually inspected BLAST results as the source for their candidate descriptions, hence biasing these references descriptions to preferably equal those of high scoring BLAST Hits (Bangalore 2013).

We conclude, that the proposed extension to AHRD might increase accuracy and reliability and help overcome problems in passing descriptions from faulty annotated database proteins. Because this applies to only very few proteins, the effort is not yet justified, which may change when more protein domain annotations become available.

12. Human Readable Descriptions for Tomato gene families

"AHRD on gene clusters" proved to provide the Biologists, the so called "Gene Family Captains", of Consortium with valuable short descriptions to enable a quick selection of those gene families they wanted to investigate further. Together with the phylogenetic trees, generated with the neighbour joining method (Saitou and Nei 1987), that come with the OrthoMCL (Van Dongen 2008) output, these gene family descriptions enabled assessing the families' evolutionary history, along with such properties like gene expansion or loss, and the emergence of new functions or them becoming obsolete.

High confidence can be put into those descriptions, where the frequency of the InterPro annotations selected by "AHRD on gene clusters" was high, since the majority of the gene family had been annotated with the chosen description. This confidence is derived from the fact, that InterProScan (Zdobnov and Apweiler 2001) has widely been applied and shown to produce reliable results (section 1.2, page 13 and Zdobnov and Apweiler 2001). To our satisfaction half of the gene families with Tomato members had frequencies (scores) surpassing 0.7 (table 9.1, page 67) and thus the assigned human readable descriptions suggest themselves as being reliable protein family characterizations.

In those cases where such high confidence in the descriptions can not readily be concluded, at least the assigned descriptions could give a rough idea of the gene family's characteristics (see table 9.3, page 68). The largest gene family "ORTHOMCL0" was described as a group of deubiquitinating enzymes (9.4, page 69), where the description score suggests that this characterization should be treated with some care. But manual inspection of the annotated InterPro Family "IPR003653" (*ebi.ac.uk/interpro/entry/IPR003653*) shows that this family is itself a member of a diverse group of "peptidases and peptidase homologues" that "are grouped into clans and families". This corresponds to the large size of the annotated gene family and suggests that the assigned description characterizes

it accordingly.

These results encourage to assess in the future the quality of gene family descriptions made by "AHRD on gene clusters", for which one could use a suitable reference set of gene families — for example a selection of the Pfam database (*Pfam - Sanger Institute*) — and evaluate the predictive quality of the assigned descriptions in a similar manner as it was done for AHRD, that is by treating the resulting descriptions as sets of atomic words, which can be true or false positives in the predictions.

Taking into account that for a typical plant genome more than 20,000 gene clusters can be identified among related species, "AHRD on gene clusters" provided useful and fast means to access these groups of homologous proteins. Thus we recommend this method for a rapid overview of clusters, but would suggest further testing before assuming a single protein from a cluster actually performs the function suggested by the cluster description, e.g. using AHRD and PhyloFun on this protein.

13. PhyloFun

13.1. Evaluation of Version 1.0

We annotated the Tomato (*Solanum lycopersicum*), and the *Medicago truncatula* predicted proteomes with Gene Ontology (GO) terms using PhyloFun (v1.0) and InterProScan (v4.5) (Zdobnov and Apweiler 2001). They proved to nicely complement each other because the union of annotations made by both covered approximately half of each proteome, while each tool showed to have its specialised expertise in annotating respective sets of GO terms, which had very little overlap (tables 10.1 and 10.4, page 69 and 72, respectively). This shows how some protein characteristics are better predictable when considering conserved protein domains while other characteristics appear to be more accurately detectable by amino acid subsequences approaching the length of complete proteins.

When GO term annotations provided by Blast2GO (Conesa and Gotz 2008) for a set of Tomato genes mainly involved with pathogen resistance were evaluated by experts from Consortium their judgement was that many of the assigned annotations were wrong, which motivated them to manually annotate those genes, and lead us to discontinue the usage of this annotation pipeline. While to out satisfaction a great part of the GO term annotations made by PhyloFun (v1.0) and InterProScan (v4.5) (Zdobnov and Apweiler 2001) was estimated as being correct.

As mentioned before (chapter I, page 12) many different methods for functional characterisation of predicted proteins have been developed (Pierri, Parisi, and Porcelli 2010; Hawkins and Kihara 2007; Rentzsch and Orengo 2009), but to our knowledge very few have been applied in high throughput environments and on genomic scales. Our results show that the combination of PhyloFun and InterProScan (Zdobnov and Apweiler 2001) annotations provide a good coverage, high fineness — measured as the mean GO level of the predicted proteins (tables 10.2 and 10.5, page 70 and 72, respectively) —, and reliable annotations.

However the usage of PhyloFun (v1.0) provided some problems as maintaining an up-to-date database and including a larger set of reference species turned out to be tedious or even impossible, a result obtained when Haili Song updated the relational database PhyloFun (v1.0) required to lookup amino acid sequences and GO annotations for found homologs (chapter 7, page 41). The resulting database was very large and it took several months to update it, partly because some data was acquired via web services that did not enable fast recovery of large batches of data. Another result of

updating this database was that reference proteins belonging to species formerly not included in the database now needed to be present in the manually curated species tree in order to enable identification of duplication and speciation events, respectively (Zmasek and Eddy 2001). This extension of the species tree turned out to be tedious and for some species impossible, simply because their exact position in the tree of life is yet unknown.

13.2. Evaluation of Version 2.0

13.2.1. Objectives

We concluded from the problems observed when preparing PhyloFun (v1.0) to annotate fresh data (see chapter 13.1, page 95), that especially if we wanted to provide the scientific community with a useful, easy to install, as well as easy to apply protein annotation tool (section 1.1, page 12) PhyloFun had to be redesigned. Apart from this motivation we aimed to avoid the propagation of annotation errors by relying only on *trustworthy* sources and also used the occasion to base the computation of GO term annotation probability distributions on empirical measurements (see part 2.4, page 22) rather than on a preconceived model (Engelhardt, Jordan, Muratore, and Brenner 2005; Engelhardt, Jordan, Srouji, and Brenner 2011).

13.2.2. Calibration

In order to enable PhyloFun to base the computation of GO term mutation probabilities on empirical assessments, calibrations were required, which we made for each GO term found in the previously described trust-UniKB protein dataset — annotated as *trustworthy*, i.e. experimentally verified or curator made (section 7.2, page 43). The resulting mutation probability lookup tables proved to have increasing mutation probabilities for increased sequence distances, thus fulfilling our expectations. While they did not significantly differ when estimated separately for the two different GO ontologies "biological process" and "cellular component" nor when split up into subsets of different GO levels (chapter 7.1, page 43), they had a significant number of outliers. In my opinion the latter demonstrates, that the approach to calibrate mutation probabilities separately for each GO term does make sense as the likelihood of a descending protein sharing its ancestors GO term differs significantly between given GO terms for any given branch length (tables 10.7–10.8, page 74). This is also supported by the observation, that GO terms of the "molecular function" ontology show approximately double mean maximum sequence distances when compared with the other two ontologies mentioned before, because cellular localization and involvement in biological processes are protein characteristics that are lost easier than molecular functions when sequence mutation occurs, and thus the computation of their annotation probability should be based on individualized mutation models.

