Computational Functional Annotation of Crop Genomics using Hierarchical Orthologous Groups

---

Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy

Alexander George Warwick Vesztrocy

Department of Genetics, Evolution and Environment
University College London

December 2019

Declaration

I, Alexander George Warwick Vesztrocy, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Alexander George Warwick Vesztrocy

September 2019

Abstract

Improving agronomically important traits, such as yield, is important in order to meet the ever growing demands of increased crop production. Knowledge of the genes that have an effect on a given trait can be used to enhance genomic selection by prediction of biologically interesting loci. Candidate genes that are strongly linked to a desired trait can then be targeted by transformation or genome editing. This application of prioritisation of genetic material can accelerate crop improvement. However, the application of this is currently limited due to the lack of accurate annotations and methods to integrate experimental data with evolutionary relationships.

Hierarchical orthologous groups (HOGs) provide nested groups of genes that enable the comparison of highly diverged and similar species in a consistent manner. Over 2,250 species are included in the OMA project, resulting in over 600,000 HOGs. This thesis provides the required methodology and a tool to exploit this rich source of information, in the HOGPROP algorithm. The potential of this is then demonstrated in mining crop genome data, from metabolic QTL studies and utilising Gene Ontology (GO) annotations as well as ChEBI terms (Chemical Entities of Biological Interest) in order to prioritise candidate causal genes.

Gauging the performance of the tool is also important. When considering GO annotations, the CAFA series of community experiments has provided the most extensive benchmarking to-date. However, this has not fully taken into account the incomplete knowledge of protein function – the open world assumption (OWA). This will require extra negative annotations, for which one such source has been identified based on expertly curated gene phylogenies. These negative annotations are then utilised in the proposed, OWA-compliant, improved framework for benchmarking. The results show that current benchmarks tend to focus on the general terms, which means that conclusions are not merely uninformative, but misleading.

Impact Statement

Improving agronomically important traits, such as yield, is important in order to meet the ever growing demands of increased crop production. Knowledge of the genes that influence traits can be used to prioritise different genetic material during breeding processes and thereby accelerate crop improvement. The continuously increasing availability of genomics and trait-association data present a unique opportunity for computational identification of gene function.

Biological interpretation of omics data for crop species is hampered by the highly redundant nature of plant genomes, originating from their complex history of duplication and hybridisation events. With almost all genes being available in several copies on multiple sub-genomes, the use of comparative genomics is essential in order to distil biological information from scattered data on single genes. Genes that have common ancestors, similar expression patterns, or are physically close to each other may share biological function.

The vast majority of the current functional annotation of genes in crop species are projected from the model plant *Arabidopsis thaliana* and thus inadequate for all aspects in species that have changed since divergence – for instance, the monocot wheat diverged from *A. thaliana* over 100 million years ago. Traditional methods to propagate functional knowledge across evolutionary related genes are unable to deal with highly redundant genomes, which makes them ill-suited to consider evolutionary close and distant species in a consistent framework. In contrast, hierarchical orthologous groups (HOGs) enable the analysis of related genes across multiple species in a precisely defined, consistent and interpretable manner. This thesis provides a new reliable method to infer and propagate high-quality functional data within and across multiple species, thereby fully exploiting existing omics data to go above and beyond simple comparisons with *A. thaliana*.

Recent efforts to improve and benchmark computational gene function prediction methods have led to the CAFA series of community experiments. These currently provide the

most comprehensive benchmark, with a time-delayed analysis leveraging newly curated, experimentally supported annotations. However, there are fundamental problems which remain unsolved. In particular, the inherent incompleteness of databases and lack of negative annotations ("open world assumption") limits current assessments of function prediction accuracy. This thesis goes some way to solve this, by providing a benchmark which permits for the open world assumption by providing a balanced test set such that methods are only rewarded for predicting terms that can be disproved. Previously, this has not been possible due to the relative paucity of negative experimental annotations. To alleviate this issue, a large number of negative annotations are derived from expert-curated annotations of protein families on phylogenetic trees. An assessment is also presented, using this framework, of the two baseline methods from CAFA (one based on BLAST and the other on annotation frequency), as well as two orthology methods. The results show that the current benchmarks typically focus more on general terms and demonstrates how this is not merely uninformative, but in fact misleading.

Acknowledgements

I would like to thank Christophe Dessimoz, my doctoral supervisor, for his continuous support and guidance throughout my studies. His passion for both the subjects of orthology and benchmarking is infectious. Also, I wish to thank Henning Redestig, my industrial supervisor, for the direction he provided early in my PhD – in particular, his vision for a tool using HOGs to integrate QTL data with GO annotations (chapter 7) was exceptionally inspiring. I would also like to thank Paola Oliveri, my secondary academic supervisor, for her continual encouragement as well as the various suggestions she made throughout my studies.

I wish to thank all past and present members of the Dessimoz research group at University College London, the University of Lausanne and ETH Zürich. In particular, I would like to thank Adrian Altenhoff for his helpful advice throughout, particularly towards the submission of predictions to the third CAFA challenge. I would also like to thank Jeremy Levy for the many interesting discussions and intellectual distractions we shared.

I would also like to thank Pascale Gaudet, Huaiyu Mi and Paul D. Thomas for providing the relevant data from PAINT, as well as for their helpful feedback on the OWA-compliant GO benchmarking. I also thank Monique Zahn for suggestions on this chapter.

Throughout the project, computation was performed on the University College London (UCL) Legion high performance computing facility (Legion@UCL), the UCL Computer Science high performance computer, as well as at the Vital-IT centre for high-performance computing of the SIB Swiss Institute of Bioinformatics. Further, I would like to acknowledge BBSRC grant BB/M015009/1 for the CASE studentship which funded this work, made in partnership with Bayer CropScience NV (now BASF Agricultural Solutions Belgium NV).

I am also forever thankful for the encouragement, love and support that my partner Amy has given me, throughout all of my studies. She has been by my side throughout my doctoral studies at University College London, my master's degree at the University of Bristol and undergraduate degree at the University of Bath. Finally, I wish to thank my parents for both the financial and emotional support that they have provided me with during my studies, as well as for their love, care and advice throughout my life.

Contents

Lists of Figures, Tables & Algorithms

List of Figures

List of Tables

List of Algorithms

Part I

Introduction

Chapter 1

Introduction

Improving agronomically important traits, such as yield, is important in order to meet the ever growing demands of increased crop production. Knowledge of the genes that have an effect on a given trait can be used to enhance genomic selection by prediction of biologically interesting loci (demonstrated in animal breeding [SCK+13], however applicable to plants also). Candidate genes that are strongly linked to a desired trait can then be targeted by transformation or genome editing. This application of prioritisation of genetic material can accelerate crop improvement. However, the application of this is currently limited due to the lack of accurate annotations and methods to integrate experimental data with evolutionary relationships.

Biological interpretation of "omics" data for crop species is restricted by the highly redundant nature of plant genomes, originating from their complex history of duplication and hybridisation events. With almost all genes being available in several copies on multiple sub-genomes, the use of comparative genomics is essential in order to distil biological information from scattered data on single genes. Genes that have common ancestors, similar expression patterns, or are physically close to each other may share biological function.

Furthermore, the vast majority of current functional annotation of genes in crop species are projected from the dicot model plant *Arabidopsis thaliana*. Thus, these are inadequate for all aspects in species that have changed since the divergence (for example, the monocot wheat diverged from *Arabidopsis thaliana* over 100 million years ago). Traditional methods to propagate functional knowledge across evolutionary related genes (for example, bidirectional BLAST hits or OrthoMCL) are unable to deal with highly redundant genomes, which makes them ill-suited to consider evolutionary close and distant species in a consistent framework. In contrast, there has been progress in the ability to compute hierarchical

orthologous groups (HOGs) [AGGD13; TGG+17] – nested groups of genes that evolved from a common ancestral gene within each clade of interest – now enabling the comparison of genes across multiple species in a precisely defined, consistent and interpretable manner.

The increasing availability of both genomic and trait-association data presents a unique opportunity for computational prediction of gene function. This thesis aims to utilise HOGs in order to exploit this abundance of data, to predict function as well as to provide methods in which to integrate many different types of data.

This chapter describes general concepts, methods and resources relevant to the work presented in this thesis, before closing with the aims and an outline of the following chapters.

1.1 Biological Function

There are two main schools of thought on how function should be defined [Tho17]: "causal-role function" and "selected-effect function". The former, first proposed by Cummins [Cum75], focusses on describing function by how some entity contributes to some overall capacity of the system in which it is contained. That is, the function of a part is relative to some system. However, one problem with this is that there is no systematic way in order to identify what the larger system should be, nor the relevant capacity of that system.

The second of the definitions, selected-effect, is derived from the aetiological definition of function given by Wright [Wri73]. This defines the function of an entity as the answer to why such an entity exists, in the first place. Similarly, this can be explained by which of the effects were selected for during evolution [Mil89; Nea91]. This has the advantage that function is derived from the history of its natural selection, which explicitly encorporates evolution into the definition. This has the advantage over the causal-role definition in that it does not require arbitrary decisions to define a containing system and capacities. This issue means that under the causal-role definition there is no general rule for distinguishing functional from accidental effects.

Many biologists, however, have continually defended the causal-role definition. This is particularly the case within the discipline of functional anatomy [AL94], which emphasises how anatomical parts function as parts of larger systems. As the selected trait can often be difficult to infer, many believe that a lack of any hypothesis for such a trait should not hinder the analysis of the mechanism in which an anatomical feature operates. As such, the search for mechanisms of operation has more recently been offered as an alternative paradigm [MDC00]. This, just as in the causal-role definition, focusses on how parts contribute to an overall system. However, it instead looks at how entities (such as proteins) perform activities, or actions that can have casual effects on other activities. That is, a function is simply an activity carried out as part of a larger mechanism. The subtle difference here is that the activity holds the role of a particular function, instead of the entity itself having a function. However, similar to the causal-role definition, no constraints are placed upon the mechanism and so it receives similar criticism with regards to the arbitrary choice of system.

The differences in these two main paradigms stem from the questions they are attempting to address. Under the selected-effects definition, the question is regarding origins [Wri73] – why does the entity exist? Whereas, for the causal-role definition the subject is operation [Cum75] – how does the entity contribute to the biological capacity of the organism to which it belongs? In most biological research today, the focus is to understand the mechanisms by which biological systems operate, rather than explaining why the parts exist altogether.

Molecular biologists [AJL+02; Mon74] have defined function as specific, co-ordinated activities which have the appearance of having been designed for a given purpose. That apparent *"purpose"* is their function. The appearance of design, however, derives from natural selection and so many biologists favour describing these as biological "programmes", in order to avoid connotations of intentional design. These programmes are modular and are present inside a nested hierarchy with other programmes. The lowest-level of such programmes is the expression of a single macro-molecule – for example, a protein. The DNA in the gene is translated into RNA, before being translated into the amino-acid chain of the protein. As this adopts a particular structure then, by simply following physical

laws, it is possible to determine how it will interact with other specific molecular entities. At a higher-level, the functions of multiple proteins are executed in a coherent, regulated manner, to accomplish a larger function. In order to discover selected-effects functions then means to identify a coherent, regulated system of activities. However, causal-role analyses, whilst they play a role in functional anatomy and molecular biology, are only candidates for evolved biological functions until they have been related to survival and reproduction.

1.2 Protein Function

Genes are contiguous regions of DNA which encode instructions for how a cell can produce a particular macro-molecule, or potentially multiple different macro-molecules. These *gene-products* (the macro-molecules) can be of two types: a non-coding RNA or, the most common, a protein.

As the function of a given protein is often context-based and can be studied from different aspects, ranging from its biochemical activity to the role of the protein in particular pathways, cells, tissues and organisms. As such, the functional role of a protein can be described in many different contexts. It can be explained in terms of: (i) the molecular function of the protein, (ii) its role in a biological pathway and (iii) its cellular location. Natural language annotations in databases and literature are too vague and unspecific to accurately describe the function of proteins. This has led to the development of several vocabularies for annotating protein function.

1.2.1 Representation of Functional Knowledge

Various protein annotation schemes have been developed [RO09; RHT00]. For instance, some of the most popular are the Enzyme Commission (EC) numbers [Web+92], Kyoto Encyclopaedia of Genes and Genomes (KEGG) [KFT+17], the Riley scheme [Ril93], MIPS Functional Catalogue (FUNCAT) [RZM+04] and the Gene Ontology (GO) [ABB+00; Gen17; Gen18]. The EC and TC schemes for enzymes and transport proteins, have traditionally been used to annotate molecular function. Involvement in biological processes or cellular pathways is, instead, annotated using the (KEGG), the Riley scheme or FUN-

CAT. Many complementary schemes also exist, for example Reactome [FJM+18] provides manually-curated biological pathways containing molecular details of cellular processes as an ordered network of molecular transformations.

The Gene Ontology

The Gene Ontology (GO), on the other hand, aims to unify the annotation of gene-products (for example, proteins), in a biologically meaningful way across all species [ABB+00; Gen17; Gen18]. A gene-product can be thought of as a molecular machine, performing a chemical action (an *activity*). Further, the gene-products resulting from different genes can combine into a larger molecular machine, called a macro-molecular *complex*. Each concept within the GO relates to the activity of a gene-product or complex, as these are the entities which undertake cellular processes. As a gene encodes a gene-product, it can be considered the source of these activities and processes. However, as it does not perform the activity itself when referring to "gene function" this is strictly speaking "gene-product function".

The GO defines a "universe" of possible functions which a gene may have, however it does not make any claims about the function of any particular genes. These exist as annotations – statements about the function of a particular gene. As biological knowledge is, typically, grossly incomplete, the GO annotation format was designed to capture partial, incomplete statements about gene function. Typical annotations associate a single GO concept (or "term") with a single gene. Together, these statements provide a snapshot of current biological knowledge.

This universe of functions is defined as three distinct aspects, representing key biological domains that are shared by organisms: *biological process*, *cellular component* and *molecular function*. In the molecular biology paradigm (as described in Section 1.1), a gene encodes a gene-product, which carries out a molecular-level process or activity (*molecular function*) in a specific location relative to the cell (*cellular component*). This particular molecular process contributes to a larger biological objective (*biological process*) comprised of multiple molecular-level processes.

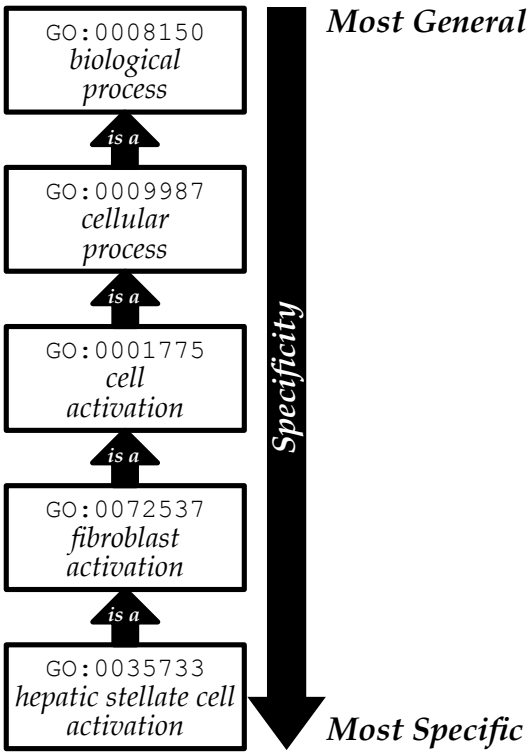


Figure 1.1: Example of the GO term for *hepatic stellate cell activation* and the parent terms (defined by the "is a" relationship) which are implied by an association with it. The closer to the root term, the less specific the GO term is. Whilst, the further away the more specific it is.

Each of these aspects are defined as directed acyclic graphs (DAGs), with each term (node in graph) having defined relationships to one, or more, other terms in the same aspect. The most common relationships are: *"is a"* / *"can be"*, *"part of"* / *"has part"* and *"regulates"* / *"regulated by"*. A simple example can be seen in Figure 1.1, for the GO term for *hepatic stellate cell activation* (`GO:0035733`). If a protein is associated with this term, the parent terms (defined by the "is a" relationship, here) are also implied.

The GO is available as single Open Biomedical Ontologies (OBO) file, containing all three aspects [GŠHD17]. It is also available in the newer Web Ontology Language (OWL) format, which contains extra semantic information – for example, it contains links to other ontologies such as the ChEBI (Chemical Entities of Biological Importance) [HOD+15].

The GO consortium provides annotations in the Gene Annotation Format (GAF), as well as the newer pair of formats: Gene-Product Association Data (GPAD) and Gene-Product Information (GPI) [HHAF+14]. These new formats permit explicit relationship types

Type	Evidence Code	Definition
Experimental	EXP	Inferred from Experiment
	IDA	Inferred from Direct Assay
	IPI	Inferred from Physical Interaction
	IMP	Inferred from Mutant Phenotype
	IGI	Inferred from Genetic Interaction
	IEP	Inferred from Expression Pattern
High Throughput	HTP	Inferred from High Throughput Experiment
	HDA	Inferred from High Throughput Direct Assay
	HMP	Inferred from High Throughput Mutant Phenotype
	HGI	Inferred from High Throughput Genetic Interaction
	HEP	Inferred from High Throughput Expression Pattern
Phylo-genetics	IBA	Inferred from Biological Aspect of Ancestor
	IBD	Inferred from Biological Aspect of Descendant
	IKR	Inferred from Key Residues
	IRD	Inferred from Rapid Divergence
Computational	ISS	Inferred from Sequence or structural Similarity
	ISO	Inferred from Sequence Orthology
	ISA	Inferred from Sequence Alignment
	ISM	Inferred from Sequence Model
	IGC	Inferred from Genomic Context
	RCA	Inferred from Reviewed Computational Analysis
Author	TAS	Traceable Author Statement
	NAS	Non-traceable Author Statement
Curator	IC	Traceable Author Statement
	ND	No Biological Data available
Elec.	IEA	Inferred from Electronic Annotation

Table 1.1: Evidence codes associated with GO annotations, to indicate the support given to a particular association.

between a gene or gene-product and a GO term, as well as having the advantage that entries with no annotations can also be expressed. Separate members of the GO consortium provide annotations for their species of interest. The UniProt-GOA database [BDH+09] is a central location to retrieve GO annotations. It provides experimental, electronic and other curated annotations from many different sources.

Different knowledge regarding gene function may be established using different means, which can be indicated by the evidence upon which the assertion is made. Both formats in which the GO annotations are provided contain meta data, including evidence codes [GŠHD17]. The 26 codes (Table 1.1) indicate the support of the annotation to a particular GO term. In the case of electronic annotations, a reference code is also required – specifying which method produced the prediction. As the terms are nested, if multiple terms are associated arising from a single source of evidence, typically only the most specific terms will be associated with the gene-product as all parent terms are implied. Thus, annotation data typically requires the propagation of GO terms towards the root of the ontology.

Negative annotations also exist – that is, with a `NOT`-qualifier in the qualifier field. Very few of these exist in practice, probably due to their considered unimportance, as well as the increased burden of proof that a biological entity does *not* have a given function, rather than it does.

The different GO concepts are designed to describe aspects (molecular activity, location of the activity and the larger biological programme to which it is part) of the functions that a gene evolved to perform – that is, selected-effect functions. GO concepts may not always be applied in this way and as a side-effect a given GO annotation may or may not be a statement about a selected-effect function.

Whilst all biological programmes (biological process) are necessarily carried out by molecular activities (molecular function), not all molecular activities contribute to a molecular programme. Thus, GO annotations which refer to biological programmes can be considered to generally reflect selected-effect functions. That means – a molecular function annotation cannot be automatically interpreted as a selected-effect function. However, as most GO annotations are made from publications describing specific, small-scale molecular biology studies that focus on a particular biological programme, most GO annotations are likely to refer to selected-effect functions.

Interpretation of GO annotations needs careful consideration, however. For example, the "protein binding" (`GO:0005515`) term is commonly thought to be noise and not necessarily part of any biological programme. Using the notion of information content, it is possible to filter terms such as these out of analyses. Information content (IC) of a GO term is defined as $\mathrm{ic}(t_i) = -\log_2(\mathbb{P}(t_i))$, that is, the logarithm of the probability ($\mathbb{P}(t_i)$) of the term – the logarithm taken base 2, by convention. The probability of the term can be estimated as the frequency of occurrences of the term, or its child terms, in a database of well-characterised proteins (for example, UniProtKB/SwissProt [Uni18]). Thus, more specific terms (see Figure 1.1) will receive a higher IC than more general terms, closer to the root.

The GO is a powerful tool, facilitating many protein analyses across diverse species due to the consistent vocabulary that it contains [RO09]. One advantage is the ability to quantitatively compare the functional similarity between proteins, using semantic similarity

measures. These measures are typically based on IC or the graph, or a combination of the two [Pes17].

1.2.2 Unabated Growth of Protein Sequences

Over many decades there has been a painstaking approach to probe a range of functional aspects of individual proteins. This approach cannot keep up with the next-generation sequencing technologies, which can measure gene expression regulation, genomic organisation and variation on a large scale [SHS13]. According to the GOLD database, hundreds of thousands of genomes have already been sequenced, including close to ten thousand eukaryotes [MSB+19].

This growth can also be seen in the proteins deposited to the UniProtKB (Figure 1.2). Between 2005 and 2015, the number of protein sequences grew exponentially, whilst those with experimental annotations grew linearly. It is the case that, at a molecular level, nearly all biological knowledge is concentrated in a handful of model species and the human. Strikingly, in UniProt-GOA [HSMM+15], over 80% of all GO annotations supported by direct experimental evidence are concentrated in just seven species. Thus, for the overwhelming majority of species, functional characterisation is almost entirely reliant on automated computational methods [CJ17].

The unabated growth of new sequences will continue, if not accelerate further. Within a decade, the Earth BioGenome consortium aims to sequence 1.5 million eukaryotic sequences [LRK+18]. Electronic annotations based on the subset of proteins with experimentally verified annotations, has gone some way to make assertions on a substantial fraction of the new sequences.

Recent developments in high-throughput experiments led to functional annotation of whole genomes, now providing up to 25% of all experimental protein annotations. However, it has been observed that this can lead to significant annotation biases [SRT+13] – they provide particularly low information content compared to low or moderate-throughput experiments and are typically biased towards a limited number of functional annotations. Thus, these experiments can only provide a partial picture of the function of a protein.

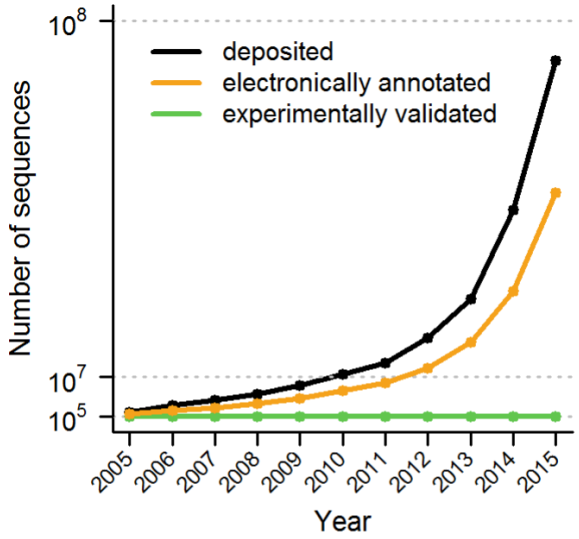


Figure 1.2: Functional annotation coverage of proteins in UniProtKB – between 2005 and 2015 the number of protein sequences deposited in UniProtKB grew exponentially (black), whilst those with experimental annotations only grew linearly (green). However, electronic methods have utilised these limited experimental annotations in order to make assertions about a substantial fraction of the new sequences (orange). Reproduction of [CJ17, Figure 1(a)]

Due to the biases involved, these are now tagged with separate high-throughput-specific evidence codes (Table 1.1).

1.3 Orthology

With newly sequenced genomes, the first step is almost always to identify, within or across-species, *homologous* regions – those which share common ancestry. Here, the focus is on genes as evolutionary and functional entities. If genes are similar within, or between, species it is clear that they must be evolutionarily related and share ancestry – that is, they are are said to be *homologues*. It is important to distinguish between two distinct classes of homologous genes [Fit70]: *orthologous* are pairs of genes that emerging after a speciation event; whereas *paralogues* result from a gene duplication event within a genome.

As an example, consider the gene tree in Figure 1.3. Examples of orthologues, emerging from a speciation event, are (x_1, y_1) or (x_2, z_1). Paralogues are those that result from a duplication event – for example (x_1, x_2) or (x_1, y_1).

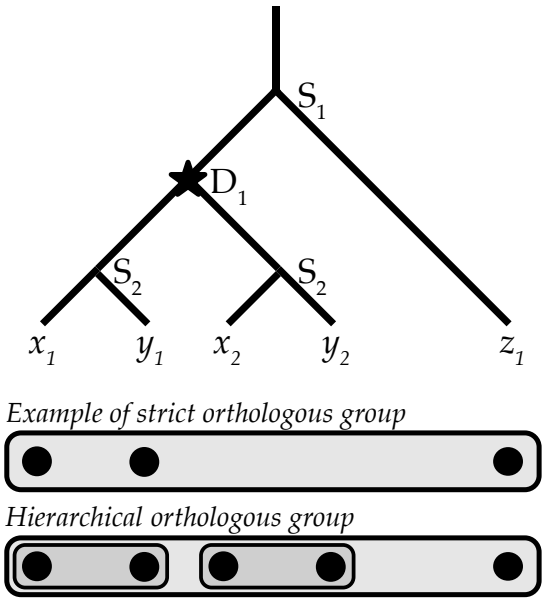


Figure 1.3: Gene tree showing orthologous and paralogous relationships. Also showing example of a strict orthologous group, as well as the corresponding hierarchical orthologous group for this gene tree. Adapted from [AD12, Figure 1] and [FGD19, Figure 1].

Fitch's original definition [Fit70] gave orthology and paralogy as relationships between two genes, depending on the type of initial evolutionary event that gave rise to the pair. The implication of this is that subsequent events, for example the duplication of one and / or the other gene would not alter the relationship. Such duplication can, however, mean that a gene can have more than one orthologous counterpart in another species. That is, orthology is not just a one-to-one relationship, but can also entail one-to-many, many-to-one and many-to-many relationships. The gene-pair (x_1, y_1) is an example of a one-to-one pair of orthologues, whereas (x_2, z_1) is a many-to-one relationship. Paralogy, likewise, can be split into in-paralogues and out-paralogues with reference to a speciation event. The gene-pair (x_1, y_2) are considered in-paralogues, with respect to the speciation event S_1, as the duplication occurred after the speciation event. However, the same pair of proteins are considered out-paralogues with respect to the speciation event S_2 or S_3.

1.3.1 Orthologous Groups

Moving beyond pairs of genes, it is possible to consider how orthology and paralogy apply to more than two species. This is not straightforward, as if gene A is orthologous to gene

B which, in turn, is orthologous to gene C, it is not possible to conclude that A and C are orthologues. That is, orthology – as well as paralogy – are not transitive relationships. This means that there is no straight-forward way to extrapolate pairwise relationships amongst groups of genes or across species. In order to alleviate the difficulty in interpreting the pairwise relationships, it is more common to use *orthologous groups* – these can be split into two main types: "strict" orthologous groups, and "hierarchical" orthologous groups.

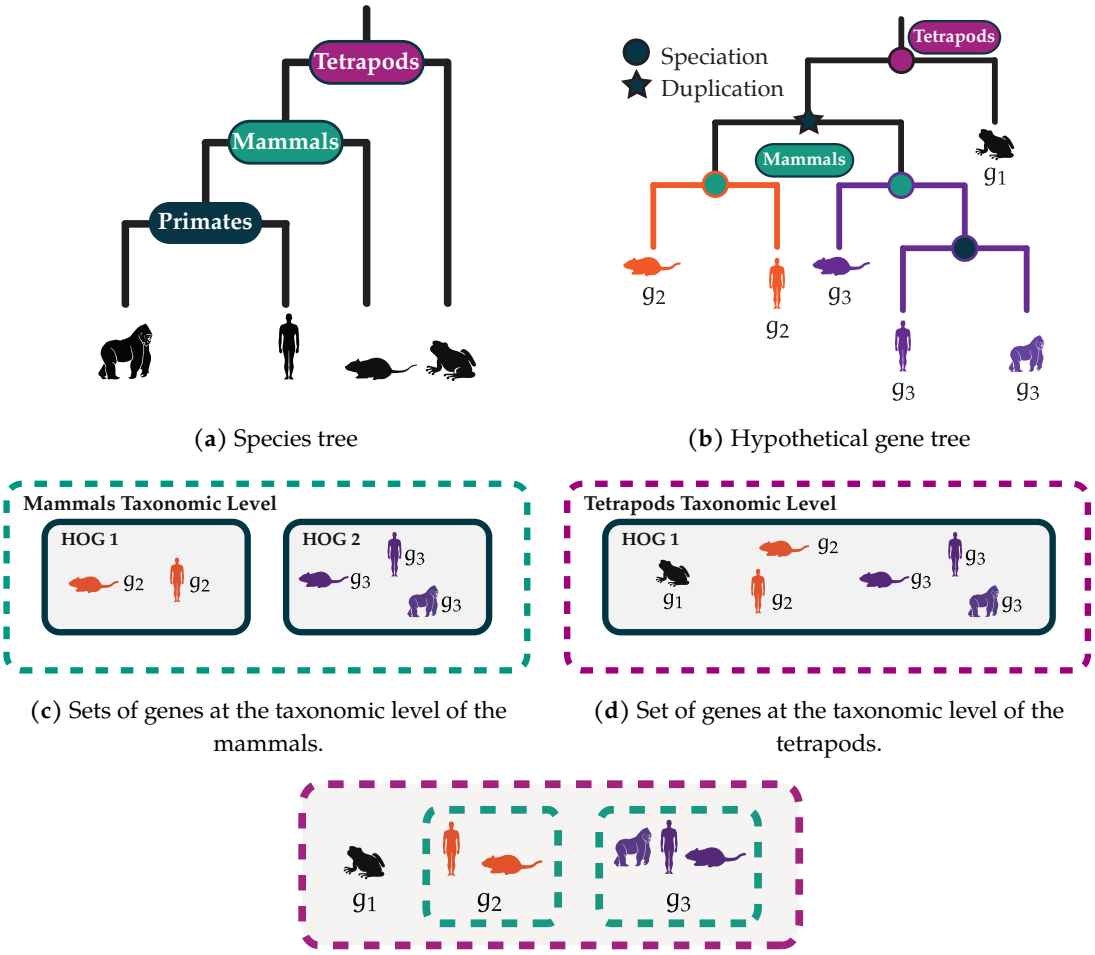
1.3.1.1 "Strict" Orthologous Groups

The first, strict groups, denote sets of genes for which every two members are orthologous. These may be simply sets of one-to-one orthologues. However, so long as paralogues are excluded from the group there is no reason to exclude other types of orthologue. For example, in Figure 1.3, x_1 and y_1 can be grouped with z_1 in a strict orthologous group.

1.3.1.2 Hierarchical Orthologous Groups (HOGs)

Hierarchical orthologous groups (HOGs), the other main kind of grouping, is a group of sets of genes arranged into a hierarchy, dependent on their location in the gene tree. Each one of these sets (sub-HOGs) shares a single common ancestor, but genes can be a member of more than one [AGGD13; TGG+17]. In the case of the OMA algorithm, these sets are then nested into an overall HOG to identify sets of genes that have descended from a common ancestral gene in a given ancestral species. In the example, as all genes x_1, x_2, y_1, y_2 and z_2 descend from the same ancestral gene in the last common ancestor of the three species (x, y and z) they are in a common HOG at that level. However, after the duplication event, $\{x_1, y_1\}$ and $\{x_2, y_2\}$ are HOGs at the lower levels defined at S_2.

In a slightly less abstract manner, consider the species and hypothetical gene tree in Figure 1.4a–b, then one possible question could be: "what gene sets exist at the taxonomic level of mammals?" This group, showing two distinct sets of genes, can be seen in Figure 1.4c. Alternatively, if the interest was in both amphibians and mammals, the HOG defines a single set of related genes (Figure 1.4d). This taxonomic scoping is the advantage of using HOGs over the flat, strict, orthologous groups. In this example, the gene tree shown in Figure 1.4b has the same structure as the HOG. If, for example, there was a series



(**a**) Species tree

(**b**) Hypothetical gene tree

(**c**) Sets of genes at the taxonomic level of the mammals.

(**d**) Set of genes at the taxonomic level of the tetrapods.

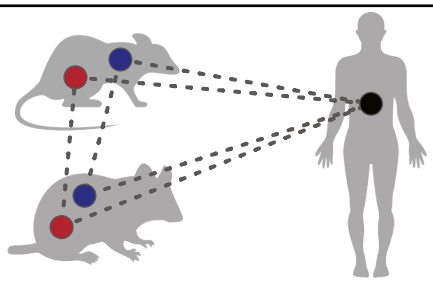
(**e**) Nesting of the gene sets.

Figure 1.4: Example of a hierarchical orthologous group (HOG).

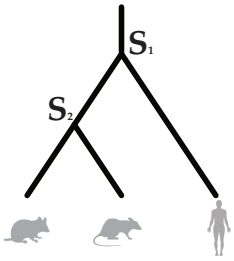
of duplication events in the gene phylogeny, the OMA HOG inference algorithm would not be able to infer the order of the duplication events [AGGD13].

To further illustrate the concept of hierarchical orthologous groups (HOGs), consider the insulin gene within the mammals (Figure 1.5a). In the human there is one copy of the gene, whilst in rodents there are two copies. The gene phylogeny in Figure 1.5c depicts the relationships between these five genes – nodes with a star represent a duplication event, whilst nodes labelled S_i indicate speciation events. Here, S_1 corresponds to the mammalian speciation and S_2 to the speciation of the rodents (Figure 1.5b).

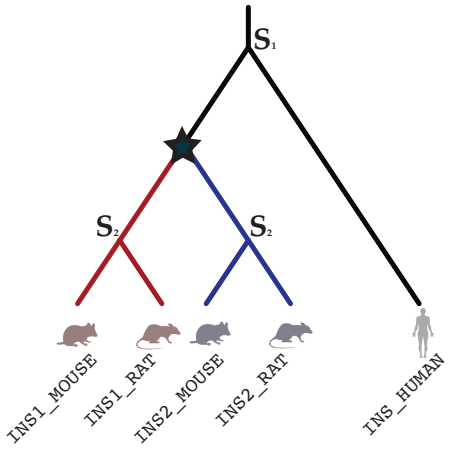
In the ancestor of all mammals, there was only one copy of the insulin gene. Therefore, all insulin genes in mammals are derived from this single gene, so all five genes should



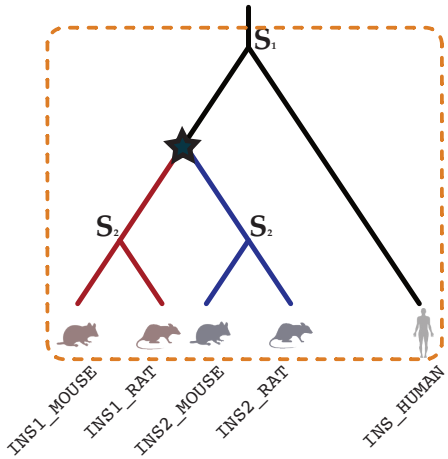
(**a**) The insulin gene in mammals – there is only a single copy in humans, whilst there are two in the rodents.



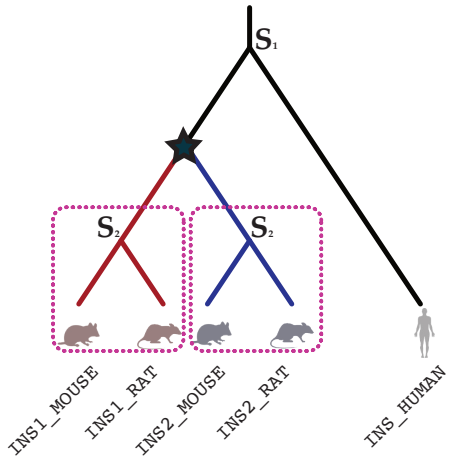
(**b**) Species phylogeny of the mammals. S_1 corresponds to the mammalian speciation event, whilst S_2 is the speciation of the rodents.



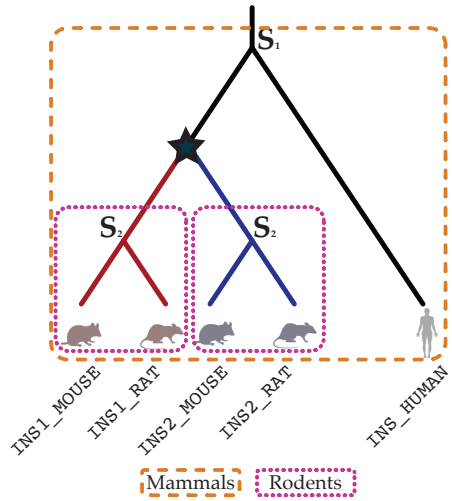
(**c**) Gene phylogeny of the insulin genes in human, mouse and rat. There are two copies in the mouse and rat, which most likely occurred due to a duplication before their most recent common ancestor.



(**d**) HOG defined at the level of the mammals, at which point there was only one copy of the insulin gene.



(**e**) HOGs defined at the level of the rodents, where there are two copies of the insulin gene due to a duplication. This results in two HOGs.



(**f**) Nesting of the groups – the "hierarchy" of the groups.

Figure 1.5: Example of hierarchical orthologous groups (HOGs) for the insulin gene in mammals (adapted from [Glo16]).

be in a single HOG (Figure 1.5d) – including the human insulin gene. HOGs contain both orthologues and paralogues. For example, INS_HUMAN and INS2_RAT are orthologues, as they are related through the speciation event S_1. INS2_MOUSE and INS2_RAT are also orthologues, related via S_2. There are also in-paralogues – those which duplicated since the speciation event at which the HOG is defined. For example, INS1_MOUSE and INS2_RAT are in-paralogues relative to the mammals, so are in the same HOG at this level.

The duplication event results in both mice and rats having two insulin genes, suggesting that their most recent common ancestor already had both copies. As each insulin gene in the extant species can be traced back to one or the other copy, this defines two HOGs at the level of the rodents (Figure 1.5e). At the rodents taxonomic level, INS1_MOUSE and INS2_RAT are out-paralogues – those that diverged at a duplication before the speciation event in question. This means that they are in *different* HOGs at this level.

The "hierarchical" nature of the framework results from the nesting of the groups, visible in Figure 1.5f. When referring to the "insulin gene in mammals", this corresponds to the collection of members in the single insulin HOG defined at the level of all mammals. Whilst this includes the two rodent copies, there is no differentiation made between them. Instead, referring to the "two rodent copies" it is necessary to consider the differences between them and so they are in separate HOGs at the level of the rodents.

1.3.1.3 Reconciled Gene Trees

As an alternative to orthologous groups, all necessary information to capture orthologous relationships is present in reconciled gene trees (sometimes referred to as "labelled" gene trees). These are rooted gene phylogenies in which the internal nodes have been labelled as speciation or duplication events.

All orthology and paralogy relationships amongst extant genes can be deduced from the label associated with their last common ancestor. That is, if their last common ancestor is a speciation node, the genes are orthologues. If it is, instead, a duplication they are paralogues.

Of course, it is also possible to directly map to a hierarchical orthologous group in order to

query sets of genes defined as orthologous at a given taxonomic level. If the reconciled gene trees are fully resolved (that is, they are strictly binary trees), then the order of duplication events is also present as well as some quantification of sequence divergence – the branch lengths.

1.3.2 Inference

Many orthology inference methods exist, however they can be classified into two main types [AD12]: graph-based methods and tree-based methods. The former relies on graphs with genes, or proteins, as the nodes and evolutionary relationships as the edges. Inference is performed to ascertain whether the edge corresponds to an orthology or paralogy, before clusters are built based on the graph. Tree-based methods, on the other hand, are based on gene / species tree reconciliation – the process of annotating all splits in a given gene tree with speciation and duplication events, given a particular phylogeny of the species included in the analysis. From this tree, all pairs of orthologous and paralogous gene can be derived: pairs which coalesce at a speciation node are orthologues, otherwise if they split at a duplication node they are paralogues.

1.3.2.1 Tree-Based Methods

As already noted, tree-based orthology inference reconstruct a gene tree for a group of homologous sequences before labelled the evolutionary events which occurred at each internal node of the phylogeny. Traditionally this involves gene-tree / species-tree reconciliation, undertaken using either a parsimony framework (for example, Forester [ZE01] or Notung [CDFC00]) or a likelihood framework (for example, GSR [ÅSAL09] or Phyldog [BSD+13]). Most reconciliation methods fix the species tree, but not all. Phyldog, for instance, attempts to infer the species tree from the collection of gene trees. Notung [CDFC00], instead, explores the local neighbourhood will consider alternative topologies for the species tree if the defined reconciliation score is improved.

More recently the method of species overlap has been developed. This labels any internal node which has the same species represented in more than one sub-tree as a duplication event [HCDDG07; HSVNH07]. This has the advantage of not making any assumptions

about the underlying taxonomy of the species included. It is also more robust to the topological diversity observed in gene trees, whereas the reconciliation methods tend to introduce more duplication events in order to explain any departure from a given species taxonomy [AD12].

A number of resources provide reconciled gene trees – for instance, PANTHER trees [MPM+16] infers reconciliation for all PANTHER families using the GIGA algorithm [Tho10] – a gene / species reconciliation method. Similarly, EnsemblCompara infers reconciled gene trees which relate all Ensembl genomes using the TreeBeST algorithm [VSUV+09]. OrthoFinder [EK15; EK18] uses DLCpar [WRBK14], in order to label the gene trees it constructs. DLCpar searches for the most parsimonious reconciliation of the gene-tree / species-tree, under a duplication-loss-(deep) coalescent (DLC) model, which addresses incognruence between the gene and species trees. On the other hand, MetaPhOrs [PHCG10] and PhylomeDB [HCCGP+13] both use the species overlap approach. For each reference species, PhylomeDB infers a gene tree starting from each protein (the "seed") and refers to the resulting set of trees as the phylome of that species.

1.3.2.2 Graph-Based Methods

Instead of using trees, graph based approaches are based on comparisons between pairs of genes within, as well as between species. The basis of all graph methods is that, for pairs of genes between two species, orthologues tend to be the least-diverged pair of sequences. This is the case, due to the orthologues being a single gene up until the speciation event, whilst paralogues are the result of earlier duplication events and so exhibit higher sequence divergence.

The bi-directional best-hit (BBH) approach [OFD+99] was the first large-scale method for orthology prediction. Under this scheme, pairs with the mutually highest alignment scores are considered orthologous. A similar approach also exists based on phylogenetic distance – reciprocal shortest distance (RSD) [WFH03]

Both BBH and RSD miss many pairs, due to many-to-many orthology relationships [DD13]. InParanoid aims to alleviate this issue, by identifying many-to-many orthology relation-

ships [SÖ15]. However, both BBH and RSD can also fail when the corresponding orthologue has been lost in both species, leading to paralogues being incorrectly identified as orthologues. The OMA algorithm introduced the use of a third-party species, which may have retained both copies, to act as a "witness of non-orthology" [DCG+05].

Another issue with BBH and RSD is that they do not generalise to the concept of orthologous groups. The COG database used "triangles" of pairwise orthologues, in order to build multi-species orthologous groups [TGNK00]. Whilst OrthoMCL uses Markov clustering [LSR03], which uses an "inflation parameter" rather than defining the type, or some evolutionary limit, of orthology which makes the groups difficult to interpret.

The main graph-based resources include, but are not limited to (in alphabetical order): eggNOG [HCSF+16], HaMStR [ESH09], InParanoid / Hieranoid [SÖ15; KRLS17], OMA [AGT+18], OrthoDB [ZTK+17], OrthoFinder [EK15; EK18], OrthoInspector [LAS+15], OrtholugeDB [WWLB13] and PANTHER [MPM+16].

1.3.2.3 Meta-Methods

A new type of orthology-inference, which utilises the results from other orthology inference methods has emerged. The idea of these meta-methods is that they can combine several individual, distinct, methods in order to produce more robust orthology predictions. This is all made possible due to the standardised formats (particularly OrthoXML [SMSS11]) defined by the "*Quest for Orthologs*" community [GDHJ+09].

Most of these methods assign a confidence score in given predicted orthology relationships. The most basic would be to give more weight to orthology relationships which are predicted by more methods. Examples of this include COMPARE [SGCM07], DIOPT [HFV+11], GET_HOMOLOGUES [CMV13] and HCOP [EWLB06]. When looking at the performance of these methods, they necessarily must have a lower recall, despite having achieved a higher precision, as they intersect the results of many methods.

However, additional post-processing has been used by some methods in order to build upon a base set of orthologues found in several methods. This leads to orthologous relationships being inferred for more sequences and so there is a possibility to improve

performance. One example is MetaPhOrs [PHCG10], which integrates phylogenetic and homology data across databases. In order to assign a confidence score, they use the number of independent sources in which an orthologous relationship exists, as well as the consistency of orthology predictions. Another method, MOSAIC [MH15], uses an iterative graph-based optimisation algorithm. It increases the number of detected orthologues 1.6-fold, identifying those which were missed by the individual methods. Another example is MARIO, which uses the intersection of the various orthology methods as seed groups to add unassigned proteins using HMM profiles [PDL14].

Depending on the required application, different meta-approaches may be appropriate. For instance, if users require high-confidence groups, methods which only combine by intersecting the orthology relationships defined by individual methods are suitable. However, if recall is important and a higher-coverage is required, the methods which perform post-processing on the results, possibly using machine-learning techniques, are better fitted to the application.

1.3.2.4 Orthology Benchmarking

One focus of the *"Quest for Orthologs"* community is to provide benchmarking of the various orthology inference methods [AB+16]. This enables better understanding of the characteristics of each method. A battery of tests is performed on the predictions given by the individual methods. This includes, but is not limited to: functional conservation, gene neighbourhood conservation, species tree discordance and comparison with gold standard gene trees.

The results of this benchmarking show that there is no one-method which solves the problem of orthology perfectly. Different methods excel in different areas and so choosing the best approach is highly dependent on the required application of orthology.

1.3.3 Applications to Automated Function Prediction

As homologous proteins tend to have similar structures and functions, inferring sequence homology of an uncharacterised protein sequence to a sequence of characterised function can provide a simple mechanism for prediction. The assumption here is that the

function has been conserved through evolution. The search for homologues is commonly performed using BLAST [CCA+09] or PSI-BLAST [AMS+97]. However, hidden Markov model (HMM) methods, such as HMMER [Edd11], have increased sensitivity to remote homologues which can increase coverage. New methods have also appeared recently, notably DIAMOND [BXH15] and MMSeqs2 [SS17b], which aim to speed up the sequence search whilst maintaining the performance of BLAST. A database of well-annotated potential homologues, such as UniProtKB/SwissProt [Uni17a], as a starting point is required. In the simplest setting, annotation transfer takes place from the lowest E-value (Expect value, e) below some defined point (for example, $e < e_{\text{accept}} := 10^{-6}$). By relaxing this parameter ($e_{\text{accept}} \to 1$) more distant homologues can be detected, yet the precision shall be negatively impacted.

There are also methods which combine the information for the top results through enrichment of the GO terms assigned to the best hits. GOtcha [MBB04] and PFP [TMSD09] are two examples of such methods. The first assigns weights to each term, taking into account statistical significance of any enrichment, based on the number of similar sequences annotated to that term. The structure of the GO is then used to update the weights on less specific terms. PFP instead uses PSI-BLAST, combining output from results with very liberal E-values. Scores also incorporate co-occurrence statistics of GO terms in the UniProtKB, so as not to over-predict.

Methods based on homology do not distinguish propagation from orthologues to that from paralogues. The pre-conceived notion is that duplicated genes lack selective pressure to maintain their original function. As such, any functional role is free to deviate from the ancestor's [GK13]. However, Nehrt *et al.* [NCRH11] found, using human and mouse functional annotations, that paralogues appear more functionally similar than orthologues. However, after controlling for confounding factors, Altenhoff *et al.* [ASRRD12] found that the then-current experimental annotations did support the so-called "orthologue conjecture", but more weakly than expected.

The groups provided by the many orthology inference methods provide the basis for transfer of GO annotations, under the assumption that members of an orthologous group are

functionally equivalent. Some more sophisticated prediction algorithms based on orthologous relationships have also been developed. For example, SIFTER [EJMB05; EJSB11] uses the gene phylogeny as a belief propagation network [Pea82] in order to propagate molecular function GO terms, using a model of molecular function evolution on the edges (branches). The eggNOG-Mapper [HCFC+17] enables users to functionally annotate proteins by propagating from a restricted type of orthologues. To do this, it uses either DIAMOND [BXH15] or HMMER [Edd11] to identify the closest orthologue in eggNOG [HCSF+16] (the "seed" orthologue) before inferring one-to-one and one-to-many relationships. Another example is the phylogenetic annotation and inference tool (PAINT) [GLLT11]. This is a semi-automated tool which enables curators to annotate the ancestral states in PANTHER [MPM+16] gene families, before uncharacterised extant genes are then annotated using function from common ancestors.

Phylogenetic profiling can also be used to predict function, using observed patterns between multiple families. It postulates that co-evolving families are functionally entwined and may, for instance, participate in the same biological process. Škunca *et al.* [ŠBK+13] introduced a method based on decision trees in a random forest-like setting, with the ability to handle multiple GO terms for each phylogenetic profile, using the orthologous groups from OMA.

1.4 Sequence Alignment

Sequence alignment, as already discussed above, is central to the identification of homologous proteins. Alignment is the procedure of comparing sequences in order to identify regions of similarity by searching for characters arranged in the same order. In the case of proteins, these characters represent the amino acids of which they are composed. Only alignments between two sequences (pairwise) shall be introduced here. It is possible, however, to compute the alignment of many sequences – a multiple sequence alignment (MSA) – usually using some heuristic.

Aligned sequences can be represented in individual rows, where gaps have been inserted such that identical or similar regions of the sequence are aligned in the columns. When

sequences are homologous, gaps in an alignment indicate either insertions or deletions. Mis-matches, on the other hand, are interpreted as point mutations at the nucleotide level in one or more lineages since the sequences diverged. This means that as the sequences become more distantly related, more gaps and mutations will be observed.

Two types of sequence alignment exist – local and global. Local alignments align regions of the sequence which share the highest similarity, resulting in one or more sub-alignments. This can be useful when aligning sequences of varying length, or when they share a conserved region or domain. On the other hand, global alignments are useful to align similar sequences of approximately the same length, as they provide an end-to-end alignment of the sequences.

