Year : 2019

Inferring Hierarchical Orthologous Groups

Train Clément

UNIL | Université de Lausanne

Faculté de biologie
et de médecine

Département de biologie computationnelle

Inferring Hierarchical Orthologous Groups

Thèse de doctorat ès sciences de la vie (PhD)

Présentée à la

Faculté de biologie et de médecine

de l'Université de Lausanne

par

Clément-Marie Train

Master de bioinformatique de l'université de Bordeaux (2015)

Jury

Prof. Edward Farmer, Président

Prof. Paul D. Thomas, Expert

Prof. Robert Waterhouse, Expert

Prof. Christophe Dessimoz, Directeur de thèse

Prof. Jérome Goudet, Co-directeur de thèse

Lausanne

2019

UNIL | Université de Lausanne

Faculté de biologie
et de médecine

Département de biologie computationnelle

Inferring Hierarchical Orthologous Groups

Thèse de doctorat ès sciences de la vie (PhD)

Présentée à la

Faculté de biologie et de médecine

de l’Université de Lausanne

par

Clément-Marie Train

Master de bioinformatique de l’université de Bordeaux (2015

Jury

Prof. Edward Farmer, President

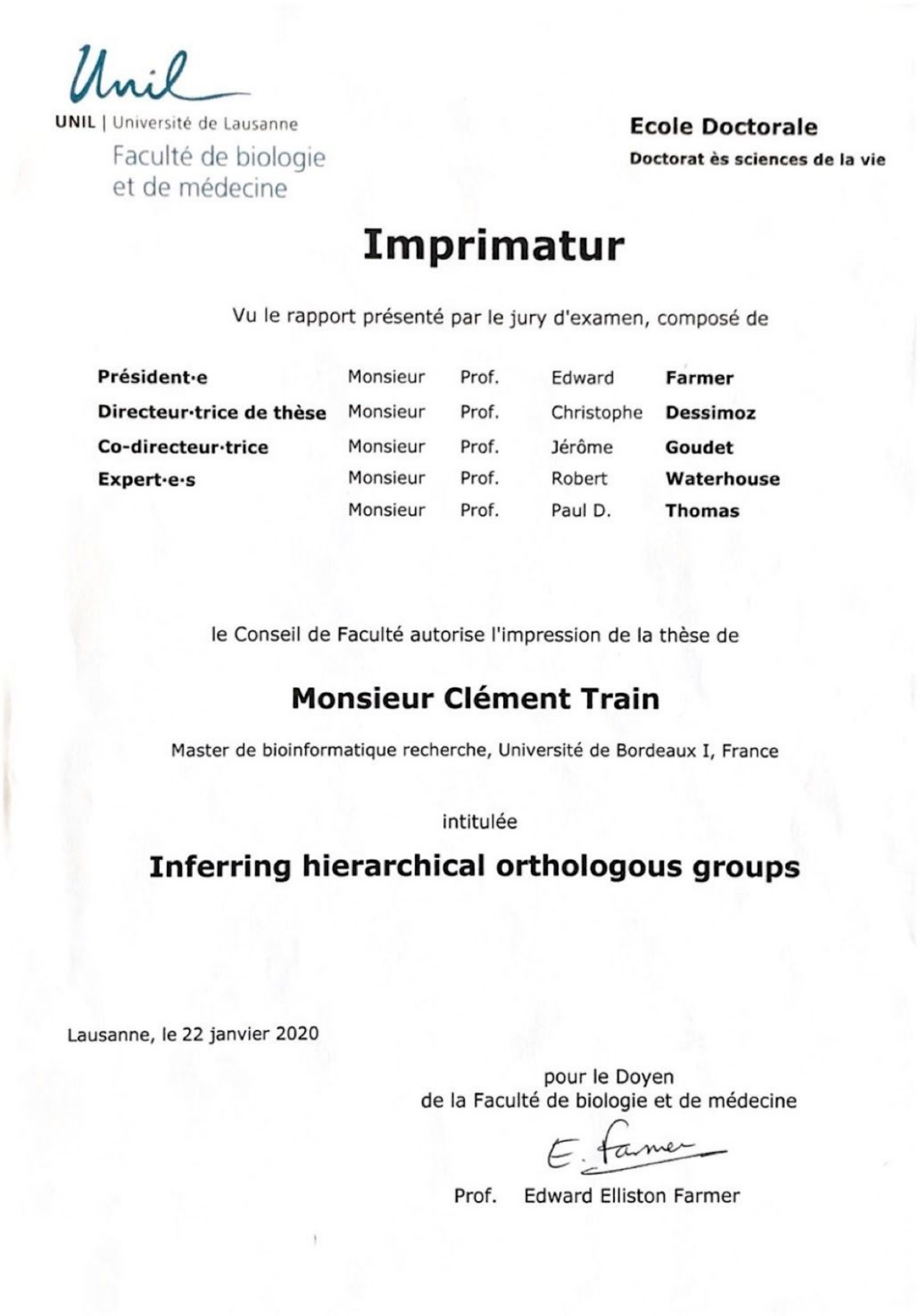
Prof. Paul D. Thomas, Expert

Prof. Robert Waterhouse, Expert

Prof. Christophe Dessimoz, Directeur de thèse

Prof. Jérome Goudet, Co-directeur de thèse

Lausanne
2019



Unil

UNIL | Université de Lausanne

Faculté de biologie
et de médecine

Ecole Doctorale

Doctorat ès sciences de la vie

Imprimatur

Vu le rapport présenté par le jury d'examen, composé de

Président·e	Monsieur	Prof.	Edward	**Farmer**
Directeur·trice de thèse	Monsieur	Prof.	Christophe	**Dessimoz**
Co-directeur·trice	Monsieur	Prof.	Jérôme	**Goudet**
Expert·e·s	Monsieur	Prof.	Robert	**Waterhouse**
	Monsieur	Prof.	Paul D.	**Thomas**

le Conseil de Faculté autorise l'impression de la thèse de

Monsieur Clément Train

Master de bioinformatique recherche, Université de Bordeaux I, France

intitulée

Inferring hierarchical orthologous groups

Lausanne, le 22 janvier 2020

pour le Doyen
de la Faculté de biologie et de médecine

E. farmer

Prof. Edward Elliston Farmer

Abstract

The reconstruction of ancestral evolutionary histories is the cornerstone of most phylogenetic analyses. Many applications are possible once the evolutionary history is unveiled, such as identifying taxonomically restricted genes (genome barcoding), predicting the function of unknown genes based on their evolutionary related genes gene ontologies, identifying gene losses and gene gains among gene families, or pinpointing the time in evolution where particular gene families