40% of the GO terms PhyloFun (v2.0) has calibrated mutation probability lookup tables for are estimated to have a mutation probability of 1.0 for any given sequence distance (chapter 10.3.1, page 74). This is because, when computing the mutation probabilities on each of these GO terms' set of protein pairs sorted by their ascending sequence distances the very first pair already did not share the respective GO term, which resulted in this overestimation of its mutation probability (table 10.7, page 74). This for the so affected "always mutating" GO terms leads to a significant decrease of sensitivity, and hence upon this point one should first focus future work on PhyloFun. Such an improvement of sensitivity could be achieved by using a "pseudo count" during computation of the GO term mutation probability tables, i.e. introducing a pair of sequence identical proteins that share the GO term annotation. Using this approach one would correctly introduce the GO term mutation probability 0 for identical proteins. As a result, even if the very next pair in the list of increasing

sequence distances does not share the respective GO term annotation, the mutation probability for the respective sequence distance would only be set to 0.5 instead of 1.0. As a practical approach to achieve this one could simply include the self matches from the sequence similarity searches which perfectly fulfill the requirement of sequence distance 0 and sharing the GO term annotation.

Another improvement of the generation of GO term mutation probability lookup tables might be the consideration of parent child relationships in the GO directed acyclic graph (GO-DAG) (Ashburner, Ball, Blake, et al. 2000), such that annotations of a child term are also treated as annotations of its parent terms. This required preprocessing of the annotations and adding parent terms to each proteins' annotations. In doing so missing annotations of parent terms in the reference set would no longer have an effect on the computation of GO term mutation probability lookup tables.

13.2.3. Tree rooting

Also an improvement to the PhyloFun (v2.0) pipeline might be rooting the generated phylogenetic trees with outgroups. Currently the message passing algorithm (Pearl 1988) is implemented to interpret the phylogenetic tree `(A,(B,C));` (figure 13.1, page 97) rooted as shown in tree "Root 1". In most cases this might be the correct interpretation, because tips "B" and "C" are clustered together, but the location of the true evolutionary root itself might be different and revealed when the tree is rooted using e.g. a convenient outgroup as shown in tree "Root 2". As the position of the root does influence where in the Bayesian Network evidence is gathered and propagated towards the query's tip (Pearl 1988) the effect of outgroup rooting on the quality of PhyloFun's predictions should be investigated.

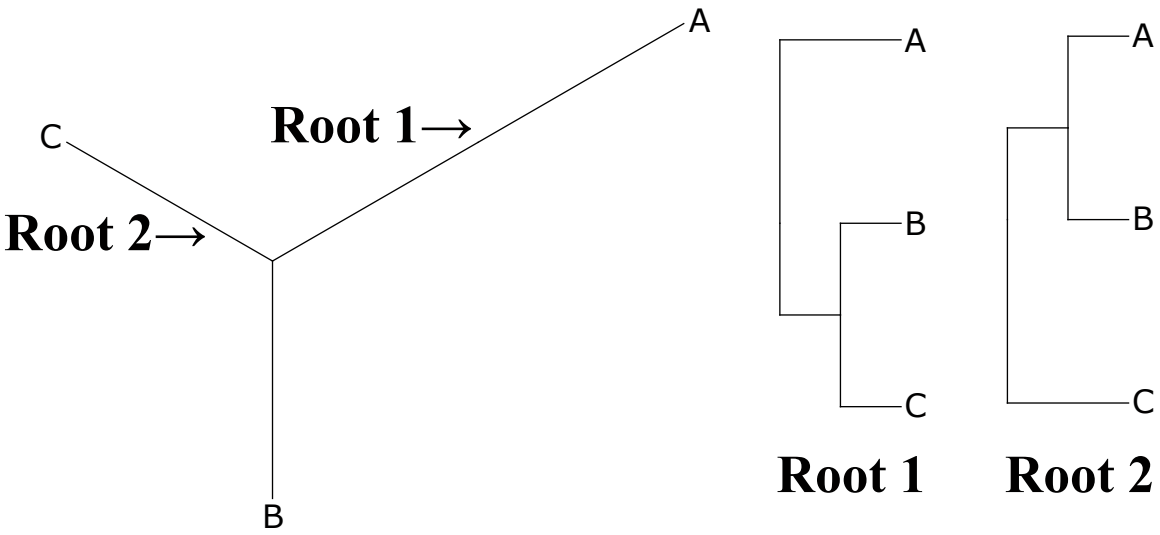

Figure 13.1.: Phylogenetic tree rooting

13.2.4. Predictive evidence

Another option, whose effect on the quality of annotations produced by PhyloFun (v2.0) could be evaluated, is the usage of "predictive" evidence at the root node of the phylogenetic tree (see chapter 3, page 23). One could measure composite annotation frequencies in a suitable reference set and extend PhyloFun (v2.0) to initialize the function annotation probability distribution of the root node to the normalized observed frequency of the composite function annotations present in the tree.

This application of "predictive" evidence, Baysian "prior" probabilities, would render rare composite function annotations less likely even if found annotated often in the query protein's homologs, while seldom annotated functions that have an high observed frequency in the reference set would receive higher posterior probabilities. Such an approach might aid PhyloFun (v2.0) to avoid being biased by annotations found in the input tree and thus to overcome for example the percolation of annotation errors made by curators (Gilks, Audit, Angelis, et al. 2002).

13.2.5. Performance

The performance of PhyloFun (v2.0) was inferred as the mean F2-Score of GO term annotations made on a randomly selected set of 1000 reference proteins with experimentally verified or at least curator made annotations (PF-test — section 7.2.4, page 47). I compared PhyloFun's performance with that of the competitive annotation methods "InterProScan" (Zdobnov and Apweiler 2001) and "Blast2GO" (Conesa and Gotz 2008), of which both the graphical user interface version and the pipeline version were executed. This evaluation was executed on different datasets (section 5.13, page 38). PhyloFun (v2.0) outperformed all of its competitors (table 10.11, page 86) and furthermore provides the user with a highly accurate and reliable annotated phylogenetic tree (e.g. figure 10.11, page 82), which shows the evolutionary relationships of the query proteins. Because its resource requirements are comparable to those of InterProScan and are in most cases not much higher than those of the Blast2GO pipeline version I recommend the usage of PhyloFun, especially when applied in a high throughput environment or on genome scale (section 1.1, page 12). I recommend PhyloFun (v2) over Blast2GO because the pipeline version of Blast2GO does not parse BLAST XML result files without custom pre processing (figure 7.2.4, page 48) and the graphical user interface version regularly stopped its execution when it was fed more then 800 query proteins.