1.4.1 Substitution Matrices

Substitution matrices are used to compute a score for the matches and mis-matches in the alignments. These contain scores, representing how likely it is for one amino acid to mutate into another over a particular evolutionary time period. Common choices are the BLOSUM substitution matrix 62 (BLOSUM62) [HH92] and the Point Accepted Mutation matrix 250 (PAM250) [DO78].

The choice of substitution matrix can have a large effect on the output alignment, as assumptions are made about the sequences in question. PAM matrices, for instance, were calibrated based on 71 groups of closely related protein sequences, sharing at least 85% similarity. BLOSUM matrices are based on observations in large set of conserved amino acid blocks, derived from more than 500 protein families. This means that whilst the PAM matrices were designed to track the evolutionary origins of proteins, the model underlying the BLOSUM matrices is constructed to identify conserved domains.

There are some other common substitution matrices used for protein alignments, including the JTT [JTT92], WAG [WG01] and LG [LG08] matrices.

1.4.2 Pairwise Alignments

The most basic method to compare two sequences is a so-called "dot matrix" analysis – a visual, time-consuming approach. The two proteins are mapped along the first and second dimensions of a matrix. The simplest visualisation colours individual cells if identical. Thus, matching segments shall appear as diagonal lines across the matrix. This time-consuming approach led to algorithms based on the dynamic programming approach [SW81; NW70], to identify this optimal path, converting the boolean relational matrix into a score matrix, in which the cells are scored according to the similarity of the pairs of amino acids associated to them. The optimal path is then that with the highest score.

These approaches are named the Smith-Waterman algorithm [SW81] and Needleman-Wunsch algorithm [NW70], which generate local and global alignments respectively. In both algorithms, a matrix is populated with scores according to the identities or similarities of the residues associated with each cell by following a scoring scheme for matches, mismatches and gaps. These are then accumulated from the lower-right corner to the upper-left corner. The highest-scoring path from the upper-left to lower-right is then traced, representing the optimal pairwise alignment of the two sequences.

The two methods populate the score matrix differently. Consider two protein sequences, $A := a_1, a_2, ..., a_n$ and $B := b_1, b_2, ..., b_m$, of length n and m, respectively. If the matrix containing the scores is denoted H, with the score H_{ij} as that between a_i and b_j.. The first row and column are initialised as 0, that is $H_{k0} = H_{0l} = 0$ for $0 \leqslant k \leqslant n$ and $0 \leqslant l \leqslant m$. Then, the scoring matrix is filled in from upper-left to lower-right, using a different formula for local and global alignments. For local [SW81], this is defined as

$$H_{ij}^{\text{local}} := \max \begin{cases} H_{i-1,j-1} + s(a_i, b_j), \\ \max_{k \geqslant 1}(H_{i-k,j} - W_k), \\ \max_{l \geqslant 1}(H_{i,j-l} - W_l), \\ 0, \end{cases}$$

and for global alignments [NW70], the minimum value of 0 is removed:

$$H_{ij}^{\text{global}} := \max \begin{cases} H_{i-1,j-1} + s(a_i, b_j), \\ \max_{k \geqslant 1}(H_{i-k,j} - W_k), \\ \max_{l \geqslant 1}(H_{i,j-l} - W_l), \end{cases}$$

where $s(a_i, b_j)$ is the score for aligning the characters a_i and b_j from some pre-defined scoring system. $W_k := kW_1$, where W_1 is the cost of a single gap. Thus, W_k is the penalty for a gap of length k in sequence A and W_l is the penalty of a gap of length l in sequence B. Instead of using linear gap penalties, an affine gap penalty can be used instead which helps to avoid scattered small gaps. That is, $W_x := ux + v$ for $u, v > 0$, where v is the gap opening penalty and u the gap extension penalty.

1.5 Orthologous MAtrix (OMA) Project

As the algorithm [DCG+05; RGD08] and output of the Orthologous MAtrix (OMA) project are central to this thesis, this section briefly introduces the algorithm before describing its current method of function prediction.

The OMA algorithm has been run on a large number of publicly available genomes, with a database accessible via the OMA browser [AŠG+15; AGT+18]. With releases roughly every six months, 2,288 genomes from across the tree of life are now included in the June 2019 release. A standalone version of the algorithm is also available [ALZ+19], enabling the analysis of custom user data.

1.5.1 Algorithm

The algorithm starts with an all-against-all alignment of every protein sequence being aligned against every other sequence – both within, and between species. The Smith-Waterman algorithm [SW81] is used to perform local alignments using the Gonnet PAM substitution matrices [GCB92]. For each alignment the matrix corresponding to the PAM distance that maximises the score is selected, by using a maximum likelihood approach [Gon94], implemented in the Darwin programming environment [GHKB00]. Alignments

scoring below 198 (typically corresponding to E-value 1.3×10^{-16}), or where one sequence is of length less than 60% of the longest, are filtered out and discarded as they are not considered significant. Sequence shorter than 50 amino acids are filtered out at this stage, as well.

These alignments are then filtered in order to identify stable pairs. That is, protein pairs in two different organisms which both have each other as the best match. For the sake of robustness, however, pairs are also retained when the score is not significantly lower than the best match.

Verification of these stable pairs is then undertaken. Usually orthologues have a higher similarity to paralogues, thus the majority of stable pairs will link two orthologous proteins. However, a problem occurs if a corresponding orthologue of a particular protein has been lost during evolution. This then leaves a stable pair to be formed between two paralogous proteins, when these two proteins should belong to two individual orthologous groups. As previously mentioned, the OMA algorithm makes use of a third-party species, which may have retained both copies, in order to act as a "witness of non-orthology" and remove these paralogous relationships from the stable pairs.

The verified pairs form an orthology graph, where proteins are represented by the vertices and stable pairs as the edges. Construction of "strict" orthologous groups ("OMA groups"), for which every pair of members is orthologous, is possible by identifying fully connected subgraphs. This is performed using an iterative algorithm, searching and then removing maximal cliques one-by-one, until no pairs remain.

Hierarchical orthologous groups ("OMA HOGs") are also constructed from the orthology graph of verified pairs. There is a one-to-one correspondence between connected components of a perfect orthology graph and HOGs [AGGD13], for a given taxonomy. However, as the orthology graph is typically noisy, containing many errors, a heuristic approach of a min-cut algorithm was proposed [AGGD13], to break down spurious orthologous relationships before identifying the HOG as the connected component. This is performed for every taxonomic range in a given reference species taxonomy, in a "top-down" manner. More recently, an updated "bottom-up" algorithm was proposed in order to improve

efficiency and to be more resilient to errors in the orthology graph. This starts, instead, at the most specific taxonomic levels and incrementally merges them towards the root [TGG+17].

1.5.2 Function Prediction using OMA Groups

The cliques of orthologues that OMA provides – OMA groups – have previously been shown to be highly coherent in terms of functional annotations [ŠBK+13]. This makes it possible to propagate GO annotations, from UniProt-GOA [BDH+09], as a means of function prediction. GO terms with experimental evidence codes (`EXP`, `IDA`, `IPI`, `IMP`, `IGI`, `IEP`) are propagated across OMA groups and predictions made available in the public release. Essentially, when more than 50% of members have a particular annotation this is propagated to the rest of the group. However, in order to avoid over-propagating clade-specific terms there is a requirement that propagated terms be seen in at least one curated annotation (literature-based) in the clade in question.

These groups have been used for function prediction in the OMA project for some time [AŠG+15]. This functionality is also available as part of the OMA standalone tool [ALZ+19].

1.5.3 Available Output

The OMA browser [AGT+18] enables users to interact with the output of the database. It also provides the output to download, for further analyses. The orthologous groups are available in both flat files containing identifiers, as well as FASTA files containing the sequences in each group.

The hierarchical orthologous groups (HOGs) are available, instead, as an OrthoXML [SMSS11] file. A tool in order to query the OrthoXML has also been developed, PyHAM [TPAD18]. This tool can infer events that occurred on particular ancestral branches of the taxonomy used to construct the HOGs, returning novel, duplicated, lost and continuing genes.

More recently, a REST API has been developed in order to allow users to interact with the

OMA database direct from their programming environment, with native clients available for both R and Python [KWAD19].

1.6 Alternative Methods of Automatic Function Prediction

There are many differing methods of function prediction. So far, the methods that have been covered have been either based on the concept of homology, or more stringently using orthologous and paralogous relationships. If sequence similarity is low, however, these methods will not perform well.

Structural similarity of proteins has also been used in order to predict gene function. Known 3-dimensional structures can be exploited in order to search for structural similarities and predict binding sites and catalytic sites [JKZT14; PCD+15]. Modelling protein structure is a complex topic. Instead, if previously characterised motifs or domains are present, then this information can be used to build gene families. These can be represented by a "signature", which can take several forms. For example, some are defined by particular arrangements of multiple, potentially discontinuous, short linear motifs. Others could be described using more general models of domain sequences – such as sequence profiles [GME87] which can be represented as hidden Markov models (HMMs) [Edd98].

The SUPERFAMILY [OSV+15] and CATH-Gene3D [SLC+15] databases store domain assignments for known protein sequences, based on the SCOP [MBHC95] and CATH [OMJ+97] schemes to classify protein structure, respectively. SUPERFAMILY has a sister project, by the name of dcGO (domain-centric GO) [FG13]. This database includes annotations to individual and multiple domains, with a batch query facility provided as "dcGO Predictor". In the case of CATH, the superfamilies have been split into functional families (FunFams) which include genes with highly similar sequences, structures and functions [DLS+15]. GO terms are then associated with each of these families. Matching against these FunFams then provides predictions. The CATH-FunFHMMer web server provides this as an automated search process for end users [DSL+15].

The SUPERFAMILY and CATH-Gene3D databases are two of the 14 distinct members in the InterPro database [FAB+17]. InterPro combines these databases, integrating predict-

ive information about protein function. Each of the members have their own biological focus, with varying methods to produce signatures as well as matching queries. The InterPro2GO mapping links protein families to the most specific GO terms which apply to each of their members [BKL+12]. The InterProScan tool [JBC+14] enables users to query sequences, which can be combined with InterPro2GO in order to provide GO term predictions.

Methods also exist which build on machine learning principles – the sequences are first transformed into a set of component features, before relating these to GO terms using supervised machine learning. This type of method is able to predict when no characterised homologous proteins can be identified. However, for each GO term (or set of GO terms), sufficient training proteins are required in order to detect patterns in the feature sets. For instance, methods can be built using the Protein Feature engineering toolkit (ProFet) [OL15], in order to extract hundreds of these features representing biophysical as well as sequence attributes. Features derived from alignments can also be used, such as similarity measures, E-values, sequence coverage or even the scores from GOtcha [SBH10; CR11; CBBJ13].

ProtFun [JGB+02; JGSB03] uses neural networks in order to transfer functional annotations between human proteins, based on similarity of their biochemical attributes. Originally the broad functional classes of the Riley scheme [Ril93] were used, more recently being extended to a subset of GO terms. FFPred [LSOJ07; MPCJ13; CMCJ16], based on support vector machines (SVMs), extends this approach by considering the strong correlation between particular molecular functions and biological processes with the lengths and positions of disordered protein regions, whilst still considering many biochemical attributes describing secondary structure, transmembrane helices, intrinsically disordered regions, signal peptides and other motifs.

More recently, a number of methods have been developed to integrate, potentially many, heterogeneous data sources with the aim of providing predictions with higher confidence [CR11; WBS12; CBBJ13] similar to that of the meta-methods of orthology inference.

CombFunc [WBS12] first uses multiple methods to predict GO terms separately before combining using a support vector machine (SVM) to make the final predictions. The individual methods used include sequence homology using BLAST / PSI-BLAST, domain-based predictions using information from InterPro, as well as predictions from protein-protein interactions and gene expression data. FunctionSpace [CBBJ13] proposes a single framework in which information from many sources is combined – sequence, gene expression, protein-protein interaction data and annotations retrieved from text-mining UniProtKB/Swiss-Prot entries. Information from all methods are combined in a probabilistic manner to provide predictions, whilst accounting for the structure of the GO. GOLabeler [YZX+18], which excelled in the third CAFA challenge [ZJB+19], combines data from five different sources using the "learning to rank" (LTR) framework. It uses many different sequence features using the ProFet toolkit, as well as specific GO-term derived features.

1.7 Assessing Automated Function Prediction

As the number of sequenced genomes rapidly grows, the majority of proteins are only annotated computationally. Assessing the quality of these annotations is necessary, if they are to be relied upon. To this end, the critical assessment of protein function annotation (CAFA) experiment [RCO+13] was undertaken, with repeat experiments taking place every few years [JOC+16; ZJB+19].

These primarily consist of three phases (Figure 1.6). The first is a prediction phase, when a set of completely and mostly uncharacterised protein sequences are released as targets, for many research groups to use their own prediction method. Predictions submitted can include the confidence, $\alpha \in (0, 1]$, a method has in its predictions. This is then followed by a target accumulation phase for the test set, which requires enough experimental annotations to accumulate in order to gauge function.

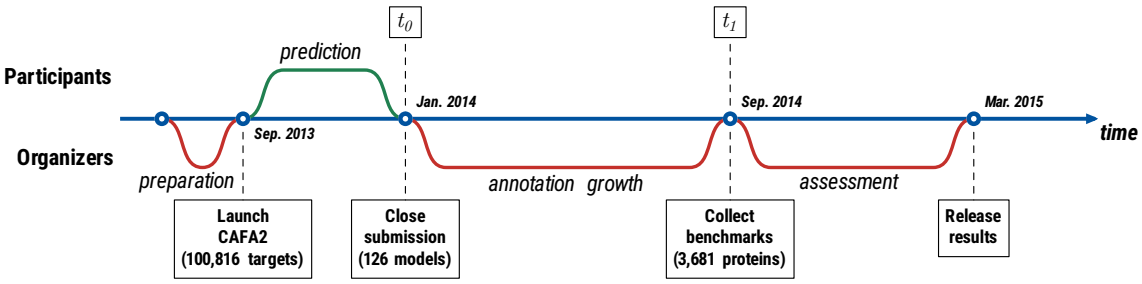


Figure 1.6: Timeline of the second CAFA experiment. Predictions are submitted based on many targets, before experimental annotations are allowed to accumulate on these. Once enough sequences have experimental annotations the benchmarking takes place. Reproduction of [JOC+16, Figure 1]

This is then followed by an analysis stage, where various metrics are calculated in order to gauge the performance of the entrants against two baseline methods. These are: the naïve predictor, which assigns confidence as the frequency of annotation of that term to annotated proteins in a given database; the BLAST predictor, which defines the confidence, for each GO term, as the maximum percentage identity to a sequence that has been annotated with the term. Entries in UniProtKB/Swiss-Prot entries with experimentally verified annotations are used for both of these baseline predictors.

The results of the second experiment showed that the top performing methods are outperforming the best methods from the first, demonstrating that automated function prediction is improving. This was again verified in the third community challenge.

1.7.1 Metrics Used

The metrics used [JOC+16; ZJB+19] are the protein-centric F_{max} and S_{min}, as well as the term-based average area under the precision-recall curve (AUC). Perfect predictors would have an F_{max} score of 1.0 and S_{min} score of 0.

The first two are single-measure evaluations of precision-recall and remaining uncertainty-misinformation curves. Both these curves are computed by varying the cut-off in confidence ($\tau \in (0, 1]$) which methods assign.

1.7.1.1 Precision-Recall

For each target, precision and recall is calculated for all cut-offs. Terms which overlap in the prediction and truth sets are considered correct (true positive). True terms which are not predicted as false negatives, whilst "over-predictions" are considered false positives. This means that, for each target $p \in \mathcal{P}$, precision (pr) is computed as

$$\mathrm{pr}_p(\tau) = \frac{\sum_t \mathbb{1}(t \in P_p(\tau) \wedge t \in T_p)}{|P_p(\tau)|},$$

and recall (rc) as

$$\mathrm{rc}_p(\tau) = \frac{\sum_t \mathbb{1}(t \in P_p(\tau) \wedge t \in T_p)}{|T_p|},$$

where $\mathbb{1}$ is the indicator function, t is an individual GO term and T_p is the truth set for protein p and $P_p(\tau)$ is the set of predicted terms for protein p with confidence $\alpha \geqslant \tau$.

For an overall measure of precision and recall, these are then averaged to obtain

$$\mathrm{pr}(\tau) = \frac{1}{|\mathcal{P}_\tau|} \sum_{p \in \mathcal{P}_\tau} \mathrm{pr}_p(\tau),$$

and

$$\mathrm{rc}(\tau) = \frac{1}{n} \sum_{p \in \mathcal{P}} \mathrm{rc}_p(\tau),$$

where n is the total number of targets and $\mathcal{P}_\tau = \{p \in \mathcal{P} : |P_p(\tau)| > 0\}$ is the number on which at least one prediction has been made with confidence $\alpha \geqslant \tau$.

The F_{max} is the maximum F_1 score on the precision-recall curve. That is,

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

1.7.1.2 Remaining Uncertainty-Misinformation

Minimum semantic distance (S_{min}) is an information-theoretic based measure which was introduced by Clark and Radivojac [CR13]. It aims to overcome some of the complexity in the structure of the GO as well as biased and incomplete experimental annotation sets.

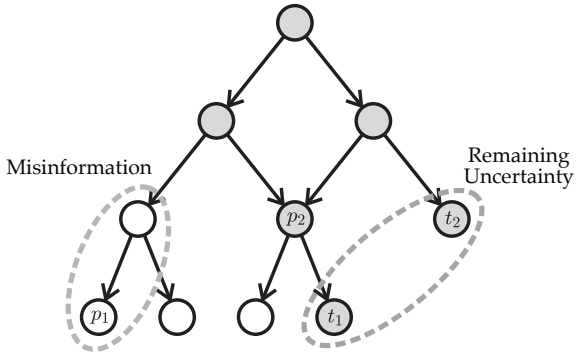


Figure 1.7: Remaining uncertainty and misinformation, given prediction terms p_1, p_2 and truth t_1, t_2. Adapted from [CR13, Figure 2].

It is based on a Bayesian network, structured according to the GO, in order to model the prior probability of GO annotations. The remaining uncertainty in a protein's annotation is defined as the knowledge which is not supplied in the predictions, whilst the misinformation is the total information of the terms considered incorrect in the predictions (see Figure 1.7).

Average remaining uncertainty (ru) is calculated as

$$\mathrm{ru}(\tau) = \frac{1}{n} \sum_{p \in \mathcal{P}} \sum_{t} \mathrm{ia}(t) \mathbb{1}(t \notin P_p(\tau) \wedge t \in T_p),$$

and misinformation (mi) as

$$\mathrm{mi}(\tau) = \frac{1}{n} \sum_{p \in \mathcal{P}} \sum_{t} \mathrm{ia}(t) \mathbb{1}(t \in P_p(\tau) \wedge t \notin T_p),$$

for a given cut-off τ, where $n := |\mathcal{P}|$ is the total number of targets, T_p is the truth set for protein p, $P_p(\tau)$ are the predicted terms for protein p with confidence $\alpha \geqslant \tau$, and $\mathrm{ia}(t)$ is the information accretion of GO term t.

This measure of information accretion is *not* the same as the information content defined earlier. Instead, it is the amount of extra information obtained by adding the term to a parent term, or set of parents terms, in an annotation.

The minimum semantic distance (S_{min}) is the equivalent of F_{max} for remaining uncertainty-misinformation curves. It is defined as the minimum distance from the origin to the curve

and is also used by CAFA to rank the prediction methods. It is defined as

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

1.7.1.3 Prediction Coverage

For each of the measures already described, F_{max} and S_{min}, the coverage of predictions is also reported to gauge how many predictions different methods make.

Benchmarks are reported in a "full" mode as well as a "reduced" mode where allowance is made for a reduced coverage, for instance when calculating recall.

1.7.1.4 Average Area Under Curve (AUC)

The term-based average AUC is based on the area under the receiver operating characteristic (ROC) curve for each term individually.

These are calculated for all terms that have at least 10 positive annotations, before the average is then taken.

1.7.2 Open World Assumption

These benchmarking metrics do not account for the "open world assumption" (OWA) [TWM+12; DvT13] underlying GO annotations. That is, functional characterisation of most proteins is typically grossly incomplete, as both experimental annotations and manual curation of annotations costly and onerous tasks. As such, absence of annotation does not imply absence of function. This leads to a systematic over-estimation of false positive predictions, which may significantly affect the results reported in the CAFA challenges [DvT13].

1.8 Overview

This section starts by posing some open problems, before describing the objectives of this thesis. It then finishes with an outline of the remaining chapters.

1.8.1 Open Problems

Several open problems arise from the subjects discussed in the previous sections. Hierarchical orthologous groups (HOGs) are more accurate depiction of evolution than "strict" orthologous groups, as well as more scalable than reconciled gene trees. They have not yet been used for functional annotation, unlike gene trees (for example, SIFTER [EJMB05; EJSB11]). Therefore, one question is

1. *How can hierarchical orthologous groups be used for function prediction? (for example, Gene Ontology)*

Further, the OMA algorithm requires an extensive, computationally intensive, all-against-all alignment before constructing the HOGs. As there are more significant matches when including closely related species, it can take longer than including distantly related species. So, another question is

2. *How can closely related species, for example cultivars, be projected onto hierarchical orthologous groups?*

Another arises in the benchmarking of predictors of functional annotations. The current CAFA metrics do not account for the open world assumption (OWA) that functional characterisation of most proteins is typically grossly incomplete. Thus, this poses the question of

3. *How can the benchmarking of function predictions account for the open world assumption of incomplete knowledge?*

1.8.2 Objectives

Hierarchical orthologous groups (HOGs) provide a framework for comparing highly diverged and similar species in a consistent manner. The overall objective of this thesis is to provide an algorithm, using the HOGs as a framework, to exploit the multitude of functional and trait-associated data – both for function prediction (for example, Gene Ontology terms), as well as when integrating multiple sources. It also aims to address the other

open problems of efficient mapping of closely related species and addressing the issue of the open world assumption in benchmarking.

The method of function prediction (HOGPROP) sits at the centre of the thesis. Paired with the k-mer based method of efficient mapping makes it possible to predict on sequences both already present in OMA, as well as closely related ones. This was necessary to submit predictions to the CAFA challenges – some species are not in OMA, but a closely related species is.

Overcoming the limitations of the current benchmarks is an overriding priority for the community. Thus, the development of an OWA-compliant benchmark was necessary in order to meet the objective of integrating the HOG-based predictions in the OMA database in order to assess their precision more accurately.

Also, another objective was to further demonstrate the usefulness of the propagation algorithm. One such application is ancestral gene ontology enrichment analyses. Another is to provide a framework for integrating many sources of data. This can then be used, for example, to prioritise candidates in QTL analyses.

An overview of the main thesis objectives is available in Figure 1.8.

1.8.3 Outline

The first subject of the thesis is the methods developed during this project. In Chapter 2, a fast function propagation method based on k-mers and the data pre-existing in the OMA database shall be presented. This was also used in order to predict on targets from the third CAFA experiment, which did not have an exact match (100% identity) in the OMA database. This finds the closest homologous sequence in the database, providing an efficient and accurate mapping for closely related sequences. If the closest homologue is a member of a HOG, this can be used to query the HOG-based function predictions from the OMA database.

Then, in Chapter 3, a propagation algorithm using the OMA HOGs (HOGPROP) is intro-

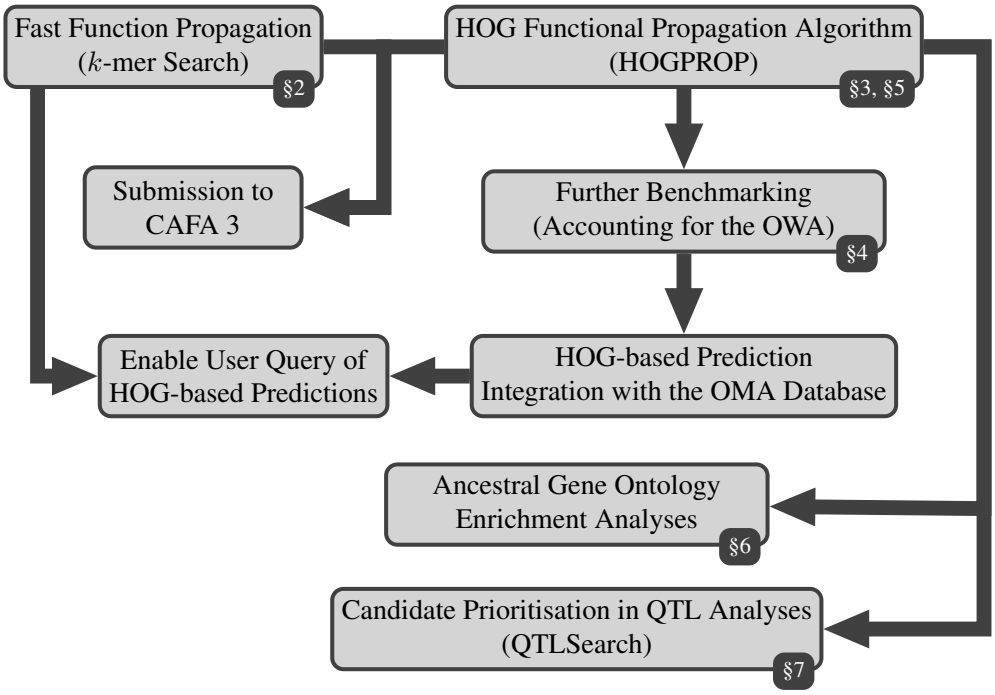


Figure 1.8: Overview of the main thesis objectives and how these link together.

duced. The benchmarking and parameter optimisation that was undertaken prior to the third CAFA experiment will also be discussed.

Benchmarking is further explored in Chapter 4, where a framework utilising explicit negative annotations is introduced which accounts for the open world assumption (OWA). However, as there are very few Gene Ontology annotations which are negatively qualified, it was necessary to supplement these with an alternative source. One such procedure is described, through derivation based on expertly annotated gene phylogenies.

Chapter 5 then describes how evolutionary distances can be fitted to the gene phylogenies, implied by HOGs. This information can then be used in a multitude of studies. Their relevance to the propagation algorithm (HOGPROP) is discussed, alongside how this data could be integrated.

The subject will then change to applications of the propagation algorithm. In Chapter 6, a method to undertake an ancestral gene ontology enrichment analysis shall be introduced. This identifies any enrichment in the functional annotations of genes across a particular branch in a taxonomy.

Then, in Chapter 7 the propagation algorithm is adapted and repurposed to predict candidate causal genes in QTL studies by combining Gene Ontology annotations across many species. This demonstrates how HOGs (and the propagation algorithm) can be used to integrate many sources of data, in a consistent manner.

Chapter 8 then closes by giving an overview of this thesis and conclusions from the presented work.

Part II

Methods

Chapter 2

Fast Function Propagation

The method presented in this chapter has been included in the OMA browser, with the fast function propagation tool featuring in the peer-reviewed publication [AGT+18]. The fast approximate mapping and function propagation tools were also made available via the REST API [KWAD19]. However, the benchmarking analyses, including comparisons with DIAMOND and MMSeqs2, are unpublished.

Homology search is a highly established technique for the propagation of functional annotations. This can be a time-consuming exercise, due to the complexity of compiling reference databases and the long computation time of the homology search. This chapter outlines work to provide a fast function propagation method using currently available data in the OMA database. The method relies on a k-mer table which is built from a suffix-array. This is then used to perform an initial k-mer mapping, before refining the order with a small number of Smith-Waterman alignments. Functional characterisation, in the form of Gene Ontology (GO) terms are then propagated from the closest sequence.

A standalone tool has been developed to perform the k-mer mapping and alignments. This is suitable for both low-memory web-server applications, as well as speed-driven standalone computation when it is possible to take full advantage of the massively multi-core machines available today.

2.1 Requirement for Fast Function Prediction Methods

The current method for predicting function in the OMA database relies on propagation through OMA orthologous groups (see Section 1.5.2). This has been implemented in the OMA standalone tool [ALZ+19]. As the orthologous groups are improved when

including more species, the functional prediction is also. However, this requires the all-against-all alignments of the proteomes, which has quadratic time complexity.

The exponential growth in the number of new genomes sequenced (for example, in the GOLD database [MSB+17]), shows that it is unlikely feasible to ever complete enough all-against-all alignments to be able to predict on many of these genomes. This, alongside the time consuming nature of experimental validation, highlights the requirement for a fast method of automated function prediction.

Most phylogenomic databases, including the Orthologous MAtrix database (OMA), do not provide a method for users to conveniently annotate large sets of protein sequences. The EggNOG database [HCSF+16] recently introduced such a tool for functional assignment. Their approach, outlined in [HCFC+17], maps sequences either through a DIAMOND search [BXH15], or via HMM profiles using HMMER [Edd11]. All gene ontology terms are then transferred from this protein's one-to-one and one-to-many orthologues.

Further, the EggNOG-mapper approach requires the storage of HMM profiles and / or DIAMOND databases. In the approach presented here the sequences already stored in the OMA database file shall be repurposed, requiring only additional indexes to be stored. This was then used to update the sequence search functionality on the OMA website, removing the requirement for legacy code.

2.2 Fast Homology Search

The fast homology search that has been developed is based on k-mer matches. This section will outline the method for the pre-computation of the k-mer lookup table, as well as the search algorithm. This has been implemented inside the (currently internal) PyOMA Python package, for interacting with the OMA database.

1. **Pre-Computed Indexes**

   (a) *Suffix array*

   Suffix arrays (occasionally known as PAT arrays [GBYS92]) are a sorted array of all suffixes of a particular string [MM93]. It is important to delimit the indi-

Suffix	Index (i)
IIDVFSRYSG$	0
IDVFSRYSG$	1
DVFSRYSG$	2
VFSRYSG$	3
FSRYSG$	4
SRYSG$	5
RYSG$	6
YSG$	7
SG$	8
G$	9
$	10

(**a**) All suffixes of the protein sequence.

Suffix	Index (i)
$	10
DVFSRYSG$	2
FSRYSG$	4
G$	9
IDVFSRYSG$	1
IIDVFSRYSG$	0
RYSG$	6
SG$	8
SRYSG$	5
VFSRYSG$	3
YSG$	7

(**b**) Ordered suffixes of the protein sequence.

Table 2.1: Suffixes and ordered suffixes of the protein sequence IIDVFSRYSG (excerpt of S100P_HUMAN) with delimiter denoted as $.

vidual sequences with a character which is unique and lexicographically smaller than any character in the sequence. Here, "$" is used to denote the delimiter. For example, the protein sequence IIDVFSRYSG would be broken down into the suffixes seen in Table 2.1a. These would then be sorted into the order seen in Table 2.1b. The suffix array would then be

$$\begin{bmatrix} 10 & 2 & 4 & 9 & 1 & 0 & 6 & 8 & 5 & 3 & 7 \end{bmatrix}.$$

It is then possible to perform an efficient search for sub-strings (using the binary search algorithm) using the suffix array as an index.

Currently, the OMA database contains over 14.6 million entries in the June 2019 release, however this is ever-expanding. After concatenation of the current sequences, it requires over 4.3×10^9 indices to reference every position. As the maximum unsigned 32 bit integer is $2^{32} \approx 4.295 \times 10^9$, it is required to use a 64 bit integer in order to reference offsets inside the entire sequence buffer.

At the time, there was no Python library which could compute suffix arrays with 64 bit indices. An optimised C implementation of the induced sorting algorithm, introduced in [NZC11], is available under the MIT license [Mor10].

As part of this work, this library was updated and tested with 64 bit indices. It was then packaged inside a Python wrapper to create the PySAIS library

[War17] – available to download from the Python Package Index (PyPI) – licensed under the MIT license.

(b) *Compute Lookup Table for* k*-mers*

The protein sequences stored in OMA have an alphabet containing 21 characters – the 20 amino acids plus `X` to denote unknown, as in the standard IUPAC alphabet. It is possible to think of the k-mers as an encoding of an integer in base 21 using this AA alphabet sorted alphabetically to denote each numeral.

For example, this would mean that the 10-mer `IIDVFSRYSG` is an encoding of the integer 5,829,799,935,146. Similarly, 53 can be encoded to the 10-mer `AAAAAAAADN`.

This encoding can permit the bucketing of the k-mers – i.e., to form a lookup table of k-mer $\mapsto$ entries in the database. The most efficient method of storage is to store the entry numbers in a single array with the offsets at which each k-mer starts in another. In terms of computation, this is efficient as the database is not updated dynamically. Instead, the indexes will be rebuilt with every release.

The sensitivity of a k-mer frequency analysis can be adjusted by altering k. However, storage space is also a factor in this decision. When $k := 6$ it is possible to convert between `AAAAAA` – `XXXXXX` to $0 - 85,766,121$. The offset array required for this can be stored in ~327MB, whereas that for $k := 7$ requires some ~6.7GB and $k := 8$ would require more than 281GB!

To create this lookup table, the suffix array is used to sort an array containing the entry numbers for each position. The offsets where each k-mer starts are located on a scan of this entry array, sorted by k-mer . The two arrays of entry numbers and k-mer start positions are then stored in a persistent manner as a `CArray`[1], made available in `PyTables` [AVP+17], enabling compression of the two arrays.

[1] `CArray` is an implementation of chunked array storage in an HDF5 database file.

Note, it is possible to directly search using the suffix array for k-mers efficiently. However, this is large and most-likely requires memory-mapping. This can be problematic in shared computing, as it is common for schedulers not to be configured in the required manner[2]. Storing the suffix array using `PyTables` and using a compression filter was severely detrimental to the performance. Further, the lookup table is likely to require less than half the storage[3].

2. **Search**

   (a) *Exact-match*

   First, the implementation checks if there is an exact-match to the query sequence in the database. This step also undergoes the same taxonomic filtering described in step 2c. If there are any accepted exact matches, the rest of step 2 shall be skipped. If multiple exact-matches are identified, one database entry is chosen at random.

   (b) *Pre-Filtering with* k*-mer Lookup*

   The k-mer lookup table, as previously defined, can be indexed by each k-mer in the particular range. The query k-mer is decoded into its respective integer (base 21), as already described. A list of entries with this k-mer can then be loaded. This is repeated for every k-mer in the query sequence, with a table of counts to the overlap with each relevant database entry stored.

   It is also possible to filter based on the percentage coverage, of k-mer matches, of the query sequence. This can further increase the speed of computation, by removing low coverage matches.

   (c) *Taxonomic Filtering* (*optional*)

   Given an NCBI taxonomic identifier (which appears inside the OMA taxonomy), the results of the k-mer lookup (or exact matches) can be filtered to only include entries from species that are below this node in the taxonomy.

   In Figure 2.1, the location of the rice genomes in the *Viridiplantae* is shown

[2] `cgroups` can be configured, with certain queue schedulers, to alleviate this issue somewhat.
[3] The number of entries in the OMA database is a long way from requiring 64 bit integers for reference.

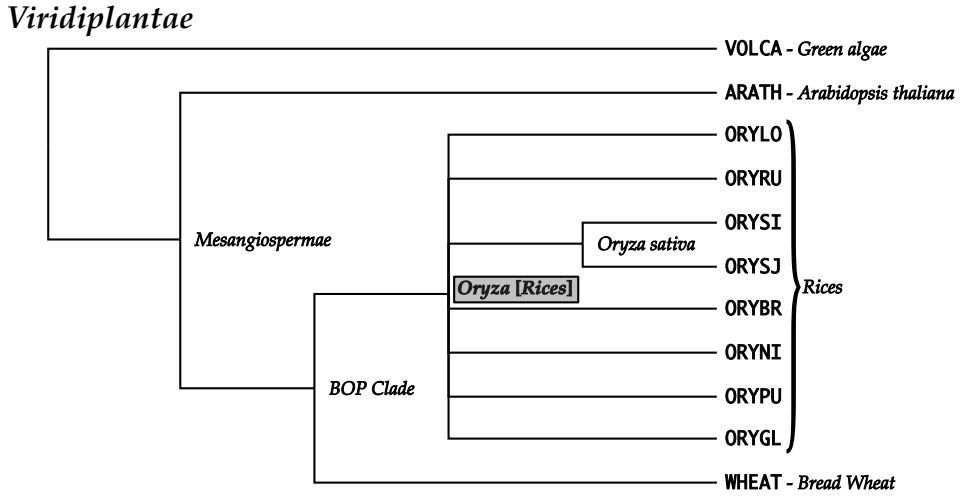


Figure 2.1: Location of rice genomes in the taxonomy of the *Viridiplantae* (many internal levels and genomes missing). If a new rice genome is being annotated, the "Oryza" node would likely be chosen as the taxonomic limit. Leaf labels are UniProt / OMA species identifiers.

(many levels and genomes are missing that are present in the OMA taxonomy). If a user input a new variety of, for example, *Oryza sativa* and did not want to propagate annotations from non-rice genomes they could set the taxonomic filter to the NCBI Taxon ID of the highlighted *Oryza* level (4527). However, this could be relaxed to enable annotations from all other plant genomes by setting it to that of *Viriplanatae* (33090).

(d) *Smith-Waterman Alignment*

The top n results (default $n := 50$), after any filtering, are aligned against the query sequence. The Smith-Waterman alignment is performed using the Striped Smith-Waterman implementation in `scikit-bio`, from the SSW Library [ZLGM13]. A single mutation matrix is used, which is `BLOSUM62` by default with gap opening penalty of 11 and extension penalty of 1 (the default parameters of `blastp`). A cut-off in the score of 50 was chosen, based on [Pea13], in order to only identify homologous sequences.

3. **Propagation**

Any Gene Ontology annotations of the top result of the Smith-Waterman alignment, or of an exact match, will be propagated to the query sequence.

This implementation was integrated into the Python library for interacting with the OMA browser database file. This has then been used to provide the functionality via the REST API to the OMA database[4] [KWAD19]. It is also behind the web service now provided in the OMA browser[5]. This enables a user to upload a FASTA formatted file containing their protein sequences of interest. They will then receive the propagated gene ontology annotations in GAF 2.1 format [GOC15; GŠHD17]. As this takes some time, it is implemented asynchronously – an email will be sent to the user when the results are available.

2.3 Case-Insensitive Identifier Search

This sequence search method, utilising the suffix array, has been re-purposed in order to provide a case-insensitive identifier search in the web browser. Instead of building the suffix array for a buffer of concatenated protein sequences, it is built for a buffer of concatenated identifiers translated to either all upper or lower case.

Then, queries are converted to the same case as the index was built for. Taking all the exact sub-string matches identified, when matching against the suffix array, these are then ordered by the length of match and returned to the user with associated entry information.

2.4 Standalone Tool

A standalone tool was also developed, in order to demonstrate how this was implemented in the OMA browser for a forthcoming publication.

This takes a user-selected set of proteomes (stored in FASTA format) and builds a database containing the sequence and the indexes described above. In this tool, it is presumed that exact-matches do not exist, so first the database is queried by k-mer lookup followed by refinement using the Smith-Waterman alignments.

The standalone tool also has the ability to load the k-mer lookup table into memory, whereas the browser queries this directly from disk. The focus of the standalone tool

[4]Documentation at `https://omabrowser.org/api/docs#sequence-list`
[5]Web service available at `https://omabrowser.org/oma/functions/`

Species	Version	Total No. Sequences	No. with Homologue
Oryza brachyantha (ORYBR)	Ensembl Plants 21	31,307	31,306
Homo sapiens (HUMAN)	Ensembl 86	30,709	30,545
Escherichia coli strain K12 (ECOLI)	Ensembl Fungi 38	4,140	4,134

Table 2.2: Species included in the benchmarking as query proteomes, against the set of 55 plant species – a closely related rice, as well as two outgroups – human and *E. coli*.

is more for speed, whereas in a lightweight server it is necessary to maintain low memory consumption.

2.5 Benchmarking

The central step of the method of fast function propagation proposed in this chapter, is the identification of the closest homologue in a particular set of species. It is only necessary to consider whether the method identifies the same closest homologue as a gold standard does in order to gauge the performance of the tool.

2.5.1 Data

From the December 2018 release of OMA, all but one of the 56 species below the *Viridiplantae* were used to build the database of subject protein sequences. The retained species, *Oryza brachyantha* (ORYBR), was then queried against this database as an example of a closely related species. To test the identification of the closest homologue in plants in more distant species (outside of *Viridiplantae*) both human and *Escherichia coli* strain K12 (ECOLI) were used.

As a gold standard for closest homologue, a full Smith-Waterman alignment was used. The subject protein with the highest score was taken, for each of the query proteins. A cut-off in the score of 50 was chosen, based on [Pea13], in order to identify the homologous sequences. If multiple entries share the highest score, then if the comparison method scores one of these as the top protein then it is considered correct. The number of sequences with a homologue detected in the 55 plant species can be seen in Table 2.2.

2.5.2 Parameter Optimisation

2.5.2.1 Method

Choosing the correct k is a trade-off between speed and accuracy – the same when choosing the number of alignments necessary to refine the k-mer hits. To analyse the choice of these parameters, the database and search was performed for $k \in \{4, 5, 6, 7\}$ and the number of alignments as 50, 250 and 1000.

It is also possible to merge amino acids in the sequence – having an effect of increasing the k-mer search space, without having to lookup the entries for multiple k-mers. Linclust [SS17a] uses a reduced alphabet of 13 characters, instead of the usual 21. This was constructed iteratively, starting with the full alphabet and merging two characters at a time which conserved the maximum mutual information, resulting in the merging of: (A, S, T), (D, N), (E, Q), (F, Y), (I, V), (K, R) and (L, M).

This reduced alphabet was implemented and included in the testing, with $k \in \{6, 7, 8\}$. Due to the reduced alphabet the k-mer size can be extended to 8 without having to use 64 bit integers.

All timing was undertaken on a GNU/Linux machine with four Intel Xeon E5-4620 (2.2GHz) (total 64 hardware threads) and 512GB RAM, making use of all hardware threads available on the machine.

2.5.2.2 Results

Figure 2.2 shows the results for each of the three query species. It is clear that when querying a more closely related species (ORYBR), as there have been less changes at the sequence level, it is possible to recover more of the relationships using only a k-mer based mapping. Further, the number of alignments that are performed to refine the matches does not appear to have much of an effect on the proportion identified, as there is usually a best hit within the first 50 hits of the k-mer mapping. The choice of k does not appear to have much of an effect, nor the use of a reduced alphabet. The 4-mers do appear to reduce

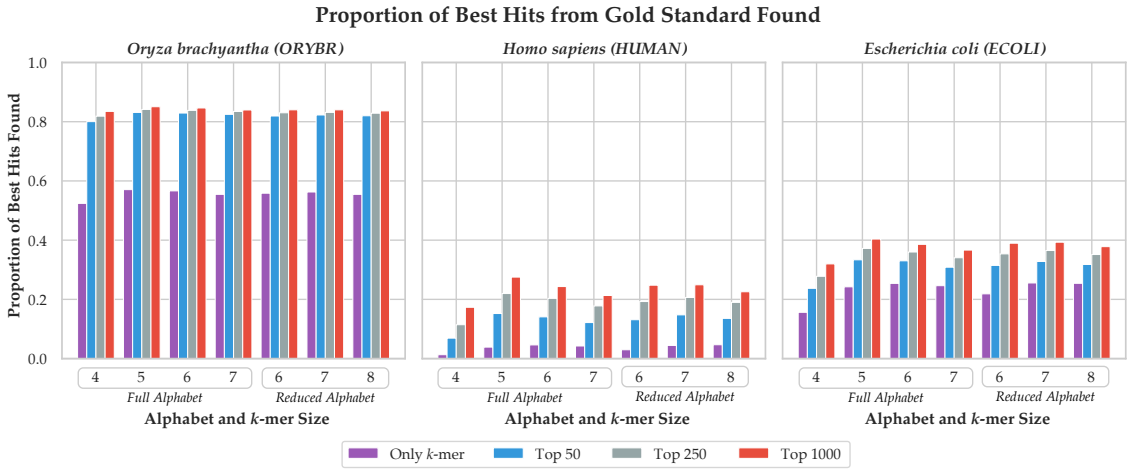


Figure 2.2: Proportion of sequences with at least one best-hit identified, when varying the k-mer size, number of alignments to refine the matches and whether a reduced amino-acid alphabet was used. The choice of k effects the proportion of best-hits identified in the outgroup proteomes, more-so than in the rice species.

the proportion when looking at k-mer only mapping, indicating that there is much more noise when mapping using a short word size.

However, when querying more distant species the proportion of best hits identified is greatly reduced, with the number of alignments in order to refine the ordering having a much greater impact. In Figure 2.3, the amount of user and CPU time as well as the maximum memory usage during the search is shown. This indicates that whilst the extra alignments do increase the proportion of best hits recovered, they do so at extreme cost. The choice of k has more of an effect when querying the human and *E. coli* proteomes, with the 4-mers recovering a lower proportion of the best hits identified again. Further, when using this short word size, the user and CPU time is massively increased – likely due to the increase in more distant proteins that are being aligned.

Comparing the full and reduced alphabets, the results are similar for k set to 5 vs 6, respectively. The same is true for 6 vs 7, where the amount of storage saved by using the reduced alphabet is approximately 88MB (down from ~327 MB). This means that the overall change in maximum memory usage and time to search is minimal.

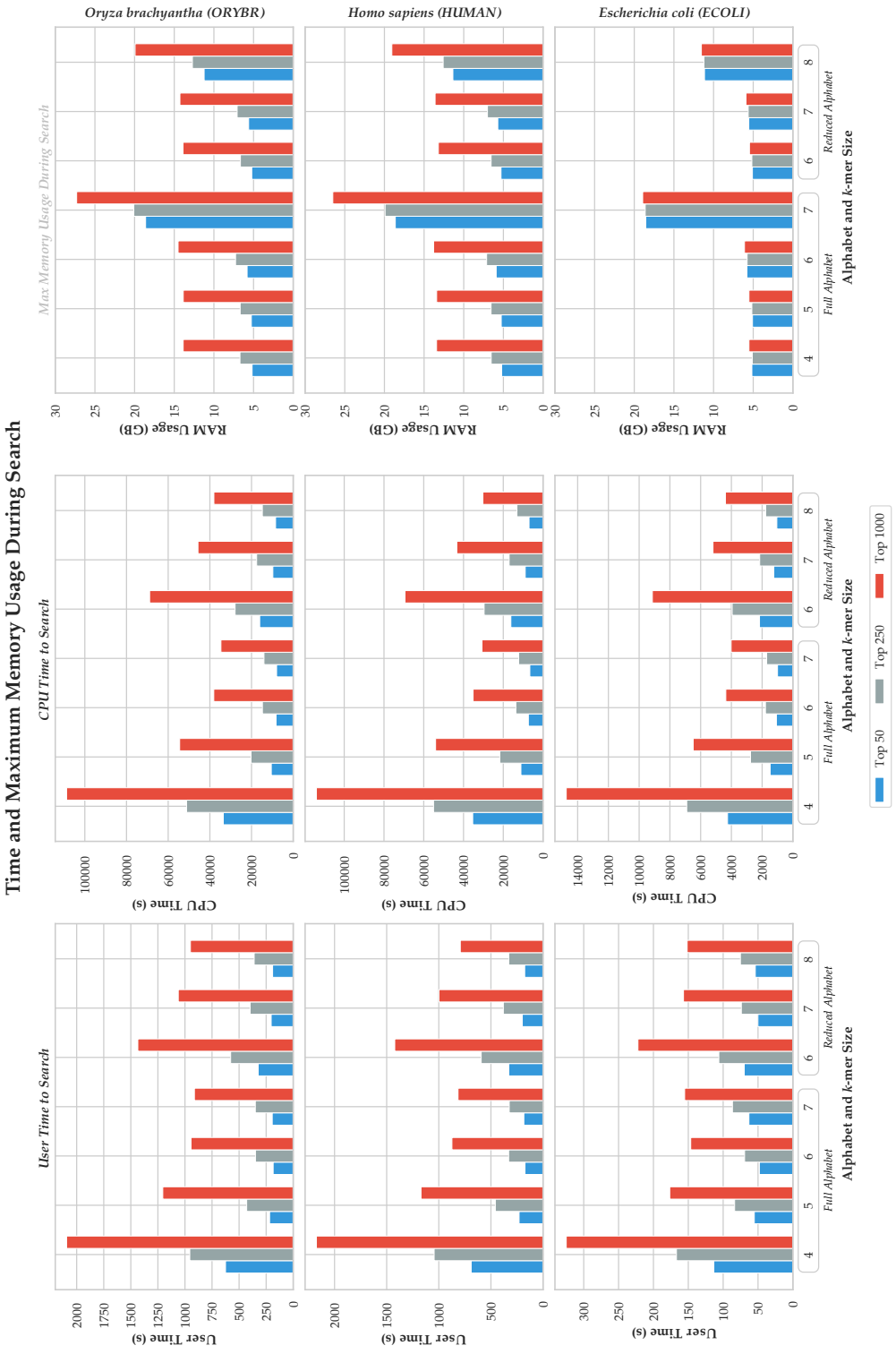


Figure 2.3: Time taken and maximum memory usage, when searching using each of the query proteomes. The k-mer size, number of alignments to refine the matches and alphabet used, were varied. This indicates that whilst the extra alignments do increase the proportion of best hits recovered, they do so at extreme cost.

Alphabet	k	Proportion of Best Hits Identified	Rank
Full	4	81.9%	7
Full	5	84.1%	1
Full	6	83.8%	2
Full	7	83.4%	3
Reduced	6	83.0%	5
Reduced	7	83.1%	4
Reduced	8	82.9%	6

Table 2.3: Proportion of best hits identified in ORYBR, when considering the alignment scores of the top 50 hits from the k-mer mapping.

Parameter Choice

As the main use case is for mapping against close species, the results from ORYBR are most relevant. It is clear that performing alignments to refine the ordering is beneficial, however the number of hits to align appears to have diminishing returns. Table 2.3 shows the actual proportion of best hits identified when aligning the top 50 hits from the k-mer mapping in ORYBR. The top-ranked parameter choice is using $k = 5$ with the full alphabet, however when using $k = 6$ the proportion identified is just 0.3% lower. As such, the decreased computation time (218.3s [User] / 10,518.32s [CPU] for $k = 5$; 185.26s [User] / 8,227.17s [CPU] for $k = 6$) with only ∼0.5GB extra RAM required at maximum memory usage means that choosing $k = 6$ is a good balance between speed and accuracy.

2.5.3 Comparison Methods

Method

Three comparison methods were chosen: BLAST [CCA+09], as well as DIAMOND [BXH15] and MMSeqs2 [SS17b] which also utilise k-mer pre-filtering and reduced amino-acid alphabets. As a gold standard, a full Smith-Waterman alignment of all pairs was utilised. As the chosen "gold standard" does not correct for any compositional bias, any corrective techniques were disabled in the comparison methods (see Table 2.4 for arguments used).