emerge (sometimes referred to as “phylostratigraphy”). Typically, the reconstruction of the evolutionary histories is limited to the inference of evolutionary relationships (homology, orthology, paralogy) and basic clustering of these orthologs. In this thesis, we adopted the concept of Hierarchical Orthology Groups (HOGs), introduced a decade ago, and proposed several improvements both to improve their inference and to use them in biological analyses such as the aforementioned applications. In addition, HOGs are a powerful framework to investigate ancestral genomes since HOGs convey information regarding gene family evolution (gene losses, gene duplications or gene gains). In this thesis, an ancestral genome at a given taxonomic level denotes the last common ancestor genome for the related taxon and its hypothetical ancestral gene composition and gene order (synteny). The ancestral genes composition and ancestral synteny for a given ancestral genome provides valuable information to study the genome evolution in terms of genomic rearrangement (duplication, translocation, deletion, inversion) or of gene family evolution (variation of the gene function, accelerate gene evolution, duplication rich clade). This thesis identifies three major open challenges that composed my three research arcs. First, inferring HOGs is complex and computationally demanding meaning that robust and scalable algorithms are mandatory to generate good quality HOGs in a reasonable time. Second, benchmarking orthology clustering without knowing the true evolutionary history is a difficult task, which requires appropriate benchmark strategies. And third, the lack of tools to handle HOGs limits their applications.

In the first arc of the thesis, I proposed two new algorithm refinements to improve orthology inference in order to produce orthologs less sensitive to gene fragmentations and imbalances in the rate of evolution among paralogous copies. In addition, I introduced version 2.0 of the GETHOGs 2.0 algorithm, which infers HOGs in a bottom up fashion, and which has been shown to be both faster and more accurate.

In the second arc, I proposed new strategies to benchmark the reconstruction of gene families using detailed cases studies based on evidence from multiple sequence alignments along with reconstructed gene trees, and to benchmark orthology using a simulation framework that provides full control of the evolutionary genomic setup. This work highlights the main challenges in current methods.