PhyloFun's usefulness is also demonstrated in three examples (section 10.3.2, page 79). Here PhyloFun (v2.0) does not loose the ability to pass on annotations found in highly similar sequences (figures 10.10 and 10.11, pages 80 and 82), but at the same time the consideration of different protein characteristics mutation probabilities enables it to make accurate predictions even when only far relatives are annotated with these (predicted "cellular component" annotation in figure 10.11, page 82). In example one for "Query_B7YZE7" (page 79) PhyloFun was able to make the very reasonable prediction that the Uniprot protein "B7YZE7" should be localized in a "voltage-gated potassium channel complex", supported by the fact, that it already has the electronically made *reference* function annotation "voltage-gated potassium channel activity". This example also shows how the less restrictive mode of PhyloFun is able to compensate for possible annotation errors, as the results obtained from this mode agree with the reference molecular function annotation, while the more restrictively made function annotation "cGMP-dependent protein kinase activity" can not be found in the reference. Although one might argue that this molecular function predicted by PhyloFun (v2.0) should not necessarily be rejected in favor of the electronically made "voltage-gated potassium channel activity" annotation, because both the missing experimental evidence as well as the relatedness of the two annotations (see chapter 10.3.2, page 79) impede deciding on a correct annotation. The second example for "Query_P38857" (page 81) also gives good evidence for the usefulness of GO term specific calibrations of their respective mutation probabilities, as the correct composite "cellular component" annotation is only found in a single relatively distant homolog, but has a much lower mutation probability. Example three for "Query_P38857" (page 83) supports the earlier suggestion to introduce zero mutation probability rows at the beginning of all GO term mutation probability lookup tables, because the *false* "biological process" annotation had such zero mutation probability on the branch directly leading to the query, while the *correct* annotation had one of 0.52, in spite of it being assigned to the query's close homolog to which it had *no* sequence distance. Hence this shows even more how

all GO term lookup tables should have the mentioned zero mutation probability row, because the fact that some already have such rows, strong evidence for correct annotations can be biased as happened in example three.

13.2.6. PhyloFun modes

PhyloFun can be run in two modes (section 7.2.3, page 47), where the less restrictive mode selects all GO term annotations that receive a higher probability than equal distribution, while the restrictive mode assigns only those GO term annotations that receive highest annotation probabilities and are annotated to a single reference homolog. The comparison of both on our test set showed, that the less restrictive mode was able to annotate approximately 3.3 times more pairwise distinct GO terms than the more restrictive one (table 10.14, page 87), and also had higher mean F2-Scores (table 10.11, page 86) as well as higher mean specificity rates (table 10.13, page 87). In spite of this we believe that PhyloFun's restrictive mode has the advantage that most likely its GO annotations *do not* contradict each other, because they are obtained *as a whole* from the reference proteins and are not mixed with annotations made to *other* references, as is the case in the less restrictive mode. Hence, in case one requires most reliable GO annotations, the restrictive mode is recommended. On the other hand, if the risk of having contradicting annotations is acceptable and a higher sensitivity is wanted, the less restrictive mode should be applied.

13.2.7. Complementary annotation methods

Both versions of PhyloFun produced GO term annotations that when compared with annotations made by InterProScan (Zdobnov and Apweiler 2001) only partially overlapped (tables 10.1, 10.4 and 10.14, pages 69, 72, and 87, respectively), while there was much more agreement with annotations made by Blast2GO (Conesa and Gotz 2008) (table 10.14, page 87). This shows how both methods PhyloFun and InterProScan nicely complement each other and help obtaining a better coverage and sensitivity when annotating query proteins. Furthermore both tools prove to have unique domains of GO term annotations they are able to make, based on the observation that the sets of terms annotated by each tool have only small intersections, even when regarding ancestor-descendant-realtionships (figures 10.6 and 10.3, pages 72 and 70 respectively). In spite of this the union of the proteomes annotated by the respective tools is only a fraction of what the sum would be (tables 10.1, 10.4, and 10.14, pages 69, 72, and 87, respectively). Meaning that even though both tools have their domain of GO terms they can annotate, the proteins they manage to assign these terms are mostly the same. In my opinion this points to one of the limitations of sequence based protein predictions: Where no significant sequence similarity can be detected, no knowledge can be transferred from well described and studied references to uncharacterized query proteins. For these cases, though usually coming at the cost of much higher resource requirements and often with the need of up to date training data (section 1.10, page 18), annotation methods based on intrinsic protein characteristics might improve coverage and sensitivity. Further study and comparison with such methods is needed to properly answer these questions.

14. Conclusion

Three protein function annotation tools were developed. All three of them — “AHRD”, ”AHRD on gene clusters”, and ”PhyloFun (v2)” — met the postulated requirements of usability and large scale applicability (section 1.1, page 12). That is, they are easy to install and use, and do not exceed reasonable resource requirements like memory demands or processor power. Thus overcoming difficulties confronted with when using many published methods on large scale (section 1.10, page 18).

AHRD aims to annotate query proteins with human readable descriptions, which are often the first contact a biologist has with proteins of interest (section 2.2, page 21). In order to assess the accuracy of the descriptions AHRD assigns, a new method was successfully developed and applied (section 11.1.1, page 91). This new accuracy measure is based on the popular “F-measure”, which is also used to evaluate the accuracy of electronically made GO term annotations (section 1.10, page 18). In the so enabled comparison with competitive methods AHRD clearly outperformed them, and, because of this, is already used in different institutes around the world.

Furthermore, its specialisation, “AHRD on gene clusters” provided the expert biologists with descriptions that enabled them to quickly scan large sets for clusters of interest (chapter 12, page 94).

The third method PhyloFun (v2) not only outperformed its competitors when annotating query proteins with Gene Ontology terms (section 13.2.5, page 98), but also provides the user with a highly reliable phylogenetic tree, that includes well studied reference proteins (section 7.2.2, page 46). GO term annotations of these reference homologs are only taken into account, and possibly passed to the query, if they are considered “trustworthy”, that is if they have experimental verification or at least are expert annotated (section 7.2, page 43).

For these reasons I recommend the combined application of “AHRD” and “PhyloFun (v2)” for protein function annotation, especially in the context of a high throughput environment or on genomic scale, as has been done successfully on the tomato and *M.truncatula* genome annotation projects. (section 1.1, page 12).

Part V.

Appendix and Bibliography

15. Electronic supplement

For reasons of space only those files are provided that can not be generated by application of the explained procedures on the respective publicly available material. Also note, that all of these supplementary files are compressed with the `bzip2` algorithm (*bzip.org*).