The substitution matrix used and gap opening / extension penalties were normalised across the methods, to the default for BLAST, DIAMOND and MMSeqs2. That is, substitution matrix: BLOSUM62; gap open penalty: 11; gap extension penalty: 1.

The raw alignment score was used in all methods, so as to order predictions. For each query the top subject protein, or proteins in the case of ties, is tested against the gold standard. If there is at least one protein in both sets, then it is considered correct.

Results

In Figure 2.4 are the results from each of the comparison methods, as a proportion of the best hits identified in the gold-standard full Smith-Waterman alignment. BLAST performs the best, however still misses off almost 20% of best hits in the case of HUMAN. More striking is the number of hits that DIAMOND misses, even in its more sensitive modes. This is still the case in ORYBR, where it identified between 65.4%–66.5% of the top hits dependent on the sensitivity setting.

On the other hand, in ORYBR, the novel method (82.9% identified) and MMseqs2 (83.0% / 83.3% identified) in both sensitivity modes perform similarly. This shows that when mapping onto a closely related species, the method presented here is performing as well as the state-of-the-art of fast protein search tools.

When considering the timing of the different methods (Figure 2.5), it was clear that BLAST is not an option for performing fast sequence placement. Therefore, BLAST is not included in the timing plots, as it took: 11,060.03s (user) / 707,842.15s (CPU) for ORYBR; 13,190.17s (user) / 844,171.03s (CPU) for HUMAN; and 916.77s (user) / 58,673.23s (CPU) for ECOLI. The multi-threading feature of `blastp` is not as effective as splitting the query sequences and searching using multiple instances (using GNU parallel [Tan11]). As such, user-time is estimated as 1/64th of the CPU time. For instance, the method presented in this chapter took 185.26s on ORYBR, which is 59.7x faster.

It is also important to consider the time to create the database for each of the methods (Figure 2.6). The method presented here takes the longest, due to the sequential suffix-array computation. However, when considering CPU time instead of user-time, DIAMOND takes a similar amount of time to this method, with MMseqs2 taking almost 2.5 times as many CPU seconds. Turning to the maximum memory consumption, the method

Method	Version	Command	Arguments
BLAST [CCA+09]	2.6.0+	`blastp`	`-seg no -comp_based_stats false`
DIAMOND [BXH15]	0.9.24	`diamond blastp`	`--comp-based-stats 0`
MMSeqs2 [SS17b]	9-d36de	`mmseqs search`	`--comp-bias-corr 0 --mask 0`

Table 2.4: Comparison methods used to gauge the performance of the fast assignment method. Command / arguments listed are the command at which the compositional bias was disabled and the arguments used to do so.

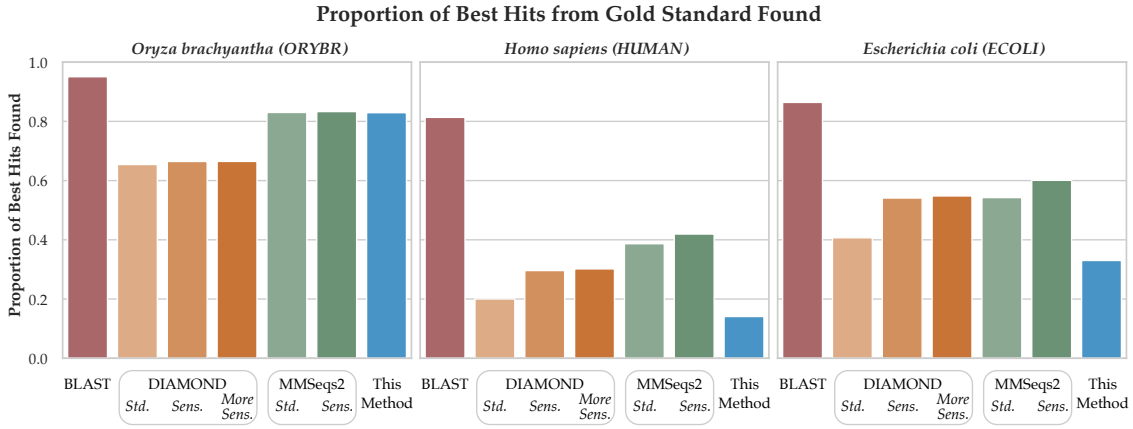


Figure 2.4: Proportion of sequences with at least one best-hit identified, between the different methods chosen for comparison. The method presented here is included with $k = 6$ using the full alphabet and refining the top 50 k-mer hits with Smith-Waterman alignments. When attempting to map the out-group proteomes, the performance of all methods is lower – apart from BLAST which still recovers >80% of the best hits from the gold standard.

presented here takes approximately the same amount of memory during database build as MMseqs2 does during search.

2.6 Discussion and Conclusions

The method and tool presented in this chapter is for mapping protein sequences to the closest homologous sequence, in a given database, before propagating functional characterisation in the form of associations to GO terms. This is particularly useful when considering closely related species, or different assemblies / releases, for which some down stream-analyses have already been performed within a database. In this case, the standalone tool has been shown to be similarly accurate as well as approximately the same speed as MMseqs2 in standard mode (`ORYBR` against 55 plant species).

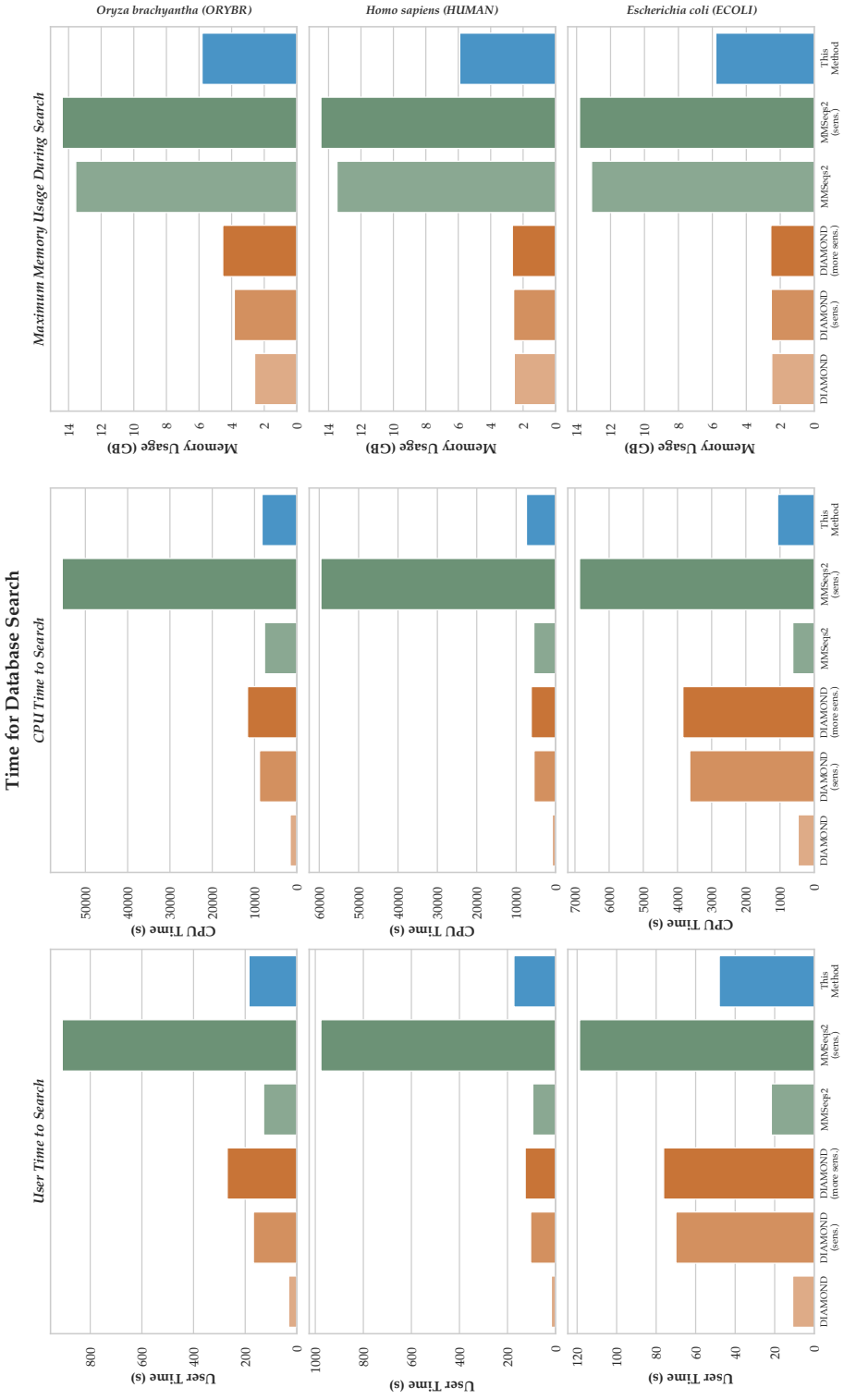


Figure 2.5: Time taken and maximum memory usage, when searching using each of the query proteomes, for each of the comparison methods, with varying parameters. Using DIAMOND and MMSeqs2 in their more sensitive modes take a lot longer than their standard parameters. In the case of a close-species (ORYBR) the method presented here is approximately the same speed as MMseqs2 in standard mode.

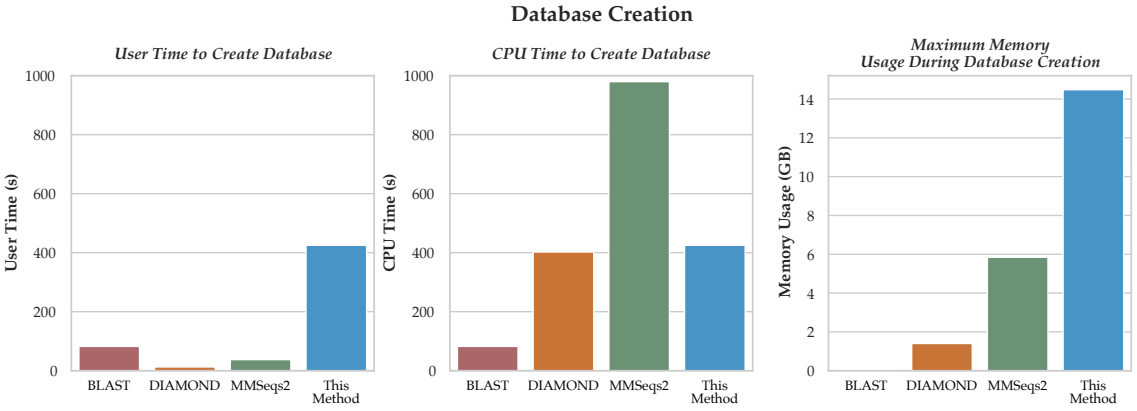


Figure 2.6: Time taken and maximum memory usage, when building the database for the 55 plant species for each of the comparison methods. MMSeqs2 takes >2x the CPU time to create the database, however is much faster when considering user time. This is due to the sequential algorithm to build the suffix-array included in the method presented here.

However, it is important to consider why this method performs worse than others when attempting to map more distant species. As the evolutionary distance increases between the species, the more mutations will have occurred. Mutations such as insertions, deletions and non-synonymous substitutions will remove the exact-matches which the k-mer based mapping is based on. The reduced alphabet from Linclust that was investigated when choosing the parameters did not show much improvement in the proportion of best hits identified. However, the use of a reduced alphabet should have the same effect as looking up multiple k-mers, widening the search space. Future studies could look into alternative reduced alphabets, as this only shows that this choice of alphabet is not beneficial. It may be that different reduced alphabets are more effective when varying the evolutionary distance.

Further, in the gold standard set of homologous best-hits used, the only requirement to be included was that the alignment score was greater than, or equal, to 50. As such, the best-hit sequences are not necessarily of similar length to the query sequence.

When considering functional propagation, there is also a trade-off between speed and accuracy – the method implemented in this chapter is much faster than the current method for propagation through OMA groups, due to the lack of the requirement for all-against-all alignments. However, it is error prone like many homology-based predictors [CJ17]. In

particular, this method makes no distinction between orthologues and paralogues. This is relevant as duplicated genes lack the selective pressure to maintain their original biological role. As such, this leads to more mutations at the nucleotide-level which enables deviation from their original function [GK13].

This projection method was used for the third CAFA community experiment [ZJB+19], for two reasons. Firstly, some of the proteins in the CAFA targets did not exist in the proteomes used in OMA (annotation mis-match). Further, some proteomes did not match exactly to a single species in the OMA browser: the *Xenopus laevis* was mapped to *Xenopus tropicalis* (XENTR, from *Ensembl* release 73); the *Escherichia coli* strain K12 was mapped to any of the strains included below the given NCBI taxonomy ID (83333).

In the future, it is expected that by using a similar technique, fast mapping and placement of a novel genome inside hierarchical orthologous groups (HOGs) will be possible. Pairing this with ancestral predictions of function, using the algorithm described in Chapter 3, will enable a more phylogenetically consistent approach.

Chapter 3

Functional Propagation using Hierarchical Orthologous Groups

The method presented in this chapter was used to submit predictions to the third CAFA challenge [ZJB+19], as part of this work. The method description and parameter optimisation are unpublished.

The algorithm behind the Orthologous Matrix (OMA) project infers orthologous genes among multiple genomes, using a pairwise optimal alignment of protein sequences [DCG+05; RGD08]. Using these orthologous relationships, two different groupings are then performed – into OMA groups and hierarchical orthologous groups (HOGs) [AGGD13; TGG+17]. These groups, as well as the current method for functional propagation, were introduced in Section 1.5.

Whilst the current method of propagation uses the functionally-coherent OMA groups, it has its shortcomings. Despite the popularity of using groups of orthologues to mean the same genes in different species, it is not possible to flatten the evolution of genes into simple groups. HOGs, instead, enable the comparison of highly diverged and similar species in a consistent manner, but have not yet been exploited in the context of functional inference. This chapter describes the work undertaken for this thesis to propagate functional knowledge through these HOGs.

3.1 HOG Propagation (HOGPROP) Algorithm

Propagating functional annotations along gene phylogenies is a classical notion (for example, [Eis98]). However, reconstructing large gene trees remains computationally demanding and error-prone. As a more scalable alternative, annotations can be propagated

across hierarchical orthologous groups (HOGs) [SGS+14]. For instance, in the case of Gene Ontology (GO) annotations, a subset (experimental and some electronic annotations [based on [ŠAD12]]) are given a score dependent on their evidence code. These terms, with scores, are then associated with the leaves of the hierarchical structure (genes), before being pushed up and pulled down the hierarchy as can be seen in Figure 3.1.

The score decays across each edge, currently set at a fixed rate of 20%, with penalisation when propagating over paralogous relationships of a double decay. Scores are combined at each node (using summation) during the up-propagation, whilst the maximum score is taken in the down-propagation. This is performed in an ontology-aware manner. That is, when dealing with ontology-based knowledge, the score associated to a particular term is also relevant to all less specific terms (parent terms) in the ontology.

Three combination methods are currently available:

- "Max" – taking the maximum of the scores;
- "Sum" – summation of the scores;
- "One-Max" – summation of the scores, with a maximum available score of 1.0.

After propagation, a score is available for every input annotation on all genes that are members of a group. A basic algorithm, similar to this, was submitted to the second CAFA experiment (team name "CBRG"), where it performed well under several criteria [JOC+16].

3.2 Benchmarking of HOGPROP for CAFA 3

For benchmarking purposes, before submitting to the third CAFA challenge, the data from the second challenge was used. The CAFA metrics of F_{max}, S_{min} and average AUC were used (see Section 1.7.1 for details). The latest version of input annotations and ontology definition were taken from the time before submission to the second experiment. However, the HOGs from the OMA release at the time of the third CAFA were used (May 2016), as the quality increases with the more genomes included and these would be used for the submission to the third CAFA.

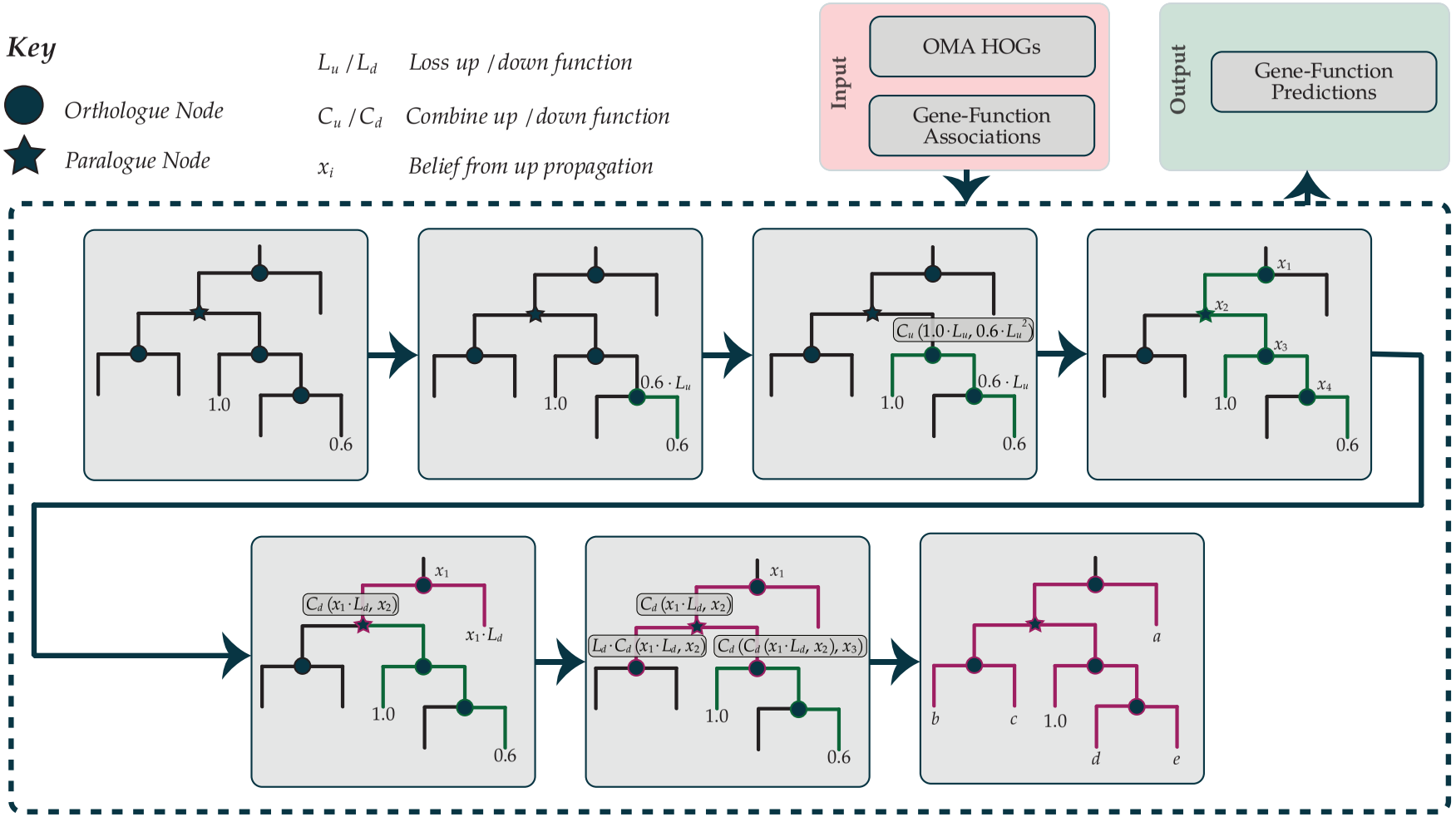


Figure 3.1: Overview of the HOGPROP algorithm, for propagating through hierarchical orthologous groups (HOGs). This visualises the propagation of a single gene-function association.

3.2.1 Input Annotation Filtering

Input annotations were taken from the UniProt-GOA, however these annotations have a varying level of evidence to support them. Three input datasets were created – purely experimental; experimental plus "trusted" electronic (IEA) annotations; experimental plus "trusted" IEA and those coming from PAINT [GLLT11]. The filtered IEA annotations are based on work by Škunca, Altenhoff and Dessimoz [ŠAD12] (see Appendix A for details).

3.2.2 Transfer Rates

The decay rate over orthologue / paralogue nodes can be set independently. For benchmarking before submitting to the third CAFA challenge, the rate of decay when entering an orthologue node was set to 20%. The penalisation of propagation over the paralogous relationships was then optimised. The transfer rate is

$$t_{orth} := \frac{100 - d_{orth}}{100},$$

where d_{orth} has been set to 20%, i.e. 0.8. The paralogue transfer rate is then defined as

$$t_{para} := t_{orth} \cdot \left(1 - \frac{p}{100}\right),$$

where p is the penalisation. This was varied from 0–100% in steps of 20%. That is, t_{para} was set to 0.8 (0%), 0.64 (20%), 0.48 (40%), 0.32 (60%), 0.16 (20%) and 0 (100%).

3.2.3 Scoring Transformation

Each of the combination methods were tested. The Max and One-Max methods can simply be rounded in order to ensure they are in the domain required for the CAFA scoring ($(0, 1]$, to two decimal places). However, the summation method does not. As the CAFA metrics are taken at the maximum / minimum from the submission, this means that the scores can be transformed in order to gain maximum resolution[1]. So, if the current annotations are set at 1.00 then the predicted annotations have the remaining 99 slots available. If there

[1]This is relevant to the Max and One-Max methods, also.

are 99 or fewer scores then it is possible to have a one-to-one mapping to the CAFA score. However, in the case of summation this was highly unlikely.

As such, an iterative procedure of binning into equal-width bins (from maximum to minimum score), with unused bins "deleted" and their score-slot pushed up towards 1.00. The last bin was then divided into the remaining slots, until there were no slots left. As there is a higher confidence in the predictions with a greater score, the lack of resolution at the top is of less importance than increasing the resolution at the bottom.

It was later found that this binning process was not optimal for spreading the scores to gain maximum resolution. Instead, the scores from each aspect can be fitted to a log skew-norm distribution and the cumulative distribution function (CDF) of the fitted distribution used to transform the scores. This results in an equal spread of scores into each of the bins, resulting in higher resolution and better results (used later in Chapter 4 for the HOGPROP methods).

3.2.4 Approximate Mapping

A number of the target sequences did not have an exact match to an entry in OMA. However, some had only a single amino acid difference due to the integration of genomic data from different sources. Coverage of the targets in each test was a problem of the method in the second CAFA. The k-mer approach described in Chapter 2 was used in order to map sequences to the closest entry in the OMA database. This meant that some still did not map to a HOG and as such there were still no predictions.

In Figure 3.2, the change in metrics based on performing the approximate mapping can be seen. In this case, the input annotations were only experimental. When looking at the scores that include the paralogue scaling, the F_{max} increases around 0.035 / 0.04 for the Biological Process (BP) / Cellular Component (CC) ontologies, respectively. For the Molecular Function ontology (MF), it increased by around 0.65.

The increase in the average AUC is very small. The S_{min} has a different story. Especially if normalised, as the S_{min} for BP terms is $\sim$29 and on the CC and MF aspects it is < 10.

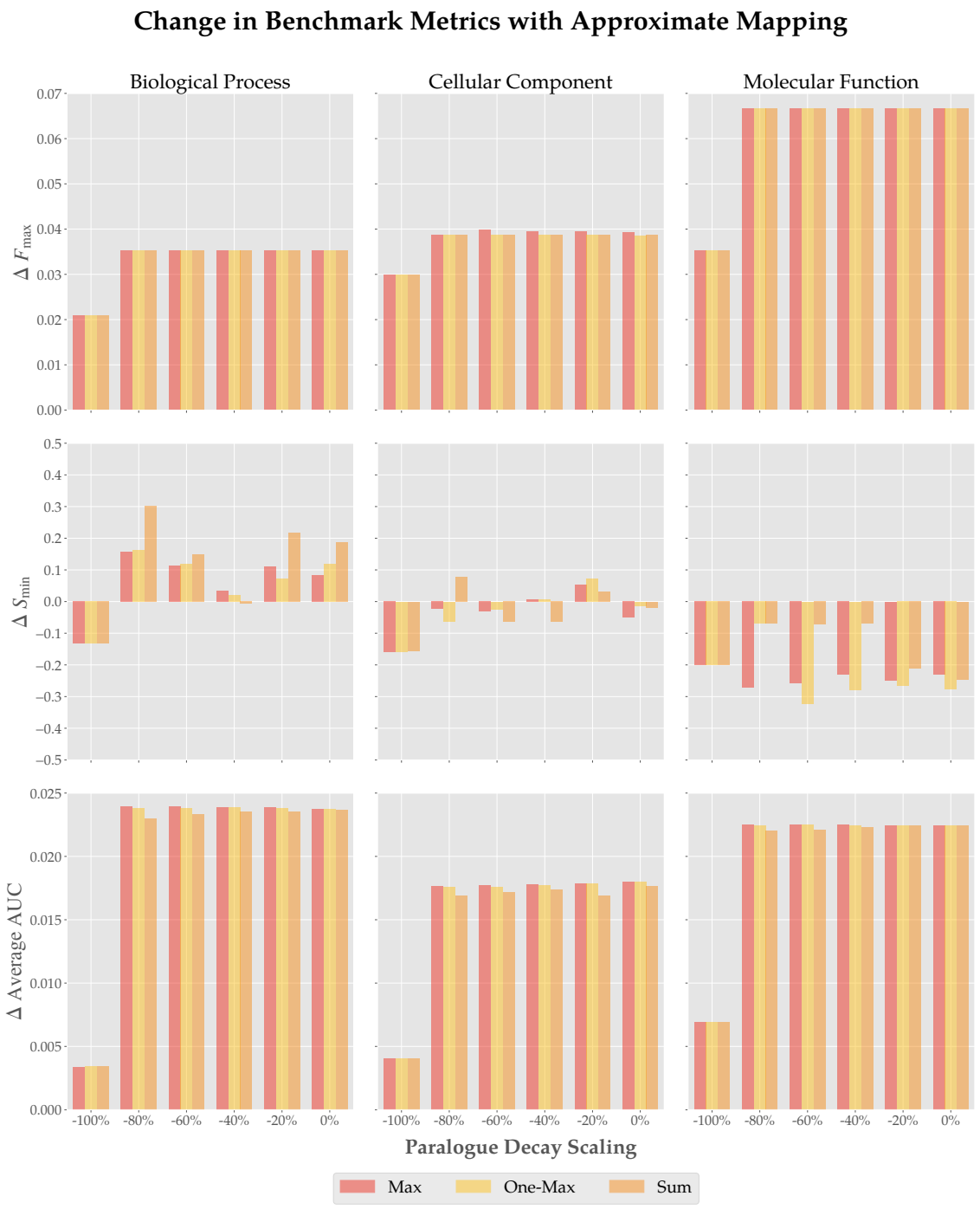


Figure 3.2: Change in metrics with approximate mapping, based on predictions made using experimental-only input annotations.

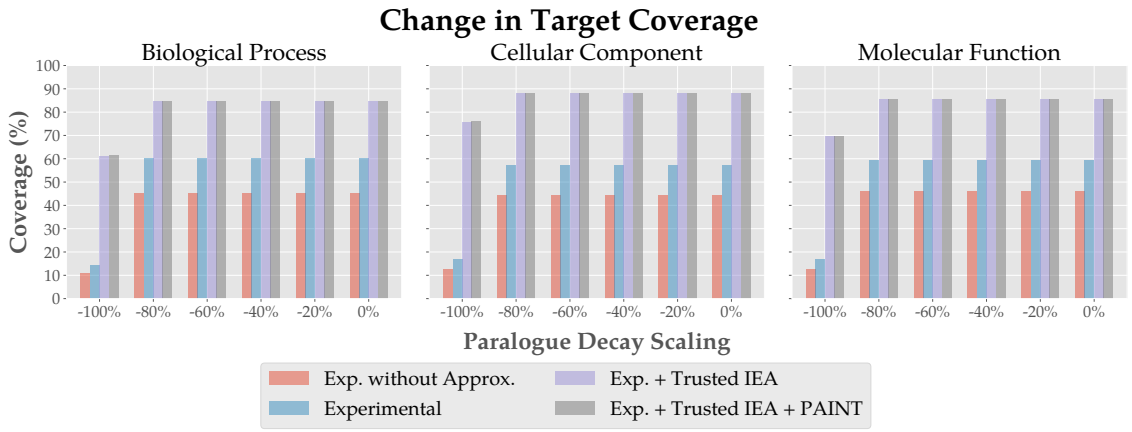


Figure 3.3: Coverage of the benchmark proteins, when varying the input annotations. All include predictions by approximate mapping, apart from the experimental without approximate mapping (included for reference).

Thus, the decrease in score is much more important than the relative increase seen in the BP aspect.

The coverage of the benchmark proteins drastically increases through approximate mapping (Figure 3.3). However, when relaxing the filtering of the input annotations to include "trusted" IEA terms then this increases further.

3.2.5 Parameter Choices

The predictions based on purely experimental input annotations were used to choose the combination method. In Figure 3.4 the metrics are shown for each aspect of the GO, for each of these methods. This shows that, in this situation, the metrics are fairly insensitive to this choice. The "Max" method was chosen, however, because it had a slight advantage in the S_{min} metric.

Then, using this combination function the paralogue decay scaling was chosen. The three initial datasets, described in Section 3.2.1 were used, with the benchmark metrics being computed on all. Results are shown in Figure 3.5. It is clear, as it was in the purely experimental case, that propagating knowledge across duplication events (the paralogue nodes) improves the scores in all situations. That is, the metrics are worse in all cases when the paralogue decay scaling is set to 100%. However, with all other values both the

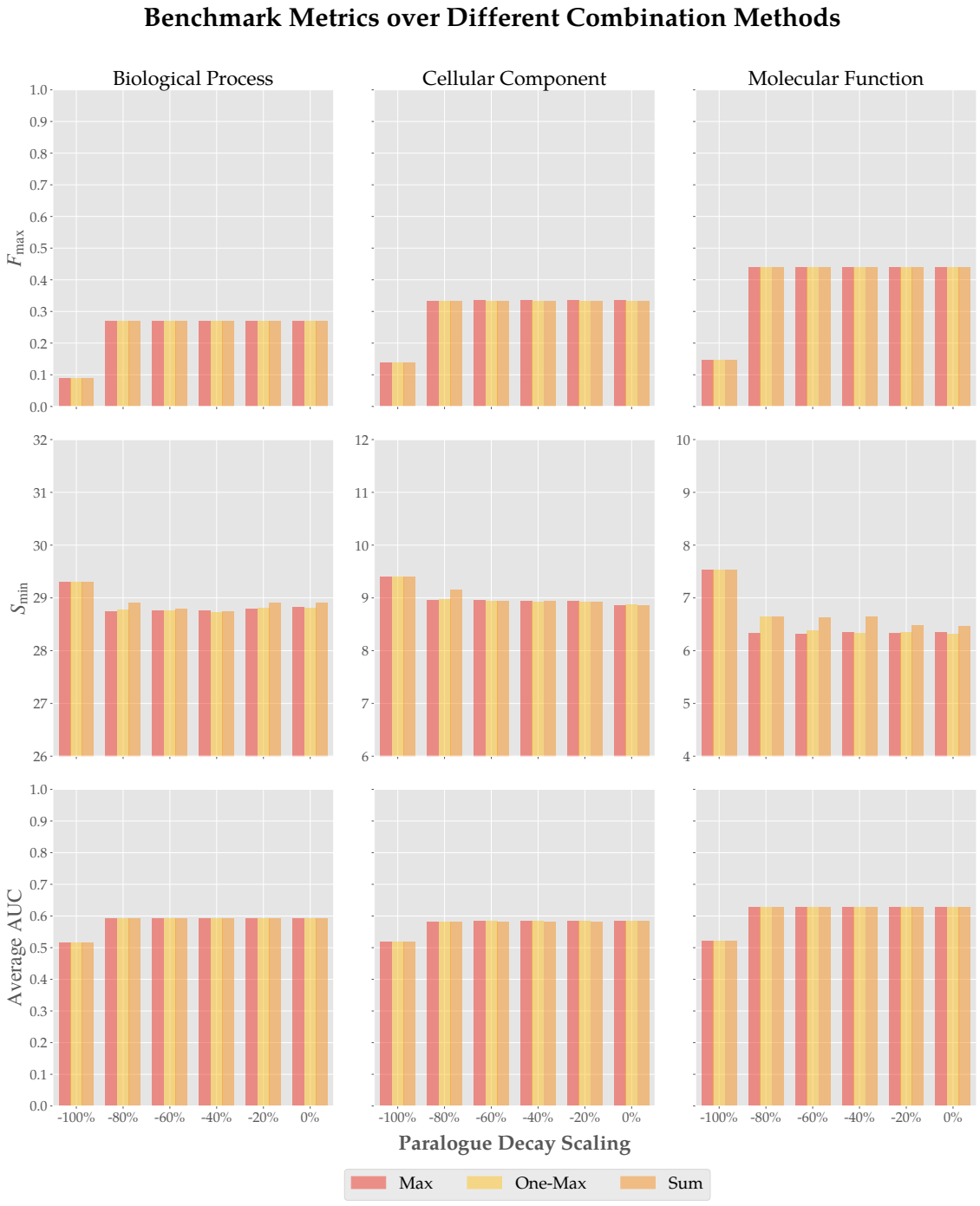


Figure 3.4: Benchmark metrics for each combination method, for each paralogue decay scaling. Input annotations purely experimental.

F_{max} and average AUC appear to be invariant. There is a very slight reduction in the S_{min} for 20% paralogue decay for the BP and MF aspects. This value was chosen.

This is in agreement with the observations of Škunca *et al.* [ŠBK+13], where orthologous and paralogous cliques were used to build phyletic profiles. It has also been observed that the divergence between paralogues is not as strong as would be expected – Altenhoff *et al.* [ASRRD12] found that the then-current experimental annotations did support the so-called "orthologue conjecture" of orthologues having greater functional similarity than paralogues, but more weakly than expected.

3.2.6 Comparison to Other CAFA 2 Submissions

Figure 3.6 shows the distributions of the metrics on all the submissions to the second CAFA challenge. The position of the predictions from HOGPROP is shown on these plots, from both the purely experimental and experimental plus "trusted" IEA input annotations. The third input annotation set was not included, as the results are very similar. The CAFA BLAST and naïve methods are also overlaid.

Focussing on the predictions from the dataset including IEA, HOGPROP is achieving around average in the F_{max} results. However, it is slightly below average for average AUC and well below average S_{min} for BP and CC terms. However, it is amongst the best S_{min} for MF terms.

3.3 Conclusions

This chapter has introduced HOGPROP, an algorithm which uses the hierarchical orthologous groups (HOGs) from the OMA project in order to predict Gene Ontology terms associated with protein sequences. This was optimised, using the CAFA benchmarks on the targets from the second CAFA. Whilst the propagation model through the structure of the HOG is simple, this appears to be effective.

A future direction for this work could be to implement a probabilistic model based on belief propagation networks, introduced by Pearl [Pea82]. A simple approach based on this has been used in the first version of SIFTER [EJMB05]. This framework was then

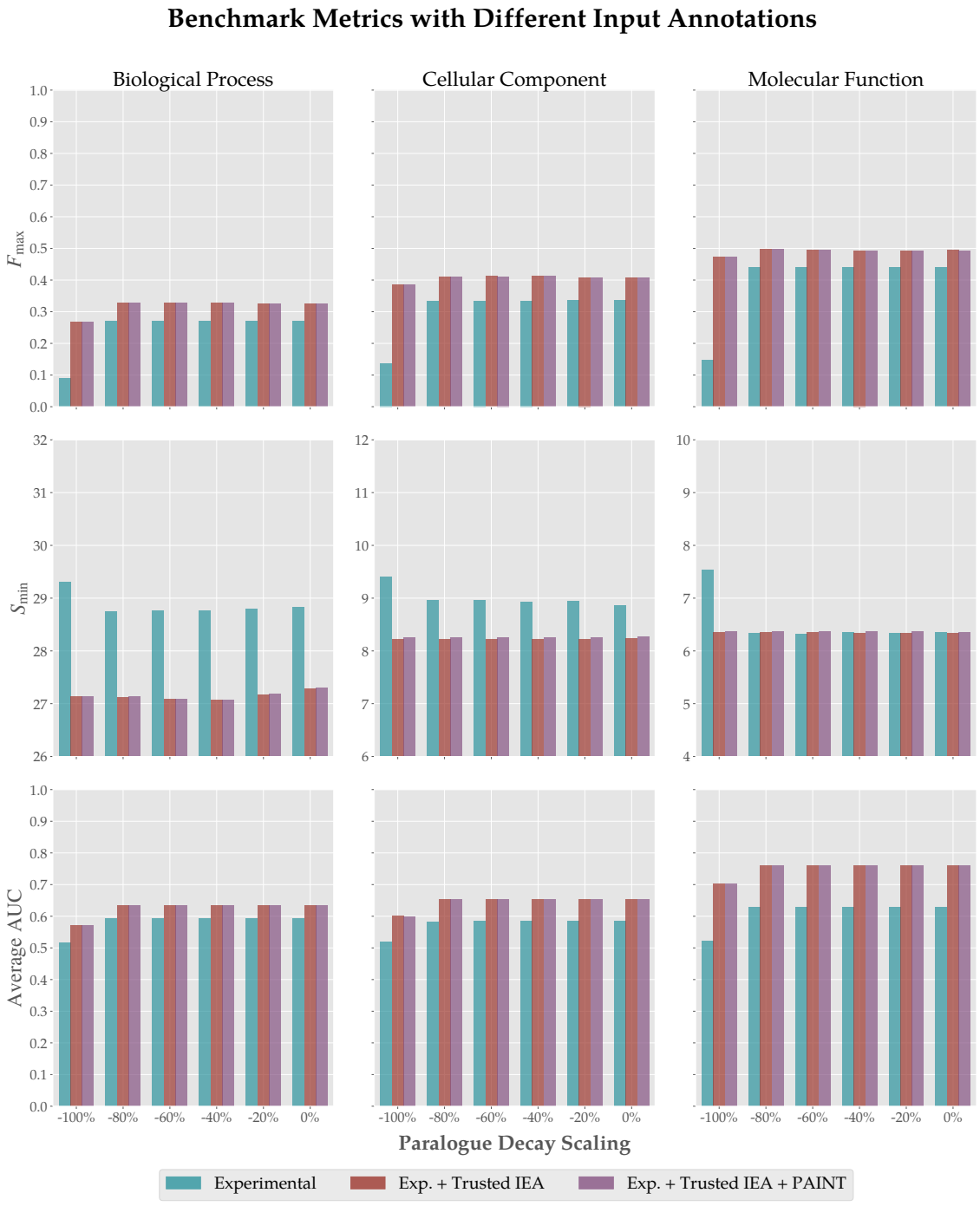


Figure 3.5: Benchmark metrics when varying the level of filtering on the input annotations. This shows that the benchmarks overall improve when given the "trusted" IEA annotations. The paralogue decay scaling was chosen as 20%. The metrics appear to be fairly insensitive to this changing. However, including the information from paralogous sequences appears to increase the performance of the tool. Each of these three filtered datasets were submitted to CAFA.

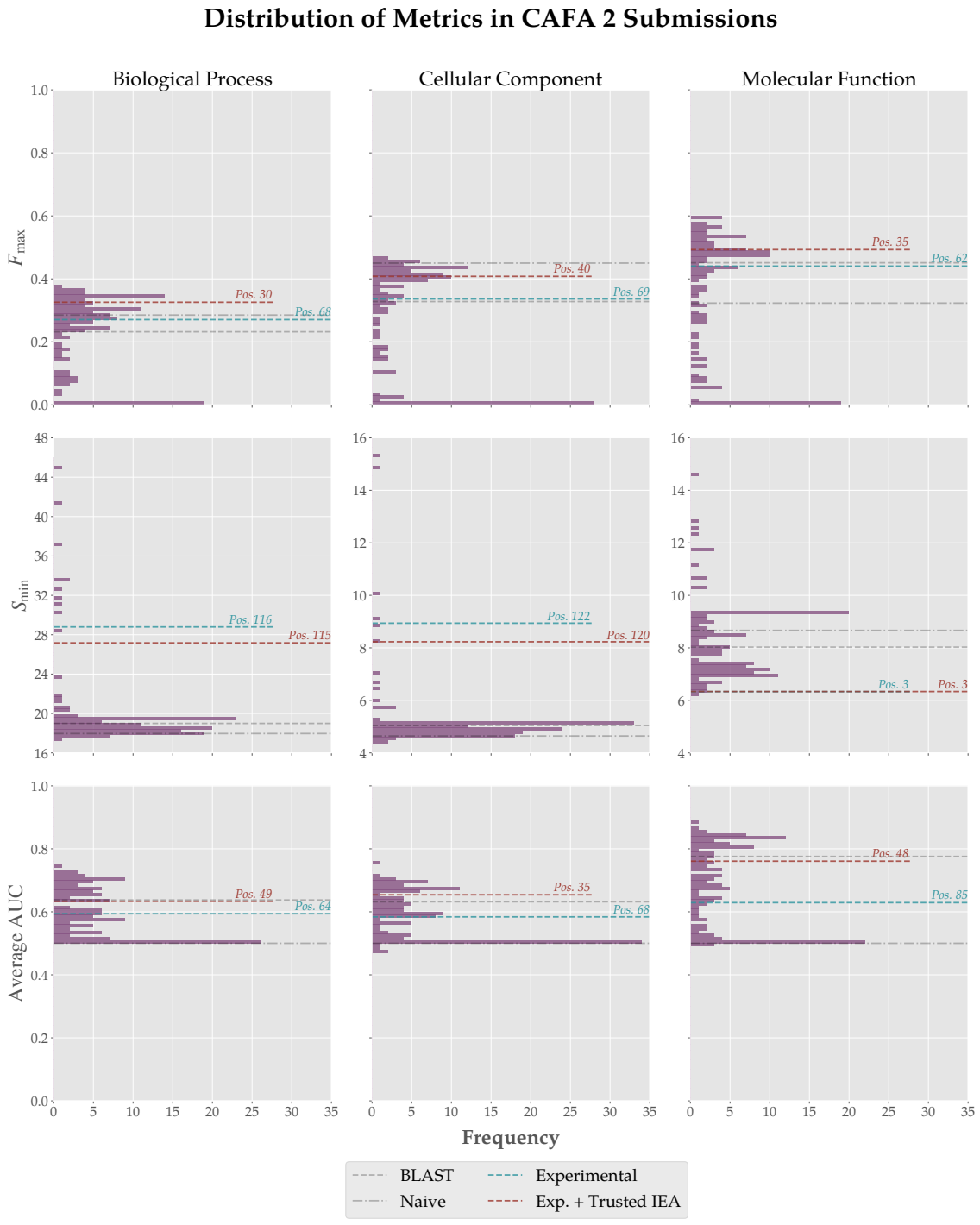


Figure 3.6: Distribution of metrics for CAFA submissions. Multiple submissions from the same team included. Total of 129 submissions. The predictions from HOGPROP were included, from both the purely experimental and experimental plus "trusted" IEA input annotations. Some extreme outliers are not visible in the S_{min} for BP terms.

extended, for their second version [EJSB11], to include extra parameters which were then estimated using generalised expectation maximisation. There may be some computational complexities when implementing this, as in the June 2019 release the largest HOG contains over 100,000 proteins. This would also require distances to be fitted to the edges within the HOG. Whilst the OMA algorithm estimates pairwise evolutionary distances, in order to construct an orthology graph, not all pairs are computed that would be required to fit distances to the HOG topology.

However, as previously mentioned, the CAFA metrics used in the benchmarking in this chapter do not account for the open world assumption (OWA) (see Section 1.7.2). This leads to systematic over-estimation of the false positives, resulting in a misleading benchmark. As such, the development of a benchmarking framework which accounts of the OWA is of high importance.

The following chapter proposes such a framework, before comparing HOGPROP to two baseline methods and GOtcha. Then, in Chapter 5, a method to fit distances to large phylogenies containing polytomies, in the context of missing pairwise distances.

Chapter 4

Benchmarking Gene Ontology Function Prediction Using Negative Annotations

The work presented in this chapter is currently under peer review.

With the ever-increasing number and diversity of sequenced species, the challenge to characterise genes with functional information is ever more important. In most species, this characterisation almost entirely relies on automated electronic methods. As such, it is critical to benchmark the various methods.

The CAFA series of community experiments provide the most comprehensive benchmark, with a time-delayed analysis leveraging newly curated experimentally supported annotations. However, the definition of a false positive in CAFA has not fully accounted for the Open World Assumption (OWA), leading to systematic underestimation of precision. The main reason for this limitation is the relative paucity of negative experimental annotations.

This chapter introduces a new, OWA-compliant benchmark based on a balanced test set of positive and negative annotations. The negative annotations are derived from expert-curated annotations of protein families on phylogenetic trees. This approach results in an increase in the average information content (IC) of negative annotations. The benchmark has been tested using the naïve and BLAST baseline methods, as well as two orthology-based methods. This new benchmark could complement existing ones in future CAFA experiments.

4.1 Motivation

According to the GOLD database, hundreds of thousands of genomes have already been sequenced, including close to ten thousand eukaryotes [MSB+19]. Within one decade, the Earth BioGenome consortium aims to sequence 1.5 million eukaryotic sequences [LRK+18]. At a molecular level, however, nearly all biological knowledge is concentrated in human and a handful of model species. Strikingly, in UniProt-GOA [HSMM+15], over 80% of all Gene Ontology annotations supported by direct experimental evidence are concentrated in just seven species. Thus, for the overwhelming majority of species, functional characterisation is almost entirely reliant on automated computational methods [CJ17].

As such, it is critical to benchmark the various computational methods. The CAFA series of community experiments have provided the most comprehensive benchmark, with a time-delayed analysis leveraging new experimentally supported annotations [RCO+13; JOC+16].

One major complication in assessing protein function predictions is that proteins typically possess multiple "functions" (*sensu* Gene Ontology [Tho17]), and knowledge of these functions, even for well-known genes in model species, is typically notably incomplete. This incomplete state of knowledge is referred to as the "open world" assumption (OWA) [TWM+12; ŠRS17]. To date, CAFA has not fully accounted for the open world assumption (OWA), leading to systematic underestimation of precision [DvT13]. For example, consider the human gene *Serotonin N-acetyltransferase* (`SNAT_HUMAN`) which controls the night / day rhythm of melatonin production in the pineal gland. A method, when this protein had no GO annotations, might have predicted *"circadian rhythm"* (`GO:0007623`), *"rhythmic process"* (`GO:0048511`) and *"indolalkylamine biosynthetic process"* (`GO:0046219`). Then, when *"circadian rhythm"* and *"rhythmic process"* were associated with this gene, they would both be considered true positives and *"indolalkylamine biosynthetic process"* as a false positive. Several years later, however, this term was associated with this protein – contradicting the assertion that it was a false positive and demonstrating the problem with assuming a "closed world" of complete knowledge.

Ideally, to be compliant with the OWA during benchmarking, explicit negative annotations are required – those that state a particular gene does *not* have a particular function – thus making it possible to classify computational predictions of the contrary as a false positive [DvT13]. Yet currently, in UniProt-GOA, less than 2.5% of all experimentally annotated proteins have a Gene Ontology annotation which is negatively qualified, indicated by the use of the "NOT" tag in the qualifier field of a GAF file [GŠHD17].

Furthermore, reasoning on ontologies when using negative annotations requires different treatment than with positive annotations. Thus, the information content associated with negative annotations needs to be computed differently. As is elaborated below, this has not been accounted for in benchmarks to date.

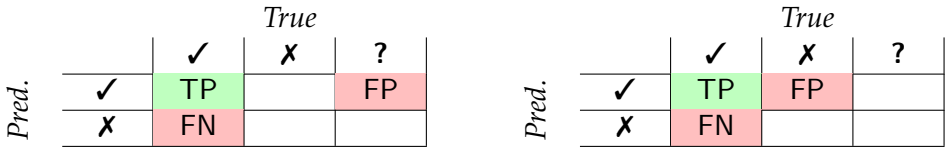
This paper introduces an approach to derive a large number of negatively qualified annotations from expertly curated gene phylogenies. Utilising these, a framework for OWA-compliant benchmarking was developed, based on a balanced test set of positive and negative annotations. This benchmark has been tested on the naïve and BLAST baseline methods, GOtcha and an orthology-based method. This new benchmarking framework could complement existing ones in future CAFA experiments.

4.2 Results

This section starts by highlighting the differences in benchmarking GO annotations with explicit negative annotations, over the current practice. Then, a derived set of negative annotations based on expertly curated gene phylogenies is presented. Next, the results of a method comparison, using these derived negative annotations to benchmark, are given.

4.2.1 Benchmarking Gene Ontology Annotation with Explicit Negative Annotations

A large amount of explicit negative annotations would help to address the Open World Assumption (OWA) in benchmarking. Further, benchmarking using these negative annotations requires different handling. It is customary to assess automated function predictors in a protein-centric sense. That is, computing some measure of quality – for example,



(**a**) Current benchmarks (**b**) Benchmarks using negative annotations

Table 4.1: Definitions of true positives, false positives and false negatives (for a single GO term on a single protein) used in (a) current benchmarks and (b) in this paper for no-knowledge targets. Current benchmarks use the lack of annotation to a particular GO term in the true annotations (symbol "**?**") to compute the set of false positive GO terms.

precision-recall – for each protein, with an average taken over the proteins tested. A set of true annotations is required, that are not available to the predictor, in order to properly assess the method. It is currently common-place to identify the false positive GO terms as those that have been predicted, but not in the set of true annotations (Table 4.1 (a)). When there are sufficient negative annotations in the true annotation set for a given protein, the false positives can then be identified as overlapping with these (Table 4.1 (b)).

Furthermore, because different terms vary in their information content (IC) – for example, a positive association with a term such as "root hair elongation" (`GO:0048767`) is more informative than the more general term "growth" (`GO:0040007`) – it is common to compute weighted precision-recall curves. For instance, Clark and Radivojac [CR13] proposed to weight by the information accretion (see Section 4.4.1 for details) – an approach which was subsequently implemented in CAFA 2 [JOC+16]. In order to compute the IC of GO terms, the probability is required – this can be estimated using the empirical annotation frequency of each term.

However, it is important to recognise that the information content of a single term is not the same if it is negatively or positively qualified. For example, it is easier to show that a gene should be annotated with the general *metabolic process* term (`GO:0008152`) than a particular metabolic process, for instance *lactose biosynthetic process* (`GO:0005989`). On the other hand, it is exceptionally challenging to show that a gene is not associated with any metabolic process, in comparison to showing that it is not involved in a very specific one. Thus, more general terms in the GO have a lower IC than more specific ones when

a positive association is made. However, the inverse is true for negatives – general terms have a greater IC than those that are more specific.