`PF-test.fasta.bz2` section 4.1.1, page 30

`ahrd_on_gene_clusters_tomato.tar.bz2` section 9, page 67

`ahrd_sim_anneal_1.txt.bz2` section 8.5, page 55

`ahrd_sim_anneal_2.txt.bz2` section 8.5, page 55

`ahrd_sim_anneal_3.txt.bz2` section 8.5, page 55

`ahrd_sim_anneal_4.tar.bz2` section 8.5, page 55

`ahrd_sim_anneal_5.txt.bz2` section 8.5, page 55

`ahrd_sim_anneal_6.tar.bz2` section 8.5, page 55

`ahrd_sim_anneal_7.txt.bz2` section 8.5, page 55

`ahrd_sim_anneal_8.txt.bz2` section 8.5, page 55

`b_graminis.fasta.bz2` section 4.1.1, page 30

`swissprot.fasta.bz2` section 4.1.1, page 30

`tomato.fasta.bz2` section 4.1.1, page 30

`trust-UniKB_accessions.txt.bz2` section 4.1.1, page 30

16. Summary

"As the number of sequenced genomes rapidly grows, the overwhelming majority of protein products can only be annotated computationally." (Radivojac, Clark, Oron, et al. 2013) With this goal, three new protein function annotation tools were developed, which produce trustworthy and concise protein annotations, are easy to obtain and install, and are capable of processing large sets of proteins with reasonable computational resource demands. Especially for high throughput analysis e.g. on genome scale, these tools improve over existing tools both in ease of use and accuracy. They are dubbed:

- Automated Assignment of Human Readable Descriptions (AHRD) (*github.com/groupschoof/AHRD*; Hallab, Klee, Srinivas, and Schoof 2014),
- AHRD on gene clusters, and
- Phylogenetic predictions of Gene Ontology (GO) terms with specific calibrations (PhyloFun v2).

"AHRD" assigns human readable descriptions (HRDs) to query proteins and was developed to mimic the decision making process of an expert curator. To this end it processes the descriptions of reference proteins obtained by searching selected databases with BLAST (Altschul, Madden, Schaffer, et al. 1997). Here, the trust a user puts into results found in each of these databases can be weighted separately. In the next step the descriptions of the found homologous proteins are filtered, removing accessions, species information, and finally discarding uninformative candidate descriptions like e.g. "putative protein". Afterwards a dictionary of meaningful words is constructed from those found in the remaining candidates. In this, another filter is applied to ignore words, not conveying information like e.g. the word "protein" itself. In a lexical approach each word is assigned a score based on its frequency in all candidate descriptions, the sequence alignment quality associated with the candidate reference proteins, and finally the already mentioned trust put into the database the reference was obtained from. Subsequently each candidate description is assigned a score, which is computed from the respective scores of the meaningful words contained in that candidate. Also incorporated into this score is the description's frequency among all regarded candidates. In the final step the highest scoring description is assigned to the query protein.

The performance of this lexical algorithm, implemented in "AHRD", was subsequently compared with that of competitive methods, which were Blast2GO and "best Blast", where the latter "best Blast" simply passes the description of the best scoring hit to the query protein. To enable this comparison of performance, and in lack of a robust evaluation procedure, a new method to measure the accuracy of textual human readable protein descriptions was developed and applied with success. In this, the accuracy of each assigned competitive description was inferred with the frequently used "F-measure", the harmonic mean of precision and recall, which we computed regarding meaningful words appearing in both the reference and the assigned descriptions as *true positives*. The results showed that "AHRD" not only outperforms its competitors by far, but also is very robust and thus does not require its users to use carefully selected parameters. In fact, AHRD's robustness was demonstrated through cross validation and use of three different reference sets.

The second annotation tool "AHRD on gene clusters" uses conserved protein domains from the InterPro database (Apweiler, Attwood, Bairoch, et al. 2000) to annotate clusters of homologous proteins.

In a first step the domains found in each cluster are filtered, such that only the most informative are retained. For example are family descriptions discarded, if more detailed sub-family descriptions are also found annotated to members of the cluster. Subsequently, the most frequent candidate description is assigned, favoring those of type "family" over "domain".

Finally the third tool "PhyloFun (v2)" was developed to annotate large sets of query proteins with terms from the Gene Ontology. This work focussed on extending the "Belief propagation" (Pearl 1988) algorithm implemented in the "Sifter" annotation tool (Engelhardt, Jordan, Muratore, and Brenner 2005; Engelhardt, Jordan, Srouji, and Brenner 2011). Jöcker had developed a phylogenetic pipeline generating the input that was fed into the Sifter program. This pipeline executes stringent sequence similarity searches in a database of selected reference proteins, and reconstruct a phylogenetic tree from the found orthologs and inparalogs. This tree is than used by the Sifter program and interpreted as a "Bayesian Network" into which the GO term annotations of the homologous reference proteins are fed as "diagnostic evidence" (Pearl 1988). Subsequently the current strength of belief, the probability of this evidence being also the true state of ancestral tree nodes, is then spread recursively through the tree towards its root, and then vice versa towards the tips. These, of course, include the query protein, which in the final step is annotated with those GO terms that have the strongest belief. Note that during this recursive belief propagation a given GO term's annotation probability depends on both the length of the currently processed branch, as well as the type of evolutionary event that took place. This event can be one of "speciation" or "duplication", such that function mutation becomes more likely on longer branches and particularly after "duplication" events. A particular goal in extending this algorithm was to base the annotation probability of a given GO term not on a preconceived model of function evolution among homologous proteins as implemented in Sifter, but instead to compute these GO term annotation probabilities based on empirical measurements. To achieve this, calibrations were computed for each GO term separately, and reference proteins annotated with a given GO term were investigated such that the probability of function loss could be assessed empirically for decreasing sequence homology among related proteins. A second goal was to overcome errors in the identification of the type of evolutionary events. These errors arose from missing knowledge in terms of true species trees, which, in version 1 of the PhyloFun pipeline, are compared with the actual protein trees in order to tell "duplication" from "speciation" events (Zmasek and Eddy 2001). As reliable reference species trees are sparse or in many cases not available, the part of the algorithm incorporating the type of evolutionary event was discarded. Finally, the third goal postulated for the development of PhyloFun's version 2 was to enable easy installation, usage, and calibration on latest available knowledge. This was motivated by observations made during the application of the first version of PhyloFun, in which maintaining the knowledge-base was almost not feasible. This obstacle was overcome in version 2 of PhyloFun by obtaining required reference data *directly* from publicly available databases.

The accuracy and performance of the new PhyloFun version 2 was assessed and compared with selected competitive methods. These were chosen based on their widespread usage, as well as their applicability on large sets of query proteins without them surpassing reasonable time and computational resource requirements. The measurement of each method's performance was carried out on a "gold standard", obtained from the Uniprot/Swissprot public database (Boeckmann, Bairoch, Apweiler, et al. 2003), of 1000 selected reference proteins, all of which had GO term annotations made by expert curators and mostly based on experimental verifications. Subsequently the performance assessment was executed with a slightly modified version of the "Critical Assessment of Function Annotation experiment (CAFA)" experiment (Radivojac, Clark, Oron, et al. 2013). CAFA compares the performance of different protein function annotation tools on a worldwide scale using a provided set of reference proteins. In this, the predictions the competitors deliver are evaluated using the already introduced "F-measure". Our performance evaluation of PhyloFun's protein annotations interestingly

showed that PhyloFun outperformed all of its competitors. Its use is recommended furthermore by the highly accurate phylogenetic trees the pipeline computes for each query and the found homologous reference proteins.