Hence, it is necessary to estimate the IC of negative annotations separately (for details, see Section 4.4.1) – ensuring to propagate term counts to children instead of the parents, unlike for positive annotations [GD17].

4.2.2 Deriving Negative Annotations from Curated Gene Phylogenies

Expert curators have annotated ancestral states in gene phylogenies with GO terms, using the *Phylogenetic Annotation and INference Tool* (PAINT) [GLLT11] on PANTHER families [MMT12]. These ancestral annotations are then propagated down the phylogeny to the extant genes. Both positive and negative (that is, "NOT"-qualified) annotations are recorded in ancestral states.

Considering an individual GO term, if a curator finds evidence that this term applies to all members of the gene family then the root node shall be annotated (Figure 4.1 (a)). However, if there is evidence that this function is not present in a particular sub-tree then a negative annotation would be assigned to an internal node (coloured red here) (Figure 4.1 (b)). This implies that the gene in question has lost a particular function on the branch leading to this node.

A curator might annotate an internal node with the term of interest, without propagating it all the way to the root (Figure 4.1 (c)). This could be motivated, for example, by a lack of experimental information outside of the sub-tree, or taxon-based constraints [DDM10; TMMT18]. Irrespective of the reason, an expert curator has deemed that there is currently a lack of evidence to annotate the root node with this term. As such, it can be argued that an automated predictor should be penalised for predicting such terms.

By scanning the PAINT annotations for such instances, it is possible to derive many pairs, (p, t), where p is a protein which is member of a family where an ancestral node, not in its direct lineage, has been annotated to a GO term t. That is, p is not covered by a PAINT annotation (positive or negative) for a GO term t, but other members of its family are.

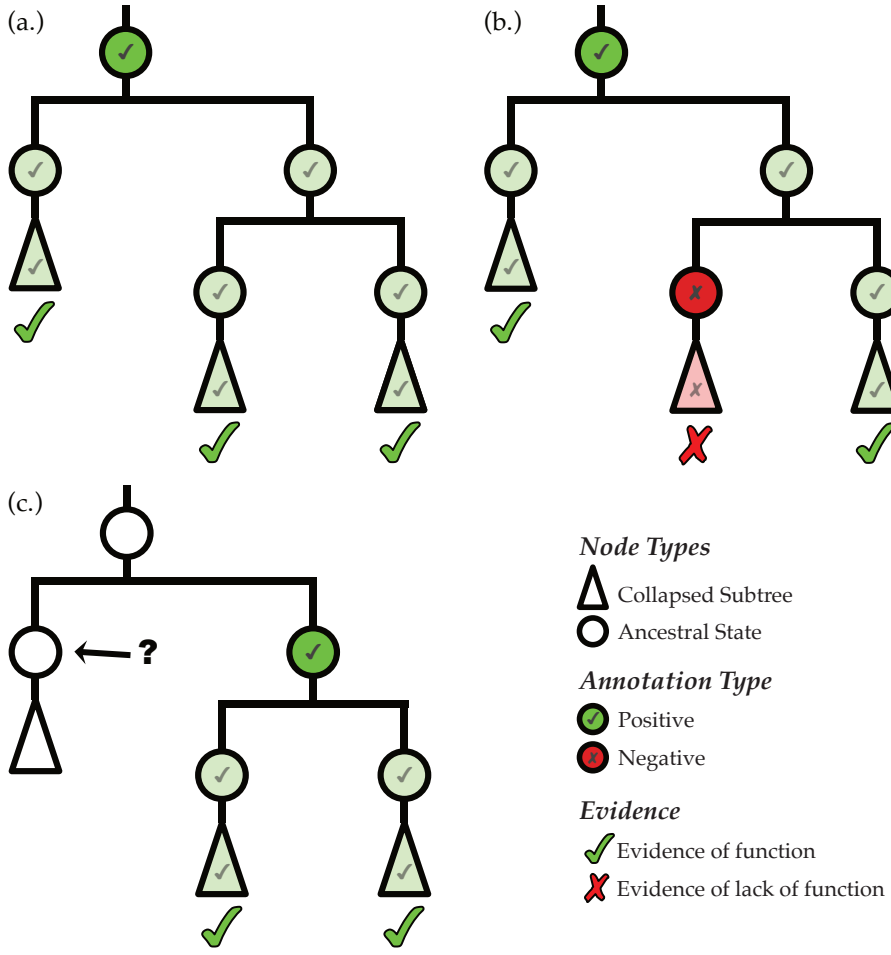


Figure 4.1: Possible locations in an example gene phylogeny where a curator can annotate a term in positive and negative way. Green and red nodes indicate positive and negative annotations, respectively. The propagated annotation on child nodes and the collapsed sub-trees are shown in lighter colour. In (a) this shows a term for which there is evidence on all sub-trees and, as such, the root node is positively annotated with the term. Then, (b) shows that if there is instead lack of evidence or evidence of lack of function in one of the sub-trees then the annotator will negatively associate the node leading to this sub-tree, however there are few such cases. If, instead, there is no information on the left hand side of the tree, as in (c), the curator would annotate a lower node than the root and leaving the left hand side (see question mark) without annotation.

The number of such pairs is shown in Figure 4.2 (a) for on each aspect of the Gene Ontology – Biological Process (BP), Cellular Component (CC) and Molecular Function (MF). In the database, only 11,633 proteins were covered by a negative annotation in UniProt-GOA – consisting of 4,911 with BP annotations, 4,619 with CC and 5,068 with MF. After including the derived negative annotations, this increased to 330,635 – 198,848 with BP, 268,831 with CC and 192,307 with MF. This is more than the number of proteins with at least one positive (non-IEA) annotation (323,438) as well as more than those with only at least one CC positive (non-IEA) annotation (266,658). Further, when including these derived negative annotations, there is an increase in average information content of the negative annotations, as can be seen in Figure 4.2 (b).

4.2.3 Balanced Benchmarking

In general, approaches to benchmarking GO annotations recognise that some aspects of function are easier to predict than others. Thus, they typically consider the Information Content (IC) of each annotation. Furthermore, since the IC for the same term varies whether it is associated positively or negatively with a given target (see above), this difference should also be taken into account. One such way to account for differences in IC amongst annotations is by weighting predictions by their IC. However, this only works up to a point: if there are no, or very few, annotations with high IC, the results will have a very large variance and thus not be particularly informative. To avoid this, it is possible to design a benchmark to test GO terms for which there are informative positive and negative examples. Henceforth, this design shall be referred to as a "weighted and balanced" benchmark.

To investigate the two approaches (weighted-only, as well as weighted and balanced), two test sets were generated that represent each case. For the weighted-only case, the test set contains 2,992 distinct protein-pairs from each annotated gene family, consisting of a protein with positive annotations (p_+) and the other with negative (p_-). True positive and false negative terms are identified with the positive protein, p_+, and false positives with p_-. For the "weighted and balanced" case, proteins were chosen for every GO term that has a positive and negative example within a protein family, resulting in 12,613 protein

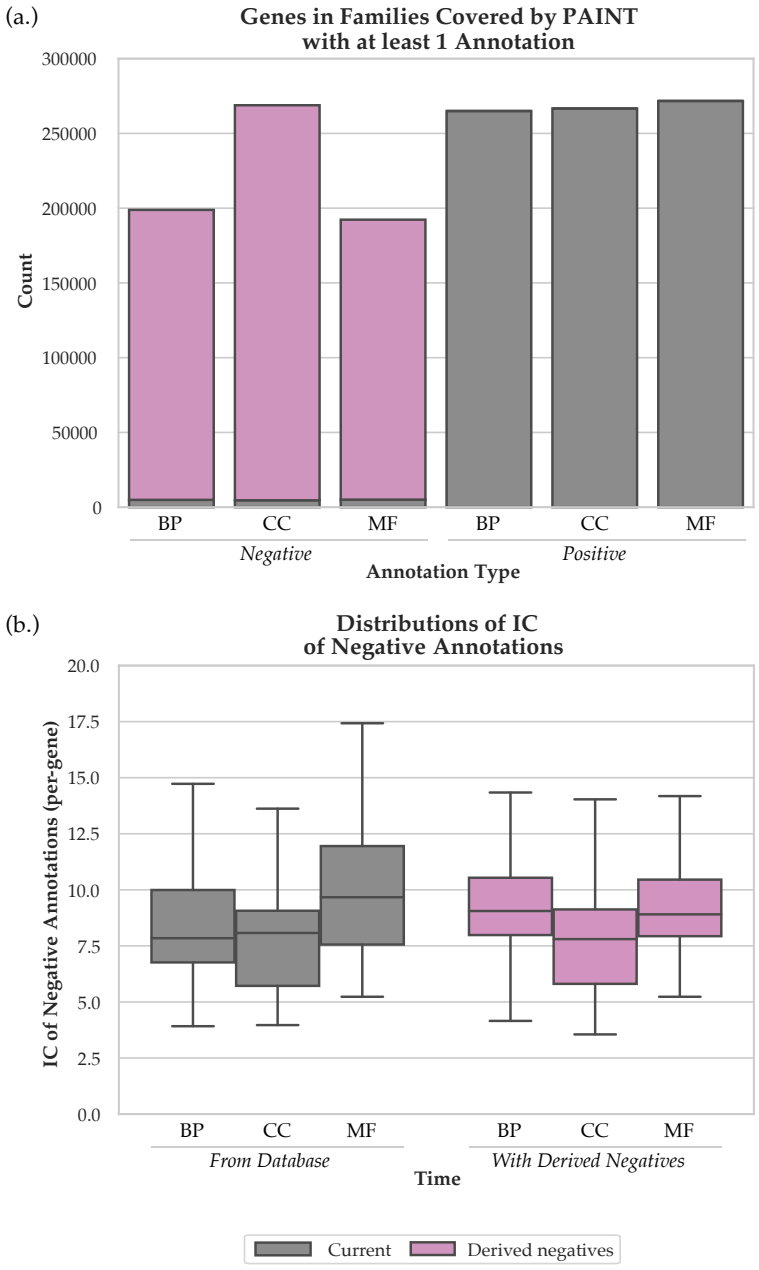


Figure 4.2: Resulting number of annotations (*a.*) and difference in average information content (*b.*) when including the curated negative annotations. (*a.*) shows the number of genes in PANTHER families covered by PAINT, with at least one non-IEA annotation. Relatively few (4,911 [BP], 4,619 [CC], 5,068 [MF]) were covered by a negative annotation in the database, increasing to 198,848 (BP), 268,831 (CC) and 192,307 (MF), with the curated negative annotations. For CC this is more than the number of proteins with at least one positive (non-IEA) annotation (266,658). (*b.*) displays the distribution of IC of negative annotations on genes in PANTHER families covered by PAINT. An increase in average IC of negative annotations is observed when including the curated negatives.

pairs (with associated GO term). In this case it is still necessary to weight, to account for variation of information among GO terms for positive or negative annotations (Figure 4.3). For details, see Section 4.4.4.

For each of the benchmarks, separately, all annotations to the target proteins were hidden during training, with all terms and proteins tested at the same time. Predictors for which it was possible to provide custom training data were used: the two baseline methods included in CAFA (naïve and BLAST), GOtcha [MBB04] and HOGPROP (`DessimozLab` in the third CAFA). All methods output a confidence score, $\alpha \in (0, 1]$. By varying the confidence cut-off (taking only annotations for which $\alpha \geqslant \tau$), a precision-recall curve was computed (for more details, see Section 4.4.4). For comparison to benchmarks under the CWA, the positive example genes from the weighted-only benchmark were used in order to identify false positives and weighting as in the CAFA weighted-precision recall benchmark.

The results are shown across the three different aspects separately (rows) with the different assessment methods in each column (Figure 4.4). The width of the curves represents the average IC of the predictions which are used to calculate the precision measures. The maximum F_1 scores (F_{max}) for each method, on each aspect, are available in Table 4.2 and also displayed as points on the curves.

The closed world assumption (CWA) benchmark recapitulates some key observations from the CAFA experiments [RCO+13; JOC+16]: naïve, which only relies on background term frequencies, performs especially well in Cellular Component terms – where most annotations are relatively general. BLAST, also considered as a baseline approach, performs worse than the non-baseline methods, even at stringent score cut-offs. Predictions for Molecular Function and Cellular Component terms are generally more accurate than for Biological Process.

However, besides the questionable discussed previously, the narrow lines in the plots indicate that most terms considered in the CWA benchmark have low Information Content (IC). This is particularly the case for the naïve method, which inherently focuses on high frequency (and thus low IC) terms.

If explicit negative annotations are used instead, the picture changes markedly. However, the first variant, which uses the weighted-only scheme, carries little information (Figure B.1 bottom row). Indeed, the naïve predictor performs with 100% precision at low recall, even better than in the CWA (Figure 4.4 left vs. middle column). This can however be explained by the complete lack of negative annotations involving general terms, reflected in the very low average IC of annotations (thin curve).

The weighted and balanced OWA benchmark provides more insight (Figure 4.4 bottom row). In the second OWA benchmark, the test set consists of pairs of proteins, a positive and negative example, for each GO term in each family containing both annotation types. This tests a predictor's ability to discriminate between homologous proteins.

With a balanced test set the naïve predictor performs much worse than in conventional CWA tests. This is because very general predictions, which are very easy to prove but near impossible to disprove, are by design not considered here. In other words, when naïve is evaluated on testable predictions, it makes many mistakes, which is reflected in this OWA benchmark. The recall is also markedly lower, which would be expected from a method inherently limited to predicting only the most frequent terms.

Likewise, results obtained for the BLAST predictor make more sense than on conventional CWA benchmarks: precision is very high where recall is low, but degrades steeply when recall increases. This makes sense, as the confidence score is based on the percentage sequence identity, high-precision-low-recall results are obtained when sequence identity is close to 100%, and where one would expect functions to be highly conserved. Increasing recall requires more permissive thresholds, which also results in more false positives.

One last finding of note is that GOtcha, a method which combines BLAST results, performs particularly well under the CWA benchmark. For instance, on the Molecular Function (MF) aspect, GOtcha achieves an F_{max} of 0.65 compared to the next best method of 0.58 (HOGPROP2). However, in the weighted and balanced OWA benchmark, it performs worse than BLAST (F_{max} of 0.52 vs 0.55 in MF). This large discrepancy appears to be due to two main factors. First, the internal scoring scheme of GOtcha strongly favours general terms. As seen with the naïve predictor, predictions of general GO terms tend

to be rewarded in conventional benchmarks. However, being practically impossible to disprove, they are by design not considered in the balanced benchmark. Secondly, given a target protein to be annotated, although GOtcha uses the E-values of BLAST matches to the target to assess the relative plausibility of the GO annotations associated with each match, it then normalises the scores obtained for each target by the maximum score of that target. As a result, predictions for a target for which the best functionally annotated BLAST match is, say, 100% identical could receive the same confidence as a prediction for a target for which the best is only 40% identical. Indeed, by removing this normalisation, a substantial improvement for GOtcha was observed in the weighted and balanced OWA benchmark (Figure B.2).

4.3 Discussion and Conclusion

Current benchmarks make an assumption that proteins are fully annotated, by identifying false positives as all the predicted terms which are not confirmed by experimentally backed annotations. Instead, to account for the open world assumption (OWA), it is necessary to utilise explicit negative annotations in order to assess GO predictions. The methodology developed in this study provides the necessary framework to benchmark using negative annotations.

To overcome the relative paucity of negative annotations (Figure 4.2), this study identified a substantial source of negative annotations derived from the expertly curated annotation of gene phylogenies in the PAINT project. After performing this procedure, when considering all genes which are members of families that have been annotated in PANTHER, there is roughly the same number of genes that have at least one positive annotation to that with at least one negative.

These derived negative annotations are of higher quality than negative electronic assertions, however less so than those that have been performed manually by an expert. One potential issue is that it requires annotators to carefully gauge the most appropriate level of specificity of the term used in annotations. If a curator, in an abundance of caution, assigns an overly general term to a subset of the gene family, the lack of this annotation will be

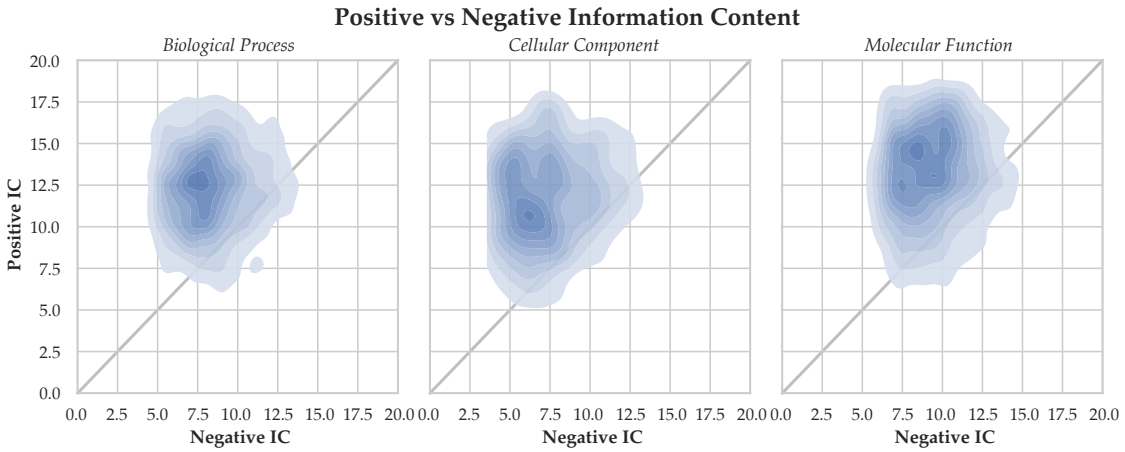


Figure 4.3: Information content of individual terms when associated positively vs negatively. There is a clear difference in the information content between the two, which motivates the weighting and balancing scheme used in this study.

	CWA			*OWA Weighted-Only*			*OWA Weighted and Balanced*		
Method	F_{max}	τ	Avg. IC	F_{max}	τ	Avg. IC	F_{max}	τ	Avg. IC
Naïve	0.25	0.14	2.96	0.59	0.01	4.73	0.14	0.01	7.09
BLAST	0.29	0.47	7.84	0.55	0.21	5.75	0.51	0.41	9.70
GOtcha	0.40	0.35	5.69	0.53	0.01	5.61	0.49	0.02	9.28
HOGPROP1	0.39	0.62	7.17	0.52	0.06	5.34	0.58	0.05	9.97
HOGPROP2	0.50	0.65	7.03	0.59	0.48	5.77	0.66	0.30	10.05

(**a**) Biological Process

	CWA			*OWA Weighted-Only*			*OWA Weighted and Balanced*		
Method	F_{max}	τ	Avg. IC	F_{max}	τ	Avg. IC	F_{max}	τ	Avg. IC
Naïve	0.41	0.30	1.51	0.61	0.03	2.45	0.30	0.01	6.08
BLAST	0.40	0.43	5.49	0.61	0.20	3.72	0.53	0.38	7.91
GOtcha	0.55	0.41	3.40	0.58	0.01	3.56	0.51	0.04	7.69
HOGPROP1	0.53	0.70	5.44	0.60	0.01	3.47	0.62	0.14	8.30
HOGPROP2	0.61	0.65	5.66	0.70	0.01	3.73	0.69	0.37	8.43

(**b**) Cellular Component

	CWA			*OWA Weighted-Only*			*OWA Weighted and Balanced*		
Method	F_{max}	τ	Avg. IC	F_{max}	τ	Avg. IC	F_{max}	τ	Avg. IC
Naïve	0.36	0.15	2.02	0.56	0.01	4.42	0.09	0.01	7.04
BLAST	0.47	0.45	7.15	0.62	0.28	6.38	0.55	0.40	10.66
GOtcha	0.65	0.37	5.96	0.61	0.07	6.23	0.52	0.04	10.42
HOGPROP1	0.53	0.72	7.30	0.63	0.01	6.34	0.64	0.03	11.12
HOGPROP2	0.58	0.67	7.56	0.69	0.01	6.57	0.69	0.23	11.07

(**c**) Molecular Function

Table 4.2: Cut-off and average information content at the point of the maximum F_1 score (F_{max}), for each method on each aspect of the Gene Ontology, as measured under the CWA, OWA weighted-only and OWA weighted and balanced benchmarks.

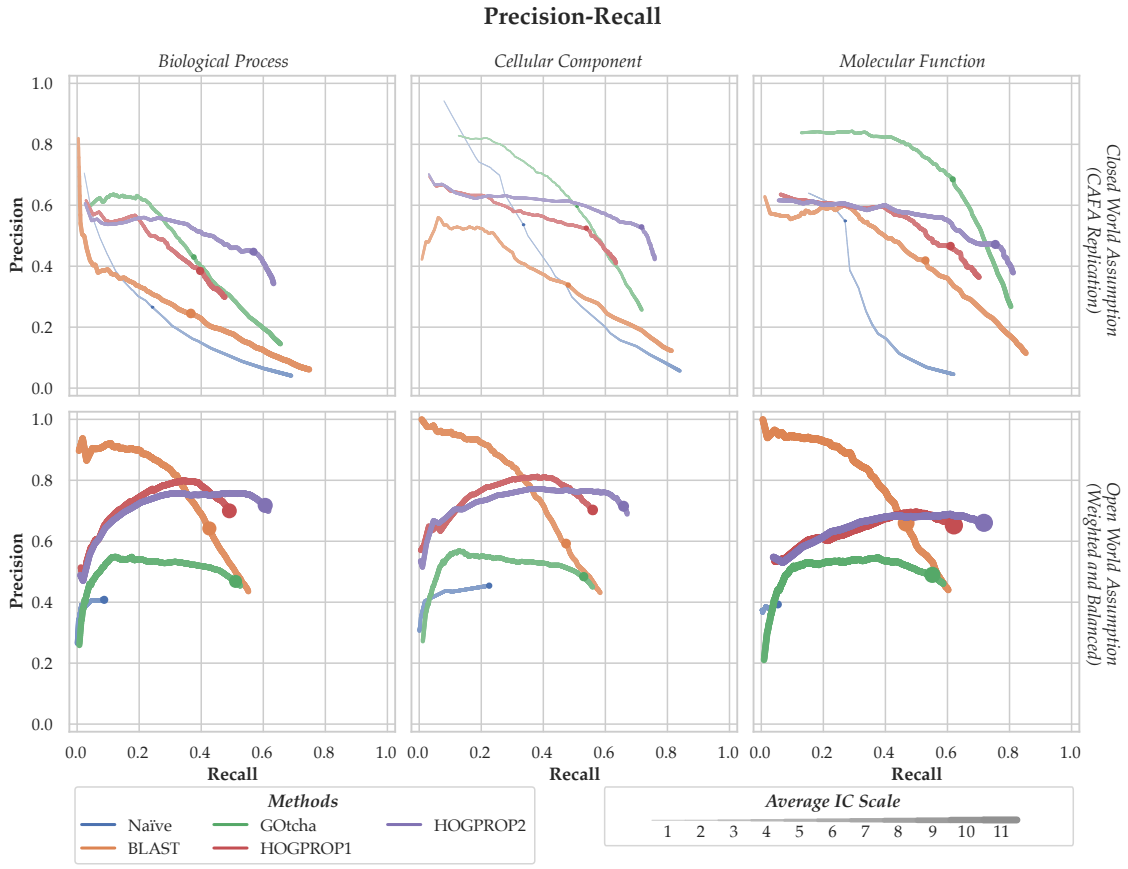


Figure 4.4: Precision-recall curves, for each aspect of the GO separately (columns) with the line-width and colour altering based on the average IC of the assessed predictions. (Top) benchmarking under the CWA – identifying false positives using unknown knowledge; (Bottom) weighted and balanced OWA-compliant benchmark, using positive and negative examples for each GO term, for which they exist. The thickness of the curves represents the average IC of the predictions which are used to calculate the precision at that point. The maximum F_1 score (F_{max}) is shown as a point on each curve – values are available in Table 4.2.

interpreted by the derivation procedure as a negative annotation with high information content. The procedure also presumes that any annotation placed lower than the root of the phylogeny has been done so deliberately. Yet, there are plausible situations when this may not be the case, such as when the underlying gene phylogeny is updated (for example, between PANTHER releases) or the inclusion of new species without a thorough review of each family. These potential pit-falls could be addressed by: (1) being cautious when choosing which terms to derive negatives for; (2) utilising date stamps for when a family's annotation set was last approved by a curator. The former has been implemented by only deriving negatives for GO terms with a *positive* IC greater than, or equal, to five – limiting the negative annotations to more specific terms. The latter is more complex and could be resolved in a future study.

There are, however, many cases where the derived negative annotations makes sense. One such case is in the PANTHER family `PTHR10686` (Figure 4.5). The root node of this family has been annotated to *transmembrane transport* (`GO:0055085`). Then, further down at the level of the *Chordata*, there is a duplication. One sub-family (green) has been annotated to have the molecular function *folic acid transmembrane transporter activity* (`GO:0008517`), whilst two other sub-families after the duplication have been annotated to have the molecular function *thiamine transmembrane transporter activity* (`GO:0015234`). It appears that after this duplication, the function has specialised to transport either folic acid or thiamine. In the weighted and balanced OWA benchmark, there were a number of tests performed on GO terms for which there are positive and negative examples in this family. For example, the *thiamine transmembrane transporter activity* (`GO:0015234`) was tested on the proteins with UniProtKB IDs `F6SXG7` (sub-family C) and `F1N2M7` (sub-family A) as positive and negative examples, respectively. Likewise, *folic acid transmembrane transproter activity* (`GO:0008517`) was tested on positive and negative examples `F1PFN8` (sub-family A) and `F6SXG7` (sub-family C), respectively. At the F_{max} point, both these paired tests show that none of the methods can correctly discriminate between these two GO terms on these sequences from the same gene family (see Section 4.3). Finally, another test was performed on *folate transmembrane transport* (`GO:0098838`), with positive and negative examples of `F7EDM0` (sub-family A) and `C3ZIU7` (not in labelled sub-families), respectively. At the F_{max}

point, both BLAST and HOGPROP2 correctly discriminate between these closely related proteins, whereas GOtcha and HOGPROP1 do not.

Despite the plethora of methods developed and submitted to the CAFA challenge, only a few of them are available as standalone software. This makes it difficult to test them on newly developed benchmarks, such as the one introduced here. Note that web-based services, while convenient for end-users, are difficult to include in such a benchmark due to the lack of control over the input – it is very important that the ontology definition and existing protein annotations are carefully controlled during training, to avoid circularity in evaluation.

Time-lapsed studies, such as CAFA, are by design less prone to this circular evaluation. However, they require a steady supply of new annotations. For the derived negative annotations introduced here, time-lapsed studies would require steady supply of gene families newly annotated by PAINT or a similar curated approach. This may seem more constraining than merely annotating individual gene targets using the literature. However, family-wise annotation is also more consistent and scalable than the inconsistent process of annotating individual targets; their value in benchmarking based on negative examples is an additional incentive for this curation effort.

Directly curated, experimentally-backed negative annotations – from expert curators – would be even more valuable than the derived negatives introduced here. Indeed, there is a great interest within automated functional annotation methods for a high-quality source of negative annotations, for both method-development and benchmarking. In particular, recent developments in, so-called, "deep learning" machine-learning methods show promising results, but heavily rely on training sets consisting of both positive and negative examples.

More specifically, this study also provides guidance to curation, by quantifying which individual terms – positive or negative – are most valuable for benchmarking. Whilst positive associations become more informative the further they are away from the root-terms, negative annotations are more informative the closer they are. Negating particularly general terms may prove prohibitively difficult to experimentally validate. This also explains why

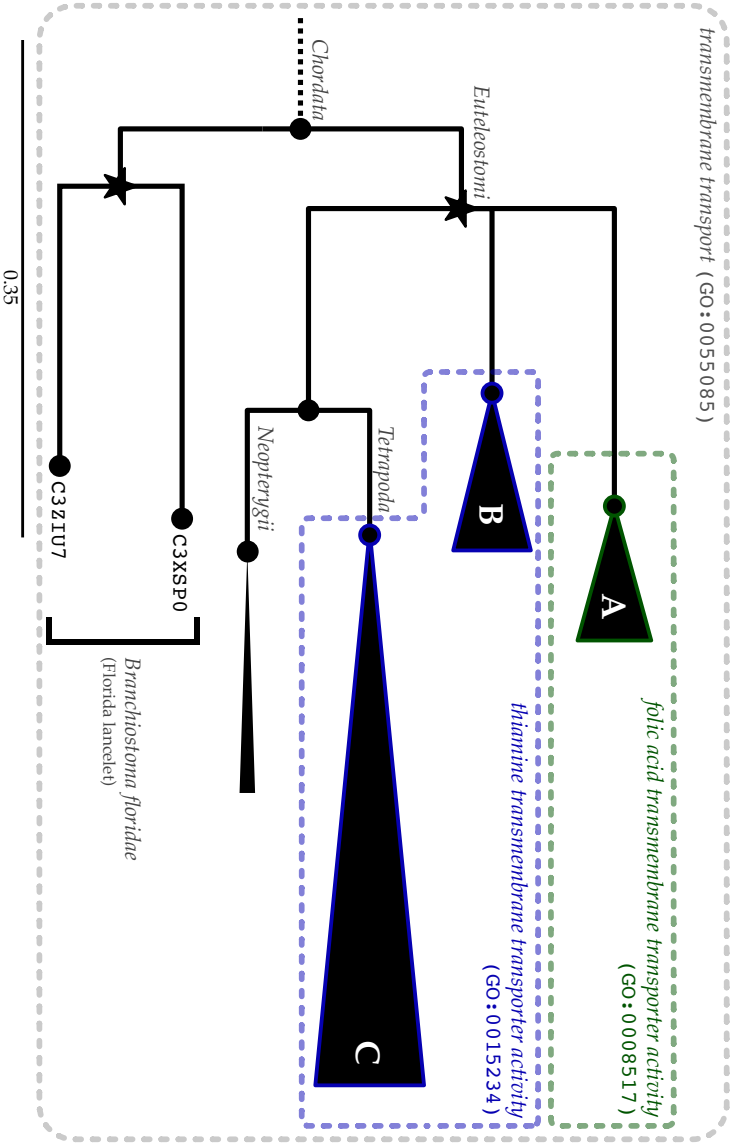


Figure 4.5: Sub-family of PANTHER family PTHR10686 – the root term is annotated to *transmembrane transport*, whilst particular sub-families have been annotated to *folic acid transmembrane transporter activity* and *thiamine transmembrane transporter activity*. This implies that, for example, proteins outside of that annotated with *folic acid transmembrane transporter activity* (green) should *not* be annotated with this term.

GO Term			Example Proteins		Method Predictions									
					Naive		BLAST		GOtcha		HOGPROP1		HOGPROP2	
ID	Name	Aspect	Positive	Negative	+	-	+	-	+	-	+	-	+	-
GO:0015234	*thiamine transmembrane transporter activity*	MF	F6SXG7 (Sub-Fam. C)	F1N2M7 (Sub-Fam. A)	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓
GO:0008517	*folic acid transmembrane transporter activity*	MF	F1PFN8 (Sub-Fam. A)	F6SXG7 (Sub-Fam. C)	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓
GO:0098838	*folate transmembrane transport*	BP	F7EDM0 (Sub-Fam. A)	C3ZIU7 (Sub-Fam. –)	✗	✗	✓	✗	✓	✓	✓	✓	✓	✗

Table 4.3: Results for subset of tests performed on PANTHER family PTHR10686 in the weighted and balanced OWA benchmark, at the F_{max} point. For each method, predictions are listed – tick indicates the method predicted, cross that it did not. Green / red colouring indicates correct / incorrect classification, respectively. Those for both *thiamine* and *folic acid transmembrane transporter activity* show that all methods fail to discriminate between these two terms. Whereas, on the term for *folate transmembrane transport* both BLAST and HOGPROP2 correctly classify the two proteins. These terms all have too low a frequency in UniProtKB/Swiss-Prot for the naïve predictor to make predictions. Proteins are referred to with UniProtKB identifiers – sub-families refer to Figure 4.5.

only using general terms in a benchmark is not merely uninformative [GŠHD17; Pes17; CR13; ŠRS17], but misleading.

When weighting by information content it is possible to correct for the difference in information *within* and *between* protein annotation sets. It does not, however, provide a balanced test – especially if only general terms are used. The balanced OWA-compliant benchmark provides a balanced test set such that methods are only rewarded for predicting terms that can be disproved. This, alongside the relatively low information content of annotations considered in the benchmark under the closed world assumption, explains why the naïve predictor performs so well in CAFA.

Finally, this work highlights the importance of the methodological details underpinning benchmarking. The absolute and relative performance of methods is enormously affected by seemingly technical decisions. Overcoming the limitations of the current benchmarks should be an overriding priority for the function prediction community.

4.4 Methods

4.4.1 Information Content Computation

Information content (IC) is estimated by computing the frequency of a particular GO term in a given database of annotations. The IC that an individual term holds is then computed as

$$\mathrm{ic}_+(t) = -\log_2(\mathbb{P}[t]),$$

where t is a single GO term and $\mathbb{P}[t]$ is the empirical probability of observing said term. The logarithm is taken base 2 by convention, with the units of information as Shannons or bits [Sha48]. Then, the IC of a set of terms $\mathcal{T}$, can be computed as

$$\mathrm{ic}_+(\mathcal{T}) = -\log_2(\mathbb{P}[\mathcal{T}]),$$

where $\mathbb{P}[\mathcal{T}]$ is estimated directly from the annotation matrix (P), taking into account for co-occurrence of the annotations. As IC of positive and negative annotations is not equal, this was computed separately, to account for the OWA. As such, analogously, the "negative" IC of a term (t) and set of terms $(\mathcal{T})$ can be calculated as

$$\mathrm{ic}_{-}(\mathrm{t}) = -\log_2(\mathbb{P}[\neg \mathrm{t}]),$$
$$\mathrm{ic}_{-}(\mathcal{T}) = -\log_2(\mathbb{P}[\neg \mathcal{T}]),$$

however $\mathbb{P}[\neg\mathcal{T}]$, the prior probability of negative associations of the set of terms in $\mathcal{T}$, would be computed directly from the negative annotation matrix (N). Note, proteins were considered annotated if they had at least one annotation in at least one aspect of the GO, lower than the root term, listed in the UniProt-GOA or the derived set of negative annotations.

Denote the sets of terms classified as true positive, false negative and false positive as $\mathsf{TP}, \mathsf{FN}, \mathsf{FP}$, respectively. In the OWA-compliant benchmarking framework, the weighted metric representing each of these is computed by calculating the IC of the terms in each set. For true positive and false negative terms, that is $\mathsf{TP}_{\mathrm{w}} = \mathrm{ic}_{+}(\mathsf{TP})$ and $\mathsf{FN}_{\mathrm{w}} = \mathrm{ic}_{+}(\mathsf{FN})$. For false positives, this is instead calculated as $\mathsf{FP}_{\mathrm{w}} = \mathrm{ic}_{-}(\mathsf{FP})$.

4.4.2 Curating Negative Annotations

Negative annotations were curated using the ancestral annotations from PAINT on PANTHER 13.1 families, provided in personal correspondence on 21st August 2018. At this time, 5,664 PANTHER families contained annotations, for which it was possible to derive at least one extra negative annotations on 2,894. In order not to make too general negative assertions, only GO terms for which the "positive" IC was greater than 5 bits were used. In particular, this was due to some erroneous annotations to more general terms in the GO which were in the process of being removed.

PANTHER families were scanned for instances of proteins where an ancestral node, not in its direct lineage, has been annotated to a particular GO term. That is, proteins which are not covered by a PAINT annotation (positive or negative) for an individual GO term, but other members of its family are.

4.4.3 Comparison Prediction Methods

Two baseline methods were used, the naïve and BLAST predictors used in CAFA 2 [JOC+16]. This was compared against GOtcha [MBB04] and HOGPROP (`DessimozLab` in CAFA 3). The benchmark set of proteins $\mathcal{P}$ was chosen subject to the routines described in Section 4.4.4. All existing knowledge on these proteins was removed from the annotation data provided to the methods. Each predictor outputs in the form (p, t, α), where $p \in \mathcal{P}$ is a protein identifier, t a GO term and $\alpha \in (0, 1]$ the method's confidence in its prediction.

4.4.3.1 Naïve Predictor

The naïve predictor assigns the same (t, α) for all $p \in \mathcal{P}$. The confidence score is the frequency of annotation of the term to annotated proteins in the database. This is computed using only experimentally verified annotations on proteins in UniProtKB/Swiss-Prot [Uni17a; Uni18].

4.4.3.2 BLAST Predictor

For each term, the confidence is defined as the maximum percentage identity to a sequence that has been annotated with this term. Again, only experimentally verified annotations on proteins in UniProtKB/Swiss-Prot [Uni17a; Uni18] were used.

4.4.3.3 GOtcha

GOtcha [MBB04] is a more sophisticated predictor, utilising not only sequence homology but also the structure of the GO whilst combining BLAST hits. Consider a target protein p, GO term t and a set of of sequences associated with said term $\mathcal{S}_t$. Then, first an r-score is computed as $r_t = -\sum_{s \in \mathcal{S}_t} \log(e(p, s))$ where $e(p, s)$ represents the E-value of the alignment between the target sequence p and sequence s. i-scores are then calculated by dividing the r-scores by the score for the root term in the relevant aspect – that is, $i_t = r_t / r_{root}$. GOtcha was included in the assessment of Clark and Radivojac [CR13] as an example of a good predictor, performing better than the baseline methods.

4.4.3.4 HOGPROP

This was submitted to the third CAFA as `DessimozLab` and utilises the hierarchical orthologous groups (HOGs) from the OMA project [AGT+18], with the same algorithm also being applied to predicting potential causal genes in QTL experiments [WDR18]. Two variants are included in this paper – HOGPROP1 uses experimentally derived annotations as well as a sub-set of the electronic annotations deemed to be "trusted" (see [WDR18] for details); HOGPROP2 uses *all* annotations, except for electronic ones which are filtered to only include the "trusted" ones.

A subset of GO annotations (including some electronic annotations {based on [ŠAD12]}) are given a score dependent on their evidence code. These terms, with scores, are then associated with the leaves of the hierarchical structure (genes), before being pushed up and pulled down the hierarchy. The score decays across each edge (fixed rate of 20%), with a penalty when propagating over paralogous relationships of a double decay. Scores are combined at each node (using summation) during the up-propagation, whilst the maximum score is taken during down-propagation.

4.4.4 Benchmarking

Two novel benchmarks were developed: the weighted-only, as well as weighted and balanced. These differ in both the calculation of precision and recall and the manner in which the test set is chosen.

Precision-recall curves were computed for both benchmarks, by varying the confidence cut-off ($\tau \in (0, 1]$) that each method reports in its predictions.

Weighted-Only

The test set includes one pair of proteins per-family, for which it is possible to choose a protein with positive annotations and one with negative annotations. This resulted in 2,292 protein-pairs used for this benchmark.

The protein pairs in the test set were chosen without stipulation on the depth or amount of information that each gene has per-aspect or overall. Weighting is then required to correct

the differences in IC – both *within* and *between* the positive and negative annotation sets. To balance within, the IC of the terms inside the particular gene set (for example, true positives) was used. Then, to balance between the positive and negative sets a normalised measure is computed for each of the gene sets (for example, normalised true positive), normalising by the total IC of the positive or negative example genes. That is, the weighted-normalised measures for computing precision and recall are

$$\widetilde{\mathsf{TP}}_{\mathrm{w}}^{\tau} = \frac{\sum_{p_+} \mathrm{ic}_+(\mathsf{TP}_{p_+}^{\tau})}{\sum_{p_+} \mathrm{ic}_+(A_{p_+}^{+})}, \; \widetilde{\mathsf{FN}}_{\mathrm{w}}^{\tau} = \frac{\sum_{p_+} \mathrm{ic}_+(\mathsf{FN}_{p_+}^{\tau})}{\sum_{p_+} \mathrm{ic}_+(A_{p_+}^{+})} \text{ and } \widetilde{\mathsf{FP}}_{\mathrm{w}}^{\tau} = \frac{\sum_{p_-} \mathrm{ic}_-(\mathsf{FP}_{p_-}^{\tau})}{\sum_{p_-} \mathrm{ic}_-(A_{p_-}^{-})},$$

where $\mathsf{TP}_{p_+}^{\tau}$, $\mathsf{FN}_{p_+}^{\tau}$ are the sets of true positive and false negative GO terms for p_+ and $\mathsf{FP}_{p_-}^{\tau}$ the false positive for p_-, both with confidence cut-off τ. $A_{p_+}^{+}$ is the truth set of positive annotations for p_+, and $A_{p_-}^{-}$ is the truth set of negative annotations for p_-.

In order to compare with a benchmark under the closed world assumption (CWA) (seen in Figure 4.4 and Figure B.1), all proteins chosen for their positive annotations were used. The CWA benchmark presented then corresponds to the weighted precision-recall benchmark in CAFA [JOC+16].

Weighted and Balanced

The test set is chosen such that for each GO term included there is a positive and negative example. For all families, for all GO terms with both positive and negative examples one protein was chosen (at random) to be the positive example and another as the negative. This resulted in 12,613 GO term and associated protein-pairs chosen.

Weighting is still required in order to correct for the difference in IC between positive and negative terms (see Figure 4.3). The following definitions for weighted and normalised true positive, false negative and false positive measures were used:

$$\widetilde{\mathsf{TP}}_{\mathrm{w}}^{\tau} = \frac{\sum_i \mathbb{1}_{\mathsf{TP}}^{\tau}(i) \cdot \mathrm{ic}_+(t_i)}{\sum_i \mathrm{ic}_+(t_i)}, \; \widetilde{\mathsf{FN}}_{\mathrm{w}}^{\tau} = \frac{\sum_i \mathbb{1}_{\mathsf{FN}}^{\tau}(i) \cdot \mathrm{ic}_+(t_i)}{\sum_i \mathrm{ic}_+(t_i)} \text{ and } \widetilde{\mathsf{FP}}_{\mathrm{w}}^{\tau} = \frac{\sum_i \mathbb{1}_{\mathsf{FP}}^{\tau}(i) \cdot \mathrm{ic}_-(t_i)}{\sum_i \mathrm{ic}_-(t_i)},$$

where

$$\mathbb{1}^{\tau}_{TP}(i) = \begin{cases} 1 & \text{if } t_i \in TP^{\tau}_{p_i^+} \\ 0 & \text{if } t_i \notin TP^{\tau}_{p_i^+} \end{cases}$$

and similarly,

$$\mathbb{1}^{\tau}_{FN}(i) = \begin{cases} 1 & \text{if } t_i \in FN^{\tau}_{p_i^+} \\ 0 & \text{if } t_i \notin FN^{\tau}_{p_i^+} \end{cases} \text{ and } \mathbb{1}^{\tau}_{FP}(i) = \begin{cases} 1 & \text{if } t_i \in FP^{\tau}_{p_i^-} \\ 0 & \text{if } t_i \notin FP^{\tau}_{p_i^-} \end{cases}$$

4.4.5 Materials

Here, the versions of each dataset and annotation pipeline are provided:

Datasets

- PAINT ancestral annotations from 21st August 2018 [GLLT11];
- PANTHER families from version 13.1 [MMT12];
- Gene Ontology definition from 1st August 2018 [ABB+00; Gen17; Gen18];
- UniProt-GOA from 16th July 2018 (release 180) [BDH+09];
- OMA hierarchical orthologous groups (HOGs), December 2018 release [AGT+18];
- For comparison methods, UniProtKB/Swiss-Prot release from August 2018 [Uni17a; Uni18]

Software

- BLAST+ version 2.6.0+ [CCA+09];
- MATLAB scripts [Jia18] used in the evaluation of CAFA 2 [JOC+16].

Chapter 5

Fitting Evolutionary Distances to Hierarchical Orthologous Groups

Reconstructing large gene trees remains computationally demanding and prone to errors. For this reason, analyses are typically limited to a few hundred genomes. As a more scalable alternative, hierarchical orthologous groups (HOGs)[1] can be used [SGS+14]. An individual HOG is a group of sets of genes arranged into a hierarchy, dependent on their location in the gene tree. Each one of these sets shares a single common ancestor, but genes can be a member of more than one set [AGGD13; TGG+17]. This enables the comparison of highly diverged and similar species in a consistent manner.

HOGs have been increasingly adopted: this hierarchical approach is used by several orthology databases including OrthoDB [ZTK+17], EggNOG [HCSF+16], HieranoidDB [KRLS17], as well as OMA [AGT+18]. In OMA, for instance, the most recent release contains over 600,000 HOGs with the largest containing over 100,000 members, across more than 2,000 species.

Reconciled gene trees have been used in many studies: for example, they have been used to recognise selective pressures acting on different areas of genomes [WNK+09]; also, it is possible to use them to assign Gene Ontology (GO) functional annotations to proteins such as in SIFTER [EJMB05; EJSB11]. However, the gene phylogeny implied by a HOG does not provide branch lengths, which are necessary in both of these example analyses.

[1] For more details on hierarchical orthologous groups, see Section 1.3.1.2.

5.1 Evolutionary Distances in OMA

Instead of using the sequence similarity as a proxy of evolutionary distance when identifying homologues, Wall, Fraser and Hirsh [WFH03] proposed to use maximum likelihood estimates of the evolutionary distance between sequence pairs. Building on this, Roth, Gonnet and Dessimoz [RGD08] developed the OMA algorithm, showing how statistical uncertainties in the estimation of the distances can be incorporated into the inference strategy.

The first step is to perform alignments between all sequences using a full Smith-Waterman alignment with a fixed PAM matrix (PAM224) in order to identify all homologous sequences. Secondly, significant alignments (those with a score > 85) are refined by searching through all PAM distances for the scoring matrix which maximises the alignment score, which are then used in order to compute the pairwise orthology graph. As such, the algorithm does not generate maximum likelihood estimates for the evolutionary distance between all pairs of sequences in the database. Later, when HOGs are inferred from the pairwise orthologues, some distant homologues may be included in the same top-level group. In the most recent releases a few families contain over 100,000 members, linking many distant homologues, which means that there are many pairwise distances which have not been computed – almost 65% in the June 2019 release.

5.2 Fitting Evolutionary Distances to HOGs

Due to the number and size of the HOGs, a method that utilises the existing pre-computed distances would be well-suited to estimating the branch lengths on the implied gene phylogeny. When pairwise distances between extant genes are known, branch lengths can be fitted using a least-squares method [BW98]. However, current software (for example, *ERaBLE* [BGS+16]) does not permit for the polytomies which the HOG topologies typically contain, whilst also requiring the full pairwise distance matrices between all extant genes.

Polytomies exist in HOGs for two reasons. Firstly, the first is that the HOG construction

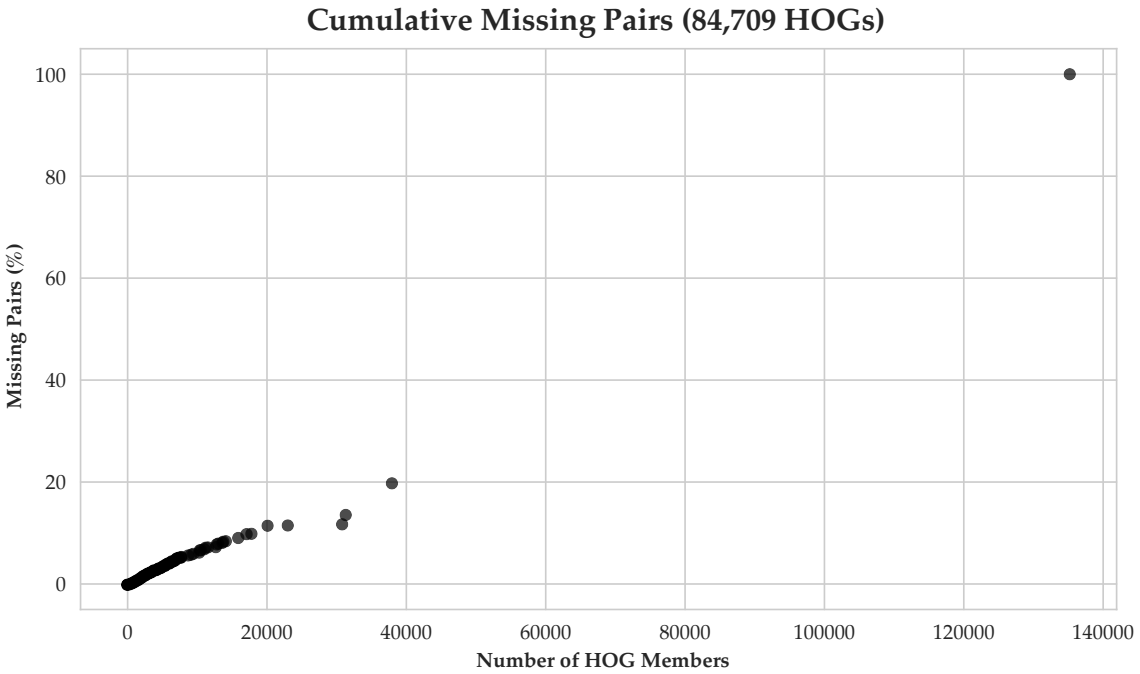


Figure 5.1: Cumulative proportion of missing pairs against size of HOG, sorted by HOG size. Total number of missing pairs 10.68×10^9. Over 80% of missing pairs are the result of the single largest HOG. Together, the three largest are responsible for 88.17% of the missing pairs.

algorithms are unable to determine the order of duplication events. The second being those that result from unresolved sections of the NCBI taxonomy which is used to provide the order of speciation events.

Further, there are more than 16.4×10^9 pairwise distances that would be required to provide a full pairwise distance matrix for every HOG. A large number ($> 10.6 \times 10^{92}$) are filtered out early, as they do not meet the alignment score threshold. Over 80% of missing pairs are the result of the single largest HOG. Together, the three largest are responsible for 88.17% of the missing pairs. It is infeasible to estimate the remaining distances even if the largest three HOGs were removed from any analyses, leaving more than 1.26×10^9 pairs remaining. The all-against-all alignment and maximum likelihood evolutionary distance estimation is the most computationally intensive part of the OMA algorithm, as of the June 2019 release accumulating more than 9 million CPU hours (over 1,035 CPU years) in the last 15 years.

[2] 5,765,402,573 of the 16,452,904,160 required are computed during the all-against-all stage of the OMA algorithm. An additional 10,687,501,587 pairs would need to be computed (64.96% missing).

5.3 Fitting Evolutionary Distances to Trees Containing Polytomies

Let $\mathcal{G}$ be the set of genes in a given family and $n := |\mathcal{G}|$ the number of genes. Evolutionary distances can be calculated between each of these genes and stored in a column vector, denoted $\mathbf{d} \in \mathbb{R}^m$, where $m = \frac{n^2-n}{2}$. For instance, suppose $\mathcal{G} = \{g_1, g_2, g_3, g_4\}$ then $\mathbf{d} = (d_{12}, d_{13}, d_{14}, d_{23}, d_{24}, d_{34})^T$.

A tree T, with the members of $\mathcal{G}$ as the leaves, can be encoded into a binary matrix $A \in \{0, 1\}$, containing the shape of the tree: columns correspond to the branches of T and rows to pairs of genes in $\mathcal{G}$. For example, if branch k is in the path between gene i and j then $A_{ij,k} = 1$.

Then, if $\mathbf{d}$ denotes the vector of gene-to-gene distances and $\mathbf{b}$ the vector of branch lengths, the relationship between them can be defined as a system of linear equations,

$$\mathbf{d} = A\mathbf{b}. \tag{5.1}$$

This is an over-determined system of linear equations, containing inconsistencies due to errors in the encoded topology as well as in the estimates of evolutionary distance. A least-squares approach can be used to find $\mathbf{b}$ which minimises the residual sum of squares. The most simple is to use an ordinary least-squares (OLS) approach,

$$\hat{\mathbf{b}} = \arg\min_{\mathbf{b}} ||(A\mathbf{b} - \mathbf{d})||_2. \tag{5.2}$$

Also, a further constraint is often added. That is, the branch lengths must be strictly greater than, or equal to, zero. So, the problem becomes an optimisation problem – minimise Equation (5.2), subject to $\mathbf{d} \geqslant \mathbf{0}$ – commonly termed the non-negative least-squares (NNLS) problem,

$$\hat{\mathbf{b}} = \arg\min_{\mathbf{b} \geqslant \mathbf{0}} ||(A\mathbf{b} - \mathbf{d})||_2. \tag{5.3}$$

The direct solution of Equation (5.2) is

$$\mathbf{b} = (A^{\mathsf{T}}A)^{-1}\mathbf{d}, \tag{5.4}$$

although this is very expensive to compute due to the large matrix inversion and would not necessarily solve Equation (5.3). It is possible to, instead, reformulate Equation (5.1) as

$$\underbrace{(A^{\mathsf{T}}A)}_{=:H}\mathbf{b} = \overbrace{A^{\mathsf{T}}\mathbf{d}}^{=:\mathbf{f}}, \tag{5.5}$$

which is then of the form $H\mathbf{b} = \mathbf{f}$, where H is symmetric positive definite and a non-negative constraint can quite easily be added to $\mathbf{b}$ in various approaches that exist to solving such optimisation problems.

For instance, Lawson and Hanson [LH95] defined an active set method which solves the Karush-Kuhn-Tucker conditions for the NNLS problem, which is included in the `scipy` Python package. However, the NNLS problem becomes challenging if a large amount of data needs to be processed, which makes the standard methods infeasible. The projected Landweber method was proposed in order to deal with large NNLS problems [JEK+06], however Franc, Hlaváč and Navara [FHN05] showed that a sequential co-ordinate-wise algorithm performed better. This algorithm is an adapted form of the classical Gauss-Seidel method of successive displacement, with a requirement at each step for the elements to be greater than or equal to zero.

The tree topology encoded in A is typically sparse. No implementation existed that enabled the large amount of data and exploited this sparsity existed – especially that could be integrated into the existing Python pipeline.

As such, the algorithm stated by [FHN05] (here termed Non-Negative Gauss-Seidel [NNGS]) was implemented in Python taking full advantage of the sparsity of A. The Numba JIT compiler was also used, which has been shown to approach the speed of C or FORTRAN code [LPS15]. The adapted algorithm for fitting branch lengths is available in Algorithm 5.1.

Algorithm 5.1: Non-Negative Gauss-Seidel for Topology Branch Fitting

Input: A (binary topological matrix) and $\mathbf{d}$ (gene-to-gene distances)

Output: $\hat{\mathbf{b}}$ (branch length estimates)

/* Form SPD matrix and RHS containing constraints */

$H := (A^T A)$;

$\mathbf{f} := A^T \mathbf{d}$;

/* Initialise */

$\mathbf{x}^{(0)} := \mathbf{0}$; /* Initial guess */

$\boldsymbol{\mu}^{(0)} := \mathbf{f}$; /* Initial direction */

$\epsilon := 10^{-6}$; /* Choose convergence criteria */

while *not* converged **do**

  converged := True;

  for $k := 1$ **to** n **do**

    /* Estimate update */

    $x_k^{(t+1)} := \max\left(0, x_k^{(t)} - \frac{\mu_k^{(t)}}{H_{k,k}}\right)$;

    /* Direction update (Note, $H = [\mathbf{h}_1, \cdots, \mathbf{h}]$) */

    $\boldsymbol{\mu}^{(t+1)} := \boldsymbol{\mu}^{(t)} + \left(x_k^{(t+1)} - x_k^{(t)}\right) \mathbf{h}_k$;

    /* Convergence check */

    if $\left(x_k^{(t+1)} - x_k^{(t)}\right) \geqslant \epsilon$ **then** converged := False;

As Franc, Hlaváč and Navara [FHN05] noted, they were unable to prove convergence of this method. However, they found that it converged quickly in practice for their problem in computer vision. This appears to be the case here, as well.

5.4 Fitting with Missing Distances

As not all pairwise distances are computed during the OMA pipeline, given the large number of extra ($> 10 \times 10^9$) alignments required to complete this it is infeasible to do so. The system defined in Equation (5.1) is over-determined. However, the equations are not necessarily consistent due to errors which occur along the pipeline. For instance, errors may occur during the sequencing and assembly of the genomes, as well as in the hierarchical clustering. To maintain the scalability of HOGs in these large analyses, it was necessary to investigate any additional error incurred when fitting with incomplete pairwise distances.

In order to ensure that the matrix A does not become singular when sub-sampling the gene-pairs, it would be advantageous to identify a basis of A. Ideally this would contain pairs for which the distances are already computed, or at least the majority are. However, this is a complex problem and further work is required in order to choose an optimal set of gene pairs.

For now, an (almost) minimal spanning set can be identified. Polytomies in the topology are resolved before identifying a set of $2n - 3$ gene pairs (where n is the number of genes in a given HOG). Extra gene pairs can then be added, in order to add further constraints to the least squares fitting.

5.4.1 Error Analysis

Method

To gauge the error induced by reducing the number of pairs over each branch, a sample of 52 HOGs with no missing pairwise evolutionary gene distances was used. An (almost) minimal spanning set of pairs was chosen for each HOG, as described in the previous section.

Extra pairs were then added at random such that a minimum number of k pairs passes through each branch, before running the non-negative least squares algorithm. This was done for k from 10 to 100 in steps of 10. For each k, the process was performed 10 times.

The error was calculated using the branch lengths resulting from fitting using *all* of the pairwise distances. The normalised root-mean-squared error (RMSE) was computed,

$$\widetilde{\text{RMSE}} = \frac{1}{||\hat{x}||_2}\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{x}_i - \hat{x}_i^{(k)}\right)^2}, \tag{5.6}$$

where $\hat{x}$ denotes the vector of branch lengths from the "full" pairwise distance fitting, with $\hat{x}_i$ and $\hat{x}_i^{(k)}$ being the i-th branch length in the "full" and reduced estimated, respectively.

Results

Figure 5.2 shows the error induced by reducing the number of pairwise distances. All 52 HOGs that were used are included within each bar. It is clear that as the minimum number of pairs per-branch increases, the error decreases. However, this starts to plateau at around $k = 60$ with an error between ~ 0.09 and 0.15, for this dataset. In Figure 5.4, the error is plotted against proportion of pairs used for fitting the branch lengths. This shows that as the proportion of the full constraints increases the error drops. If a user can accept some degree of error and their use case is, instead, interested in the order of the branches, then reducing the number of pairs used during the fitting may be of interest.

The extra pairs, on top of the base reduced pairs, are chosen at random and discarded if they do not increase the required number of pairs on a particular branch. The number of pairs grows linearly with k, as can be seen in Figure 5.5 (confidence interval of 95% displayed). The reduction in the number of constraints also speeds up the iterative solver, as can be seen in Figure 5.3. When the fewest pairs were used this could result in more than 50x speed-up.

5.5 Conclusions

This chapter has shown that evolutionary distances can be fitted to hierarchical orthologous groups. Previously there was no such tool that could fit distances to such large topologies including polytomies. Further, investigating the error induced by reducing the number of pairs used during the least-squares process shows that if a small amount of error is acceptable then the number of pairs in the tree can be decreased.

Fitting evolutionary distances to the hierarchical orthologous groups enables a multitude of new analyses to be performed. In particular, it would be possible to utilise these in the HOGPROP algorithm for functional annotation, proposed in Chapter 3. This could be as simple as being able to discriminate between the different branches after a duplication event, with the assumption that the function would diverge with the observed increase in evolutionary distance. A more complex approach could be implemented, instead, similar

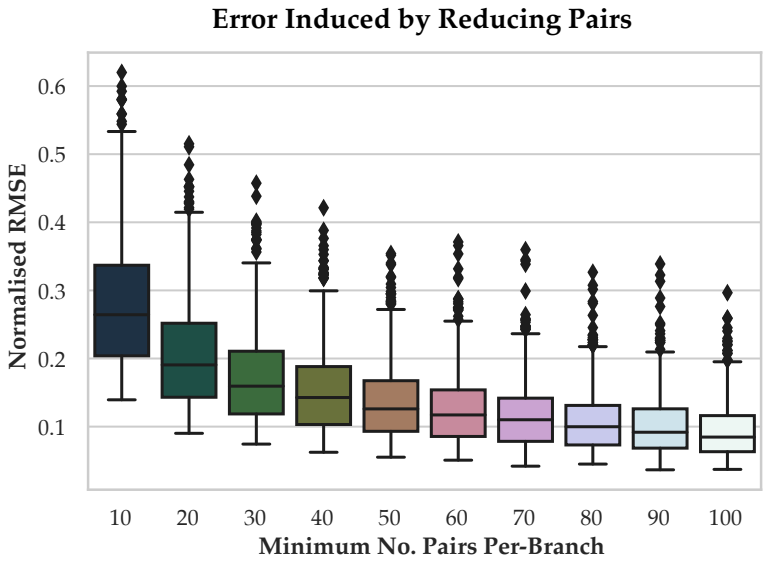


Figure 5.2: Error incurred on branch lengths, on a set of 52 HOGs, when reducing the number of pairwise distances used *vs.* fitting with all of the pairs, measured as normalised RMSE. The number of pairs is varied by choosing at random, such that a minimum number (k) pairs passes through each branch, and performed 10 times to take an average of the error induced with k pairs for a given HOG.

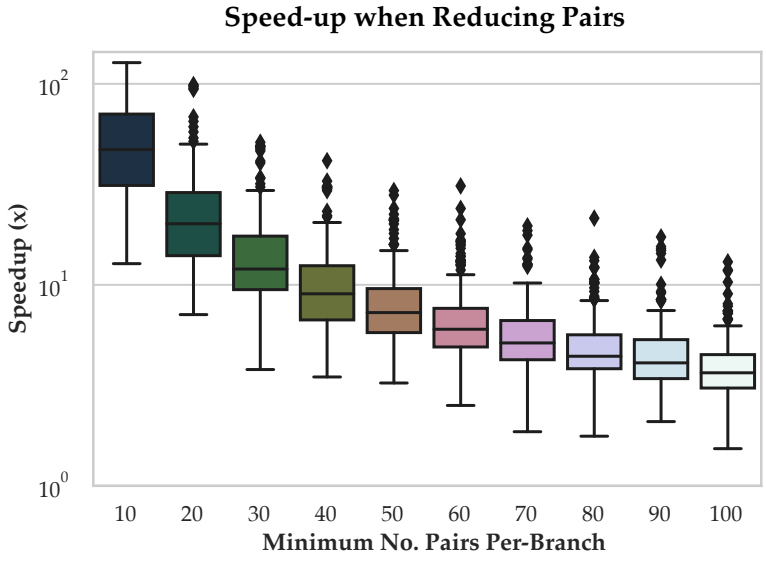


Figure 5.3: Speed-up on a set of 52 HOGs when reducing the number of pairwise distances used in estimating the branch lengths *vs.* fitting with all of the pairs. The number of pairs is varied by choosing at random, such that a minimum number (k) pairs passes through each branch, and performed 10 times to take an average for each HOG.

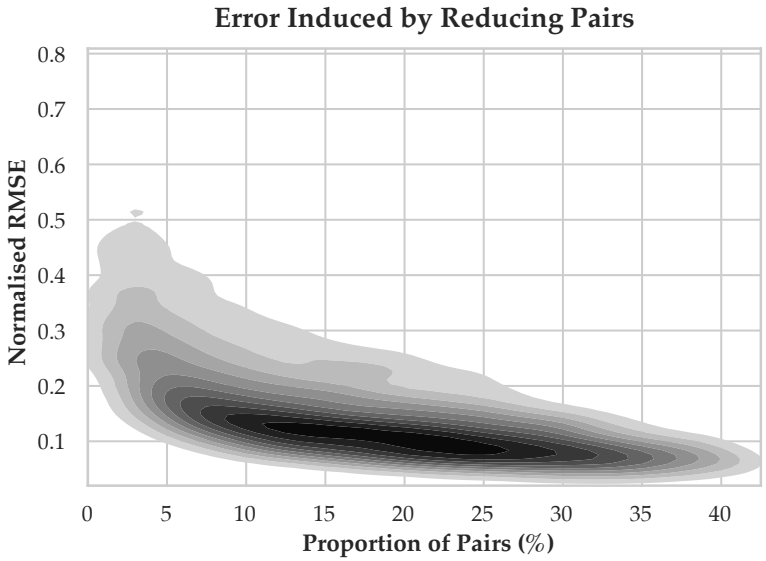


Figure 5.4: Density plot of the error incurred on branch lengths, on a set of 52 HOGs, based on the proportion of pairs used when reducing the number of pairwise distances *vs.* fitting with the all pairs, measured as normalised RMSE. Same data as in Figure 5.2.

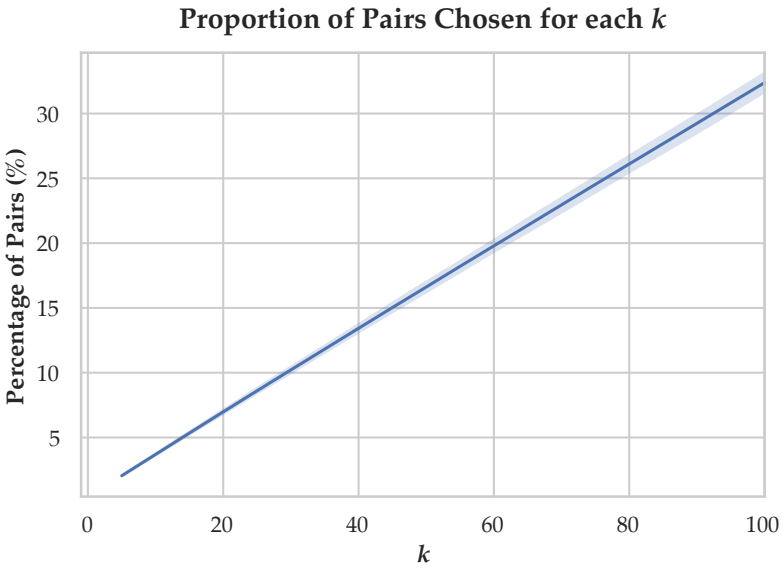


Figure 5.5: Proportion of pairs chosen on each HOG for each sub-sample based against the minimum to choose for each edge (k). The number of pairs grows linearly with k, however the variability is much higher the more pairs per-branch chosen.

to that of SIFTER [EJMB05; EJSB11] – using a belief propagation network model, introduced by Pearl [Pea82].

Part III

Applications

Chapter 6

Ancestral Gene Ontology Enrichment Analyses

The biological interpretation of gene sets which have some common property of interest is typically accomplished by undertaking Gene Ontology (GO) enrichment analyses. For example, if a set of genes were either up- or down-regulated in a particular experiment, a GO enrichment study could indicate likely biological processes or molecular functions involved.

The hierarchical orthologous groups (HOGs) from OMA provide a framework for identifying genes related through evolutionary events. Using the annotations provided by HOGPROP, (introduced in Chapter 3) an enrichment study can be performed on, for example, the genes that were lost over a particular branch in the tree of life.

This chapter starts by describing the methods for undertaking a GO enrichment analysis. Then a case study is presented, performing an ancestral GO enrichment analysis with a dataset from a study on the barn owl (*Tyto alba*).

6.1 Performing Gene Ontology Enrichment Analyses

> *This section includes reference to a book chapter [WD17] written as work towards this thesis, as well as to a co-authored paper presenting the GOATOOLS Python library [KZP+18] to perform GO enrichment analyses.*

One of the most common analyses performed with GO annotations is an enrichment (or depletion) analyses – that is, to find over- and under-represented terms in a given gene-set (*e.g.*, novel genes in a given species) in comparison to a background gene-set (*e.g.*, all genes

in a given species). Fisher's exact test is commonly used to perform these GO enrichment analyses. Multiple testing correction is often required due to the nested nature of GO terms. However, there are alternative methods that take into account the relationship of GO terms.

One such approach is the topology-based algorithms developed by Alexa, Rahnenführer and Lengauer [ARL06]. These are the *elim* and *weight* algorithms, which calculate a score for the term depending on the relevance of child terms. Another method is the *parent-child* approach, which is based on Fisher's exact test. The probability of an individual term being over-represented is instead conditioned on properties of the "parent" terms of the term in question [GBRV07].

Early in this project, a contribution was made to the Gene Ontology Handbook [DŠ17]. This chapter[1] [WD17] provides a tutorial on how to use the GO within the Python programming language.

This tutorial entails querying the GO graph, retrieving annotations, performing gene enrichment analyses and computing basic semantic similarity between GO terms. This work used `GOATOOLS` [KZP+18] – a Python library for GO analyses, which enables enables users to parse the GO, read annotation files and compute over- and under-represented terms using Fisher's exact test (also permitting for multiple correction testing). There are alternative libraries available to perform similar analyses, such as `topGO` for `R`. `topGO` also provides the *parent-child*, *weight*, *elim* and a hybrid method based on *weight* and *elim*.

However, at the time of writing this contribution, the `GOATOOLS` library was in a more basic state than it is today. This lead to contributions to the code base and tutorials of `GOATOOLS`.

6.2 Ancestral Gene Ontology Enrichment Analyses

The work presented in this section shall be included in a forthcoming publication, focussing on the Tyto alba (Barn owl) genome sequenced at the University of Lausanne, Switzerland.

[1] This chapter has been included in Appendix C.

Using the HOGs provided by OMA, the location can be identified of the origination, duplication and loss of genes over a particular branch in the taxonomy. A GO enrichment study can then be performed in order to identify if each of these sets of genes are associated with, for example, gain in a particular biological functions.

6.2.1 Motivation

Tyto alba (barn owl) was sequenced and assembled through a collaboration of research groups at the Department of Ecology and Evolution at the University of Lausanne, Switzerland. After gene prediction, it was deemed interesting to identify the genes that were gained, duplicated and lost and compute any functional enrichment present.

6.2.2 Method

The OMA algorithm (using the OMA standalone tool [ALZ+19]) was run with two alternative novel assemblies of a newly sequenced barn owl (*Tyto alba*), from Europe, as well as a previously sequenced North American individual, from the BGI [ZLL+14]. An array of 11 other avian species and three model species, as an out-group, were also included in the analysis. These species, with the location of *Tyto alba*, can be seen in the phylogeny shown in Figure 6.1. This array of species aids the construction of the HOGs, whilst the inclusion of extra model species aids the propagation of functional annotations in more conserved families.

There are, however, few functional annotations to the existing *Tyto alba* assembly. Any proteins, with annotations, that matched with 100% identity, had their annotations propagated to the novel assemblies. Furthermore, the HOGPROP algorithm outlined in Chapter 3 was used for further functional annotation. This work was undertaken before the extensive benchmarking in Chapter 4, so a conservative approach to the HOGPROP score cut-off was used. A raw-score of higher than 0.7 was taken from the summation method, with a score decay of 15% used. For instance, this would result in prediction if a single sister-protein was annotated with a particular term (with a score of 0.7225). As input annotations, experimental and some filtered[2] electronic annotations were included from the UniProt-GOA

[2]Described in Chapter 3, listed in Appendix A.

database. The April 2017 release of the UniProt-GOA database was used for this analysis, with the GO definition file from the 28th April 2017.

The gene-sets, implied by the HOGs, that represent changes on the branch to the three *Tyto alba* assemblies (highlighted branch in Figure 6.1) were computed – comprising of novel, duplicated, lost and same. These were then used, individually, as foreground sets in GO enrichment / depletion studies. As a background set, the set of all extant genes in the *Tyto alba* was used, except in the case of the lost genes where the set of all extant genes at the next level up in the tree was used (in *Haliaeetus leucocephalus*, *Haliaeetus albicilla* and *Cathartes aura*).

The analyses were performed using the `topGO` package for `R`. The p-values for both the `parentchild` and `weight01` were used, with a cut-off of 0.05 (under both methods) taken to be significant in order to provide high-confidence results.

6.2.3 Results

Full results of the GO enrichment analysis, for each gene-set and aspect of the GO, are available in Appendix D (Tables D.1 to D.12). These tables contain p-values for both the `weight01` and `parentchild` methods from `topGO`, however only `parentchild` p-values are stated here.

There are many enriched terms for each of the gene-sets. For example, phototransduction (`GO:0007602`) is significantly enriched in the lost genes (0.002) and less so in the duplicated gene-set (0.047). Photoreceptor activity (`GO:0009881`) is also significantly enriched in the lost gene-set (0.003), likewise with kidney development (0.018). The duplicated genes are also enriched in the term "olfactory receptor activity" (`GO:0004984`) (< 0.001).

The genes associated with these light-sensing related terms may be interesting to investigate further, as Barn owls are part of a small number of totally nocturnal, non-echolocatory, flying birds [Mar82] (unlike the closest other species included in this analysis).

There is likely further interesting insight from the results of this analysis, for instance many of the gene-sets are enriched in one or more metabolic processes (novel – "meta-

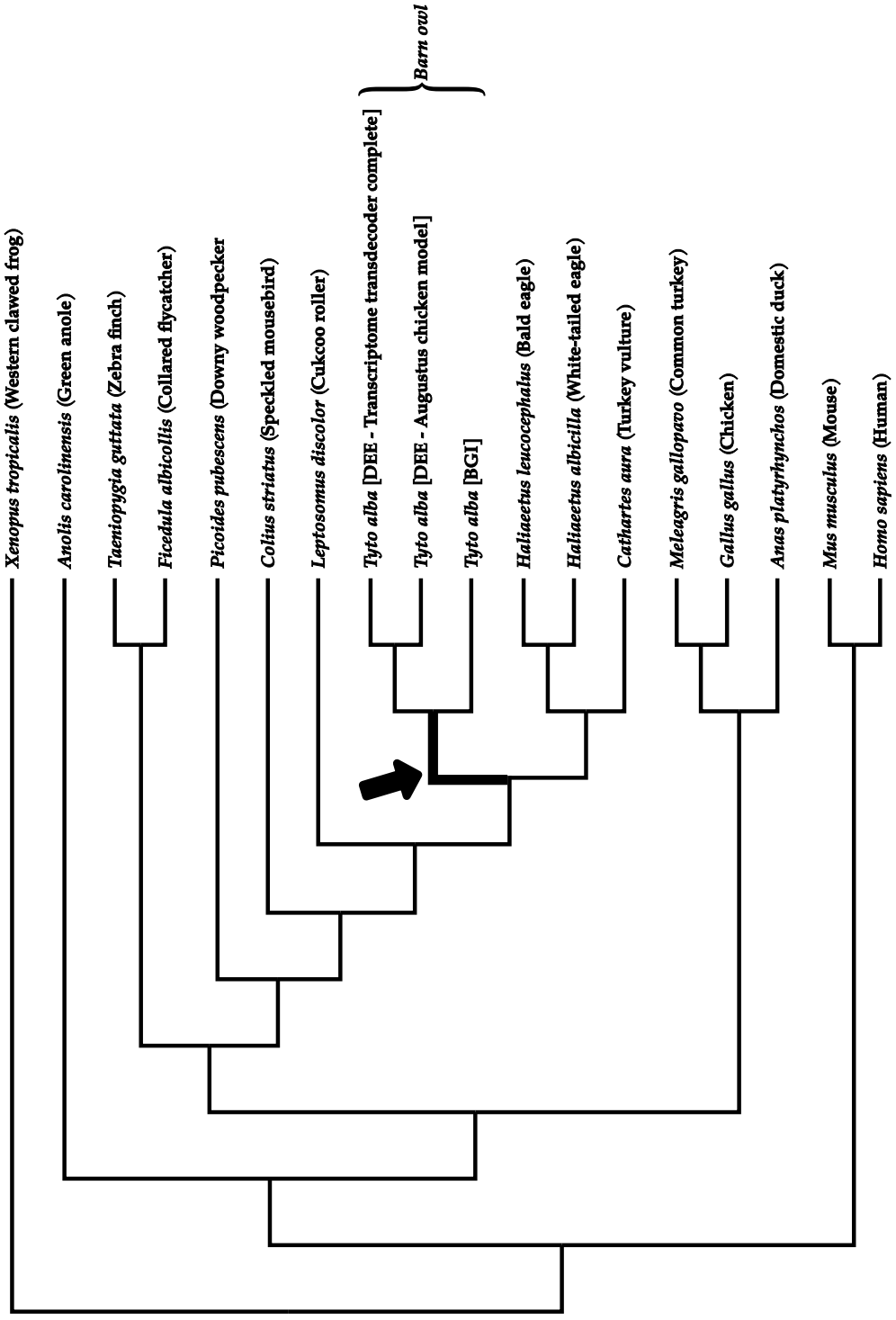


Figure 6.1: Location of the *Tyto alba* (Barn owl) in the 16-species set used in the OMA standalone analysis. Highlighted branch indicates that leading to the three *Tyto alba* assemblies.

bolic process" [GO:008152], "carbohydrate metabolic process" [GO:0005975]; duplicated – "glutamine metabolic process" [GO:0006541]) and the novel gene-set is enriched in the term for autophagy (GO:0006914).

6.2.4 Further Analysis of Lost Gene-Set

Multiple cases of gene fragments being present in the genome, but not in the proteome were identified after this analysis was performed. To provide a more robust analysis, it was important to categorise the witnesses of gene-loss into three categories:

1. Present in the annotation of the genome as a protein-coding gene, as:
   (a) non-fragment;
   (b) fragment.
2. Present in the genome but not in the annotation, as:
   (a) non-fragment;
   (b) fragment.
3. Not found in either.

In order to find those in category one, a single (protein) BLAST database was built using the three *Tyto alba* proteomes uses for the OMA computation. Sequences from each of the other species were then queried against this using blastp from the BLAST+ package [CCA+09]. Hits including sequences of less than 50 amino acids long were filtered out. Significant hits taken as those with E-value $\leqslant 10^{-6}$. In order to classify whether the hit was fragmentary or not, query and database sequence alignment coverage was checked. If the coverage for both query and database sequences was $\geqslant 60\%$, the hit was labelled as non-fragmentary. However, for $< 60\%$ coverage for either query or database sequence and it was classified as a fragment.

When sorting the rest between category two and three, two (DNA) BLAST databases were built for each of the BGI and DEE assembled genomes. The protein sequences of the witnesses of gene loss, from each of the other avian species, were then queried against these two databases using tblastn. No short sequences (< 50 amino acids) would be present, as OMA pre-filters these out. Again, significant hits were taken as those with

Description	Genes	HOGs	Top-Level HOGs
1*a* – Full in proteome	3,988	2,356	2,172
1*b* – Fragment in proteome	1,846	1,375	1,347
2*a* – Full in genome	211	129	125
2*b* – Fragment in genome	106	79	78
3 – Not found in either	67	48	42

Table 6.1: Categorisation of the witnesses of gene loss.

Description	Genes	HOGs	Top-Level HOGs
Full in proteome (diff. best hit)	3,450	1,834	1,650
Full in proteome (same best hit)	538	522	522
Full in proteome (total)	3,988	2,356	2,172

Table 6.2: Sub-categorisation of witnesses of gene loss, present in full in the proteome.

E-value $\leqslant 10^{-6}$. Only query sequence alignment coverage was filtered to decide if hits were fragmentary or not, using the same cut-off of 60%.

The full results of this categorisation can be seen in Table 6.1. Most of the witnesses were placed in category 1*a* and 1*b*. Relatively few (around 5%) were in category two, with only 67 ending up in the third category.

Currently the only certain gene losses are those in category three – not found in either the genome or proteome. However, the first category (containing many of the so-called witnesses of gene loss) underwent further sub-division, in order to classify further probable genuine gene losses. To do this, a reciprocal `BLAST` was undertaken on the protein sequences. That is, all *Tyto alba* protein sequences queried against three databases for each of the other species at the next level (*Haliaeetus leucocephalus*, *Haliaeetus albicilla* and

Description	Genes	HOGs	Top-Level HOGs
Fragment in proteome (diff. best hit)	1,396	927	899
Fragment in proteome (same best hit)	450	448	448
Fragment in proteome (total)	1,846	1,375	1,347

Table 6.3: Sub-categorisation of witnesses of gene loss, present as a fragment in the proteome.

Cathartes aura). If the non-owl sequence matches a gene in *Tyto alba*, but the owl gene has a better match back to the other species, it is an indication that the orthologue was lost in *Tyto alba*. However, careful counting is required – if a HOG has *any* extant sequence with the same best hit to a *Tyto alba* sequence, it is included in the "same best hit" category. Results for those found full in the proteome are shown in Table 6.2 and fragments in the proteome shown in Table 6.3.

The GO enrichment analysis was then re-run with the extant genes that are members of: category one, with different best-hits; category three ("not found in either"). The full results from this analysis can also be found in Appendix D (Tables D.13 to D.15).

After this filtering, there are many more significant terms. Some of the previous terms related to light-sensing are not present. However, "olfactory receptor activity" (`GO:0004984`) is now present (< 0.001). There are now terms such as "post-embryonic development" (`GO:0009791`) and "heart development" (`GO:0007507`) which are very significant (both < 0.001).

6.3 Conclusions

This chapter has introduced ancestral Gene Ontology (GO) enrichment analyses, using the functional predictions from the HOGPROP algorithm. In the case-study, many light-sensing related biological process and molecular function terms were significantly enriched. This is understandable, as the barn owl is a totally nocturnal and non-echolocatory flying bird. This demonstrates that performing a GO enrichment study on sets of genes related through evolutionary events can identify possible drivers of genetic adaptation.

Chapter 7

Prioritising Candidate Genes Causing QTL using Hierarchical Orthologous Groups

The work presented in this chapter was presented in the proceedings track of ECCB 2018 and published in Bioinformatics: Warwick Vesztrocy, Dessimoz and Redestig [*WDR18*]

A KEY GOAL IN PLANT BIOTECHNOLOGY APPLICATIONS is the identification of genes associated to particular phenotypic traits (for example: yield, fruit size, root length). Quantitative Trait Loci (QTL) studies identify genomic regions associated with a trait of interest. However, to infer potential causal genes in these regions, each of which can contain hundreds of genes, this data is usually intersected with prior functional knowledge of the genes. This process is however laborious, particularly if the experiment is performed in a non-model species, and the statistical significance of the inferred candidates is typically unknown.

This chapter introduces QTLSearch, a method and software tool to search for candidate causal genes in QTL studies by combining Gene Ontology annotations across many species, leveraging hierarchical orthologous groups (HOGs). The usefulness of this approach is demonstrated by re-analysing two metabolic QTL studies: one in *Arabidopsis thaliana*, the other in *Oryza sativa* subsp. *indica*. Even after controlling for statistical significance, QTLSearch inferred potential causal genes for more QTL than BLAST-based functional propagation against UniProtKB/Swiss-Prot, and for more QTL than in the original studies.

QTLSearch is distributed under the LGPLv3 license. It is available to install from the Python Package Index (as `qtlsearch`), with the source available from

`https://bitbucket.org/alex-warwickvesztrocy/qtlsearch`.

7.1 Introduction

Identification of variants of genes that are linked to differences in phenotypic traits is a first step in many plant biotechnology applications. By creating mapping populations, characterising and genotyping the individuals of these, it is often possible to find trait-associated regions of chromosomes – so-called Quantitative Trait Loci (QTL). However, a single QTL can typically contain hundreds, if not thousands, of genes. Thus, from a single study, it is rarely straight-forward to pinpoint the causal gene (if there is one at all) and multiple evidence is typically required.

Wide QTL can be broken down by performing additional experiments using higher-resolution genetic maps. A faster complementary approach is to annotate the genes in the target species with known associations to the trait of interest (for example, involvement in relevant pathways or biological processes), and searching for overlap with the genes inside a given QTL [LSM+09; GCG+13; CDT+12; BNSPD14]. This approach has aided the identification of several verified causal genes – for example, the `AT5G50950` fumarase [LMS+08; BRL+11] – demonstrating its merit.

Propagating gene-function annotations across and within species whilst taking evolutionary distance into account, alongside ensuring to control for chance co-occurrence, is difficult. This is particularly the case for non-model species that may have little or no curated annotations available. Currently, there are no dedicated tools to facilitate this analysis, potentially leading important insight to be missed.

This paper presents QTLSearch – a method and tool which aims to recommend genes that are plausible candidates for causing an observed QTL, by identifying the intersection of those associated with a given trait based on an evolutionary analysis and one or more QTL analyses (Figure 7.1). That is, QTLSearch is a method for integrating data from public re-

sources (for example, as Gene Ontology annotations) with the genomic regions identified during a QTL experiment. Gene families, in the form of hierarchical orthologous groups (HOGs) from the Orthologous MAtrix project (OMA) [AGT+18], enable reasoning over complex nested homologies in a consistent framework. By integrating functional inference with homology mapping, it is possible to differentiate the confidence in orthologous and paralogous relationships when propagating functional knowledge.

This method takes existing functional annotations (in an ontology-aware manner). As such, traits measured in QTL experiments need to be mapped to relevant terms. For instance, if the trait of interest was an abundance of the metabolite *Galactose*, this could be mapped to the Gene Ontology (GO) term for "*Galactose bio-synthetic process*" (`GO:0046369`), as well as to the ChEBI term for *Galactose* (`CHEBI:28260`). Existing gene annotations to this GO and ChEBI term would then be mapped to the trait and propagated through HOGs, using the HOGPROP algorithm.

This propagated knowledge is then used to find genes, with an evidence trail, that are located in QTL for a given trait and homologous with another gene, possibly in a different species, that via functional annotations is known to be associated with that same trait.

While QTLSearch is applicable to any type of QTL studies, this chapter shall demonstrate the usefulness of this method, using two metabolic QTL studies in *Oryza sativa* subsp. *indica* from Gong *et al.* [GCG+13] and *Arabidopsis thaliana* from Lisec *et al.* [LSM+09], each reporting several QTL for a large number of metabolite abundances (phenotypic traits). This shows that QTLSearch can find some similar results to those found in the more manual efforts, reported in the original studies. Furthermore, it also provides additional insight which was not reported in those studies.

7.2 Methods

QTLSearch is underpinned by the HOGPROP algorithm, which uses the hierarchical orthologous groups (HOGs) from the Orthologous MAtrix project (OMA) [AGT+18] in order to predict Gene Ontology (GO) terms. The framework has been extended to permit propagation of general gene-labels (traits), resulting in a per-label score for each gene. A

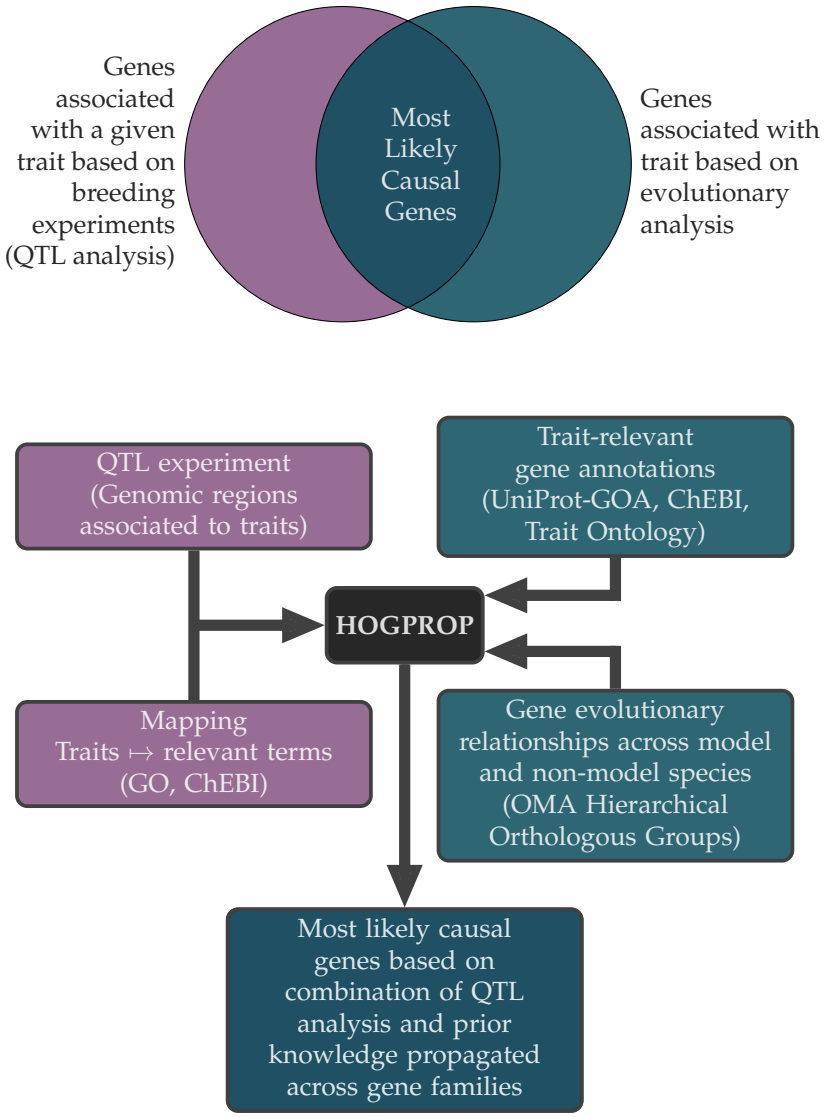


Figure 7.1: Conceptual overview of QTLSearch – to identify the most likely causal genes, by identifying the intersection of genes associated with a given trait based on an evolutionary analysis and QTL analyses.

high-level description of the HOGPROP algorithm is available in Chapter 3, Section 3.1. This section describes the adaptions required to the HOGPROP algorithm in order to implement QTLSearch, before describing the datasets used in this study and the method of comparison.

7.2.1 Required Adaptations to the Original HOGPROP Algorithm

This section will look at each of the adaptations required, in turn, to re-purpose the HOGPROP algorithm to search for trait-associated genes.

7.2.1.1 Scoring

Let a single QTL be defined simply by its co-ordinates. That is, the triple (C, s, e), where s and e denote the start and end positions on the chromosome (C) of the QTL, respectively. If a chromosome is of n genes in length, it shall be denoted as a set of n genes. That is,

$$C := \{g_i : 1 \leqslant i \leqslant n\}.$$

The genes that lie completely, or partially, within a QTL are then defined as

$$\mathcal{Q}^{C}_{(s,e)} := \{g_i : g_i \in C, s < g_i^{end}, e > g_i^{start}\},$$

where g_i^{start} and g_i^{end} are the start and end positions of gene g_i.

Then, let the score associated to a particular gene at time t be denoted as $\mathcal{S}^{t}_{g_i}$. Initially (i.e., at $t = 0$), each gene within a QTL is associated with the trait of interest with a uniform scoring, of

$$\mathcal{S}^{0}_{g_i} := \frac{1}{|\mathcal{Q}|}.$$

Functional annotations can be given as input to the HOGPROP algorithm with varying initial scores. For example, in the case of the UniProt-GOA, experimentally derived annotations are currently initialised with a score of 1.0, whilst "trusted" electronic annotations (based on [ŠAD12], see Table A.1) are given a score of 0.95.

For each QTL, individually, these scores are associated with the genes, at the leaves of the

Metabolite	GO Term	ChEBI Term
Serine	GO:0006564	CHEBI:17822
Glucose	GO:0006094	CHEBI:17234
Inositol	GO:0006021	CHEBI:24848
Fructose	GO:0046370	CHEBI:24848
Galactose	GO:0046369	CHEBI:28260
Glycine	GO:0006545	CHEBI:15428

Table 7.1: The six metabolites and their mapped GO and ChEBI terms used to find the distribution of finding at least one spurious candidate in *Arabidopsis thaliana*.

HOGs. The scores are then propagated up and down the hierarchy, after-which (*i.e.*, at $t = 1$) the observed score increase for each gene in the QTL,

$$\Delta \mathcal{S}_{g_i} = \mathcal{S}^1_{g_i} - \mathcal{S}^0_{g_i} = \mathcal{S}^1_{g_i} - \frac{1}{|\mathcal{Q}|},$$

is stored. This reflects the uniform probability of causal trait-association under the assumption that variation in a single gene is resulting in the observed QTL. This then gives an ordering to the genes in a particular QTL, to which extent they are associated with the trait of interest.

7.2.1.2 Controlling for Significance

A large QTL has a much greater chance to randomly overlap with genes with direct annotations, or have a close homologue with a relevant labelling. The narrower a QTL is, the smaller the chance of a spurious coincidence between a QTL and genes annotated as relevant for a given trait.

In order to illustrate this issue, genes in *Arabidopsis thaliana* (Ensembl Plants 20 / TAIR10) were annotated with association of the abundance of six metabolites (the traits) using annotations to the GO and cross references between UniProtKB and ChEBI terms, listed in Table 7.1. Looking at every possible sliding window, for window sizes varying from just 5 genes up to 2500 genes, the number of times at least one gene is associated with the trait was computed. It shows that for typical QTL lengths, the probability of finding at least one spurious candidate can be substantial (Figure 7.2).

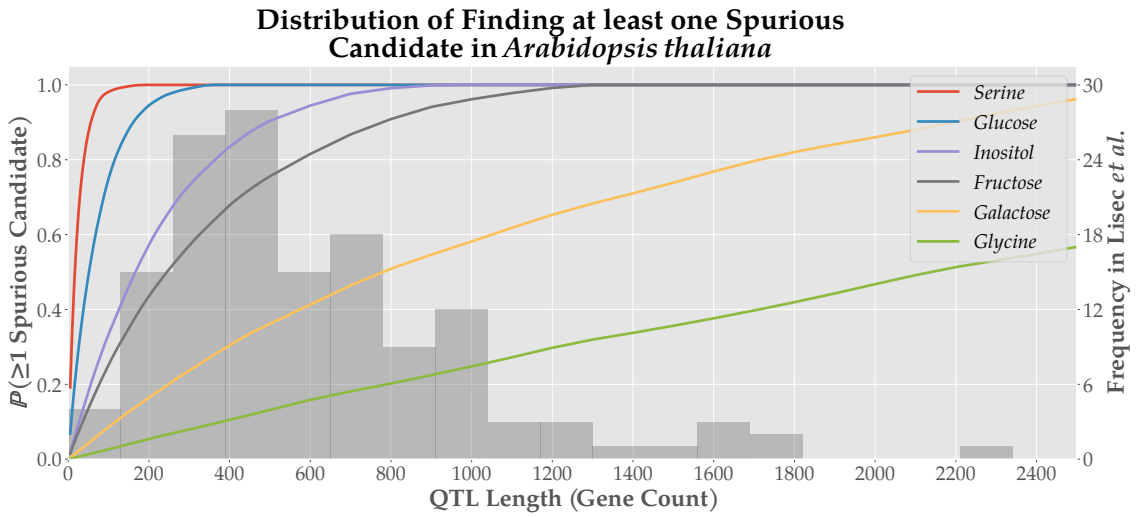


Figure 7.2: Probability of finding at least one spurious candidate in *Arabidopsis thaliana*, for six metabolites, as a function of QTL length (left y-axis). In the background, histogram of the distribution of QTL lengths reported in [LSM+09] (right y-axis).

To account for this, QTLSearch can compute an empirical distribution of score increases per QTL-trait pairing, through the randomisation of the co-ordinates of the QTL. The sampling of the co-ordinates is based on gene-count – both the chromosome and location on the locus are sampled. This feature gives the ability to report empirical p-values, which enable the control of significance. If the p-value estimation is enabled, by default the number of resamples is set to 1,000.

When the aim of the QTL study is to search for candidate genes for a given trait among several QTL, it additionally becomes important to correct for the increase of false positive gene-trait associations. While the distribution of score-increases under the null-hypothesis depends strongly on the distribution and number of trait-associated genes, both of which are fixed, the tests become dependent meaning correction for multiple testing is not straightforward. Leaving the investigation of a more suitable approach for a future study, tests reported here are corrected for falsely reporting at least one significant gene-trait association, *i.e.* the smallest p-value from each QTL, using Benjamini-Hochberg false discovery rate adjusted p-values[1] [BH95]. The unadjusted shall be denoted as p, with those adjusted as p_{BH}.

[1]Benjamini-Hochberg false discovery rate adjusted p-values (p_{BH}) were computed using the `p.adjust` function in the `R` programming environment.

		Author Reported		*Mapped to GO / ChEBI*	
Dataset	Species	Metabolites	QTL	Metabolites	QTL
Lisec *et al.*	*Arabidopsis thaliana*	50	141	35	107
Gong *et al.*	*Oryza sativa* subsp. *indica*	302	1,260	121	638

Table 7.2: Statistics of the number of QTL that could be mapped to GO and / or ChEBI terms from the two datasets in *Arabidopsis thaliana* and *Oryza sativa* subsp. *indica* [LSM+09; GCG+13].

7.2.1.3 Software Package

QTLSearch is implemented as a Python package and is freely distributed under the LGPLv3 license, requiring Python 3.6 or later. It has been published on the Python Package Index (PyPI). Thus, it is installable using `pip` by issuing the command

```
pip install qtlsearch
```

The source code is available from

`https://bitbucket.org/alex-warwickvesztrocy/qtlsearch`.

As the software has been published under an open-source license, it is possible to add extra parsers for alternative data-sources with relative ease.

7.2.2 Datasets

To demonstrate the usefulness of QTLSearch, two datasets from metabolic QTL studies [LSM+09; GCG+13] have been used. The dataset from Lisec *et al.* contains 141 QTL (with full co-ordinates) linked to 50 different metabolites in *Arabidopsis thaliana*, whilst the Gong *et al.* dataset consists of 1,260 QTL linked to the abundance of 302 metabolites in *Oryza sativa* subsp. *indica*. However, co-ordinates (as well as the authors' predictions) were based on *Oryza sativa* subsp. *japonica*.

Hierarchical orthologous groups (HOGs) were taken from the September 2014 release of OMA, so that the MSU version 6 of *Oryza sativa* subsp. *japonica* was included. The UniProt-

GOA [BDH+09] release from February 2018 was used, alongside the Gene Ontology definition from 25th March 2018 [ABB+00; Gen17]. External references from the ChEBI to UniProt entries were taken on 28th March 2018 [HOD+15].

QTLSearch requires a mapping of the GO and ChEBI terms to map to the trait of interest, in this case the relevant metabolites. For initial scores originating from functional annotations in the UniProt-GOA database, initial scores are set at 1.0 for experimentally derived annotations and 0.95 for certain electronic annotations[2] Those arising from a cross-reference to the ChEBI are included with an initial score of 1.0. Genes with multiple sources are given the maximum of the initial scores.

Many of the metabolites measured in the studies could not straight-forwardly be mapped to a GO term, so some were mapped to more general (however, still relevant) terms. ChEBI associations were only included when an exact match to the compound was possible. For the mapping between metabolic traits and GO and / or ChEBI terms used, see Tables E.1 and E.2. Table 7.2 shows the proportion of metabolites and QTL that have been mapped from each of the studies.

7.2.3 Comparison Method – Naïve BLAST

As well as comparing QTLSearch to the candidates that the respective authors reported, a comparison in performance was made to a naïve BLAST method. This takes the protein sequence for every gene inside the QTL and performs a BLAST against the entire UniProtKB/Swiss-Prot database (February 2018 release [Uni17a]), using the NCBI BLAST+ tool [CCA+09] and the GNU Parallel tool in order to exploit parallelism in the search [Tan11].

Candidate genes are predicted, as potentially causal to the abundance of a metabolite, if any of the top 10 hits, with an E-value of below 10^{-6} has a GO annotation (in the UniProt-GOA database [BDH+09] [February 2018 release]) or cross-reference to a relevant ChEBI term, which is included in the mapping of metabolite to GO / ChEBI terms. Other e-value

[2]Electronic annotations (IEA evidence code) are filtered based on [ŠAD12]. See Table A.1 for filtering used.

cut-offs (10^{-3}, 10^{-9}, 10^{-12}) gave similar results in this study. Further, the GO annotations are filtered to the same level as for QTLSearch.

7.3 Results

To illustrate the usefulness of QTLSearch, data from two previous metabolic QTL studies was re-analysed – one in *Arabidopsis thaliana* [LSM+09], the other in *Oryza sativa* subsp. *indica* [GCG+13] – in which candidate causal genes were identified for a subset of the QTL using ad hoc methods. First, aggregate results are presented, before looking at an example from each of these datasets.

7.3.1 Number of Predictions

Lisec *et al.* identified 141 QTL. For 67 of these, they inferred at least one candidate gene. In comparison, QTLSearch was able to identify at least one candidate gene for 76 QTL with $p_{BH} < 0.01$ (85 for $p < 0.01$), and a further 29 QTL when relaxing the significance to $p_{BH} < 0.05$ (20 for $p < 0.05$) – see Figure 7.3. However, the BLAST against UniProtKB/Swiss-Prot identified a candidate gene for 72 QTL. The limiting factor for QTLSearch was the number of metabolites which could be associated to Gene Ontology or ChEBI terms (available for 107 of the 141 QTL).