In conclusion, three new premium tools addressing important matters in the computational prediction of protein function were developed and, in two cases, their performance assessed. Here, both AHRD and PhyloFun (v2) outperformed their competitors. Further arguments for the usage of all three tools are, that they are easy to install and use, as well as being reasonably resource demanding. Because of these results the publications of AHRD and PhyloFun (v2) are in preparation, even while AHRD already is applied by different researchers worldwide.

17. Acknowledgements

For his expertise, guidance, patience, the rich liberty in choosing my favorite work conditions, and especially the great many opportunities to participate in conferences around the globe, I full heartedly thank my supervisor Heiko Schoof.

This project would not have been possible without the great atmosphere, all the advise and fruitful discussions, and the hard work of my colleagues. Particularly Kathrin Klee, Ulrike Goebel, Mythri Bangalore, Jens Warfsmann, Girish Srinivas, Nahal Ahmadinejad, Michael Plümer, Xue Dong, Fabian Hoffmann, Haili Song, Anika and Andreas Jöcker, Ellen Laurenzen, Ute von Ciriacy-Wantrup, Manual Spannagl, Mohamed Zouine, and all members of the Tomato and the Medicago Genome Projects.

For all the personal support I am deeply grateful to my family: Mohammed Hallab, Christine Hallab-Schmitz, Amina Hallab, and Leila Hallab. For their constant motivation and the persistent readiness to take my mind of things, all my friends shall be thanked. Also, for constantly providing the opportunity to blow of steam, I bow to all the members of our Dojo “Tsunami Köln”, especially Jörg Reuss. Furthermore, my special thanks go out to Frank Fischer for a decade of productive work and friendship. Finally, for keeping Cologne — but not always me — safe from harm intended by such villains like Joker Goat, I convey my gratefulness to Bat Sheep.

Commonly known is the story about the poor stressed PhD candidate, typing in the last words just a minute before actually printing the thesis. Thus let it be known, that this one is no exception. If due to the spectre of imminent printing some dear and indeed praiseworthy contributer has been forgotten in these acknowledgements, I humbly ask for his or her forgiveness, stressing that my, at the moment of writing, still purely hypothetical memory lapse was caused by time itself, the ever scapegoated devilish culprit.

Bibliography

Altschul, S F, T L Madden, A A Schaffer, et al. (Sept. 1997). "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs." In: *Nucleic Acids Research* 25.17, pp. 3389–3402.

ant.apache.org. URL: `http://ant.apache.org/` (visited on 06/11/2013).

Apweiler, R., T. K. Attwood, A. Bairoch, et al. (Dec. 2000). "InterPro—an integrated documentation resource for protein families, domains and functional sites". In: *Bioinformatics* 16.12, pp. 1145–1150.

Arabidopsis Genome Initiative (Dec. 2000). "Analysis of the genome sequence of the flowering plant Arabidopsis thaliana". In: *Nature* 408.6814, pp. 796–815.

Arnold, Roland, Thomas Rattei, Patrick Tischler, et al. (Jan. 2005). "SIMAP—The similarity matrix of proteins". In: *Bioinformatics* 21.suppl 2, pp. ii42–ii46.

Ashburner, Michael, Catherine A. Ball, Judith A. Blake, et al. (May 2000). "Gene Ontology: tool for the unification of biology". In: *Nature genetics* 25.1, pp. 25–29.

Asur, Sitaram, Duygu Ucar, and Srinivasan Parthasarathy (July 2007). "An ensemble framework for clustering protein–protein interaction networks". In: *Bioinformatics* 23.13, pp. i29–i40.

Bader, Gary D and Christopher WV Hogue (Jan. 2003). "An automated method for finding molecular complexes in large protein interaction networks". In: *BMC Bioinformatics* 4, p. 2.

Bairoch, A and R Apweiler (Jan. 2000). "The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000". In: *Nucleic acids research* 28.1, pp. 45–48.

Bangalore, Mythri (Feb. 2013). "Integrating protein domain architecture into Automatic assignment of Human Readable Descriptions (AHRD)". Master Thesis. Bonn: Rheinische Friedrich-Wilhelms-Universität, Department of Life Science Informatics.

Battistuzzi, Fabia U, Andreia Feijao, and S Blair Hedges (Nov. 2004). "A genomic timescale of prokaryote evolution: insights into the origin of methanogenesis, phototrophy, and the colonization of land". In: *BMC Evolutionary Biology* 4, p. 44.

Battistuzzi, Fabia U. and S. Blair Hedges (Feb. 2009). "A Major Clade of Prokaryotes with Ancient Adaptations to Life on Land". In: *Molecular Biology and Evolution* 26.2, pp. 335–343.

Boeckmann, Brigitte, Amos Bairoch, Rolf Apweiler, et al. (Jan. 2003). "The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003". In: *Nucleic Acids Research* 31.1, pp. 365–370.

Breiman, Leo (Aug. 1996). "Bagging Predictors". In: *Machine Learning* 24.2, pp. 123–140.

– (Oct. 2001). "Random Forests". In: *Machine Learning* 45.1, pp. 5–32.

Brun, Christine, Francois Chevenet, David Martin, et al. (2004). "Functional classification of proteins for the prediction of cellular function from a protein-protein interaction network". In: *Genome Biology* 5.1, R6.

Brun, Christine, Carl Herrmann, and Alain Guenoche (July 2004). "Clustering proteins from interaction networks for the prediction of cellular functions". In: *BMC Bioinformatics* 5, p. 95.

bzip.org. URL: `http://www.bzip.org/` (visited on 05/27/2014).

Cai, C Z, L Y Han, Z L Ji, and Y Z Chen (Apr. 2004). "Enzyme family classification by support vector machines". In: *Proteins* 55.1, pp. 66–76.

Castresana, J (Apr. 2000). "Selection of conserved blocks from multiple alignments for their use in phylogenetic analysis". In: *Molecular biology and evolution* 17.4, pp. 540–552.

Chen, Xue-Wen and Mei Liu (Dec. 2005). "Prediction of protein-protein interactions using random decision forest framework". In: *Bioinformatics (Oxford, England)* 21.24, pp. 4394–4400.

Claudel-Renard, Clotilde, Claude Chevalet, Thomas Faraut, and Daniel Kahn (Nov. 2003). "Enzyme-specific profiles for genome annotation: PRIAM". In: *Nucleic Acids Research* 31.22, pp. 6633–6639.

compbio.dundee.ac.uk/gotcha/gotcha.php. URL: `http://www.compbio.dundee.ac.uk/gotcha/gotcha.php` (visited on 04/24/2014).

Conesa, Ana and Stefan Gotz (2008). "Blast2GO: A Comprehensive Suite for Functional Analysis in Plant Genomics". In: *International Journal of Plant Genomics* 2008.