In the study by Gong *et al.*, 1,260 QTL were identified with the authors inferring at least one candidate gene for 142 QTL. This lower proportion was likely due to the practical difficulties of analysing a much larger set of QTL using a labour-intensive ad hoc approach. By contrast, on this dataset, QTLSearch identified candidate genes for substantially more QTL than the original study (259 with $p_{BH} < 0.01$ [360 for $p < 0.01$] and 518 with $p_{BH} < 0.05$ [same for $p < 0.01$]; Figure 7.3). The naïve BLAST search also performed much better for this dataset (338 QTL), finding candidate genes for a comparable number of QTL as QTLSearch, albeit without control for significance. Again, the limiting factor lies in the number of metabolites that could be associated with GO terms, which capped the number of QTL possible to predict using these methods to 638 out of 1,260.

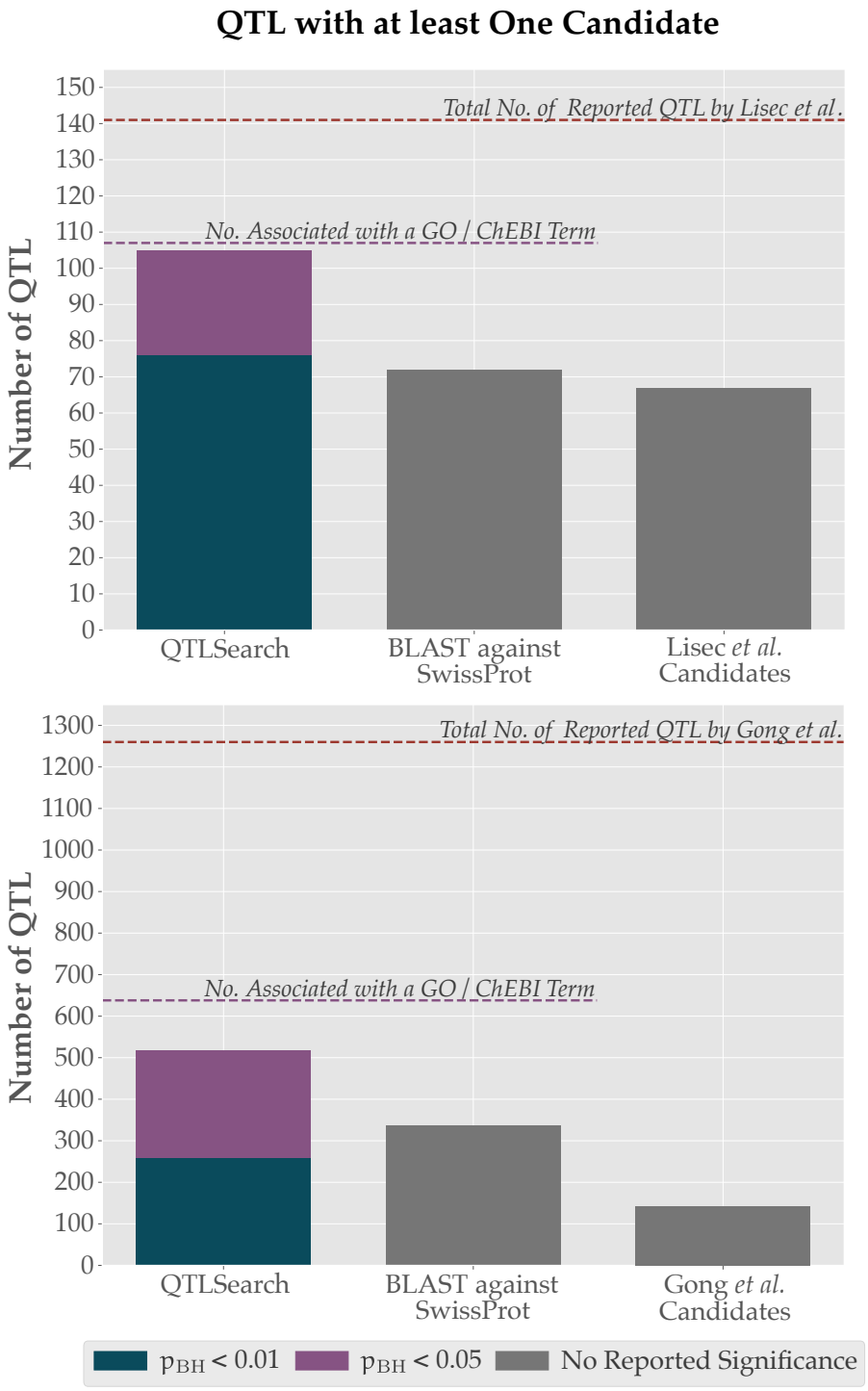


Figure 7.3: Proportion of QTL with at least one candidate from Lisec *et al.* (left) and Gong *et al.* (right) for each method.

7.3.2 Overlap in Predictions with Original Studies

An assessment of the overlap between predictions from the original studies and the two automated approaches was also performed (Figure 7.4). Authors of both studies gave multiple candidates for a subset of the QTL they reported. Here, the overlap is determined based on if a method predicted at least one of these. However, both QTLSearch and the BLAST method may have predicted more candidate genes than this.

When looking at the Lisec *et al.* dataset, both QTLSearch and BLAST find a candidate for the majority of the QTL, with QTLSearch finding a candidate for all when relaxing to the 5% level. BLAST agrees with the authors for half of the QTL. However, there is substantial disagreement in the predicted candidate genes for both methods.

As for the Gong *et al.* dataset, the authors reported either one or two candidates per QTL, with many having two candidates. QTLSearch only finds a candidate for just over half of the QTL which Gong *et al.* gave a prediction, at the 1% level (Figure 7.4). The proportion increases to roughly two thirds at the 5% level. There is also substantial disagreement in the predicted candidate genes. A similar picture emerges when comparing the BLAST results to the original authors' predictions.

7.3.3 Examples

Example from Lisec *et al.*

In the dataset from Lisec *et al.*, there is a QTL associated with the abundance of *Galactose* which is approximately 2.3 Mbp in length, containing just 13 genes. This particular metabolite was associated with both the *"Galactose bio-synthetic process"* (`GO:0046369`) GO term, as well as to the ChEBI term for *Galactose* (`CHEBI:28260`).

There were no predictions for this particular QTL from the authors, however QTLSearch finds two results with $p < 0.01$ – see Table 7.3. The first of these (`ARATH16826`) has a direct annotation in the ChEBI and is also found by the naïve BLAST method described in Section 7.2.3. The second, `ARATH16587`, is not. This OMA identifier maps to the UniProtKB entry `Q9SBA7`, which has a recommended protein name of *"Sugar transport protein*

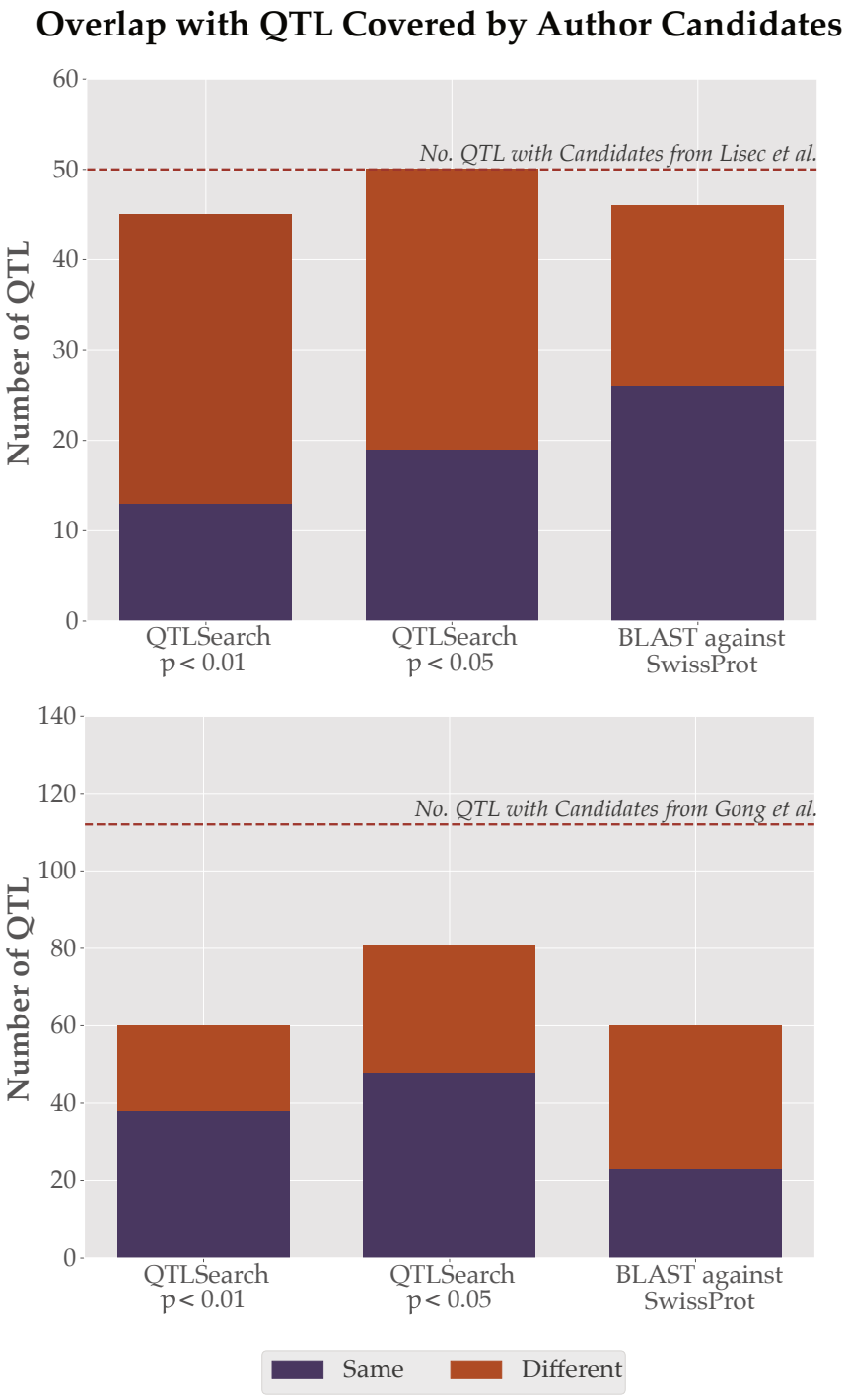


Figure 7.4: Overlap with the candidate genes reported by Lisec *et al.* (left) and Gong *et al.* (right), for QTLSearch (at 1% and 5% significance levels) and the naïve BLAST method.

OMA ID	QTLSearch Increase	QTLSearch p-value	Direct Annotation	Found by BLAST	Author Candidate
ARATH16826	0.996764	0.003126	ChEBI	✔	✘
ARATH16587	0.375134	0.003916	✘	✘	✘

Table 7.3: Table of significantly associated genes for a QTL in the Lisec *et al.* dataset, associated with *Galactose*.

OMA ID	QTLSearch Increase	QTLSearch p-value	Direct Annotation	Found by BLAST	Author Candidate
ORYSJ56351	1.980263	0.000021	✘	✘	✘
ORYSJ56362	1.494041	0.000048	UniProt-GOA	✘	✔
ORYSJ56358	0.638598	0.000260	✘	✔	✘
ORYSJ56359	0.638598	0.000260	✘	✔	✔
ORYSJ56355	0.541781	0.000418	✘	✔	✘

Table 7.4: Significantly associated genes for a QTL in the Gong *et al.* dataset, associated with *Chrysoeriol c-hexoside*.

8" [Uni17b]. Figure 7.5 shows a visualisation of the propagation from ARATH09154, which leads to the increase in score for ARATH16587.

Example from Gong *et al.*

Gong *et al.* associated a region approximately 1.03Mbp in length, containing 146 genes with the abundance of *Chrysoeriol c-hexoside* (a flavanoid). As the GO is not particularly detailed in this area, this was associated with the generic "*Flavonoid biosynthetic process*" (GO:0009813) GO term.

All candidate causal genes, reported by QTLSearch (with $p < 0.01$) are located in the same hierarchical orthologous group (HOG) (HOG:0164195) – see Table 7.4. These are all listed as *"Chalcone and stilbene synthases"* in their relevant UniProtKB entries [Uni17a], which catalyse the first committed step in the flavonoid synthesis pathway [TYSN+07].

Only three of these five were fount by the naïve BLAST method, with only one having a direct annotation in UniProt-GOA.

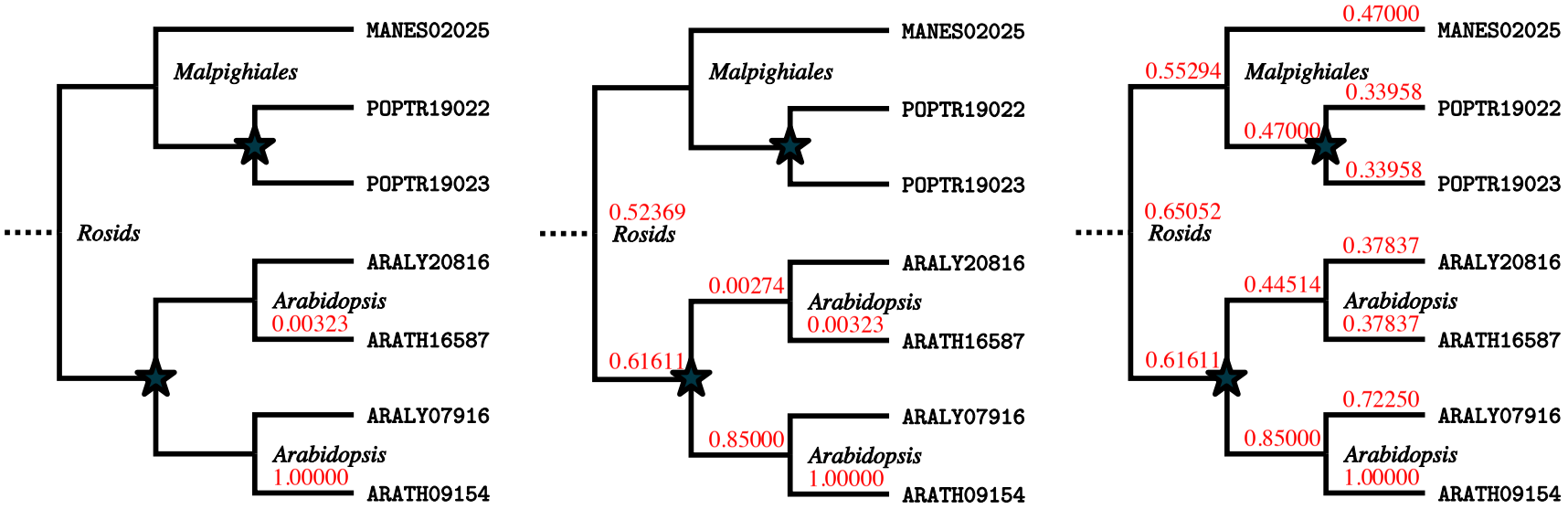


Figure 7.5: Visualisation of the propagation of annotation of `ARATH09154` to `CHEBI:28260` (*Galactose*), which leads to an increase in the score for `ARATH16587`. (Left) before propagation; (Middle) after up-propagation; (Right) After both up-propagation and down-propagation. *Note*: this hierarchical orthologous group (HOG) extends above the level of the *Rosids*.

7.4 Discussion

QTLSearch provides a method for identifying the intersection of genes associated with a given trait based on an evolutionary analysis and QTL analyses. The hierarchical orthologous groups (HOGs) from OMA are at the centre of this, providing a consistent framework to reason over complex nested homologies. Instead of the potentially painstaking manual efforts usually required, QTLSearch provides a prioritised list of candidate genes causing the QTL by integrating annotation data, potentially from many sources.

It is clear that QTLSearch has the ability to predict potentially causal genes for many of the QTL reported in the studies used, especially when accepting at the nominal 5% significance level. Despite this, the naïve BLAST method (described in Section 7.2.3) appears to overlap further with the candidates predicted by Lisec *et al.* However, BLAST is simply a search to the most similar gene, whereas QTLSearch is able to take into account the fine-grained evolutionary history encoded inside the HOGs. When more than one gene is predicted by QTLSearch, this enables the ordering of these based on the evidence trail. Further, the BLAST method does not take into account the probability of homology with genes with a direct annotation, shown in Section 7.2.1.2 to be more of an issue than may be expected.

For both of the datasets, QTLSearch predicts at least one candidate gene for more QTL than the naïve BLAST method. Experimental validation of these would be costly. However, the examples shown in Section 7.3.3 give plausibility to the results.

QTLSearch heavily relies on the existence of functional annotations and a map between these and the metabolites in question. Functional annotations can either be direct annotations to the species in the QTL analysis, or to closely related species. However, if there are no high-quality experimental annotations it is unlikely that either method will give useful results.

When considering the Lisec *et al.* dataset, it rapidly became clear that there were too few Gene Ontology (GO) annotations at an acceptable level of evidence. This motivated the inclusion of the ChEBI as an additional source of information. The mapping performed between ChEBI and the metabolites that was adopted is however keyword-based and thus

quite coarse. For instance, many of the cross-references from ChEBI for *Serine* are likely to be to *serine protein kinase*, which would be irrelevant to the question at hand. Refining the mapping should improve further the performance of QTLSearch for the metabolite QTL use-case. Similarly, it would be possible to extend the framework to include biological pathway information from databases such as Reactome [FJM+18] or KEGG [KFT+17], possibly in an automated manner. However, this inclusion of knowledge from the ChEBI has meant that rather than simply loading GO annotations, the parser has been designed to be modular. Due to the open-source license, this enables easier inclusion of annotations from further sources.

Looking beyond metabolite QTL studies, agronomical or physiological traits, for plants or animals alike, could also be analysed using QTLSearch by generating databases of genes that are associated with traits using, for example, the Trait Ontology (TO) [SMS+12], before searching for co-incidence between QTL and genes homologous to genes in those lists.

Here, just as in the metabolite QTL setting, the use of ontologies is attractive. Instead of manually having to keep track of the relationship between terms, for example, "*kernel size*" and "*fruit size*" or "*branched chain amino acid biosynthesis*" and "*valine biosynthesis*", the ontology provides the necessary "is a" relationships in order to directly use both annotations in an appropriate manner.

Likewise, this framework could also accommodate additional types of data, such as gene expression data. In the context of human genetics, several tools have been recently introduced to integrate expression alongside annotations [ARP+14; WTBP17; SFZ+17]. These frameworks, however, do not naturally extend to other species. For plants, the possibility to include gene expression data is particularly interesting as it provides a straight-forward means to add prior knowledge to the nature of the causal gene(s). For example, via a grafting experiment it may be known that the sought gene is expressed in a given tissue, and are therefore searching for genes in a QTL for given trait *and* annotated to certain biological processes *and* expressed in that tissue.

One limitation of QTLSearch that hampers the use of continuous data such as gene ex-

pression is that the current scoring mechanism in the propagation algorithm is not probabilistic, and as such the confidence values propagated along the hierarchical orthologous groups (HOGs) are not directly interpretable. Adoption of a probabilistic method similar to [EJSB11] is planned. Meanwhile, results from the second CAFA [JOC+16], as well as preliminary results from the third CAFA, have shown that the current scoring method is competitive in the field of GO prediction.

Another limitation lies in the relatively high computational cost of estimating p-values, which is currently implemented as a permutation test. The runtime scales approximately linearly with the number of resamples required (default of 1,000). This means that most of the time is spent on computing the empirical distribution. It may be possible to parameterise this, which would greatly decrease runtime. Meanwhile, it is possible to skip computation of the empirical distribution, which will still result in an ordered list of candidates.

Nevertheless, already in its current form, QTLSearch is a compelling alternative to the ad hoc approaches of typical QTL studies in plants. A fully automated framework also has clear advantages in terms of reproducibility.

Part IV

Conclusions

Chapter 8

Conclusions

Central to this thesis is HOGPROP, the function prediction algorithm introduced in Chapter 3, which propagates functional annotations through the hierarchical orthologous groups (HOGs) of the OMA project. This method was originally designed to predict Gene Ontology terms (GO) for associated protein sequences, with predictions submitted to the third CAFA challenge. Before submission the CAFA metrics, using data from the second CAFA, parameters were optimised – the decay in score through a duplication event (paralogue node), as well as the combination function (for example, summation). It was found that propagating functional annotations from paralogues improved all metrics, but more so when only experimental annotations were used.

When considering the targets for the third CAFA, there were two cases where proteins did not exist in the OMA database. For some species, there is an annotation mis-match between the proteome included in the OMA database and that used to generate the CAFA targets. Also, one species was ambiguously defined (*Escherichia coli* strain K12) and one was not included (*Xenopus laevis*), but a closely related species was (*Xenopus tropicalis*). So as to predict on targets from these species, a projection method was developed and proposed in Chapter 2, consisting of an initial k-mer filtering before refining the order of the matches with Smith-Waterman alignments. This method was also integrated into the back-end of the OMA database, in order to provide fast protein search and GO prediction tools to the front-end user. To showcase how this was integrated, a standalone package to provide the homology search was developed. Benchmarking shows that when mapping a closely related species this is similarly accurate, as well as approximately the same speed, as MMSeqs2 in standard mode.

In Chapter 4, a framework for benchmarking under the open world assumption (OWA)

was introduced, based on a balanced set of positive and negative examples for GO terms. Current benchmarks make an assumption that proteins are fully annotated, by identifying false positives as all the predicted terms which are not confirmed by experimentally backed annotations, resulting in a systematic over-estimation of false positive predictions. As such, this work was not just important to benchmark HOGPROP, but also for the wider community. The framework relies on a large set of negatively qualified annotations. In order to overcome the relative paucity of negative annotations, a substantial number were derived from the expertly curated annotation of gene phylogenies in the PAINT project. The balanced OWA-compliant benchmark provides a balanced test set such that methods are only rewarded for predicting terms that can be disproved. This, alongside the relatively low information content of annotations considered in the benchmark under the closed world assumption, explains why the naïve baseline predictor (based only on term frequency) performs so well in CAFA. It also demonstrates how it is necessary to avoid only using general terms as the results are not merely uninformative, but misleading.

Whilst the propagation model introduced in Chapter 3 is simple, it appears to be effective. Nevertheless, inspired by SIFTER, a future direction would be to implement a probabilistic model based on belief propagation networks. This would, however, require there to be some distance defined between each of the evolutionary events in the gene phylogenies implied by the HOGs. To this end, Chapter 5 presents a method – using a least-squares approach – to fit evolutionary distances to such large phylogenies, possibly containing many polytomies, from the pairwise distances already computed in the OMA algorithm. The method can also make further allowances for the likely missing pairs, which were below the cut-off in the OMA all-against-all alignments. It was shown that when reducing the proportion of pairs used in the fitting process, only a small relative error is incurred.

The subject then turned to applications of the HOGPROP algorithm. The first, presented in Chapter 6, is an ancestral Gene Ontology (GO) enrichment analysis. This demonstrated that an enrichment study could be performed on ancestral genomes – for example, on the genes that were lost over a particular branch in the tree of life. The case study considered the barn owl (*Tyto alba*), showing that many light-sensing related biological process and molecular function GO terms were significantly enriched. This is understandable, as the

barn owl is a totally nocturnal and non-echolocatory flying bird. This demonstrates that performing a GO enrichment study on sets of genes related through evolutionary events can identify possible drivers of genetic adaptation.

Chapter 7 then presented a method (QTLSearch) which demonstrated how HOGs can be used in order to exploit the multitude of functional and trait-associated data. This is an adaptation of the HOGPROP algorithm, in order to provide a tool to integrate many sources of data in order to prioritising candidate genes which caused QTL. This is demonstrated with the re-analysis of two studies, each of which reported several QTL for a large number of metabolite abundances (phenotypic traits) in two different species. QTLSearch found similar results to those found in the more manual efforts, reported in the original studies. This means that it is a compelling alternative to the ad hoc approaches of typical QTL studies in plants. It also provides additional insight which was not reported in those studies, with the fully automated framework also having a clear advantage in its reproducibility.

This thesis has shown how the HOGs from the OMA project can be used as a framework in order to exploit the increasing availability of both genomic and trait-association data. They have been used to predict function, as well as to provide a method in which to integrate many different types of data. In the future, the method developed to estimate branch lengths could be used to extend the propagation method, by implementing a probabilistic model based on belief propagation networks.

The problem of benchmarking under the open world assumption of incomplete knowledge has also been addressed. This required a large source of negative annotations, which were generated from expertly annotated gene phylogenies. In this balanced benchmark, methods are no longer rewarded for predicting terms that can be disproved. This framework is now ready for other sources of negative annotations, or can be used in a time-lapsed study with a steady supply of gene families newly annotated by PAINT or a similar curated approach.

References

[ÅSAL09] Ö. Åkerborg, B. Sennblad, L. Arvestad and J. Lagergren, 'Simultaneous bayesian gene tree reconstruction and reconciliation analysis', *Proceedings of the National Academy of Sciences*, vol. 106, no. 14, pp. 5714–5719, 2009. DOI: 10.1073/pnas.0806251106.

[AJL+02] B. Alberts *et al.*, 'Protein function', in *Molecular Biology of the Cell. 4th edition*, Garland Science, 2002.

[ARL06] A. Alexa, J. Rahnenführer and T. Lengauer, 'Improved scoring of functional groups from gene expression data by decorrelating GO graph structure', *Bioinformatics*, vol. 22, no. 13, pp. 1600–1607, 2006. DOI: 10.1093/bioinformatics/btl140.

[AVP+17] F. Alted *et al.*, *PyTables – Hierarchical datasets in Python*, 2017. [Online]. Available: http://www.pytables.org.

[AD12] A. M. Altenhoff and C. Dessimoz, 'Inferring Orthology and Paralogy', in *Evolutionary Genomics: Statistical and Computational Methods, Volume 1*, M. Anisimova, Ed. Totowa, NJ: Humana Press, 2012, pp. 259–279, ISBN: 978-1-61779-582-4. DOI: 10.1007/978-1-61779-582-4_9.

[ASRRD12] A. M. Altenhoff, R. A. Studer, M. Robinson-Rechavi and C. Dessimoz, 'Resolving the Ortholog Conjecture: Orthologs Tend to Be Weakly, but Significantly, More Similar in Function than Paralogs', *PLoS Computational Biology*, vol. 8, no. 5, pp. 1–10, May 2012. DOI: 10.1371/journal.pcbi.1002514.

[AGGD13] A. M. Altenhoff, M. Gil, G. H. Gonnet and C. Dessimoz, 'Inferring Hierarchical Orthologous Groups from Orthologous Gene Pairs', *PLoS ONE*, vol. 8, no. 1, pp. 1–11, Jan. 2013. DOI: 10.1371/journal.pone.0053786.

[AŠG+15] A. M. Altenhoff *et al.*, 'The OMA orthology database in 2015: function predictions, better plant support, synteny view and other improvements', *Nucleic Acids Research*, vol. 43, no. D1, pp. D240–D249, 2015. DOI: 10.1093/nar/gku1158.

[AB+16] A. M. Altenhoff *et al.*, 'Standardized benchmarking in the quest for orthologs', *Nature Methods*, vol. 13, no. 5, pp. 425–430, Apr. 2016. DOI: 10.1038/nmeth.3830.

[AGT+18] A. M. Altenhoff *et al.*, 'The OMA orthology database in 2018: retrieving evolutionary relationships among all domains of life through richer web

and programmatic interfaces', *Nucleic Acids Research*, vol. 46, no. D1, pp. D477–D485, 2018. DOI: 10.1093/nar/gkx1019.

[ALZ+19] A. M. Altenhoff *et al.*, 'Oma standalone: Orthology inference among public and custom genomes and transcriptomes', *Genome Research*, 2019. DOI: 10.1101/gr.243212.118.

[AMS+97] S. F. Altschul *et al.*, 'Gapped BLAST and PSI-BLAST: a new generation of protein database search programs', *Nucleic Acids Research*, vol. 25, no. 17, pp. 3389–3402, 1997. DOI: 10.1093/nar/25.17.3389.

[AL94] R. Amundson and G. V. Lauder, 'Function without purpose: The uses of causal role function in evolutionary biology', *Biology & Philosophy*, vol. 9, no. 4, pp. 443–469, Oct. 1994. DOI: 10.1007/bf00850375.

[ARP+14] M. Arnold, J. Raffler, A. Pfeufer, K. Suhre and G. Kastenmüller, 'SNiPA: an interactive, genetic variant-centered annotation browser', *Bioinformatics*, vol. 31, no. 8, pp. 1334–1336, 2014. DOI: 10.1093/bioinformatics/btu779.

[ABB+00] M. Ashburner *et al.*, 'Gene Ontology: tool for the unification of biology', *Nature Genetics*, vol. 25, 25 EP –, May 2000. DOI: 10.1038/75556.

[BNSPD14] J. W. Bargsten, J.-P. Nap, G. F. Sanchez-Perez and A. D. van Dijk, 'Prioritization of candidate genes in QTL regions based on associations between traits and biological processes', *BMC plant biology*, vol. 14, no. 1, p. 330, 2014. DOI: 10.1186/s12870-014-0330-3.

[BDH+09] D. Barrell *et al.*, 'The GOA database in 2009 – an integrated Gene Ontology Annotation resource', *Nucleic Acids Research*, vol. 37, pp. D396–D403, 2009. DOI: 10.1093/nar/gkn803.

[BH95] Y. Benjamini and Y. Hochberg, 'Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing', *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 57, no. 1, pp. 289–300, 1995. DOI: 10.2307/2346101.

[BGS+16] M. Binet, O. Gascuel, C. Scornavacca, E. J. P. Douzery and F. Pardi, 'Fast and accurate branch lengths estimation for phylogenomic trees', *BMC Bioinformatics*, vol. 17, no. 1, p. 23, Jan. 2016, ISSN: 1471-2105. DOI: 10.1186/s12859-015-0821-8.

[BSD+13] B. Boussau *et al.*, 'Genome-scale coestimation of species and gene trees', *Genome research*, vol. 23, no. 2, pp. 323–330, 2013. DOI: 10.1101/gr.141978.112.

[BRL+11] Y. Brotman *et al.*, 'Identification of enzymatic and regulatory genes of plant metabolism through QTL analysis in Arabidopsis', *Journal of plant physiology*, vol. 168, no. 12, pp. 1387–1394, 2011. DOI: 10.1016/j.jplph.2011.03.008.

[BW98] D. Bryant and P. Waddell, 'Rapid Evaluation of Least-Squares and Minimum-Evolution Criteria on Phylogenetic Trees', *Molecular Biology and Evolution*, vol. 15, no. 10, pp. 1346–1346, Oct. 1998. DOI: 10.1093/oxfordjournals.molbev.a025863.

[BXH15] B. Buchfink, C. Xie and D. H. Huson, 'Fast and sensitive protein alignment using diamond', *Nat Meth*, vol. 12, no. 1, pp. 59–60, Jan. 2015. DOI: 10.1038/nmeth.3176.

[BKL+12] S. Burge *et al.*, 'Manual GO annotation of predictive protein signatures: the InterPro approach to GO curation', *Database*, vol. 2012, bar068, 2012. DOI: 10.1093/database/bar068.

[CCA+09] C. Camacho *et al.*, 'BLAST+: architecture and applications', *BMC Bioinformatics*, vol. 10, no. 1, p. 421, Dec. 2009. DOI: 10.1186/1471-2105-10-421.

[CDT+12] C. Chen *et al.*, 'PICARA, an Analytical Pipeline Providing Probabilistic Inference about A Priori Candidates Genes Underlying Genome-Wide Association QTL in Plants', *PLoS ONE*, vol. 7, no. 11, e46596, Nov. 2012. DOI: 10.1371/journal.pone.0046596.

[CDFC00] K. Chen, D. Durand and M. Farach-Colton, 'Notung: A program for dating gene duplications and optimizing gene family trees', *Journal of Computational Biology*, vol. 7, no. 3-4, pp. 429–447, 2000. DOI: 10.1089/106652700750050871.

[CR11] W. T. Clark and P. Radivojac, 'Analysis of protein function and its prediction from amino acid sequence', *Proteins: Structure, Function, and Bioinformatics*, vol. 79, no. 7, pp. 2086–2096, Apr. 2011. DOI: 10.1002/prot.23029.

[CR13] ——, 'Information-theoretic evaluation of predicted ontological annotations', *Bioinformatics*, vol. 29, no. 13, pp. i53–i61, Jun. 2013. DOI: 10.1093/bioinformatics/btt228.

[CMV13] B. Contreras-Moreira and P. Vinuesa, 'GET_HOMOLOGUES, a versatile software package for scalable and robust microbial pangenome analysis', *Applied and Environmental Microbiology*, vol. 79, no. 24, pp. 7696–7701, Oct. 2013. DOI: 10.1128/aem.02411-13.

[CJ17] D. Cozzetto and D. T. Jones, 'Computational methods for annotation transfers from sequence', in *The Gene Ontology Handbook*, C. Dessimoz and N. Škunca, Eds. New York, NY: Springer New York, 2017, pp. 55–67. DOI: 10.1007/978-1-4939-3743-1_5.

[CBBJ13] D. Cozzetto, D. W. Buchan, K. Bryson and D. T. Jones, 'Protein function prediction by massive integration of evolutionary analyses and multiple

data sources', in *BMC bioinformatics*, BioMed Central, vol. 14, 2013, S1. DOI: 10.1186/1471-2105-14-S3-S1.

[CMCJ16] D. Cozzetto, F. Minneci, H. Currant and D. T. Jones, 'FFPred 3: feature-based function prediction for all Gene Ontology domains', *Scientific reports*, vol. 6, p. 31 865, 2016.

[Cum75] R. Cummins, 'Functional Analysis', *Journal of Philosophy*, vol. 72, pp. 741–765, 20 Jan. 1975. DOI: 10.2307/2024640.

[DD13] D. A. Dalquen and C. Dessimoz, 'Bidirectional best hits miss many orthologs in duplication-rich clades such as plants and animals', *Genome biology and evolution*, vol. 5, no. 10, pp. 1800–1806, 2013. DOI: 10.1093/gbe/evt132.

[DSL+15] S. Das *et al.*, 'CATH FunFHMMer web server: protein functional annotations using functional family assignments', *Nucleic Acids Research*, vol. 43, no. W1, W148–W153, 2015. DOI: 10.1093/nar/gkv488.

[DLS+15] S. Das *et al.*, 'Functional classification of CATH superfamilies: a domain-based approach for protein function annotation', *Bioinformatics*, vol. 31, no. 21, pp. 3460–3467, 2015. DOI: 10.1093/bioinformatics/btv398.

[DO78] S. R. M. Dayhoff M. O. and B. Orcutt, 'A model of evolutionary change in proteins', in *Atlas of Protein Sequence and Structure*, 3, M. Dayhoff, Ed., vol. 5, Washington DC, USA: Natl. Biomed. Res. Found., 1978, pp. 345–352.

[DDM10] J. I. Deegan, E. C. Dimmer and C. J. Mungall, 'Formalization of taxon-based constraints to detect inconsistencies in annotation and ontology development', *BMC bioinformatics*, vol. 11, no. 1, p. 530, 2010. DOI: 10.1186/1471-2105-11-530.

[DŠ17] C. Dessimoz and N. Škunca, Eds., *The Gene Ontology Handbook*. Springer New York, 2017, ISBN: 978-1-4939-3743-1. DOI: 10.1007/978-1-4939-3743-1.

[DvT13] C. Dessimoz, N. Škunca and P. D. Thomas, 'CAFA and the Open World of protein function predictions', *Trends in genetics: TIG*, vol. 29, no. 11, pp. 609–610, Nov. 2013, ISSN: 0168-9525. DOI: 10.1016/j.tig.2013.09.005.

[DCG+05] C. Dessimoz *et al.*, 'OMA, A Comprehensive, Automated Project for the Identification of Orthologs from Complete Genome Data: Introduction and First Achievements', in *Comparative Genomics*, A. McLysaght and D. H. Huson, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 61–72, ISBN: 978-3-540-31814-9. DOI: 10.1007/11554714_6.

[ESH09] I. Ebersberger, S. Strauss and A. von Haeseler, 'HaMStR: profile hidden markov model based search for orthologs in ESTs', *BMC evolutionary biology*, vol. 9, no. 1, p. 157, 2009. DOI: 10.1186/1471-2148-9-157.

[Edd98] S. R. Eddy, 'Profile hidden Markov models', *Bioinformatics*, vol. 14, no. 9, pp. 755–763, 1998. DOI: 10.1093/bioinformatics/14.9.755.

[Edd11] S. R. Eddy, 'Accelerated profile hmm searches', *PLOS Computational Biology*, vol. 7, no. 10, pp. 1–16, Oct. 2011. DOI: 10.1371/journal.pcbi.1002195.

[Eis98] J. A. Eisen, 'Phylogenomics: improving functional predictions for uncharacterized genes by evolutionary analysis', *Genome research*, vol. 8, no. 3, pp. 163–167, 1998. DOI: 10.1101/gr.8.3.163.

[EK15] D. M. Emms and S. Kelly, 'OrthoFinder: solving fundamental biases in whole genome comparisons dramatically improves orthogroup inference accuracy', *Genome biology*, vol. 16, no. 1, p. 157, 2015. DOI: 10.1186/s13059-015-0721-2.

[EK18] ——, 'OrthoFinder2: fast and accurate phylogenomic orthology analysis from gene sequences', *BioRxiv*, p. 466 201, 2018. DOI: 10.1101/466201.

[EJMB05] B. E. Engelhardt, M. I. Jordan, K. E. Muratore and S. E. Brenner, 'Protein Molecular Function Prediction by Bayesian Phylogenomics', *PLoS Computational Biology*, vol. 1, no. 5, Oct. 2005. DOI: 10.1371/journal.pcbi.0010045.

[EJSB11] B. E. Engelhardt, M. I. Jordan, J. R. Srouji and S. E. Brenner, 'Genome-scale phylogenetic function annotation of large and diverse protein families', *Genome research*, vol. 21, no. 11, pp. 1969–1980, 2011. DOI: 10.1101/gr.104687.109.

[EWLB06] T. A. Eyre, M. W. Wright, M. J. Lush and E. A. Bruford, 'HCOP: A searchable database of human orthology predictions', *Briefings in Bioinformatics*, vol. 8, no. 1, pp. 2–5, May 2006. DOI: 10.1093/bib/bbl030.

[FJM+18] A. Fabregat *et al.*, 'The Reactome Pathway Knowledgebase', *Nucleic Acids Research*, vol. 46, no. D1, pp. D649–D655, 2018. DOI: 10.1093/nar/gkx1132.

[FG13] H. Fang and J. Gough, 'dcGO: database of domain-centric ontologies on functions, phenotypes, diseases and more', *Nucleic Acids Research*, vol. 41, no. D1, pp. D536–D544, 2013. DOI: 10.1093/nar/gks1080.

[FGD19] R. Fernández, T. Gabaldón and C. Dessimoz, 'Orthology: Definitions, inference, and impact on species phylogeny inference', *arXiv preprint arXiv:1903.04530*, 2019.

[FAB+17] R. D. Finn *et al.*, 'InterPro in 2017—beyond protein family and domain annotations', *Nucleic Acids Research*, vol. 45, no. D1, pp. D190–D199, 2017. DOI: 10.1093/nar/gkw1107.

[Fit70] W. M. Fitch, 'Distinguishing homologous from analogous proteins', *Systematic zoology*, vol. 19, no. 2, pp. 99–113, 1970. DOI: 10.2307/2412448.

[FHN05] V. Franc, V. Hlaváč and M. Navara, 'Sequential coordinate-wise algorithm for the non-negative least squares problem', in *Computer Analysis of Images and Patterns*, A. Gagalowicz and W. Philips, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 407–414, ISBN: 978-3-540-32011-1. DOI: 10.1007/11556121_50.

[GK13] T. Gabaldón and E. V. Koonin, 'Functional and evolutionary implications of gene orthology', *Nature Reviews Genetics*, vol. 14, no. 5, pp. 360–366, 2013. DOI: 10.1038/nrg3456.

[GDHJ+09] T. Gabaldón *et al.*, 'Joining forces in the quest for orthologs', *Genome biology*, vol. 10, no. 9, p. 403, 2009. DOI: 10.1186/gb-2009-10-9-403.

[GD17] P. Gaudet and C. Dessimoz, 'Gene ontology: Pitfalls, biases, and remedies', in *The Gene Ontology Handbook*, C. Dessimoz and N. Škunca, Eds. New York, NY: Springer New York, 2017, pp. 189–205. DOI: 10.1007/978-1-4939-3743-1_14.

[GLLT11] P. Gaudet, M. S. Livstone, S. E. Lewis and P. D. Thomas, 'Phylogenetic-based propagation of functional annotations within the Gene Ontology consortium', *Briefings in Bioinformatics*, vol. 12, no. 5, pp. 449–462, Aug. 2011. DOI: 10.1093/bib/bbr042.

[GŠHD17] P. Gaudet, N. Škunca, J. C. Hu and C. Dessimoz, 'Primer on the gene ontology', in *The Gene Ontology Handbook*, C. Dessimoz and N. Škunca, Eds. New York, NY: Springer New York, 2017, pp. 25–37. DOI: 10.1007/978-1-4939-3743-1_3.

[GOC15] Gene Ontology Consortium. (2015). Gene Association File (GAF) Format 2.1.

[Gen17] ——, 'Expansion of the Gene Ontology knowledgebase and resources', *Nucleic Acids Research*, vol. 45, no. D1, pp. D331–D338, 2017. DOI: 10.1093/nar/gkw1108.

[Gen18] ——, 'The gene ontology resource: 20 years and still going strong', *Nucleic acids research*, vol. 47, no. D1, pp. D330–D338, 2018. DOI: 10.1093/nar/gky1055.

[Glo16] N. Glover, *What are hierarchical orthologous groups (hogs)?*, https://www.youtube.com/watch?v=5p5x5gxzhZA, Dec. 2016.

[GCG+13] L. Gong *et al.*, 'Genetic analysis of the metabolome exemplified using a rice population', *Proceedings of the National Academy of Sciences*, vol. 110, no. 50, pp. 20 320–20 325, 2013. DOI: 10.1073/pnas.1319681110.

[GCB92] G. Gonnet, M. Cohen and S. Benner, 'Exhaustive matching of the entire protein sequence database', *Science*, vol. 256, no. 5062, pp. 1443–1445, Jun. 1992. DOI: 10.1126/science.1604319.

[GHKB00] G. H. Gonnet, M. T. Hallett, C. Korostensky and L. Bernardin, 'Darwin v. 2.0: An interpreted computer language for the biosciences', *Bioinformatics*, vol. 16, no. 2, pp. 101–103, Feb. 2000. DOI: 10.1093/bioinformatics/16.2.101.

[Gon94] G. Gonnet, 'A tutorial introduction to computational biochemistry using darwin', Mar. 1994.

[GBYS92] G. H. Gonnet, R. A. Baeza-Yates and T. Snider, 'New indices for text: Pat trees and pat arrays.', *Information Retrieval: Data Structures & Algorithms*, vol. 66, p. 82, 1992.

[GME87] M Gribskov, A. D. McLachlan and D Eisenberg, 'Profile analysis: Detection of distantly related proteins', *Proceedings of the National Academy of Sciences*, vol. 84, no. 13, pp. 4355–4358, 1987. DOI: 10.1073/pnas.84.13.4355.

[GBRV07] S. Grossmann, S. Bauer, P. N. Robinson and M. Vingron, 'Improved Detection of Overrepresentation of Gene-Ontology Annotations with Parent–Child Analysis', *Bioinformatics*, vol. 23, no. 22, pp. 3024–3031, Nov. 2007, ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btm440.

[HOD+15] J. Hastings *et al.*, 'ChEBI in 2016: Improved services and an expanding collection of metabolites', *Nucleic acids research*, vol. 44, no. D1, pp. D1214–D1219, 2015. DOI: 10.1093/nar/gkv1031.

[HH92] S. Henikoff and J. G. Henikoff, 'Amino acid substitution matrices from protein blocks.', *Proceedings of the National Academy of Sciences*, vol. 89, no. 22, pp. 10 915–10 919, Nov. 1992. DOI: 10.1073/pnas.89.22.10915.

[HFV+11] Y. Hu *et al.*, 'An integrative approach to ortholog prediction for disease-focused and other functional studies', *BMC Bioinformatics*, vol. 12, no. 1, p. 357, 2011. DOI: 10.1186/1471-2105-12-357.

[HCDDG07] J. Huerta-Cepas, H. Dopazo, J. Dopazo and T. Gabaldón, 'The human phylome', *Genome biology*, vol. 8, no. 6, R109, 2007. DOI: 10.1186/gb-2007-8-6-r109.

[HCCGP+13] J. Huerta-Cepas, S. Capella-Gutierrez, L. P. Pryszcz, M. Marcet-Houben and T. Gabaldon, 'PhylomeDB v4: zooming into the plurality of evolutionary histories of a genome', *Nucleic acids research*, vol. 42, no. D1, pp. D897–D902, 2013. DOI: 10.1093/nar/gkt1177.

[HCSF+16] J. Huerta-Cepas *et al.*, 'Eggnog 4.5: A hierarchical orthology framework with improved functional annotations for eukaryotic, prokaryotic and viral sequences', *Nucleic Acids Research*, vol. 44, no. D1, pp. D286–D293, 2016. DOI: 10.1093/nar/gkv1248.

[HCFC+17] J. Huerta-Cepas *et al.*, 'Fast genome-wide functional annotation through orthology assignment by eggnog-mapper', *Molecular Biology and Evolution*, vol. 34, no. 8, pp. 2115–2122, 2017. DOI: 10.1093/molbev/msx148.

[HHAF+14] R. P. Huntley *et al.*, 'A method for increasing expressivity of Gene Ontology annotations using a compositional approach', *BMC Bioinformatics*, vol. 15, no. 1, p. 155, May 2014, ISSN: 1471-2105. DOI: 10.1186/1471-2105-15-155.

[HSMM+15] R. P. Huntley *et al.*, 'The GOA database: gene Ontology annotation updates for 2015', *Nucleic acids research*, vol. 43, no. Database issue, pp. D1057–63, Jan. 2015. DOI: 10.1093/nar/gku1113.

[JKZT14] M. P. Jacobson, C. Kalyanaraman, S. Zhao and B. Tian, 'Leveraging structure for enzyme function prediction: Methods, opportunities, and challenges', *Trends in Biochemical Sciences*, vol. 39, no. 8, pp. 363–371, Aug. 2014. DOI: 10.1016/j.tibs.2014.05.006.

[JGSB03] L. J. Jensen, R. Gupta, H.-H. Staerfeldt and S. Brunak, 'Prediction of human protein function according to gene ontology categories', *Bioinformatics*, vol. 19, no. 5, pp. 635–642, Mar. 2003. DOI: 10.1093/bioinformatics/btg036.

[JGB+02] L. Jensen *et al.*, 'Prediction of human protein function from post-translational modifications and localization features', *Journal of Molecular Biology*, vol. 319, no. 5, pp. 1257–1265, Jun. 2002. DOI: 10.1016/s0022-2836(02)00379-0.

[Jia18] Y. Jiang, *PFP - Matlab functions for protein function prediction*, Jan. 2018. [Online]. Available: https://github.com/yuxjiang/pfp.

[JOC+16] Y. Jiang *et al.*, 'An expanded evaluation of protein function prediction methods shows an improvement in accuracy', *Genome biology*, vol. 17, no. 1, p. 184, Sep. 2016. DOI: 10.1186/s13059-016-1037-6.

[JEK+06] B. Johansson *et al.*, 'The application of an oblique-projected landweber method to a model of supervised learning', *Mathematical and Computer Modelling*, vol. 43, no. 7-8, pp. 892–909, 2006. DOI: 10.1016/j.mcm.2005.12.010.

[JTT92] D. T. Jones, W. R. Taylor and J. M. Thornton, 'The rapid generation of mutation data matrices from protein sequences', *Bioinformatics*, vol. 8, no. 3, pp. 275–282, 1992. DOI: 10.1093/bioinformatics/8.3.275.

[JBC+14] P. Jones *et al.*, 'InterProScan 5: genome-scale protein function classification', *Bioinformatics*, vol. 30, no. 9, pp. 1236–1240, 2014. DOI: 10.1093/bioinformatics/btu031.

[KRLS17] M. Kaduk, C. Riegler, O. Lemp and E. L. L. Sonnhammer, 'HieranoiDB: a database of orthologs inferred by Hieranoid', *Nucleic Acids Research*, vol. 45, no. D1, pp. D687–D690, 2017. DOI: 10.1093/nar/gkw923.

[KWAD19] K. Kaleb, A. Warwick Vesztrocy, A. Altenhoff and C. Dessimoz, 'Expanding the orthologous matrix (OMA) programmatic interfaces: REST API and the OmaDB packages for r and python', *F1000Research*, vol. 8, p. 42, Mar. 2019. DOI: 10.12688/f1000research.17548.2.

[KFT+17] M. Kanehisa, M. Furumichi, M. Tanabe, Y. Sato and K. Morishima, 'KEGG: new perspectives on genomes, pathways, diseases and drugs', *Nucleic Acids Research*, vol. 45, no. D1, pp. D353–D361, 2017. DOI: 10.1093/nar/gkw1092.

[KZP+18] D. Klopfenstein *et al.*, 'GOATOOLS: A Python library for Gene Ontology analyses', *Scientific reports*, vol. 8, no. 1, p. 10 872, 2018. DOI: 10.1038/s41598-018-28948-z.

[KBC+17] Ł. Kreft, A. Botzki, F. Coppens, K. Vandepoele and M. Van Bel, 'PhyD3: a phylogenetic tree viewer with extended phyloXML support for functional genomics data visualization', *Bioinformatics*, vol. 33, no. 18, pp. 2946–2947, 2017. DOI: 10.1093/bioinformatics/btx324.

[LPS15] S. K. Lam, A. Pitrou and S. Seibert, 'Numba: A LLVM-based Python JIT compiler', in *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, ACM, 2015, p. 7. DOI: 10.1145/2833157.2833162.

[LH95] C. L. Lawson and R. J. Hanson, *Solving least squares problems*. Siam, 1995, vol. 15, ISBN: 978-0-89871-356-5. DOI: 10.1137/1.9781611971217.

[LG08] S. Q. Le and O. Gascuel, 'An Improved General Amino Acid Replacement Matrix', *Molecular Biology and Evolution*, vol. 25, no. 7, pp. 1307–1320, Mar. 2008, ISSN: 0737-4038. DOI: 10.1093/molbev/msn067.

[LRK+18] H. A. Lewin *et al.*, 'Earth BioGenome Project: Sequencing life for the future of life', *Proceedings of the National Academy of Sciences of the United States of America*, vol. 115, no. 17, pp. 4325–4333, Apr. 2018. DOI: 10.1073/pnas.1720115115.

[LSR03] L. Li, C. J. Stoeckert and D. S. Roos, 'OrthoMCL: identification of ortholog groups for eukaryotic genomes', *Genome research*, vol. 13, no. 9, pp. 2178–2189, 2003. DOI: 10.1101/gr.1224503.

[LAS+15] B. Linard *et al.*, 'OrthoInspector 2.0: Software and database updates', *Bioinformatics*, vol. 31, no. 3, pp. 447–448, 2015. DOI: 10.1093/bioinformatics/btu642.

[LMS+08] J. Lisec *et al.*, 'Identification of metabolic and biomass QTL in Arabidopsis thaliana in a parallel analysis of RIL and IL populations', *The Plant Journal*, vol. 53, no. 6, pp. 960–972, 2008. DOI: 10.1111/j.1365-313X.2007.03383.x.

[LSM+09] J. Lisec *et al.*, 'Identification of heterotic metabolite QTL in Arabidopsis thaliana RIL and IL populations', *The Plant Journal*, vol. 59, no. 5, pp. 777–788, 2009. DOI: 10.1111/j.1365-313X.2009.03910.x.

[LSOJ07] A. Lobley, M. B. Swindells, C. A. Orengo and D. T. Jones, 'Inferring function using patterns of native disorder in proteins', *PLoS Computational Biology*, vol. 3, no. 8, e162, 2007. DOI: 10.1371/journal.pcbi.0030162.

[MDC00] P. Machamer, L. Darden and C. F. Craver, 'Thinking about mechanisms', *Philosophy of science*, vol. 67, no. 1, pp. 1–25, 2000. DOI: 10.1086/392759.

[MH15] M. C. Maher and R. D. Hernandez, 'Rock, paper, scissors: Harnessing complementarity in ortholog detection methods improves comparative genomic inference', *G3 (Bethesda, Md.)*, vol. 5, no. 4, pp. 629–638, Feb. 2015. DOI: 10.1534/g3.115.017095.

[MM93] U. Manber and G. Myers, 'Suffix arrays: A new method for on-line string searches', *SIAM Journal on Computing*, vol. 22, no. 5, pp. 935–948, Oct. 1993. DOI: 10.1137/0222058.

[MBB04] D. M. Martin, M. Berriman and G. J. Barton, 'GOtcha: a new method for prediction of protein function assessed by the annotation of seven genomes', *BMC Bioinformatics*, vol. 5, no. 1, p. 178, Nov. 2004. DOI: 10.1186/1471-2105-5-178.

[Mar82] G. R. Martin, 'An owl's eye: Schematic optics and visual performance instrix aluco l.', *Journal of comparative physiology*, vol. 145, no. 3, pp. 341–349, Sep. 1982, ISSN: 1432-1351. DOI: 10.1007/BF00619338.

[MPM+16] H. Mi, S. Poudel, A. Muruganujan, J. T. Casagrande and P. D. Thomas, 'PANTHER version 10: expanded protein families and functions, and analysis tools', *Nucleic Acids Research*, vol. 44, no. D1, pp. D336–D342, 2016. DOI: 10.1093/nar/gkv1194.

[Mil89] R. G. Millikan, 'In defense of proper functions', *Philosophy of science*, vol. 56, no. 2, pp. 288–302, 1989. DOI: 10.1086/289488.

[MPCJ13] F. Minneci, D. Piovesan, D. Cozzetto and D. T. Jones, 'FFPred 2.0: Improved homology-independent prediction of gene ontology terms for

eukaryotic protein sequences', *PLoS ONE*, vol. 8, no. 5, L. Kurgan, Ed., e63754, May 2013. DOI: 10.1371/journal.pone.0063754.

[Mon74] J. Monod, 'On Chance and Necessity', in *Studies in the Philosophy of Biology*, Springer, 1974, pp. 357–375. DOI: 10.1007/978-1-349-01892-5_20.

[Mor10] Y. Mori, *sais* (*Version 2.4.1*), Aug. 2010. DOI: 10.5281/zenodo.3438081. [Online]. Available: https://sites.google.com/site/yuta256/sais.

[MSB+17] S. Mukherjee *et al.*, 'Genomes OnLine Database (GOLD) v.6: data updates and feature enhancements', *Nucleic Acids Research*, vol. 45, no. D1, pp. D446–D456, 2017. DOI: 10.1093/nar/gkw992.

[MSB+19] S. Mukherjee *et al.*, 'Genomes OnLine database (GOLD) v.7: updates and new features', *Nucleic acids research*, vol. 47, no. D1, pp. D649–D659, Jan. 2019. DOI: 10.1093/nar/gky977.

[MMT12] A. Muruganujan, H. Mi and P. D. Thomas, 'PANTHER in 2013: modeling the evolution of gene function, and other gene attributes, in the context of phylogenetic trees', *Nucleic Acids Research*, vol. 41, no. D1, pp. D377–D386, Nov. 2012, ISSN: 0305-1048. DOI: 10.1093/nar/gks1118.

[MBHC95] A. G. Murzin, S. E. Brenner, T. Hubbard and C. Chothia, 'SCOP: A structural classification of proteins database for the investigation of sequences and structures', *Journal of Molecular Biology*, vol. 247, no. 4, pp. 536–540, 1995. DOI: 10.1016/S0022-2836(05)80134-2.

[Nea91] K. Neander, 'The teleological notion of 'function'', *Australasian Journal of Philosophy*, vol. 69, no. 4, pp. 454–468, 1991. DOI: 10.1080/00048409112344881.

[NW70] S. B. Needleman and C. D. Wunsch, 'A general method applicable to the search for similarities in the amino acid sequence of two proteins', *Journal of Molecular Biology*, vol. 48, no. 3, pp. 443–453, Mar. 1970. DOI: 10.1016/0022-2836(70)90057-4.

[NCRH11] N. L. Nehrt, W. T. Clark, P. Radivojac and M. W. Hahn, 'Testing the ortholog conjecture with comparative functional genomic data from mammals', *PLoS Computational Biology*, vol. 7, no. 6, e1002073, Jun. 2011. DOI: 10.1371/journal.pcbi.1002073.

[NZC11] G. Nong, S. Zhang and W. H. Chan, 'Two Efficient Algorithms for Linear Time Suffix Array Construction', *IEEE Trans. Comput.*, vol. 60, no. 10, pp. 1471–1484, Oct. 2011, ISSN: 0018-9340. DOI: 10.1109/TC.2010.188.

[OSV+15] M. E. Oates *et al.*, 'The SUPERFAMILY 1.75 database in 2014: a doubling of data', *Nucleic Acids Research*, vol. 43, no. D1, pp. D227–D233, 2015. DOI: 10.1093/nar/gku1041.

[OL15] D. Ofer and M. Linial, 'ProFET: Feature engineering captures high-level protein functions', *Bioinformatics*, vol. 31, no. 21, pp. 3429–3436, 2015. DOI: 10.1093/bioinformatics/btv345.

[OMJ+97] C. Orengo *et al.*, 'CATH – a hierarchic classification of protein domain structures', *Structure*, vol. 5, no. 8, pp. 1093–1109, 1997. DOI: 10.1016/S0969-2126(97)00260-8.

[OFD+99] R. Overbeek, M. Fonstein, M. D'souza, G. D. Pusch and N. Maltsev, 'The use of gene clusters to infer functional coupling', *Proceedings of the National Academy of Sciences*, vol. 96, no. 6, pp. 2896–2901, 1999. DOI: 10.1073/pnas.96.6.2896.

[Pea82] J. Pearl, 'Reverend Bayes on Inference Engines: A Distributed Hierarchical Approach', in *Proceedings of the Second AAAI Conference on Artificial Intelligence*, ser. AAAI'82, Pittsburgh, Pennsylvania, USA: AAAI Press, 1982, pp. 133–136.

[Pea13] W. R. Pearson, 'An introduction to sequence similarity ("homology") searching', *Current protocols in bioinformatics*, vol. 42, no. 1, pp. 3–1, 2013. DOI: 10.1002/0471250953.bi0301s42.

[PDL14] C. Pereira, A. Denise and O. Lespinet, 'A meta-approach for improving the prediction and the functional annotation of ortholog groups', *BMC Genomics*, vol. 15, no. Suppl 6, S16, 2014. DOI: 10.1186/1471-2164-15-s6-s16.

[Pes17] C. Pesquita, 'Semantic similarity in the gene ontology', in *The Gene Ontology Handbook*, C. Dessimoz and N. Škunca, Eds. New York, NY: Springer New York, 2017, pp. 161–173, ISBN: 978-1-4939-3743-1. DOI: 10.1007/978-1-4939-3743-1_12.

[PCD+15] D. Petrey *et al.*, 'Template-based prediction of protein function', *Current Opinion in Structural Biology*, vol. 32, pp. 33–38, Jun. 2015. DOI: 10.1016/j.sbi.2015.01.007.

[PHCG10] L. P. Pryszcz, J. Huerta-Cepas and T. Gabaldon, 'Metaphors: Orthology and paralogy predictions from multiple phylogenetic evidence using a consistency-based confidence score', *Nucleic acids research*, vol. 39, no. 5, e32–e32, 2010. DOI: 10.1093/nar/gkq953.

[RCO+13] P. Radivojac *et al.*, 'A large-scale evaluation of computational protein function prediction', *Nature methods*, vol. 10, no. 3, pp. 221–227, Mar. 2013. DOI: 10.1038/nmeth.2340.

[RO09] R. Rentzsch and C. A. Orengo, 'Protein function prediction – the power of multiplicity', *Trends in Biotechnology*, vol. 27, no. 4, pp. 210–219, 2009, ISSN: 0167-7799. DOI: 10.1016/j.tibtech.2009.01.002.

[Ril93] M. Riley, 'Functions of the gene products of Escherichia coli.', *Microbiology and Molecular Biology Reviews*, vol. 57, no. 4, pp. 862–952, 1993.

[RHT00] S. C. G. Rison, T. C. Hodgman and J. M. Thornton, 'Comparison of functional annotation schemes for genomes', *Functional & Integrative Genomics*, vol. 1, no. 1, pp. 56–69, May 2000. DOI: 10.1007/s101420000005.

[RGD08] A. C. Roth, G. H. Gonnet and C. Dessimoz, 'Algorithm of OMA for large-scale orthology inference', *BMC Bioinformatics*, vol. 9, no. 1, p. 518, Dec. 2008, ISSN: 1471-2105. DOI: 10.1186/1471-2105-9-518.

[RZM+04] A. Ruepp *et al.*, 'The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes', *Nucleic acids research*, vol. 32, no. 18, pp. 5539–5545, 2004. DOI: 10.1093/nar/gkh894.

[SGCM07] D. Salgado, G. Gimenez, F. Coulier and C. Marcelle, 'COMPARE, a multi-organism system for cross-species data comparison and transfer of information', *Bioinformatics*, vol. 24, no. 3, pp. 447–449, Dec. 2007. DOI: 10.1093/bioinformatics/btm599.

[SMSS11] T. Schmitt, D. N. Messina, F. Schreiber and E. L. Sonnhammer, 'Letter to the Editor: SeqXML and OrthoXML: standards for sequence and orthology information', *Briefings in Bioinformatics*, vol. 12, no. 5, pp. 485–488, Sep. 2011. DOI: 10.1093/bib/bbr025.

[SRT+13] A. M. Schnoes, D. C. Ream, A. W. Thorman, P. C. Babbitt and I. Friedberg, 'Biases in the experimental annotations of protein function and their effect on our understanding of protein function space', *PLoS computational biology*, vol. 9, no. 5, e1003063, 2013. DOI: 10.1371/journal.pcbi.1003063.

[Sha48] C. E. Shannon, 'A Mathematical Theory of Communication', *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, Jul. 1948, ISSN: 0005-8580. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[SMS+12] R. Shrestha *et al.*, 'Bridging the phenotypic and genetic data useful for integrated breeding through a data annotation using the Crop Ontology developed by the crop communities of practice', *Frontiers in physiology*, vol. 3, p. 326, 2012. DOI: 10.3389/fphys.2012.00326.

[SLC+15] I. Sillitoe *et al.*, 'CATH: comprehensive structural and functional annotations for genome sequences', *Nucleic Acids Research*, vol. 43, no. D1, pp. D376–D381, 2015. DOI: 10.1093/nar/gku947.

[ŠAD12] N. Škunca, A. Altenhoff and C. Dessimoz, 'Quality of computationally inferred gene ontology annotations', *PLoS Computational Biology*, vol. 8, no. 5, e1002533, 2012. DOI: 10.1371/journal.pcbi.1002533.

[ŠRS17] N. Škunca, R. J. Roberts and M. Steffen, 'Evaluating computational gene ontology annotations', in *The Gene Ontology Handbook*, C. Dessimoz and N. Škunca, Eds. New York, NY: Springer New York, 2017, pp. 97–109. DOI: 10.1007/978-1-4939-3743-1_8.

[ŠBK+13] N. Škunca *et al.*, 'Phyletic Profiling with Cliques of Orthologs Is Enhanced by Signatures of Paralogy Relationships', *PLoS Computational Biology*, vol. 9, no. 1, pp. 1–14, Jan. 2013. DOI: 10.1371/journal.pcbi.1002852.

[SW81] T. F. Smith and M. S. Waterman, 'Comparison of biosequences', *Advances in Applied Mathematics*, vol. 2, no. 4, pp. 482–489, Dec. 1981. DOI: 10.1016/0196-8858(81)90046-4.

[SCK+13] W. Snelling *et al.*, 'BREEDING AND GENETICS SYMPOSIUM: Networks and pathways to guide genomic selection', *Journal of Animal Science*, vol. 91, no. 2, pp. 537–552, 2013. DOI: 10.2527/jas.2012-5784.

[SBH10] A. Sokolov and A. Ben-Hur, 'Hierarchical classification of gene ontology terms using the GOstruct method', *Journal of bioinformatics and computational biology*, vol. 8, no. 2, pp. 357–376, 2010. DOI: 10.1142/S0219720010004744.

[SÖ15] E. L. Sonnhammer and G. Östlund, 'InParanoid 8: orthology analysis between 273 proteomes, mostly eukaryotic', *Nucleic Acids Research*, vol. 43, no. D1, pp. D234–D239, 2015. DOI: 10.1093/nar/gku1203.

[SGS+14] E. L. Sonnhammer *et al.*, 'Big data and other challenges in the quest for orthologs', *Bioinformatics*, vol. 30, no. 21, pp. 2993–2998, 2014. DOI: 10.1093/bioinformatics/btu492.

[SHS13] W. W. Soon, M. Hariharan and M. P. Snyder, 'High-throughput sequencing for biology and medicine', *Molecular systems biology*, vol. 9, no. 1, 2013. DOI: 10.1038/msb.2012.61.

[SFZ+17] D. Stacey *et al.*, 'ProGeM: A framework for the prioritisation of candidate causal genes at molecular quantitative trait loci', *bioRxiv*, 2017. DOI: 10.1101/230094.

[SS17a] M. Steinegger and J. Soding, 'Linclust: Clustering protein sequences in linear time', *bioRxiv*, 2017. DOI: 10.1101/104034.

[SS17b] M. Steinegger and J. Söding, 'Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets', *Nature biotechnology*, vol. 35, no. 11, p. 1026, 2017. DOI: 10.1038/nbt.3988.

[TMMT18] H. Tang, C. J. Mungall, H. Mi and P. D. Thomas, 'Gotaxon: Representing the evolution of biological functions in the gene ontology', *arXiv preprint arXiv:1802.06004*, 2018.

[Tan11] O. Tange, 'GNU Parallel - The Command-Line Power Tool', *;login: The USENIX Magazine*, vol. 36, no. 1, pp. 42–47, Feb. 2011. DOI: 10.5281/zenodo.16303.

[TGNK00] R. L. Tatusov, M. Y. Galperin, D. A. Natale and E. V. Koonin, 'The COG database: a tool for genome-scale analysis of protein functions and evolution', *Nucleic acids research*, vol. 28, no. 1, pp. 33–36, 2000. DOI: 10.1093/nar/28.1.33.

[Tho10] P. D. Thomas, 'GIGA: a simple, efficient algorithm for gene tree inference in the genomic age', *BMC bioinformatics*, vol. 11, no. 1, p. 312, 2010. DOI: 10.1186/1471-2105-11-312.

[Tho17] P. D. Thomas, 'The gene ontology and the meaning of biological function', in *The Gene Ontology Handbook*, C. Dessimoz and N. Škunca, Eds. New York, NY: Springer New York, 2017, pp. 15–24. DOI: 10.1007/978-1-4939-3743-1_2.

[TWM+12] P. D. Thomas *et al.*, 'On the Use of Gene Ontology Annotations to Assess Functional Similarity among Orthologs and Paralogs: A Short Report', *PLoS computational biology*, vol. 8, no. 2, e1002386, 2012. DOI: 10.1371/journal.pcbi.1002386.

[TYSN+07] T. Tohge, K. Yonekura-Sakakibara, R. Niida, A. Watanabe-Takahashi and K. Saito, 'Phytochemical genomics in Arabidopsis thaliana: a case study for functional identification of flavonoid biosynthesis genes', *Pure and Applied Chemistry*, vol. 79, no. 4, pp. 811–823, 2007. DOI: 10.1351/pac200779040811.

[TGG+17] C.-M. Train, N. M. Glover, G. H. Gonnet, A. M. Altenhoff and C. Dessimoz, 'Orthologous Matrix (OMA) algorithm 2.0: more robust to asymmetric evolutionary rates and more scalable hierarchical orthologous group inference', *Bioinformatics*, vol. 33, no. 14, pp. i75–i82, 2017. DOI: 10.1093/bioinformatics/btx229.

[TPAD18] C.-M. Train, M. Pignatelli, A. Altenhoff and C. Dessimoz, 'iHam and pyHam: Visualizing and processing hierarchical orthologous groups', *Bioinformatics*, vol. 35, no. 14, R. Schwartz, Ed., pp. 2504–2506, Dec. 2018. DOI: 10.1093/bioinformatics/bty994.

[TMSD09] H. Troy, C. Meghana, L. Stanislav and K. Daisuke, 'PFP: Automated prediction of gene ontology functional annotations with confidence scores using protein sequence data', *Proteins: Structure, Function, and Bioinformatics*, vol. 74, no. 3, pp. 566–582, 2009. DOI: 10.1002/prot.22172.

[Uni17a] UniProt Consortium, 'UniProt: the universal protein knowledgebase', *Nucleic Acids Research*, vol. 45, no. D1, pp. D158–D169, 2017. DOI: 10.1093/nar/gkw1099.

[Uni17b] ——, *UniProtKB – Q9SBA7 (STP8_ARATH)*, https://www.uniprot.org/uniprot/Q9SBA7, Accessed: 13-03-2018, Dec. 2017. [Online]. Available: https://www.uniprot.org/uniprot/Q9SBA7 (visited on 13th Mar. 2018).

[Uni18] ——, 'UniProt: A worldwide hub of protein knowledge', *Nucleic acids research*, vol. 47, no. D1, pp. D506–D515, 2018. DOI: 10.1093/nar/gky1049.

[HSVNH07] R. T. Van der Heijden, B. Snel, V. Van Noort and M. A. Huynen, 'Orthology prediction at scalable resolution by phylogenetic tree analysis', *BMC bioinformatics*, vol. 8, no. 1, p. 83, 2007. DOI: 10.1186/1471-2105-8-83.

[VSUV+09] A. J. Vilella *et al.*, 'EnsemblCompara GeneTrees: Complete, duplication-aware phylogenetic trees in vertebrates', *Genome research*, vol. 19, no. 2, pp. 327–335, 2009. DOI: 10.1101/gr.073585.107.

[WFH03] D. P. Wall, H. B. Fraser and A. E. Hirsh, 'Detecting putative orthologs', *Bioinformatics*, vol. 19, no. 13, pp. 1710–1711, Sep. 2003. DOI: 10.1093/bioinformatics/btg213.

[War17] A. Warwick Vesztrocy, *PySAIS 1.0.7*, Jun. 2017. DOI: 10.5281/zenodo.3437933. [Online]. Available: https://bitbucket.org/alex-warwickvesztrocy/pysais.

[WD17] A. Warwick Vesztrocy and C. Dessimoz, 'A Gene Ontology Tutorial in Python', in *The Gene Ontology Handbook*, C. Dessimoz and N. Škunca, Eds. New York, NY: Springer New York, 2017, pp. 221–229, ISBN: 978-1-4939-3743-1. DOI: 10.1007/978-1-4939-3743-1_16.

[WDR18] A. Warwick Vesztrocy, C. Dessimoz and H. Redestig, 'Prioritising Candidate Genes Causing QTL using Hierarchical Orthologous Groups', *Bioinformatics*, vol. 34, no. 17, pp. i612–i619, 2018. DOI: 10.1093/bioinformatics/bty615.

[WBS12] M. N. Wass, G. Barton and M. J. E. Sternberg, 'CombFunc: Predicting protein function using heterogeneous data sources', *Nucleic Acids Research*, vol. 40, no. W1, W466–W470, May 2012. DOI: 10.1093/nar/gks489.

[WTBP17] K. Watanabe, E. Taskesen, A. Bochoven and D. Posthuma, 'Functional mapping and annotation of genetic associations with FUMA', *Nature communications*, vol. 8, no. 1, p. 1826, 2017. DOI: 10.1038/s41467-017-01261-5.

[Web+92] E. C. Webb *et al.*, *Enzyme nomenclature 1992. Recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the Nomenclature and Classification of Enzymes.* Ed. 6. Academic Press, 1992.

[WG01] S. Whelan and N. Goldman, 'A general empirical model of protein evolution derived from multiple protein families using a maximum-likelihood approach', *Molecular Biology and Evolution*, vol. 18, no. 5, pp. 691–699, May 2001. DOI: 10.1093/oxfordjournals.molbev.a003851.

[WWLB13] M. D. Whiteside, G. L. Winsor, M. R. Laird and F. S. L. Brinkman, 'OrtholugeDB: a bacterial and archaeal orthology resource for improved comparative genomic analysis', *Nucleic Acids Research*, vol. 41, no. D1, pp. D366–D376, 2013. DOI: 10.1093/nar/gks1241.

[WNK+09] Y. I. Wolf, P. S. Novichkov, G. P. Karev, E. V. Koonin and D. J. Lipman, 'The universal distribution of evolutionary rates of genes and distinct characteristics of eukaryotic genes of different apparent ages', *Proceedings of the National Academy of Sciences*, vol. 106, no. 18, pp. 7273–7280, 2009. DOI: 10.1073/pnas.0901808106.

[Wri73] L. Wright, 'Function', *Philosophical Review*, vol. 82, pp. 139–168, 1973.

[WRBK14] Y.-C. Wu, M. D. Rasmussen, M. S. Bansal and M. Kellis, 'Most parsimonious reconciliation in the presence of gene duplication, loss, and deep coalescence using labeled coalescent trees', *Genome research*, vol. 24, no. 3, pp. 475–486, 2014. DOI: 10.1101/gr.161968.113.

[YZX+18] R. You *et al.*, 'GOLabeler: improving sequence-based large-scale protein function prediction by learning to rank', *Bioinformatics*, bty130, 2018. DOI: 10.1093/bioinformatics/bty130.

[ZTK+17] E. M. Zdobnov *et al.*, 'OrthoDB v9.1: cataloging evolutionary and functional annotations for animal, fungal, plant, archaeal, bacterial and viral orthologs', *Nucleic Acids Research*, vol. 45, no. D1, pp. D744–D749, 2017. DOI: 10.1093/nar/gkw1119.

[ZLL+14] G Zhang *et al.*, 'Genomic data of the Barn owl (Tyto alba)', *GigaScience Database*, 2014. DOI: 10.5524/101039.

[ZLGM13] M. Zhao, W.-P. Lee, E. P. Garrison and G. T. Marth, 'SSW Library: An SIMD Smith-Waterman C/C++ Library for Use in Genomic Applications', *PLOS ONE*, vol. 8, no. 12, Dec. 2013. DOI: 10.1371/journal.pone.0082138.

[ZJB+19] N. Zhou *et al.*, 'The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens', *bioRxiv*, 2019. DOI: 10.1101/653105.

[ZE01] C. M. Zmasek and S. R. Eddy, 'A simple algorithm to infer gene duplication and speciation events on a gene tree', *Bioinformatics*, vol. 17, no. 9, pp. 821–828, 2001. DOI: 10.1093/bioinformatics/17.9.821.

Appendices

Appendix A

Gene Ontology Annotation Filtering

This appendix contains a table of the GO evidence codes used in HOGPROP / QTLSearch. IBA / IBD evidence codes (from PAINT) were also used in QTLSearch. Relatively few existed in the release used for the pre-CAFA 3 benchmarking performed in Chapter 3, nevertheless these were included in the testing. They were not, however, included in the training set for the methods included in the benchmarking using negative examples derived from the PAINT annotations, due to circularity.

Table A.1: Filtering of GO evidence codes, based on Škunca, Altenhoff and Dessimoz [ŠAD12] with the addition of IBA / IBD evidence codes. Those listed are included as initial functional knowledge in HOGPROP / QTLSearch.

Evidence Code	Reference Code (if relevant)	Initial Score	Description
EXP		1.0	Inferred from Experimental
IDA		1.0	Inferred from Direct Assay
IPI		1.0	Inferred from Physical Interaction
IMP		1.0	Inferred from Mutant Phenotype
IGI		1.0	Inferred from Genetic Interaction
IEP		1.0	Inferred from Expression Pattern
IBA		0.95	Inferred from Biological aspect of Ancestor
IBD		0.95	Inferred from Biological aspect of Descendant
IEA	2	0.95	Gene Ontology annotation through association of InterPro records with GO terms.
IEA	3	0.95	Gene Ontology annotation based on Enzyme Commission mapping.

Continued on next page

Table A.1 – *continued from previous page*

Evidence Code	Reference Code (if relevant)	Initial Score	Description
IEA	4	0.95	Gene Ontology annotation based on Swiss-Prot keyword mapping.
IEA	23	0.95	Gene Ontology annotation based on Swiss-Prot Subcellular Location vocabulary mapping.
IEA	37	0.95	Gene Ontology annotation based on manual assignment of UniProtKB keywords in UniProtKB/Swiss-Prot entries.
IEA	38	0.95	Gene Ontology annotation based on automatic assignment of UniProtKB keywords in UniProtKB/TrEMBL entries.
IEA	39	0.95	Gene Ontology annotation based on the manual assignment of UniProtKB Subcellular Location terms in UniProtKB/Swiss-Prot entries.
IEA	40	0.95	Gene Ontology annotation based on the automatic assignment of UniProtKB Subcellular Location terms in UniProtKB/TrEMBL entries.
IEA	42	0.95	Gene Ontology annotation through association of InterPro records with GO terms, accompanied by conservative changes to GO terms applied by UniProt.
IEA	45	0.95	Gene Ontology annotation based on UniProtKB/TrEMBL entries keyword mapping, accompanied by conservative changes to GO terms applied by UniProt.
IEA	46	0.95	Gene Ontology annotation based on UniProtKB/TrEMBL Subcellular Location vocabulary mapping, accompanied by conservative changes to GO terms applied by UniProt.

Appendix B

Supplementary Plots for Chapter 4

This appendix contains supplementary plots for Chapter 4 – the results of: the weighted-only benchmark; disabling score normalisation in GOtcha.

B.1 Weighted Only Benchmark (Closed World Assumption)

Figure B.1 contains the results for the weighted-only benchmark, as described in Section 4.4.4.

B.2 GOtcha Score Normalisation

GOtcha [MBB04] performs normalisation to the scores (r-scores), in order to achieve a relative score (i-scores) for each GO term (Section 4.4.3.3). However, this means that scores are not comparable between target proteins. By disabling this normalisation the performance is improved in the weighted and balanced OWA benchmark. However, it still does not perform as well as the BLAST baseline method.

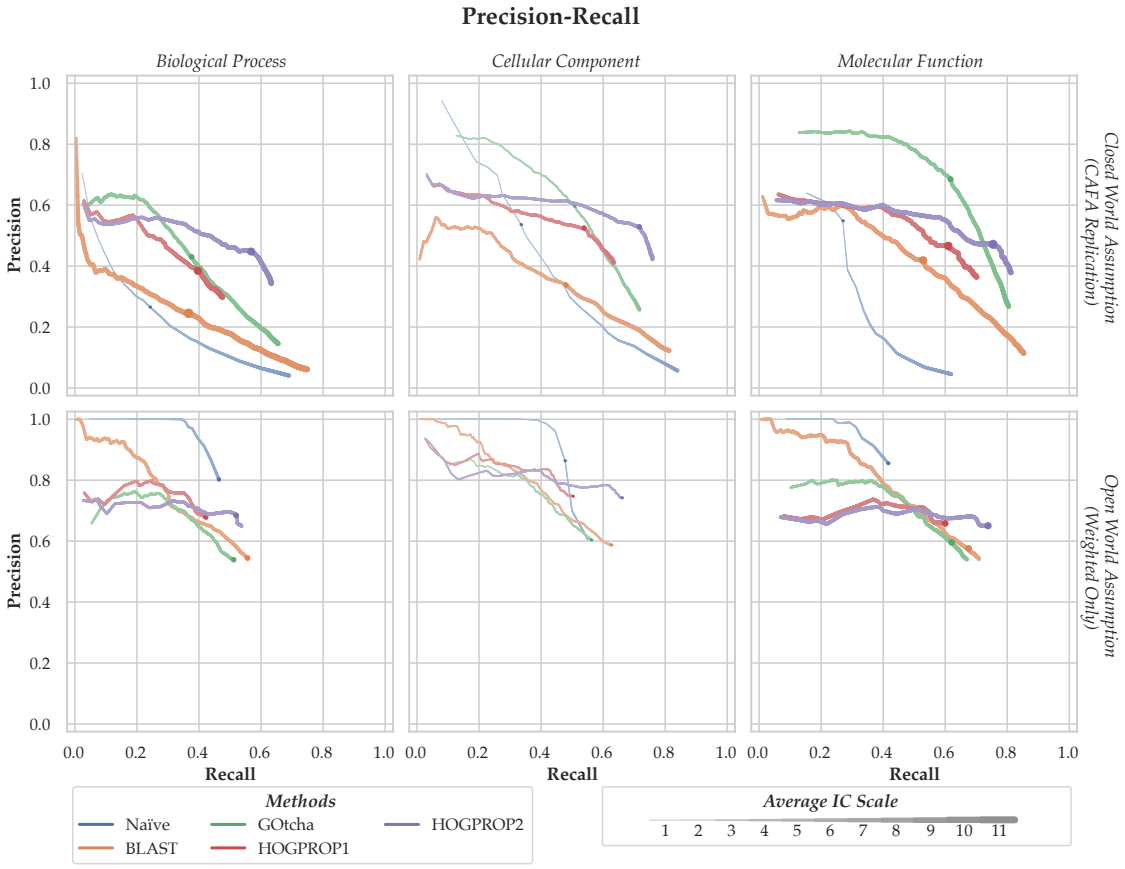


Figure B.1: Precision-recall curves, for each aspect of the GO separately (columns) with the line-width and colour altering based on the average IC of the assessed predictions. (Top) benchmarking under the CWA—identifying false positives using unknown knowledge; (Bottom) weighted OWA-compliant benchmark. The thickness of the curves represents the average IC of the predictions which are used to calculate the precision at that point. The maximum F_1 score (F_{max}) is shown as a point on each curve – values are available in Table 4.2.

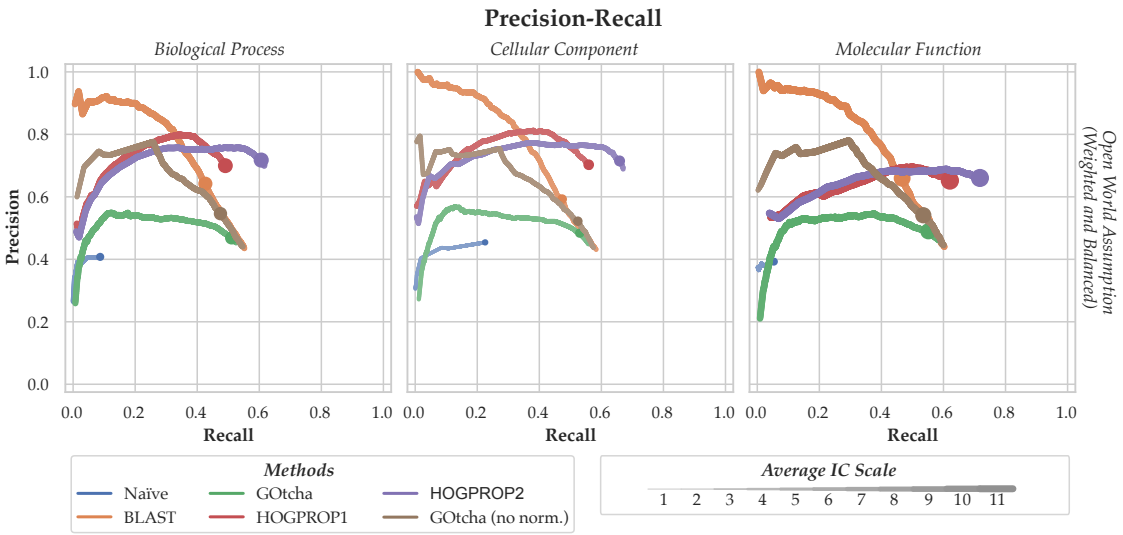


Figure B.2: Precision-recall curves, for each aspect of the GO separately (columns) with the line-width and colour altering based on the average IC of the assessed predictions. Weighted OWA-compliant benchmark, including GOtcha with score normalisation disabled and a log-skew-norm transformation of the scores used instead (as for HOGPROP). The thickness of the curves represents the average IC of the predictions which are used to calculate the precision at that point. The maximum F_1 score (F_{max}) is shown as a point on each curve. GOtcha without normalisation achieved 0.51 on BP ($\tau = 0.60$), 0.52 on CC ($\tau = 0.45$) and 0.54 on MF ($\tau = 0.57$). The values for the other methods are available in Table 4.2.

Appendix C

A Gene Ontology Tutorial in Python

This appendix is a reproduction of the peer-reviewed book chapter [*WD17*].

The following pages contain the Gene Ontology tutorial for the Python programming language [WD17], originally written as a contribution to the gene ontology handbook [DŠ17].

Abstract

This chapter is a tutorial on using Gene Ontology resources in the Python programming language. This entails querying the Gene Ontology graph, retrieving Gene Ontology annotations, performing gene enrichment analyses, and computing basic semantic similarity between GO terms. An interactive version of the tutorial, including solutions, is available at http://gohandbook.org.

Key words Gene Ontology, Tutorial, Python

1 Introduction

One of the main goals of developing a formal ontology is to facilitate computational analysis. The purpose of this chapter is to provide a hands-on introduction to handling GO terms and GO annotations in Python. This tutorial also shows how Python can be used to perform GO term enrichment analyses, as well as how to compute the similarity between GO terms.

This tutorial uses Python, but other popular languages commonly used to perform GO analyses include Java, R, Perl, and Matlab. The Gene Ontology consortium website maintains a list of software libraries, accessible from

ftp://ftp.geneontology.org/pub/go/www/GO.tools_by_type.software.shtml

An interactive version of this tutorial, with model solutions to all the questions, is available from the book homepage at http://gohandbook.org.

2 Querying the Gene Ontology

A fundamental first step is to retrieve the Gene Ontology and analyse that structure (Chap. 3 [1]).

One convenient Python package available to query the GO is GOATOOLS [2]. This package can read the GO structure stored in OBO format, which is available from the GO website (see Chap. 11 [3]). After loading this file, it is possible to traverse the GO structure, search for particular GO terms, and find out which other terms they are related to and how.

This package is available on the Python Package Index (PyPI), a standard repository of python libraries. As such, it is possible to install it locally using the command[1]:

```
pip install goatools
```

The GOATOOLS package contains the functions necessary to parse the GO in OBO format, to query it, and to visualise the ontology. Using the function `obo_parser.GODag()` from GOATOOLS, the GO file can be loaded. Each GO term in the resulting object is an instance of the `GOTerm` class, which contains many useful attributes, such as:

- `GOTerm.name`: textual definition;
- `GOTerm.namespace`: the ontology the term belongs to (i.e., Molecular Function [MF], Biological Process [BP], or Cellular Component [CC]);
- `GOTerm.parents`: list of parent terms;
- `GOTerm.children`: list of children terms;
- `GOTerm.level`: shortest distance to the root node;

Exercise 2.1
Download the GO basic file in OBO format (`go-basic.obo`), and load the GO using the function `obo_parser.GODag()` from GOATOOLS. Using this library, answer the following questions:

(a) What is the name of the GO term `GO:0048527`?

(b) What are the immediate parent(s) of the term `GO:0048527`?

(c) What are the immediate children of the term `GO:0048527`?

(d) Recursively find all the parent and child terms of the term `GO:0048527`. *Hint*: use your solutions to the previous two questions, with a recursive loop.

(e) How many GO terms have the word "*growth*" in their name?

(f) What is the deepest common ancestor term of `GO:0048527` and `GO:0097178`?

(g) Which GO terms regulate `GO:0007124` (pseudohyphal growth)? *Hint*: load the relationship tags and look for terms which define regulation.

[1] GOATOOLS version 0.6.4 was used to write this tutorial and the exercises. To install this exact version, use `pip install goatools==0.6.4`

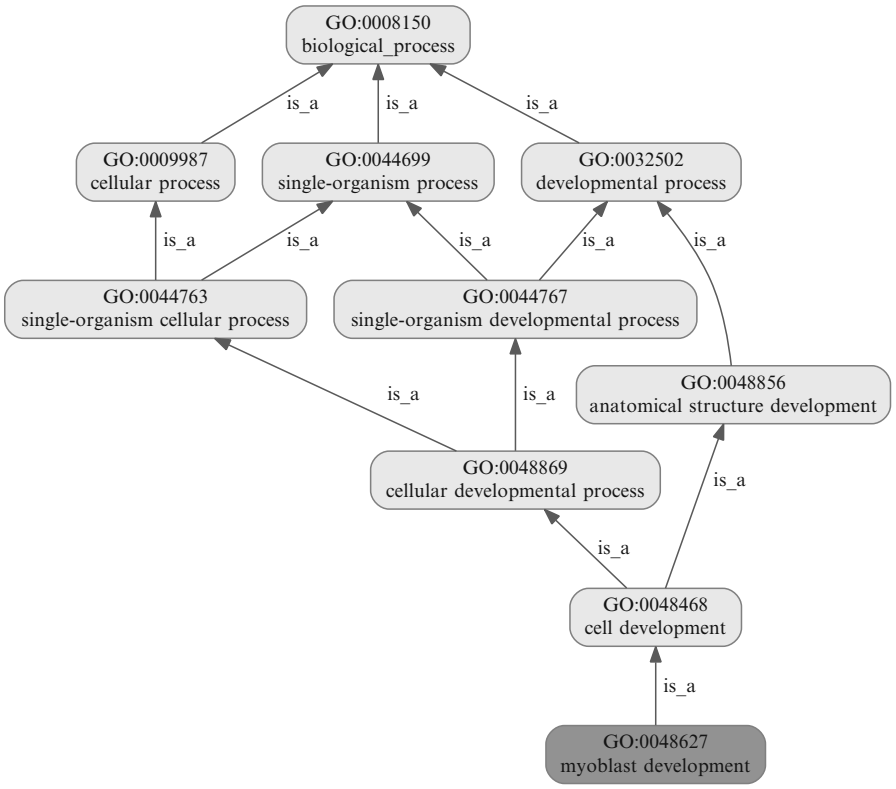


Fig. 1 Selected parts of the Gene Ontology can be visualised using the GOATOOLS library [2]

Exercise 2.2
Using the visualisation function in the GOATOOLS library, answer the following questions:

(a) Produce a figure similar to that in Fig. 1, for the GO term `GO:0097190`. From the visualisation, what is the name of this term?

(b) Using this figure, what is the most specific term that is in the parent terms of both `GO:0097191` (extrinsic apoptotic signalling pathway) and `GO:0038034` (signal transduction in absence of ligand)? This is also referred to as the lowest common ancestor (see Chap. 12 [4]).

Furthermore, other tag-value lines such as the "relationships" can be loaded with an optional argument of, e.g., `optional_attrs=['relationship']`.

The GOATOOLS library also includes functions to visualise the GO graph. For instance, it is possible to depict the location of a particular GO term in the ontology using the method `GOTerm.draw_lineage()`. For example, the plot in Fig. 1 showing the lineage of the GO term `GO:0048527` was created using this function.

As an alternative to GOATOOLS and OBO files, it is possible to retrieve information relating to a specific term from a web service. One such service is the EMBL-EBI QuickGO resource (see

Chap. 11; [3, 5]), which can provide descriptive information about GO terms in OBO-XML format. It is possible to request this OBO-XML file over HTTP, using a URL of the form

http://www.ebi.ac.uk/QuickGO/GTerm?id=<GO_ID>&format=oboxml

where `<GO_ID>` is replaced with the GO identifier for the term of interest. In Source Code 2.1, an example function to automate this in Python is listed, which uses the urllib library to request the OBO-XML and the xmltodict library to parse the XML into an easy to use dictionary structure. Both libraries are available to install using `pip`, if required. Note that the future library was used to ensure that the function is both Python 2 and 3 compatible.

The dictionary structure that is returned can vary based on what information is available in the database. One example of an information-rich term is `GO:0043065`. A visualisation of the dictionary

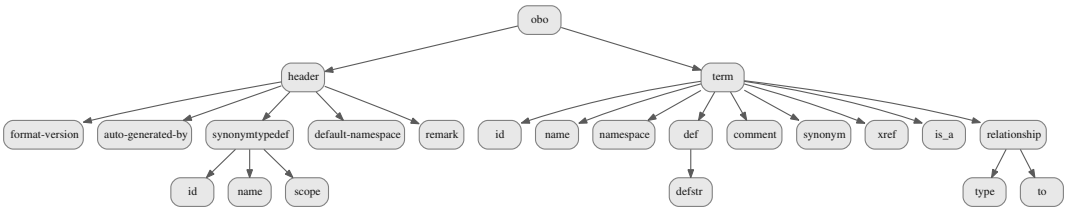


Fig. 2 Visualisation of the keys in the hierarchical dictionary structure returned by `get_oboxml('GO:0043065')`

Source Code 2.1. `get_oboxml()` function for Python 2 and 3.

```
from future.standard_library import install_aliases
install_aliases()
from urllib.request import urlopen
import xmltodict

def get_oboxml(go_id):
    """
        This function retrieves the OBO-XML for a
        given Gene Ontology term, using EMBL-EBI's
        QuickGO browser.
        Input: go_id - a valid Gene Ontology ID,
        e.g. GO:0048527.
    """
    quickgo_url= "http://ebi.ac.uk/QuickGO/GTerm?id="+
    go_id+"&format=oboxml"
    oboxml = urlopen(quickgo_url)

    # Check the response
    if(oboxml.getcode() == 200):
        obodict = xmltodict.parse(oboxml.read())
        return obodict
    else:
        raise ValueError("Couldn't receive OBOXML
        from QuickGO. Check URL and try again.")
```

structure for this term, created with the `visualisedictionary` package available from PyPI (using `pip`), has been included in Fig. 2.

The main advantage of using a web service, such as QuickGO, is that there is no requirement to download and parse the entire Gene Ontology structure; only the information required is retrieved. This is therefore more efficient if only a few particular terms are involved in an analysis. By contrast, for analyses involving many terms, the file-based approach described above is more suitable.

Exercise 2.3
Using the function `get_oboxml()`, listed in Source Code 2.1, answer the following questions:

(a) Find the name and description of the GO term `GO:0048527` (lateral root development). *Hint*: print out the dictionary returned by the function and study its structure, or use the visualisation in Fig. 2.

(b) Look at the difference in the OBO-XML output for the GO terms `GO:00048527` (lateral root development) and `GO:0097178` (ruffle assembly), then generate a table of the synonymous relationships of the term `GO:0097178`.

3 Retrieving GO Annotations

This section looks at manipulating the Gene Association File (GAF) standard, using a parser from the BioPython package [6].

Firstly, a GAF file, which contains GO annotations, shall be downloaded from the UniProt-GOA database [7]. Their website (https://www.ebi.ac.uk/GOA/downloads) lists a number of variants. For this tutorial the reduced GAF file containing only the gene association data for *Arabidopsis thaliana* is going to be used.

Annotations from GAF files can be loaded into a Python dictionary using an iterator from the BioPython package (`Bio.UniProt.GOA.gafiterator`). Source Code 3.1 shows a simple example of this being used, in order to print out the protein ID for each annotation.

Source Code 3.1

```
from Bio.UniProt.GOA import gafiterator
import gzip

# filename = <LOCATION OF GAF FILE>
filename = 'gene_association.goa_arabidopsis.gz'

with gzip.open(filename, 'rt') as fp:
    for annotation in gafiterator(fp):
        # Output annotated protein ID
        print(annotation['DB_Object_ID'])
```

Recall that the latest GAF standard, version 2.1, has 17 tab-delimited fields, which are described in detail in Chap. 3 [1]. Some of them include:

- `'DB'`: the protein database;
- `'DB_Object_ID'`: protein ID;
- `'Qualifier'`: annotation qualifier (such as NOT);
- `'GO_ID'`: GO term;
- `'Evidence'`: evidence code.

Exercise 3.1

(a) Find the total number of annotations for *Arabidopsis thaliana* with NOT qualifiers. What is this as a percentage of the total number of annotations for this species?

(b) How many genes (of *Arabidopsis thaliana*) have the annotation `GO:0048527` (lateral root development)?

(c) Generate a list of annotated proteins which have the word "growth" in their name.

(d) There are 21 evidence codes used in the Gene Ontology project. As discussed in Chap. 3 [1], many of these are inferred, either by curators or automatically. Find the counts of each evidence code in the *Arabidopsis thaliana* annotation file.

4 GO Enrichment or Depletion Analysis

As discussed in detail in Chap. 13 [8] one of the most common analyses performed on GO data is an enrichment (or depletion) analysis. In this tutorial, the `GOEnrichmentStudy()` function available in the GOATOOLS library (which has been seen in section 2) will be used.

The `GOEnrichmentStudy()` function requires the following arguments:

1. the background set of terms (also known as the "population set"), passed as a list of GO term IDs;
2. associations between proteins IDs and GO term IDs, passed as a dictionary with protein IDs as the keys and sets of associated GO terms as the values;
3. the Gene Ontology structure, i.e., the output by the `obo_parser()` function from GOATOOLS;
4. whether annotations should be propagated to all parent terms, (defined in terms of `is_a` tags, only), indicated by setting the optional boolean parameter `propagate_counts` to `True` (default) or `False`;

5. the significance level, indicated by setting the optional parameter `alpha` to the desired cut-off (default: 0.05);
6. the foreground set of terms (also known as "study set"), indicated by setting the parameter `study` to a list of GO term IDs;
7. the list of method(s) to be used to assess significance, indicated by setting the parameter `methods` to a list containing one or several of these elements:
   (a) `"bonferroni"`: Fisher's exact test with Bonferroni correction for multiple testing;
   (b) `"sidak"`: Fisher's exact test with Šidák correction for multiple testing;
   (c) `"holm"`: Fisher's exact test with Holm–Bonferroni correction for multiple testing;
   (d) `"fdr"`: Fisher's exact test, controlling the false discovery rate (see Chap. 13 [8]).

The function returns the list of over-represented and under-represented GO terms in the population set, compared to the background set.

Exercise 4.1
Perform an enrichment analysis using the list of genes with the "growth" keyword from exercise 3.1.c. Use the *Arabidopsis thaliana* annotation set as background, also from exercise 3.1, and the GO structure from exercise 2.1.

(a) Which GO term is most significantly enriched or depleted? Does this make sense?

(b) How many terms are enriched, when using the Bonferroni corrected p-value ≤ 0.01?

(c) How many terms are enriched, when using the false discovery rate (a.k.a. q-value) ≤ 0.01?

5 Computing Basic Semantic Similarities Between GO Terms

In this section, the focus is on computing semantic similarity between GO terms, based on ideas presented in detail in Chap. 12 [4]. Semantic similarity measures enable us to quantify the functional similarity of genes annotated with GO terms.

Recall that semantic similarity measures are broadly separated in two categories: graph-based and information-theoretic measures. The former relies only on the structure of the Gene Ontology graph, whilst the latter also accounts for the information content of the terms.

One graph-based measure of semantic similarity, presented in Chap. 12 [4], is the inverse of the number of edges separating two

terms. It is possible to compute the minimum number of edges separating two terms (t_1, t_2) by first finding the deepest common ancestor (t_{DCA}). Then the difference in depth between each term and the deepest common ancestor can be used to calculate the minimum distance between the terms. i.e.,

$$\text{min\_distance}(t_1, t_2) = \text{depth}(t_1) + \text{depth}(t_2) - 2 \times \text{depth}(t_{DCA})$$

Further, one example of an information-theoretic measure (see Chap. 12 [4]) is Resnik's similarity measure—the information content of the most informative common ancestor of the two terms in question. The information content of a term is defined as the negative logarithm of its probability, which can be estimated from the frequency of the term in the annotation database of choice.

Exercise 5.1

(a) `GO:0048364` (root development) and `GO:0044707` (single-multicellular organism process) are two GO terms taken from Fig. 1. Calculate the semantic similarity between them based on the inverse of the semantic distance (number of branches separating them).

(b) Calculate the information content (IC) of the GO term `GO:0048364` (root development), based on the frequency of observation in *Arabidopsis thaliana*.

(c) Calculate the Resnik similarity measure between the same two terms as in part a.

Acknowledgements

We thank Adrian Altenhoff, Debra Klopfenstein, and Haibao Tang for helpful feedback on the tutorial. CD acknowledges Swiss National Science Foundation grant 150654 and UK BBSRC grant BB/M015009/1. Open Access charges were funded by the University College London Library, the Swiss Institute of Bioinformatics, the Agassiz Foundation, and the Foundation for the University of Lausanne.