Conesa, Ana, Stefan Gotz, Juan Miguel García-Gómez, et al. (Sept. 2005). "Blast2GO: a universal tool for annotation, visualization and analysis in functional genomics research". In: *Bioinformatics* 21.18, pp. 3674–3676.

Consortium, The Tomato Genome (May 2012). "The tomato genome sequence provides insights into fleshy fruit evolution". In: *Nature* 485.7400, pp. 635–641.

Copley, R. R., C. P. Ponting, J. Schultz, and P. Bork (2003). "Sequence analysis of multidomain proteins: Past perspectives and future directions". In: *Protein Modules and Protein-Protein Interactions*. Ed. by J. Janin and S. J. Wodak. Vol. 61. San Diego: Elsevier Academic Press Inc, pp. 75–98.

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

downloads.yeastgenome.org. URL: `http://downloads.yeastgenome.org/sequence/S288C_reference/orf_protein/` (visited on 04/28/2014).

Durbin, Richard (Apr. 1998). *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press.

ebi.ac.uk/interpro/entry/IPR003653. URL: `http://www.ebi.ac.uk/interpro/entry/IPR003653` (visited on 06/11/2013).

ebi.ac.uk/Tools/webservices/services/dbfetch_rest. URL: `http://www.ebi.ac.uk/Tools/webservices/services/dbfetch_rest` (visited on 06/11/2013).

Eddelbuettel, Dirk (2013). *Seamless R and C++ Integration with Rcpp*. New York: Springer.

Eddelbuettel, Dirk and Romain François (2011). "Rcpp: Seamless R and C++ Integration". In: *Journal of Statistical Software* 40.8, pp. 1–18.

Eddy, Sean R (Oct. 2011). "Accelerated Profile HMM Searches". In: *PLoS computational biology* 7.10, e1002195.

Engelhardt, Barbara E, Michael I Jordan, Kathryn E Muratore, and Steven E Brenner (Oct. 2005). "Protein Molecular Function Prediction by Bayesian Phylogenomics". In: *PLoS Comput Biol* 1.5, e45.

Engelhardt, Barbara E., Michael I. Jordan, John R. Srouji, and Steven E. Brenner (Nov. 2011). "Genome-scale phylogenetic function annotation of large and diverse protein families". In: *Genome Research* 21.11, pp. 1969–1980.

European Bioinformatics Institute (EBI) mirror of the Gene Ontology MySQL database. URL: `http://www.geneontology.org/GO.database.shtml` (visited on 06/11/2013).

Felsenstein, Joseph (2004). *Inferring phylogenies*. Sunderland, Mass.: Sinauer Associates.

Fiser, András and Andrej Sali (2003). "Modeller: generation and refinement of homology-based protein structure models". In: *Methods in enzymology* 374, pp. 461–491.

Freund, Yoav and Robert E Schapire (Aug. 1997). "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting". In: *Journal of Computer and System Sciences* 55.1, pp. 119–139.

Gascuel, O (July 1997). "BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data". In: *Molecular biology and evolution* 14.7, pp. 685–695.

Ge, Guangtao and G. William Wong (June 2008). "Classification of premalignant pancreatic cancer mass-spectrometry data using decision tree ensembles". In: *BMC Bioinformatics* 9.1, p. 275.

geneontology.org/GO.evidence. URL: `http://www.geneontology.org/GO.evidence.shtml` (visited on 05/02/2014).

Gilks, Walter R., Benjamin Audit, Daniela De Angelis, et al. (Dec. 2002). "Modeling the percolation of annotation errors in a database of protein sequences". In: *Bioinformatics* 18.12, pp. 1641–1649.

Gille, C, A Goede, R Preissner, et al. (June 2000). "Conservation of substructures in proteins: interfaces of secondary structural elements in proteasomal subunits". In: *Journal of molecular biology* 299.4, pp. 1147–1154.

github.com/groupschoof/AHRD. URL: `https://github.com/groupschoof/AHRD` (visited on 06/11/2013).

github.com/groupschoof/AHRD_on_gene_clusters. URL: `https://github.com/groupschoof/AHRD_on_gene_clusters` (visited on 06/11/2013).

github.com/groupschoof/PhyloFun. URL: `https://github.com/groupschoof/PhyloFun` (visited on 06/11/2013).

GNU sed (stream editor). URL: `http://www.gnu.org/software/sed/` (visited on 06/11/2013).

Guan, Yuanfang, Chad L Myers, David C Hess, et al. (2008). "Predicting gene function in a hierarchical context with an ensemble of classifiers". In: *Genome biology* 9 Suppl 1, S3.

Guindon, Stéphane, Jean-François Dufayard, Vincent Lefort, et al. (Mar. 2010). "New Algorithms and Methods to Estimate Maximum-Likelihood Phylogenies: Assessing the Performance of PhyML 3.0". In: *Systematic Biology* 59.3, pp. 307–321.

Hallab, Asis, Kathrin Klee, Girish Srinivas, and Heiko Schoof (2014). "AHRD — Automatic assignment of Human Readable Descriptions". In: *PLOS Computational Biology*. In preparation. Klee and Hallab are equally contributing authors.

Hardin, Corey, Taras V Pogorelov, and Zaida Luthey-Schulten (Apr. 2002). "Ab initio protein structure prediction". In: *Current opinion in structural biology* 12.2, pp. 176–181.

Hawkins, Troy, Meghana Chitale, Stanislav Luban, and Daisuke Kihara (Feb. 2009). "PFP: Automated prediction of gene ontology functional annotations with confidence scores using protein sequence data". In: *Proteins* 74.3, pp. 566–582.

Hawkins, Troy and Daisuke Kihara (Feb. 2007). "Function prediction of uncharacterized proteins". In: *Journal of bioinformatics and computational biology* 5.1, pp. 1–30.

Højsgaard, Søren (2012). "Graphical Independence Networks with the gRain Package for R". In: *Journal of Statistical Software* 46.10, pp. 1–26.

Huala, Eva, Allan W. Dickerman, Margarita Garcia-Hernandez, et al. (Jan. 2001). "The Arabidopsis Information Resource (TAIR): a comprehensive database and web-based information retrieval, analysis, and visualization system for a model plant". In: *Nucleic Acids Research* 29.1, pp. 102–105.

Højsgaard, S. (2012). "Graphical Independence Networks with the gRain package for R". In: *Journal of Statistical Software* 46, 1–26.

Jaillon, Olivier, Jean-Marc Aury, Benjamin Noel, et al. (Sept. 2007). "The grapevine genome sequence suggests ancestral hexaploidization in major angiosperm phyla". In: *Nature* 449.7161, pp. 463–467.

java.com. URL: `http://www.java.com/en/` (visited on 06/11/2013).

Jensen, L J, R Gupta, H-H Staerfeldt, and S Brunak (Mar. 2003). "Prediction of human protein function according to Gene Ontology categories". In: *Bioinformatics (Oxford, England)* 19.5, pp. 635–642.

Jones, David T (Mar. 2007). "Improving the accuracy of transmembrane protein topology prediction using evolutionary information". In: *Bioinformatics (Oxford, England)* 23.5, pp. 538–544.

junit.org. URL: `http://junit.org/` (visited on 06/11/2013).

Jöcker, Anika (2009). "Automatic and manual functional annotation in a distributed web service environment". PhD thesis. Cologne, Germany: Universität zu Köln. URL: `http://kups.ub.uni-koeln.de/2717/`.

Kanehisa, Minoru and Susumu Goto (Jan. 2000). "KEGG: Kyoto Encyclopedia of Genes and Genomes". In: *Nucleic Acids Research* 28.1, pp. 27–30.

Karp, Peter D, Suzanne Paley, and Pedro Romero (2002). "The Pathway Tools software". In: *Bioinformatics (Oxford, England)* 18 Suppl 1, S225–232.

Katoh, Kazutaka, Kazuharu Misawa, Kei-ichi Kuma, and Takashi Miyata (July 2002). "MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform". In: *Nucleic Acids Research* 30.14, pp. 3059–3066.

Khan, Salim, Gang Situ, Keith Decker, and Carl J Schmidt (Dec. 2003). "GoFigure: automated Gene Ontology annotation". In: *Bioinformatics (Oxford, England)* 19.18, pp. 2484–2485.

kiharalab.org/web/pfp.php. URL: `http://kiharalab.org/web/pfp.php` (visited on 04/24/2014).

Kimura, M (Feb. 1968). "Evolutionary rate at the molecular level". In: *Nature* 217.5129, pp. 624–626.

Kirkpatrick, S, Jr Gelatt C D, and M P Vecchi (May 1983). "Optimization by simulated annealing". In: *Science (New York, N.Y.)* 220.4598, pp. 671–680.

Kosiol, Carolin and Nick Goldman (Feb. 2005). "Different Versions of the Dayhoff Rate Matrix". In: *Molecular Biology and Evolution* 22.2, pp. 193–199.

Koski, L B and G B Golding (June 2001). "The closest BLAST hit is often not the nearest neighbor". In: *Journal of molecular evolution* 52.6, pp. 540–542.

Lee, Bum J., Moon S. Shin, Young J. Oh, et al. (Aug. 2009). "Identification of protein functions using a machine-learning approach based on sequence-derived properties". In: *Proteome Science* 7.1, p. 27.

Lee, Byungwook and Doheon Lee (Dec. 2009). "Protein comparison at the domain architecture level". In: *BMC Bioinformatics* 10.Suppl 15, S5.

Lerat, Emmanuelle, Vincent Daubin, and Nancy A Moran (Oct. 2003). "From gene trees to organismal phylogeny in prokaryotes: the case of the gamma-Proteobacteria". In: *PLoS biology* 1.1, E19.

Li, Li, Jr Stoeckert Christian J, and David S Roos (Sept. 2003). "OrthoMCL: identification of ortholog groups for eukaryotic genomes". In: *Genome research* 13.9, pp. 2178–2189.

Lin, Kui, Lei Zhu, and Da-Yong Zhang (Sept. 2006). "An Initial Strategy for Comparing Proteins at the Domain Architecture Level". In: *Bioinformatics* 22.17, pp. 2081–2086.

Lipman, D J and W R Pearson (Mar. 1985). "Rapid and sensitive protein similarity searches". In: *Science (New York, N.Y.)* 227.4693, pp. 1435–1441.

Martin, David MA, Matthew Berriman, and Geoffrey J. Barton (Nov. 2004). "GOtcha: a new method for prediction of protein function assessed by the annotation of seven genomes". In: *BMC Bioinformatics* 5.1, p. 178.

May, Patrick, Stefanie Wienkoop, Stefan Kempa, et al. (May 2008). "Metabolomics- and Proteomics-Assisted Genome Annotation and Analysis of the Draft Metabolic Network of Chlamydomonas reinhardtii". In: *Genetics* 179.1, pp. 157–166.

McGinnis, Scott and Thomas L. Madden (July 2004). "BLAST: at the core of a powerful and diverse set of sequence analysis tools". In: *Nucleic Acids Research* 32.Web Server issue, W20–W25.

Messih, Mario Abdel, Meghana Chitale, Vladimir B Bajic, et al. (Sept. 2012). "Protein domain recurrence and order can enhance prediction of protein functions". In: *Bioinformatics (Oxford, England)* 28.18, pp. i444–i450.

Needleman, Saul B. and Christian D. Wunsch (Mar. 1970). "A general method applicable to the search for similarities in the amino acid sequence of two proteins". In: *Journal of Molecular Biology* 48.3, pp. 443–453.

Nelson, K E, R A Clayton, S R Gill, et al. (May 1999). "Evidence for lateral gene transfer between Archaea and bacteria from genome sequence of Thermotoga maritima". In: *Nature* 399.6734, pp. 323–329.

Nugent, Timothy and David T. Jones (May 2009). "Transmembrane protein topology prediction using support vector machines". In: *BMC Bioinformatics* 10.1, p. 159.

Pages, H., P. Aboyoun, R. Gentleman, and S. DebRoy (2013). *Biostrings: String objects representing biological sequences, and matching algorithms*. R package version 2.26.2.

Pearl, Judea (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.

Pearson, W R and D J Lipman (Apr. 1988). "Improved tools for biological sequence comparison". In: *Proceedings of the National Academy of Sciences of the United States of America* 85.8, pp. 2444–2448.

Pfam - Sanger Institute. URL: `http://pfam.sanger.ac.uk/` (visited on 06/11/2013).

Pierri, Ciro Leonardo, Giovanni Parisi, and Vito Porcelli (Sept. 2010). "Computational approaches for protein function prediction: a combined strategy from multiple sequence alignment to molecular docking-based virtual screening". In: *Biochimica et biophysica acta* 1804.9, pp. 1695–1712.

Poole, Rebecca L (2007). "The TAIR database". In: *Methods in molecular biology (Clifton, N.J.)* 406, pp. 179–212.

Price, Morgan N., Paramvir S. Dehal, and Adam P. Arkin (July 2009). "FastTree: Computing Large Minimum Evolution Trees with Profiles instead of a Distance Matrix". In: *Molecular Biology and Evolution* 26.7, pp. 1641–1650.

Project, International Rice Genome Sequencing (Aug. 2005). "The map-based sequence of the rice genome". In: *Nature* 436.7052, pp. 793–800.

Quinlan, J. R. (Mar. 1986). "Induction of decision trees". In: *Machine Learning* 1.1, pp. 81–106.

R Core Team (2012). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. URL: `http://www.R-project.org/`.

Radivojac, Predrag, Wyatt T. Clark, Tal Ronnen Oron, et al. (Mar. 2013). "A large-scale evaluation of computational protein function prediction". In: *Nature Methods* 10.3, pp. 221–227.

Rattei, Thomas, Roland Arnold, Patrick Tischler, et al. (Jan. 2006). "SIMAP: the similarity matrix of proteins". In: *Nucleic Acids Research* 34.suppl 1, pp. D252–D256.