References

1. Gaudet P, Škunca N, Hu JC, Dessimoz C (2016) Primer on the gene ontology. In: Dessimoz C, Škunca N (eds) The gene ontology handbook. Methods in molecular biology, vol 1446. Humana Press. Chapter 3
2. Tang H, Klopfenstein D, Pedersen B et al (2015) GOATOOLS: tools for gene ontology, Zenodo
3. Munoz-Torres M, Carbon S (2016) Get GO! retrieving GO data using AmiGO, QuickGO, API, files, and tools. In: Dessimoz C, Škunca N (eds) The gene ontology handbook. Methods in molecular biology, vol 1446. Humana Press. Chapter 11
4. Pesquita C (2016) Semantic similarity in the gene ontology. In: Dessimoz C, Škunca N (eds) The gene ontology handbook. Methods in molecular biology, vol 1446. Humana Press. Chapter 12
5. Binns D, Dimmer E, Huntley R et al (2009) QuickGO: a web-based tool for Gene Ontology searching. Bioinformatics 25:3045–3046
6. Cock PJA, Antao T, Chang JT et al (2009) Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics 25:1422–1423
7. Huntley RP, Sawford T, Mutowo-Meullenet P et al (2015) The GOA database: gene Ontology annotation updates for 2015. Nucleic Acids Res 43:D1057–63
8. Bauer S (2016) Gene-category analysis. In: Dessimoz C, Škunca N (eds) The gene ontology handbook. Methods in molecular biology, vol 1446. Humana Press. Chapter 13

Appendix D

Ancestral Gene Ontology Enrichment Analysis Results

This appendix contains the results from the ancestral Gene Ontology enrichment analysis presented in chapter 6.

D.1 Novel Genes – GO Enrichment Analysis

Table D.1: Significantly enriched ($p < 0.05$) Gene Ontology terms in the Biological Process aspect, for the novel gene-set.

GO Term	Description	Weight01 (Fisher)	Parent-Child (Fisher)
GO:0008152	metabolic process	0.012	< 0.001
GO:0007264	small GTPase mediated signal transduction	0.008	< 0.001
GO:0051301	cell division	< 0.001	< 0.001
GO:0005975	carbohydrate metabolic process	< 0.001	< 0.001
GO:0006914	autophagy	0.007	< 0.001
GO:0006412	translation	0.003	0.001
GO:0006260	DNA replication	0.007	0.003
GO:0051726	regulation of cell cycle	0.025	0.003
GO:0006413	translational initiation	0.013	0.021
GO:0006099	tricarboxylic acid cycle	0.018	0.027
GO:0006811	ion transport	0.006	0.044

Table D.2: Significantly enriched ($p < 0.05$) Gene Ontology terms in the Cellular Component aspect, for the novel gene-set.

GO Term	Description	Weight01 (Fisher)	Parent-Child (Fisher)
GO:0070062	extracellular exosome	< 0.001	< 0.001
GO:0045211	postsynaptic membrane	0.028	< 0.001
GO:0005794	Golgi apparatus	0.021	< 0.001
GO:0014069	postsynaptic density	0.042	< 0.001

Continued on next page

Table D.2 – *continued from previous page*

GO Term	Description	Weight01 (Fisher)	Parent-Child (Fisher)
GO:0048471	perinuclear region of cytoplasm	0.018	0.004
GO:0005578	proteinaceous extracellular matrix	0.001	0.004
GO:0005789	endoplasmic reticulum membrane	0.033	0.007
GO:0005743	mitochondrial inner membrane	0.008	0.008

Table D.3: Significantly enriched ($p < 0.05$) Gene Ontology terms in the Molecular Function aspect, for the novel gene-set.

GO Term	Description	Weight01 (Fisher)	Parent-Child (Fisher)
GO:0016301	kinase activity	< 0.001	< 0.001
GO:0016746	transferase activity, transferring acyl groups	0.030	< 0.001
GO:0016491	oxidoreductase activity	0.003	< 0.001
GO:0003676	nucleic acid binding	0.011	< 0.001
GO:0003735	structural constituent of ribosome	0.012	< 0.001
GO:0005525	GTP binding	0.003	< 0.001
GO:0042803	protein homodimerization activity	0.005	< 0.001
GO:0005516	calmodulin binding	0.002	0.003
GO:0043565	sequence-specific DNA binding	0.005	0.007
GO:0016887	ATPase activity	0.003	0.008
GO:0005102	receptor binding	0.002	0.008
GO:0008146	sulfotransferase activity	< 0.001	0.013
GO:0004722	protein serine/threonine phosphatase activity	0.002	0.016
GO:0003723	RNA binding	0.025	0.020
GO:0000287	magnesium ion binding	0.017	0.028
GO:0003684	damaged DNA binding	0.017	0.040

D.2 Gene Duplications – GO Enrichment Analysis

Table D.4: Significantly enriched ($p < 0.05$) Gene Ontology terms in the Biological Process aspect, for the duplicated gene-set.

GO Term	Description	Weight01 (Fisher)	Parent-Child (Fisher)
GO:0018026	peptidyl-lysine monomethylation	< 0.001	< 0.001
GO:0007216	G-protein coupled glutamate receptor signaling pathway	< 0.001	< 0.001
GO:0006426	glycyl-tRNA aminoacylation	< 0.001	< 0.001
GO:0051865	protein autoubiquitination	< 0.001	< 0.001
GO:0051262	protein tetramerization	< 0.001	< 0.001
GO:0007340	acrosome reaction	0.003	< 0.001
GO:0018298	protein-chromophore linkage	< 0.001	< 0.001
GO:0032324	molybdopterin cofactor biosynthetic process	< 0.001	< 0.001

Continued on next page

Table D.4 – *continued from previous page*

GO Term	Description	Weight01 (Fisher)	Parent-Child (Fisher)
GO:0006541	glutamine metabolic process	< 0.001	< 0.001
GO:0006406	mRNA export from nucleus	0.002	0.001
GO:0097056	selenocysteinyl-tRNA(Sec) biosynthetic process	0.002	0.002
GO:0000289	nuclear-transcribed mRNA poly(A) tail shortening	< 0.001	0.002
GO:0046951	ketone body biosynthetic process	0.004	0.004
GO:0060070	canonical Wnt signaling pathway	0.003	0.004
GO:0055070	copper ion homeostasis	< 0.001	0.005
GO:0033209	tumor necrosis factor-mediated signaling pathway	< 0.001	0.006
GO:0006626	protein targeting to mitochondrion	0.003	0.006
GO:0055117	regulation of cardiac muscle contraction	< 0.001	0.008
GO:0071482	cellular response to light stimulus	< 0.001	0.011
GO:0017158	regulation of calcium ion-dependent exocytosis	0.002	0.012
GO:0044571	[2Fe-2S] cluster assembly	< 0.001	0.013
GO:0007172	signal complex assembly	0.009	0.015
GO:0006777	Mo-molybdopterin cofactor biosynthetic process	0.013	0.015
GO:0050796	regulation of insulin secretion	< 0.001	0.015
GO:0046416	D-amino acid metabolic process	0.007	0.016
GO:0002027	regulation of heart rate	0.007	0.017
GO:0015893	drug transport	0.017	0.020
GO:0006289	nucleotide-excision repair	0.004	0.021
GO:0008360	regulation of cell shape	0.013	0.032
GO:0007250	activation of NF-kappaB-inducing kinase activity	0.002	0.033
GO:0007205	protein kinase C-activating G-protein coupled receptor signaling pathway	0.013	0.033
GO:0005981	regulation of glycogen catabolic process	< 0.001	0.036
GO:0006779	porphyrin-containing compound biosynthetic process	0.011	0.036
GO:0006836	neurotransmitter transport	0.023	0.037
GO:0007214	gamma-aminobutyric acid signaling pathway	0.015	0.038
GO:0030513	positive regulation of BMP signaling pathway	0.004	0.044
GO:0007602	phototransduction	0.038	0.047

Table D.5: Significantly enriched ($p < 0.05$) Gene Ontology terms in the Cellular Component aspect, for the duplicated gene-set.

GO Term	Description	Weight01 (Fisher)	Parent-Child (Fisher)
GO:0000786	nucleosome	< 0.001	< 0.001
GO:0005829	cytosol	< 0.001	< 0.001
GO:0019898	extrinsic component of membrane	0.025	< 0.001
GO:0016459	myosin complex	< 0.001	< 0.001
GO:0005834	heterotrimeric G-protein complex	< 0.001	< 0.001
GO:0000164	protein phosphatase type 1 complex	< 0.001	< 0.001
GO:0031251	PAN complex	< 0.001	0.001
GO:0005669	transcription factor TFIID complex	0.003	0.005
GO:0017119	Golgi transport complex	0.018	0.005
GO:0008021	synaptic vesicle	0.014	0.027
GO:0042734	presynaptic membrane	0.018	0.030

Continued on next page

Table D.5 – *continued from previous page*

GO Term	Description	Weight01 (Fisher)	Parent-Child (Fisher)
GO:0016012	sarcoglycan complex	0.018	0.032

Table D.6: Significantly enriched ($p < 0.05$) Gene Ontology terms in the Molecular Function aspect, for the duplicated gene-set.

GO Term	Description	Weight01 (Fisher)	Parent-Child (Fisher)
GO:0004984	olfactory receptor activity	< 0.001	< 0.001
GO:0046982	protein heterodimerization activity	< 0.001	< 0.001
GO:0004820	glycine-tRNA ligase activity	< 0.001	< 0.001
GO:0004871	signal transducer activity	< 0.001	< 0.001
GO:0005125	cytokine activity	0.037	< 0.001
GO:0004519	endonuclease activity	0.022	< 0.001
GO:0008504	monoamine transmembrane transporter activity	< 0.001	< 0.001
GO:0003951	NAD+ kinase activity	< 0.001	< 0.001
GO:0003883	CTP synthase activity	0.003	0.001
GO:0001604	urotensin II receptor activity	0.018	0.001
GO:0009881	photoreceptor activity	< 0.001	0.001
GO:0005509	calcium ion binding	0.008	0.001
GO:0008195	phosphatidate phosphatase activity	0.004	0.003
GO:0016785	transferase activity, transferring selenium-containing groups	0.003	0.004
GO:0004252	serine-type endopeptidase activity	0.020	0.004
GO:0008013	beta-catenin binding	0.007	0.006
GO:0003774	motor activity	< 0.001	0.007
GO:0004109	coproporphyrinogen oxidase activity	< 0.001	0.009
GO:0051015	actin filament binding	0.003	0.009
GO:0003697	single-stranded DNA binding	0.004	0.009
GO:0008158	hedgehog receptor activity	0.005	0.010
GO:0004523	RNA-DNA hybrid ribonuclease activity	< 0.001	0.011
GO:0004359	glutaminase activity	< 0.001	0.012
GO:0015238	drug transmembrane transporter activity	0.018	0.015
GO:0004791	thioredoxin-disulfide reductase activity	0.018	0.015
GO:0004067	asparaginase activity	< 0.001	0.017
GO:0004996	thyroid-stimulating hormone receptor activity	0.003	0.018
GO:0004579	dolichyl-diphosphooligosaccharide-protein glycotransferase activity	< 0.001	0.018
GO:0031071	cysteine desulfurase activity	< 0.001	0.018
GO:0008092	cytoskeletal protein binding	0.021	0.021
GO:0004143	diacylglycerol kinase activity	0.019	0.026
GO:0016300	tRNA (uracil) methyltransferase activity	0.005	0.026
GO:0016702	oxidoreductase activity, acting on single donors with incorporation of molecular oxygen, incorporation of two atoms of oxygen	0.011	0.031
GO:0008349	MAP kinase kinase kinase kinase activity	0.013	0.033
GO:0016934	extracellular-glycine-gated chloride channel activity	0.036	0.035

Continued on next page

Table D.6 – ***continued from previous page***

GO Term	Description	Weight01 (Fisher)	Parent-Child (Fisher)
GO:0008897	holo-[acyl-carrier-protein] synthase activity	< 0.001	0.036

D.3 Gene Losses – GO Enrichment Analysis

Table D.7: Significantly enriched ($p < 0.05$) Gene Ontology terms in the Biological Process aspect, for the lost gene-set.

GO Term	Description	Weight01 (Fisher)	Parent-Child (Fisher)
GO:0007602	phototransduction	< 0.001	0.002
GO:0018298	protein-chromophore linkage	0.003	0.004
GO:0034767	positive regulation of ion transmembrane transport	0.011	0.008
GO:0001822	kidney development	0.019	0.018
GO:0015701	bicarbonate transport	0.011	0.042
GO:0035385	Roundabout signaling pathway	0.041	0.047

Table D.8: Significantly enriched ($p < 0.05$) Gene Ontology terms in the Cellular Component aspect, for the lost gene-set.

GO Term	Description	Weight01 (Fisher)	Parent-Child (Fisher)
GO:0036064	ciliary basal body	< 0.001	< 0.001
GO:0005604	basement membrane	0.003	0.015
GO:0005814	centriole	0.045	0.035

Table D.9: Significantly enriched ($p < 0.05$) Gene Ontology terms in the Molecular Function aspect, for the lost gene-set.

GO Term	Description	Weight01 (Fisher)	Parent-Child (Fisher)
GO:0009881	photoreceptor activity	0.002	0.003
GO:0038021	leptin receptor activity	0.004	0.012
GO:0048040	UDP-glucuronate decarboxylase activity	0.004	0.023
GO:0050429	calcium-dependent phospholipase C activity	0.007	0.039

D.4 Continuing Genes – GO Enrichment Analysis

Table D.10: Significantly enriched ($p < 0.05$) Gene Ontology terms in the Biological Process aspect, for the continuing gene-set.

GO Term	Description	Weight01 (Fisher)	Parent-Child (Fisher)
GO:0008152	metabolic process	0.012	< 0.001
GO:0007264	small GTPase mediated signal transduction	0.008	< 0.001
GO:0051301	cell division	< 0.001	< 0.001

Continued on next page

Table D.10 – *continued from previous page*

GO Term	Description	Weight01 (Fisher)	Parent-Child (Fisher)
GO:0005975	carbohydrate metabolic process	< 0.001	< 0.001
GO:0006914	autophagy	0.007	< 0.001
GO:0006412	translation	0.003	0.001
GO:0006260	DNA replication	0.007	0.003
GO:0051726	regulation of cell cycle	0.025	0.003
GO:0006413	translational initiation	0.013	0.021
GO:0006099	tricarboxylic acid cycle	0.018	0.027
GO:0006811	ion transport	0.006	0.044

Table D.11: Significantly enriched $(p < 0.05)$ Gene Ontology terms in the Cellular Component aspect, for the continuing gene-set.

GO Term	Description	Weight01 (Fisher)	Parent-Child (Fisher)
GO:0070062	extracellular exosome	< 0.001	< 0.001
GO:0045211	postsynaptic membrane	0.028	< 0.001
GO:0005794	Golgi apparatus	0.021	< 0.001
GO:0014069	postsynaptic density	0.042	< 0.001
GO:0048471	perinuclear region of cytoplasm	0.018	0.004
GO:0005578	proteinaceous extracellular matrix	0.001	0.004
GO:0005789	endoplasmic reticulum membrane	0.033	0.007
GO:0005743	mitochondrial inner membrane	0.008	0.008

Table D.12: Significantly enriched $(p < 0.05)$ Gene Ontology terms in the Molecular Function aspect, for the continuing gene-set.

GO Term	Description	Weight01 (Fisher)	Parent-Child (Fisher)
GO:0016301	kinase activity	< 0.001	< 0.001
GO:0016746	transferase activity, transferring acyl groups	0.030	< 0.001
GO:0016491	oxidoreductase activity	0.003	< 0.001
GO:0003676	nucleic acid binding	0.011	< 0.001
GO:0003735	structural constituent of ribosome	0.012	< 0.001
GO:0005525	GTP binding	0.003	< 0.001
GO:0042803	protein homodimerization activity	0.005	< 0.001
GO:0005516	calmodulin binding	0.002	0.003
GO:0043565	sequence-specific DNA binding	0.005	0.007
GO:0016887	ATPase activity	0.003	0.008
GO:0005102	receptor binding	0.002	0.008
GO:0008146	sulfotransferase activity	< 0.001	0.013
GO:0004722	protein serine/threonine phosphatase activity	0.002	0.016
GO:0003723	RNA binding	0.025	0.020
GO:0000287	magnesium ion binding	0.017	0.028
GO:0003684	damaged DNA binding	0.017	0.040

D.5 Lost Genes after Filtering – GO Enrichment Analysis

This section contains the results of the GO enrichment analysis for the lost gene-set, after filtering out possible errors in the assembly and post-processing, as described in section 6.2.4.

Table D.13: Significantly enriched ($p < 0.05$) Gene Ontology terms in the Biological Process aspect, for the lost gene-set after filtering.

GO Term	Description	Weight01 (Fisher)	Parent-Child (Fisher)
GO:0009791	post-embryonic development	< 0.001	< 0.001
GO:0001649	osteoblast differentiation	< 0.001	< 0.001
GO:0043524	negative regulation of neuron apoptotic process	< 0.001	< 0.001
GO:0007507	heart development	< 0.001	< 0.001
GO:0051321	meiotic cell cycle	< 0.001	< 0.001
GO:0008406	gonad development	0.040	< 0.001
GO:0070269	pyroptosis	< 0.001	< 0.001
GO:0071222	cellular response to lipopolysaccharide	< 0.001	< 0.001
GO:0001570	vasculogenesis	< 0.001	< 0.001
GO:0070374	positive regulation of ERK1 and ERK2 cascade	< 0.001	< 0.001
GO:0006836	neurotransmitter transport	< 0.001	< 0.001
GO:0019915	lipid storage	< 0.001	< 0.001
GO:0007218	neuropeptide signaling pathway	< 0.001	< 0.001
GO:0060055	angiogenesis involved in wound healing	< 0.001	< 0.001
GO:0042423	catecholamine biosynthetic process	< 0.001	< 0.001
GO:0006665	sphingolipid metabolic process	< 0.001	< 0.001
GO:0006334	nucleosome assembly	< 0.001	< 0.001
GO:0006355	regulation of transcription, DNA-templated	< 0.001	< 0.001
GO:0007269	neurotransmitter secretion	< 0.001	< 0.001
GO:0006099	tricarboxylic acid cycle	< 0.001	< 0.001
GO:0045900	negative regulation of translational elongation	0.002	< 0.001
GO:0009396	folic acid-containing compound biosynthetic process	0.003	< 0.001
GO:0032729	positive regulation of interferon-gamma production	< 0.001	< 0.001
GO:0070098	chemokine-mediated signaling pathway	< 0.001	< 0.001
GO:0006426	glycyl-tRNA aminoacylation	0.002	< 0.001
GO:0016540	protein autoprocessing	0.002	< 0.001
GO:0006468	protein phosphorylation	0.002	< 0.001
GO:0007585	respiratory gaseous exchange	< 0.001	< 0.001
GO:0006888	ER to Golgi vesicle-mediated transport	< 0.001	< 0.001
GO:0019882	antigen processing and presentation	< 0.001	< 0.001
GO:0002407	dendritic cell chemotaxis	< 0.001	< 0.001
GO:0043550	regulation of lipid kinase activity	0.002	< 0.001
GO:0006700	C21-steroid hormone biosynthetic process	0.001	0.001
GO:0018298	protein-chromophore linkage	< 0.001	0.001
GO:0045943	positive regulation of transcription from RNA polymerase I promoter	< 0.001	0.001
GO:0070527	platelet aggregation	< 0.001	0.001
GO:0090398	cellular senescence	0.010	0.001

Continued on next page

Table D.13 – *continued from previous page*

GO Term	Description	Weight01 (Fisher)	Parent-Child (Fisher)
GO:0018107	peptidyl-threonine phosphorylation	< 0.001	0.001
GO:0072661	protein targeting to plasma membrane	< 0.001	0.002
GO:0045494	photoreceptor cell maintenance	0.001	0.002
GO:0055117	regulation of cardiac muscle contraction	< 0.001	0.002
GO:0006898	receptor-mediated endocytosis	0.016	0.002
GO:0006839	mitochondrial transport	0.029	0.003
GO:0034767	positive regulation of ion transmembrane transport	0.002	0.003
GO:0035584	calcium-mediated signaling using intracellular calcium source	< 0.001	0.004
GO:1902476	chloride transmembrane transport	0.002	0.004
GO:0008219	cell death	0.003	0.005
GO:0007229	integrin-mediated signaling pathway	< 0.001	0.005
GO:0090286	cytoskeletal anchoring at nuclear membrane	0.013	0.006
GO:0006270	DNA replication initiation	0.011	0.006
GO:0006541	glutamine metabolic process	0.032	0.006
GO:0031122	cytoplasmic microtubule organization	0.032	0.008
GO:0015689	molybdate ion transport	0.002	0.008
GO:0042744	hydrogen peroxide catabolic process	0.010	0.009
GO:0006260	DNA replication	0.013	0.009
GO:0043966	histone H3 acetylation	0.003	0.010
GO:0019510	S-adenosylhomocysteine catabolic process	0.001	0.010
GO:0006909	phagocytosis	0.007	0.011
GO:0009615	response to virus	0.003	0.011
GO:2000813	negative regulation of barbed-end actin filament capping	0.021	0.012
GO:0042254	ribosome biogenesis	< 0.001	0.014
GO:0006013	mannose metabolic process	0.011	0.015
GO:0042787	protein ubiquitination involved in ubiquitin-dependent protein catabolic process	0.048	0.016
GO:1902475	L-alpha-amino acid transmembrane transport	0.002	0.016
GO:0061436	establishment of skin barrier	< 0.001	0.017
GO:0030514	negative regulation of BMP signaling pathway	0.022	0.018
GO:0000038	very long-chain fatty acid metabolic process	0.030	0.018
GO:0000289	nuclear-transcribed mRNA poly(A) tail shortening	0.001	0.019
GO:0006626	protein targeting to mitochondrion	0.022	0.020
GO:0000737	DNA catabolic process, endonucleolytic	0.005	0.020
GO:0090090	negative regulation of canonical Wnt signaling pathway	0.041	0.021
GO:0044571	[2Fe-2S] cluster assembly	0.001	0.022
GO:0032786	positive regulation of DNA-templated transcription, elongation	0.013	0.024
GO:0035458	cellular response to interferon-beta	< 0.001	0.024
GO:0061337	cardiac conduction	< 0.001	0.025
GO:0015908	fatty acid transport	0.026	0.026
GO:0007172	signal complex assembly	0.022	0.029
GO:0030433	ubiquitin-dependent ERAD pathway	0.027	0.029
GO:0045766	positive regulation of angiogenesis	0.007	0.035

Continued on next page

Table D.13 – *continued from previous page*

GO Term	Description	Weight01 (Fisher)	Parent-Child (Fisher)
GO:2000042	negative regulation of double-strand break repair via homologous recombination	0.048	0.036
GO:0015914	phospholipid transport	0.034	0.037
GO:0006427	histidyl-tRNA aminoacylation	0.048	0.038
GO:0030501	positive regulation of bone mineralization	0.002	0.038
GO:0071025	RNA surveillance	0.013	0.044
GO:0019229	regulation of vasoconstriction	0.017	0.044
GO:0048266	behavioral response to pain	< 0.001	0.048

Table D.14: Significantly enriched $(p < 0.05)$ Gene Ontology terms in the Cellular Component aspect, for the lost gene-set after filtering.

GO Term	Description	Weight01 (Fisher)	Parent-Child (Fisher)
GO:0005829	cytosol	< 0.001	< 0.001
GO:0031225	anchored component of membrane	< 0.001	< 0.001
GO:0030496	midbody	< 0.001	< 0.001
GO:0009986	cell surface	< 0.001	< 0.001
GO:0005758	mitochondrial intermembrane space	< 0.001	< 0.001
GO:0016529	sarcoplasmic reticulum	< 0.001	< 0.001
GO:0070062	extracellular exosome	< 0.001	< 0.001
GO:0016324	apical plasma membrane	< 0.001	< 0.001
GO:0048471	perinuclear region of cytoplasm	< 0.001	< 0.001
GO:0000777	condensed chromosome kinetochore	< 0.001	< 0.001
GO:0005741	mitochondrial outer membrane	< 0.001	< 0.001
GO:0005654	nucleoplasm	< 0.001	< 0.001
GO:0005886	plasma membrane	< 0.001	< 0.001
GO:0005635	nuclear envelope	< 0.001	< 0.001
GO:0043209	myelin sheath	< 0.001	< 0.001
GO:0016459	myosin complex	< 0.001	< 0.001
GO:0009897	external side of plasma membrane	< 0.001	< 0.001
GO:0032580	Golgi cisterna membrane	< 0.001	< 0.001
GO:0005634	nucleus	< 0.001	< 0.001
GO:0043025	neuronal cell body	< 0.001	< 0.001
GO:0016323	basolateral plasma membrane	< 0.001	< 0.001
GO:0005730	nucleolus	< 0.001	< 0.001
GO:0005739	mitochondrion	< 0.001	< 0.001
GO:0015629	actin cytoskeleton	< 0.001	< 0.001
GO:0014069	postsynaptic density	< 0.001	< 0.001
GO:0030027	lamellipodium	< 0.001	< 0.001
GO:0055037	recycling endosome	< 0.001	< 0.001
GO:0016363	nuclear matrix	< 0.001	< 0.001
GO:0005887	integral component of plasma membrane	0.026	< 0.001
GO:0031901	early endosome membrane	< 0.001	< 0.001
GO:0005811	lipid particle	< 0.001	< 0.001
GO:0005759	mitochondrial matrix	< 0.001	< 0.001
GO:0005813	centrosome	< 0.001	< 0.001

Continued on next page

Table D.14 – *continued from previous page*

GO Term	Description	Weight01 (Fisher)	Parent-Child (Fisher)
GO:0001669	acrosomal vesicle	< 0.001	< 0.001
GO:0005769	early endosome	0.029	< 0.001
GO:0001518	voltage-gated sodium channel complex	< 0.001	< 0.001
GO:0030018	Z disc	< 0.001	< 0.001
GO:0045095	keratin filament	0.005	0.001
GO:0030688	preribosome, small subunit precursor	< 0.001	0.001
GO:0070469	respiratory chain	0.001	0.001
GO:0031105	septin complex	< 0.001	0.001
GO:0031251	PAN complex	0.001	0.001
GO:0042555	MCM complex	0.001	0.002
GO:1904813	ficolin-1-rich granule lumen	< 0.001	0.003
GO:0000930	gamma-tubulin complex	0.002	0.003
GO:0017119	Golgi transport complex	0.001	0.003
GO:0014731	spectrin-associated cytoskeleton	< 0.001	0.004
GO:0005581	collagen trimer	0.001	0.005
GO:0043204	perikaryon	< 0.001	0.008
GO:0016605	PML body	< 0.001	0.012
GO:0030175	filopodium	< 0.001	0.013
GO:0030426	growth cone	0.001	0.014
GO:0032587	ruffle membrane	0.001	0.014
GO:0043197	dendritic spine	< 0.001	0.017
GO:0000164	protein phosphatase type 1 complex	0.013	0.017
GO:0005881	cytoplasmic microtubule	0.015	0.019
GO:0016514	SWI/SNF complex	< 0.001	0.027
GO:0033588	Elongator holoenzyme complex	0.023	0.028
GO:0070772	PAS complex	0.002	0.032
GO:0030008	TRAPP complex	0.028	0.034
GO:0034993	LINC complex	0.018	0.036
GO:0045211	postsynaptic membrane	< 0.001	0.042
GO:0005743	mitochondrial inner membrane	< 0.001	0.043
GO:0072487	MSL complex	0.049	0.048

Table D.15: Significantly enriched ($p < 0.05$) Gene Ontology terms in the Molecular Function aspect, for the lost gene-set after filtering.

GO Term	Description	Weight01 (Fisher)	Parent-Child (Fisher)
GO:0016503	pheromone receptor activity	< 0.001	< 0.001
GO:0005516	calmodulin binding	< 0.001	< 0.001
GO:0004984	olfactory receptor activity	< 0.001	< 0.001
GO:0051082	unfolded protein binding	< 0.001	< 0.001
GO:0051117	ATPase binding	< 0.001	< 0.001
GO:0042826	histone deacetylase binding	< 0.001	< 0.001
GO:0005149	interleukin-1 receptor binding	< 0.001	< 0.001
GO:0008022	protein C-terminus binding	< 0.001	< 0.001
GO:0017124	SH3 domain binding	< 0.001	< 0.001
GO:0004252	serine-type endopeptidase activity	< 0.001	< 0.001

Continued on next page

Table D.15 – *continued from previous page*

GO Term	Description	Weight01 (Fisher)	Parent-Child (Fisher)
GO:0005509	calcium ion binding	< 0.001	< 0.001
GO:0008083	growth factor activity	< 0.001	< 0.001
GO:0031418	L-ascorbic acid binding	< 0.001	< 0.001
GO:0042803	protein homodimerization activity	< 0.001	< 0.001
GO:0004722	protein serine/threonine phosphatase activity	< 0.001	< 0.001
GO:0043565	sequence-specific DNA binding	0.031	< 0.001
GO:0005200	structural constituent of cytoskeleton	< 0.001	< 0.001
GO:0046983	protein dimerization activity	0.018	< 0.001
GO:0008321	Ral guanyl-nucleotide exchange factor activity	< 0.001	< 0.001
GO:0000287	magnesium ion binding	< 0.001	< 0.001
GO:0004348	glucosylceramidase activity	< 0.001	< 0.001
GO:0005178	integrin binding	< 0.001	< 0.001
GO:0005212	structural constituent of eye lens	< 0.001	< 0.001
GO:0016901	oxidoreductase activity, acting on the CH-OH group of donors, quinone or similar compound as acceptor	0.036	< 0.001
GO:0004047	aminomethyltransferase activity	0.003	< 0.001
GO:0004820	glycine-tRNA ligase activity	0.002	< 0.001
GO:0001227	transcriptional repressor activity, RNA polymerase II transcription regulatory region sequence-specific binding	0.015	< 0.001
GO:0004990	oxytocin receptor activity	0.007	< 0.001
GO:0005525	GTP binding	< 0.001	< 0.001
GO:0004326	tetrahydrofolylpolyglutamate synthase activity	0.001	0.002
GO:0004721	phosphoprotein phosphatase activity	0.037	0.002
GO:0004126	cytidine deaminase activity	0.003	0.002
GO:0003682	chromatin binding	< 0.001	0.002
GO:0004511	tyrosine 3-monooxygenase activity	< 0.001	0.002
GO:0004713	protein tyrosine kinase activity	< 0.001	0.002
GO:0004004	ATP-dependent RNA helicase activity	0.003	0.003
GO:0051015	actin filament binding	< 0.001	0.003
GO:0004012	phospholipid-translocating ATPase activity	0.034	0.004
GO:0003774	motor activity	< 0.001	0.004
GO:0008349	MAP kinase kinase kinase kinase activity	0.012	0.004
GO:0004359	glutaminase activity	0.014	0.005
GO:0030144	alpha-1,6-mannosylglycoprotein 6-beta-N-acetylglucosaminyltransferase activity	0.016	0.005
GO:0031071	cysteine desulfurase activity	0.001	0.006
GO:0030507	spectrin binding	0.002	0.006
GO:0004016	adenylate cyclase activity	0.016	0.006
GO:0005030	neurotrophin receptor activity	0.006	0.006
GO:0008158	hedgehog receptor activity	0.006	0.006
GO:0004843	thiol-dependent ubiquitin-specific protease activity	0.050	0.007
GO:0005506	iron ion binding	< 0.001	0.007
GO:0008017	microtubule binding	< 0.001	0.008
GO:0001609	G-protein coupled adenosine receptor activity	0.006	0.009
GO:0004996	thyroid-stimulating hormone receptor activity	0.002	0.009
GO:0015098	molybdate ion transmembrane transporter activity	0.003	0.009
GO:0046982	protein heterodimerization activity	< 0.001	0.010

Continued on next page

Table D.15 – *continued from previous page*

GO Term	Description	Weight01 (Fisher)	Parent-Child (Fisher)
GO:0004181	metallocarboxypeptidase activity	< 0.001	0.010
GO:0016712	oxidoreductase activity, acting on paired donors, with incorporation or reduction of molecular oxygen, reduced flavin or flavoprotein as one donor, and incorporation of one atom of oxygen	0.007	0.012
GO:0043395	heparan sulfate proteoglycan binding	0.016	0.013
GO:0030246	carbohydrate binding	0.003	0.016
GO:0004579	dolichyl-diphosphooligosaccharide-protein glycotransferase activity	< 0.001	0.018
GO:0001594	trace-amine receptor activity	0.006	0.020
GO:0003708	retinoic acid receptor activity	0.020	0.021
GO:0019825	oxygen binding	0.015	0.022
GO:0001078	transcriptional repressor activity, RNA polymerase II core promoter proximal region sequence-specific binding	< 0.001	0.023
GO:0030429	kynureninase activity	< 0.001	0.024
GO:0008430	selenium binding	0.020	0.025
GO:0004030	aldehyde dehydrogenase [NAD(P)+] activity	0.030	0.026
GO:0017147	Wnt-protein binding	0.023	0.026
GO:0004329	formate-tetrahydrofolate ligase activity	0.012	0.030
GO:0008113	peptide-methionine (S)-S-oxide reductase activity	0.014	0.032
GO:0004712	protein serine/threonine/tyrosine kinase activity	0.015	0.032
GO:0003735	structural constituent of ribosome	0.002	0.036
GO:0004672	protein kinase activity	< 0.001	0.039
GO:0008237	metallopeptidase activity	0.040	0.040
GO:0008093	cytoskeletal adaptor activity	0.016	0.042
GO:0008386	cholesterol monooxygenase (side-chain-cleaving) activity	0.001	0.045
GO:0000980	RNA polymerase II distal enhancer sequence-specific DNA binding	< 0.001	0.048
GO:0038062	protein tyrosine kinase collagen receptor activity	0.023	0.048

Appendix E

Trait Mappings

This appendix contains tables of the trait mappings required to use QTLSearch on each of the datasets, as described in Chapter 7.

E.1 Trait Mappings from Lisec *et al.* [LSM+09]

Table E.1: Mapping from metabolite to GO and / or ChEBI terms for the dataset from Lisec *et al.* [LSM+09].

	GO		ChEBI	
Trait	*Term*	*Name*	*Term*	*Name*
4-aminobutyric acid	GO:0009449	gamma-aminobutyric acid biosynthetic process	CHEBI:16865	gamma-aminobutyric acid
alpha-tocopherol	GO:0010189	vitamin E biosynthetic process	CHEBI:22470	alpha-tocopherol
ascorbic acid	GO:0019853	L-ascorbic acid biosynthetic process	CHEBI:22652	ascorbic acid
aspartic acid	GO:0006532	aspartate biosynthetic process	CHEBI:22660	aspartic acid
beta-alanine	GO:0019483	beta-alanine biosynthetic process	CHEBI:16958	beta-alanine
cellobiose	GO:2000891	cellobiose metabolic process	CHEBI:17057	cellobiose
cholesterol	GO:0006695	cholesterol biosynthetic process	CHEBI:16113	cholesterol
citrulline	GO:0000052	citrulline metabolic process	CHEBI:18211	citrulline
fructose	GO:0046370	fructose biosynthetic process	CHEBI:28757	fructose
fructose 6-phosphate	GO:0046370	fructose biosynthetic process	CHEBI:88003	fructose 6-phosphate
fucosterol	GO:0016126	sterol biosynthetic process	CHEBI:27865	fucosterol
galactonic acid	GO:0034192	D-galactonate metabolic process	CHEBI:24149	galactonic acid
galactose	GO:0046369	galactose biosynthetic process	CHEBI:28260	galactose

Continued on next page

Table E.1 – *continued from previous page*

	GO		ChEBI	
Trait	*Term*	*Name*	*Term*	*Name*
glucose	GO:0006094	gluconeogenesis	CHEBI:17234	glucose
glucose 6-phosphate	GO:0006094	gluconeogenesis	CHEBI:14314	D-glucose 6-phosphate
glycerol	GO:0006114	glycerol biosynthetic process	CHEBI:17754	glycerol
glycerol 3-phosphate	GO:0006114	glycerol biosynthetic process		
glycine	GO:0006545	glycine biosynthetic process	CHEBI:15428	glycine
hydroxyproline	GO:0019472	4-hydroxyproline biosynthetic process	CHEBI:24741	hydroxyproline
inositol	GO:0006021	inositol biosynthetic process	CHEBI:24848	inositol
lysine	GO:0009085	lysine biosynthetic process	CHEBI:25094	lysine
methionine	GO:0071265	L-methionine biosynthetic process	CHEBI:16811	methionine
nicotinic acid	GO:1901849	nicotinate biosynthetic process	CHEBI:15940	nicotinic acid
phenylalanine	GO:0009094	L-phenylalanine biosynthetic process	CHEBI:28044	phenylalanine
proline	GO:0055129	L-proline biosynthetic process	CHEBI:26271	proline
raffinose	GO:0033529	raffinose biosynthetic process	CHEBI:16634	raffinose
salicylic acid	GO:0009697	salicylic acid biosynthetic process	CHEBI:16914	salicylic acid
serine	GO:0006564	L-serine biosynthetic process	CHEBI:17822	serine
sinapic acid (cis)	GO:0033497	sinapate biosynthetic process	CHEBI:76350	cis-sinapic acid
sinapic acid (trans)	GO:0033497	sinapate biosynthetic process	CHEBI:15714	trans-sinapic acid
sucrose	GO:0005986	sucrose biosynthetic process	CHEBI:17992	sucrose
threonine	GO:0009088	threonine biosynthetic process	CHEBI:26986	threonine
trehalose	GO:0005992	trehalose biosynthetic process	CHEBI:27082	trehalose
tyrosine	GO:0006571	tyrosine biosynthetic process	CHEBI:18186	tyrosine
xylose	GO:0042842	D-xylose biosynthetic process	CHEBI:18222	xylose

E.2 Trait Mappings from Gong *et al.* [GCG+13]

Table E.2: Mapping from metabolite to GO and / or ChEBI terms for the dataset from Gong *et al.* [GCG+13].

	GO		ChEBI	
Trait	*Term*	*Name*	*Term*	*Name*
(+)-dehydrovomifoliol	GO:0016114	terpenoid biosynthetic process	CHEBI:4372	(6S)-dehydrovomifoliol
(+)-threo-9,10-dihydroxystearic acid	GO:0006633	fatty acid biosynthetic process	CHEBI:49254	(S,S)-9,10-dihydroxyoctadecanoic acid
12-hydroxyarachidonic acid	GO:0006633	fatty acid biosynthetic process	CHEBI:19138	12-HETE
16-hydroxy-hexadecanoic acid	GO:0006633	fatty acid biosynthetic process	CHEBI:55328	juniperic acid
2″, 6″-o-diacetyloninin	GO:0009813	flavonoid biosynthetic process		
24-hydroxytetracosanoic acid	GO:0006633	fatty acid biosynthetic process	CHEBI:76930	omega-hydroxytetracosanoic acid
3′, 4′, 5′-dihydrotricetin o-hexosyl-o-hexoside	GO:0009813	flavonoid biosynthetic process		
3, 5, 7-trihydroxy-6-methoxy-4′-prenyloxyflavone	GO:0009813	flavonoid biosynthetic process		
3-ketosphinganine	GO:0006633	fatty acid biosynthetic process		
4′-o-methylpuerarin	GO:0009813	flavonoid biosynthetic process		
4-geranyloxy-5-methyl coumarin	GO:0009805	coumarin biosynthetic process		
5-caffeoylquinic acid methyl ester	GO:0009699	phenylpropanoid biosynthetic process		
5-hydroxy-l-tryptophan	GO:0000162	tryptophan biosynthetic process	CHEBI:17780	5-hydroxy-L-tryptophan
6,8-dihydroxy-5,7-dimethoxycoumarin	GO:0009805	coumarin biosynthetic process		
6-prenylnaringenin	GO:0009813	flavonoid biosynthetic process	CHEBI:27566	6-prenylnaringenin
9,10-epoxy-18-hydroxy-octadecanoic acid	GO:0006633	fatty acid biosynthetic process		

Continued on next page

Table E.2 – ***continued from previous page***

Trait	GO *Term*	GO *Name*	ChEBI *Term*	ChEBI *Name*
akd 2b1	GO:0006633	fatty acid biosynthetic process		
acetosyringone	GO:0072391	phenol biosynthetic process	CHEBI:2404	acetosyringone
acteoside	GO:0072391	phenol biosynthetic process	CHEBI:132853	acteoside
aliarin	GO:0009813	flavonoid biosynthetic process		
apigenin c-pentoside	GO:0009813	flavonoid biosynthetic process	CHEBI:131755	apigenin C-pentoside
axillarin	GO:0009813	flavonoid biosynthetic process	CHEBI:2941	axillarin
ayanin	GO:0009813	flavonoid biosynthetic process		
c-hexosyl-c-pentosyl-apigenin	GO:0009813	flavonoid biosynthetic process		
c-hexosyl-apigenin o-caffeoylhexoside	GO:0009813	flavonoid biosynthetic process		
c-hexosyl-apigenin o-hexosyl-o-hexoside	GO:0009813	flavonoid biosynthetic process		
c-hexosyl-apigenin o-hexosyl-o-hexosyl-o-hexoside	GO:0009813	flavonoid biosynthetic process		
c-hexosyl-apigenin o-p-coumaroylhexoside	GO:0009813	flavonoid biosynthetic process		
c-hexosyl-chrysin o-feruloylhexoside	GO:0009813	flavonoid biosynthetic process		
c-hexosyl-chrysoeriol o-p-coumaroylhexoside	GO:0009813	flavonoid biosynthetic process		
c-hexosyl-chrysoeriol o-feruloylhexoside	GO:0009813	flavonoid biosynthetic process		
c-hexosyl-chrysoeriol o-hexoside	GO:0009813	flavonoid biosynthetic process		

Continued on next page

Table E.2 – ***continued from previous page***

Trait	GO *Term*	GO *Name*	ChEBI *Term*	ChEBI *Name*
c-hexosyl-luteolin o-hexoside	GO:0009813	flavonoid biosynthetic process		
c-hexosyl-luteolin o-p-coumaroylhexoside	GO:0009813	flavonoid biosynthetic process		
c-hexosyl-luteolin o-pentoside	GO:0009813	flavonoid biosynthetic process		
c-hexosyl-methylchrysoeriol	GO:0009813	flavonoid biosynthetic process		
c-hexosyl-naringenin o-hexosyl-o-hexoside	GO:0009813	flavonoid biosynthetic process		
c-hexosyl-naringenin o-p-coumaroylhexoside	GO:0009813	flavonoid biosynthetic process		
c-pentosyl-apeignin o-feruloylhexoside	GO:0009813	flavonoid biosynthetic process		
c-pentosyl-apigenin o-rutinoside	GO:0009813	flavonoid biosynthetic process		
c-pentosyl-apigenin o-caffeoylhexoside	GO:0009813	flavonoid biosynthetic process		
c-pentosyl-apigenin o-hexoside	GO:0009813	flavonoid biosynthetic process		
c-pentosyl-apigenin o-p-coumaroylhexoside	GO:0009813	flavonoid biosynthetic process		
c-pentosyl-chrysoeriol o-feruloylhexoside	GO:0009813	flavonoid biosynthetic process		
c-pentosyl-chrysoeriol o-hexoside	GO:0009813	flavonoid biosynthetic process		
c-pentosyl-luteolin o-hexoside	GO:0009813	flavonoid biosynthetic process		
c-rhamnosyl-apigenin o-hexoside	GO:0009813	flavonoid biosynthetic process		

Continued on next page

Table E.2 – ***continued from previous page***

Trait	GO *Term*	GO *Name*	ChEBI *Term*	ChEBI *Name*
cafestol	GO:0009813	flavonoid biosynthetic process	CHEBI:3291	cafestol
caohuoside d	GO:0046246	terpene biosynthetic process		
chryso-obtusin-o-hexoside	GO:0009813	flavonoid biosynthetic process		
chrysoeriol 5-o-hexoside	GO:0009813	flavonoid biosynthetic process		
chrysoeriol 7-o-hexoside	GO:0009813	flavonoid biosynthetic process		
chrysoeriol c-hexoside	GO:0009813	flavonoid biosynthetic process		
chrysoeriol c-hexoside derivative	GO:0009813	flavonoid biosynthetic process		
chrysoeriol o-malonyhexoside	GO:0009813	flavonoid biosynthetic process		
chrysoeriol o-rutinoside	GO:0009813	flavonoid biosynthetic process		
crotonoside	GO:0042451	purine nucleoside biosynthetic process	CHEBI:3927	Crotonoside
cyanidin 3-o-pentoside	GO:0009718	anthocyanin-containing compound biosynthetic process		
cymarin	GO:0046246	terpene biosynthetic process	CHEBI:4037	Cymarin
daidzein o-hexoside	GO:0009813	flavonoid biosynthetic process		
delphinidin o-hexoside	GO:0009718	anthocyanin-containing compound biosynthetic process		
deoxyguanosine	GO:0009163	nucleoside biosynthetic process		
ephemeranthoside	GO:0046246	terpene biosynthetic process		
epicatechin o-hexoside	GO:0009813	flavonoid biosynthetic process		
eriodictyol c-hexoside	GO:0009813	flavonoid biosynthetic process		
fructose 1,6-diphosphate	GO:0046370	fructose biosynthetic process		
gibberellin a12	GO:0009686	gibberellin biosynthetic process	CHEBI:30088	gibberellin A12
gibberellin a15	GO:0009686	gibberellin biosynthetic process	CHEBI:29590	gibberellin A15 (diacid form)

Continued on next page

Table E.2 – ***continued from previous page***

	GO		ChEBI	
Trait	*Term*	*Name*	*Term*	*Name*
gibberellin a53	GO:0009686	gibberellin biosynthetic process	CHEBI:27433	gibberellin A53
kaempferol derivative	GO:0009813	flavonoid biosynthetic process	CHEBI:28499	kaempferol
kievitone	GO:0009813	flavonoid biosynthetic process	CHEBI:16832	kievitone
kolavic acid	GO:0046246	terpene biosynthetic process		
lpc(1-acyl 12:1)	GO:0045017	glycerolipid biosynthetic process		
lpc(1-acyl 14:1)	GO:0045017	glycerolipid biosynthetic process	CHEBI:67054	lysophosphatidylcholine 14:1
lpc(1-acyl 16:0)	GO:0045017	glycerolipid biosynthetic process	CHEBI:64563	lysophosphatidylcholine 16:0
lpc(1-acyl 16:1)	GO:0045017	glycerolipid biosynthetic process	CHEBI:64560	lysophosphatidylcholine 16:1
lpc(1-acyl 16:2)	GO:0045017	glycerolipid biosynthetic process	CHEBI:67055	lysophosphatidylcholine 16:2
lpc(1-acyl 18:0)	GO:0045017	glycerolipid biosynthetic process	CHEBI:64561	lysophosphatidylcholine 18:0
lpc(1-acyl 18:2)	GO:0045017	glycerolipid biosynthetic process	CHEBI:64549	lysophosphatidylcholine 18:2
lpc(1-acyl 20:4)	GO:0045017	glycerolipid biosynthetic process	CHEBI:64568	lysophosphatidylcholine 20:4
lpc(1-acyl 24:4)	GO:0045017	glycerolipid biosynthetic process		
leu-ala-gly-lys	GO:0006520	cellular amino acid metabolic process		
leu-pro	GO:0006520	cellular amino acid metabolic process	CHEBI:73580	Leu-Pro
n2, n2-dimethyguanosine	GO:0009163	nucleoside biosynthetic process		
nicotianamine	GO:0030418	nicotianamine biosynthetic process	CHEBI:25520	nicotianamine
o-acetyl-l-serine	GO:0006520	cellular amino acid metabolic process	CHEBI:17981	O-acetyl-L-serine
o-malonylhexoside derivative	GO:0009813	flavonoid biosynthetic process		

Continued on next page

Table E.2 – ***continued from previous page***

Trait	GO *Term*	GO *Name*	ChEBI *Term*	ChEBI *Name*
o-methylchrysoeriol o-hexoside	GO:0009813	flavonoid biosynthetic process		
o-methylquercetin o-hexoside	GO:0009813	flavonoid biosynthetic process		
o-methylapigenin c-hexoside	GO:0009813	flavonoid biosynthetic process		
o-methylapigenin c-pentoside	GO:0009813	flavonoid biosynthetic process		
o-methylnaringenin c-pentoside	GO:0009813	flavonoid biosynthetic process		
phytocassane a	GO:0046246	terpene biosynthetic process	CHEBI:72664	(+)-phytocassane A
phytocassane c	GO:0046246	terpene biosynthetic process	CHEBI:72668	(+)-phytocassane C
polygodial	GO:0006694	steroid biosynthetic process	CHEBI:8305	Polygodial
spinacetin	GO:0009813	flavonoid biosynthetic process		
succinyladenosine	GO:0046086	adenosine biosynthetic process	CHEBI:71169	succinyladenosine
sucrose	GO:0005986	sucrose biosynthetic process	CHEBI:17992	sucrose
tricin 4′-o-(syringyl alcohol)ether	GO:0009813	flavonoid biosynthetic process		
tricin 4′-o-(syringyl alcohol)ether o-hexoside	GO:0009813	flavonoid biosynthetic process		
tricin 4′-o-(syringyl alcohol)ether derivative	GO:0009813	flavonoid biosynthetic process		
tricin 5-o-hexoside	GO:0009813	flavonoid biosynthetic process		
tricin 7-o-hexoside	GO:0009813	flavonoid biosynthetic process		
tricin o-hexoside derivative	GO:0009813	flavonoid biosynthetic process		
tricin o-hexosyl-o-hexoside	GO:0009813	flavonoid biosynthetic process		
tricin o-malonylhexoside	GO:0009813	flavonoid biosynthetic process		

Continued on next page

Table E.2 – ***continued from previous page***

Trait	GO *Term*	GO *Name*	ChEBI *Term*	ChEBI *Name*
tricin o-malonylhexoside derivative	GO:0009813	flavonoid biosynthetic process		
tricin o-rhamnosyl-o-malonylhexoside	GO:0009813	flavonoid biosynthetic process		
tricin o-rutinoside	GO:0009813	flavonoid biosynthetic process		
tricin o-sinapoylpentoside	GO:0009813	flavonoid biosynthetic process		
tricin derivative	GO:0009813	flavonoid biosynthetic process		
tricin-o-glucoside derivative	GO:0009813	flavonoid biosynthetic process		
tricin-o-hexoside derivative	GO:0009813	flavonoid biosynthetic process		
vitamin a	GO:0035238	vitamin A biosynthetic process	CHEBI:12777	vitamin A
vitamin b2	GO:0009231	riboflavin biosynthetic process	CHEBI:17015	riboflavin
di-c,c-hexosyl-apigenin	GO:0009813	flavonoid biosynthetic process		
di-c,c-hexosyl-apigenin derivative	GO:0009813	flavonoid biosynthetic process		
di-c,c-hexosyl-chrysoeriol	GO:0009813	flavonoid biosynthetic process		
di-c,c-hexosyl-luteolin	GO:0009813	flavonoid biosynthetic process		
di-c,c-pentosyl-apigenin	GO:0009813	flavonoid biosynthetic process		
di-c,c-pentosyl-luteolin	GO:0009813	flavonoid biosynthetic process		