Rentzsch, Robert and Christine A Orengo (Apr. 2009). "Protein function prediction–the power of multiplicity". In: *Trends in biotechnology* 27.4, pp. 210–219.

Rijsbergen, C. J. Van (1979). *Information Retrieval*. 2nd. Newton, MA, USA: Butterworth-Heinemann.

Rost, B (Feb. 1999). "Twilight zone of protein sequence alignments". In: *Protein engineering* 12.2, pp. 85–94.

Rougemont, Jacques and Pascal Hingamp (Apr. 2003). "DNA microarray data and contextual analysis of correlation graphs". In: *BMC Bioinformatics* 4, p. 15.

Ruepp, Andreas, Alfred Zollner, Dieter Maier, et al. (2004). "The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes". In: *Nucleic acids research* 32.18, pp. 5539–5545.

Saitou, N and M Nei (July 1987). "The neighbor-joining method: a new method for reconstructing phylogenetic trees". In: *Molecular biology and evolution* 4.4, pp. 406–425.

Samanta, Manoj Pratim and Shoudan Liang (Oct. 2003). "Predicting protein functions from redundancies in large-scale protein interaction networks". In: *Proceedings of the National Academy of Sciences* 100.22, pp. 12579–12583.

Satuluri, Venu, Srinivasan Parthasarathy, and Duygu Ucar (2010). "Markov clustering of protein interaction networks with improved balance and scalability". In: *Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology*. BCB '10. New York, NY, USA: ACM, 247–256. (Visited on 07/15/2013).

Schapire, Robert E. and Yoram Singer (Dec. 1999). "Improved Boosting Algorithms Using Confidence-rated Predictions". In: *Machine Learning* 37.3, pp. 297–336.

Schliep, K.P. (2011). "phangorn: phylogenetic analysis in R". In: *Bioinformatics* 27.4. R package version 1.7-1, pp. 592–593.

Schmutz, Jeremy, Steven B Cannon, Jessica Schlueter, et al. (Jan. 2010). "Genome sequence of the palaeopolyploid soybean". In: *Nature* 463.7278, pp. 178–183.

Schwikowski, B, P Uetz, and S Fields (Dec. 2000). "A network of protein-protein interactions in yeast". In: *Nature biotechnology* 18.12, pp. 1257–1261.

Shannon, C. E. (1948). "A mathematical theory of communication". In: *The Bell System Technical Journal* 27.1, 379–423,623–656.

Shimodaira, H. and M. Hasegawa (Aug. 1999). "Multiple Comparisons of Log-Likelihoods with Applications to Phylogenetic Inference". In: *Molecular Biology and Evolution* 16.8, p. 1114.

Sippl, M J (Aug. 1993a). "Boltzmann's principle, knowledge-based mean fields and protein folding. An approach to the computational determination of protein structures". In: *Journal of computer-aided molecular design* 7.4, pp. 473–501.

– (Dec. 1993b). "Recognition of errors in three-dimensional structures of proteins". In: *Proteins* 17.4, pp. 355–362.

Skolnick, Jeffrey, Yang Zhang, Adrian K Arakaki, et al. (2003). "TOUCHSTONE: a unified approach to protein structure prediction". In: *Proteins* 53 Suppl 6, pp. 469–479.

Smith, T F and M S Waterman (Mar. 1981). "Identification of common molecular subsequences". In: *J. Mol. Biol.* 147.1, pp. 195–197.

Spanu, Pietro D, James C Abbott, Joelle Amselem, et al. (Dec. 2010). "Genome expansion and gene loss in powdery mildew fungi reveal tradeoffs in extreme parasitism". In: *Science (New York, N.Y.)* 330.6010, pp. 1543–1546.

Talavera, Gerard and Jose Castresana (Aug. 2007). "Improvement of phylogenies after removing divergent and ambiguously aligned blocks from protein sequence alignments". In: *Systematic biology* 56.4, pp. 564–577.

udgenome.ags.udel.edu/frm_go.html. URL: `http://udgenome.ags.udel.edu/frm_go.html` (visited on 04/24/2014).

uniprot.org/uniprot/B7YZE7. URL: `http://www.uniprot.org/uniprot/B7YZE7` (visited on 06/11/2013).

uniprot.org/uniprot/P38857. URL: `http://www.uniprot.org/uniprot/P38857` (visited on 10/18/2013).

uniprot.org/uniprot/Q792F9. URL: `http://www.uniprot.org/uniprot/Q792F9` (visited on 10/18/2013).

Van Dongen, Stijn (Jan. 2008). "Graph Clustering Via a Discrete Uncoupling Process". In: *SIAM Journal on Matrix Analysis and Applications* 30.1, pp. 121–141.

Wang, Minglei, Ying-Ying Jiang, Kyung Mo Kim, et al. (Jan. 2011). "A universal molecular clock of protein folds and its power in tracing the early history of aerobic metabolism and planet oxygenation". In: *Molecular biology and evolution* 28.1, pp. 567–582.

Waterman, M.S, T.F Smith, and W.A Beyer (June 1976). "Some biological sequence metrics". In: *Advances in Mathematics* 20.3, pp. 367–387.

Webb, E. C. (1992). "Enzyme nomenclature 1992. Recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the Nomenclature and Classification of Enzymes." In: Ed. 6, xiii + 863 pp.

Weir, J T and D Schluter (May 2008). "Calibrating the avian molecular clock". In: *Molecular ecology* 17.10, pp. 2321–2328.

wikipedia/FASTA_format. URL: `http://en.wikipedia.org/wiki/FASTA_format` (visited on 06/11/2013).

Wilson, C A, J Kreychman, and M Gerstein (Mar. 2000). "Assessing annotation transfer for genomics: quantifying the relations between protein sequence, structure and function through traditional and probabilistic scores". In: *Journal of molecular biology* 297.1, pp. 233–249.

yaml.org. URL: `http://yaml.org/` (visited on 06/11/2013).

Yang, Z (Nov. 2000). "Maximum likelihood estimation on large phylogenies and analysis of adaptive evolution in human influenza virus A". In: *Journal of molecular evolution* 51.5, pp. 423–432.

Young, Nevin D., Frédéric Debellé, Giles E. D. Oldroyd, et al. (Dec. 2011). "The Medicago genome provides insight into the evolution of rhizobial symbioses". In: *Nature* 480.7378, pp. 520–524.

Zdobnov, Evgeni M. and Rolf Apweiler (Sept. 2001). "InterProScan – an integration platform for the signature-recognition methods in InterPro". In: *Bioinformatics* 17.9, pp. 847–848.

Zehetner, Gunther (July 2003). "OntoBlast function: from sequence similarities directly to potential functional annotations by ontology terms". In: *Nucleic Acids Research* 31.13, pp. 3799–3803.

Zmasek, Christian M. and Sean R. Eddy (Sept. 2001). "A simple algorithm to infer gene duplication and speciation events on a gene tree". In: *Bioinformatics* 17.9, pp. 821–828.