Third, I created pyHam (**py**thon **H**OG **a**nalysis **m**ethod), iHam (**i**nteractive **H**OG **a**nalysis **m**ethod) and GTM (**G**raph - **T**ree - **M**ultiple sequence alignment)—a collection of tools to process, manipulate and visualise HOGs. pyHam offers an easy way to handle and work with HOGs using simple python coding. Embedded at its heart are two visualisation tools to synthesise HOG-derived information: iHam that allow interactive browsing of HOG structure and a tree based visualisation called tree profile that pinpoints evolutionary events induced by the HOGs on a species tree. In addition, I develop GTM an interactive web based visualisation tool that combine for a given gene family (or set of genes) the related sequences, gene tree and orthology graph.

In this thesis, I show that HOGs are a useful framework for phylogenetics, with considerable work done to produce robust and scalable inferences. Another important aspect is that our inferences are benchmarked using manual case studies and automated verification using simulation or reference Quest for Orthologs Benchmarks. Lastly, one of the major advances was the conception and implementation of tools to manipulate and visualise HOG. Such tools have already proven useful when investigating HOGs for developmental reasons or for downstream analysis.

Ultimately, the HOG framework is amenable to integration of all aspects which can reasonably be expected to have evolved along the history of genes and ancestral genome reconstruction.

Résumé

La reconstruction de l'histoire évolutive ancestrale est la pierre angulaire de la majorité des analyses phylogénétiques. Nombreuses sont les applications possibles une fois que l'histoire évolutive est révélée, comme l'identification de gènes restreints taxonomiquement (barcoding de génome), la prédiction de fonction pour les gènes inconnus en se basant sur les ontologies des gènes relatifs evolutionnairement, l'identification de la perte ou de l'apparition de gènes au sein de familles de gènes ou encore pour dater au cours de l'évolution l'apparition de famille de gènes (phylostratigraphie). Généralement, la reconstruction de l'histoire évolutive se limite à l'inférence des relations évolutives (homologie, orthologie, paralogie) ainsi qu'à la construction de groupes d'orthologues simples. Dans cette thèse, nous adoptons le concept des groupes hiérarchiques d'orthologues (HOGs en anglais pour Hierarchical Orthology Groups), introduit il y a plus de 10 ans, et proposons plusieurs améliorations tant bien au niveau de leurs inférences que de leurs utilisations dans les analyses biologiques susmentionnées. Cette thèse a pour but d'identifier les trois problématiques majeures qui composent mes trois axes de recherches. Premièrement, l'inférence des HOGs est complexe et nécessite une puissance computationnelle importante ce qui rend obligatoire la création d'algorithmes robustes et efficients dans l'espace temps afin de maintenir une génération de résultats de qualité rigoureuse dans un temps raisonnable. Deuxièmement, le contrôle de la qualité du groupement des orthologues est une tâche difficile si on ne connaît l'histoire évolutive réelle ce qui nécessite la mise en place de stratégies de contrôle de qualité adaptées. Tertio, le manque d'outils pour manipuler les HOGs limite leur utilisation ainsi que leurs applications.

Dans le premier axe de ma thèse, je propose deux nouvelles améliorations de l'algorithme pour l'inférence des orthologues afin de pallier à la sensibilité de l'inférence vis à vis de la fragmentation des gènes et de l'asymétrie du taux d'évolution au sein de paralogues. De plus,

j'introduis la version 2.0 de l'algorithme GETHOGs qui utilise une nouvelle approche de type 'bottom-up' afin de produire des résultats plus rapides et plus précis.

Dans le second axe, je propose de nouvelles stratégies pour contrôler la qualité de la reconstruction des familles de gènes en réalisant des études de cas manuels fondés sur des preuves apportées par des alignement multiples de séquences et des reconstructions d'arbres géniques, et aussi pour contrôler la qualité de l'orthologie en simulant l'évolution de génomes afin de pouvoir contrôler totalement le matériel génétique produit. Ce travail met en avant les principales problématiques des méthodes actuelles.

Dans le dernier axe, je montre pyHam, iHam et GTM - une panoplie d'outils que j'ai créée afin de faciliter la manipulation et la visualisation des HOGs en utilisant un programmation simple en python. Deux outils de visualisation sont directement intégrés au sein de pyHam afin de pouvoir synthétiser l'information véhiculée par les HOGs: iHam permet d'interactivement naviguer dans les HOGs ainsi qu'une autre visualisation appelée "tree profile" utilisant un arbre d'espèces où sont localisés les événements révolutionnaires contenus dans les HOGs. En sus, j'ai développé GTM un outil interactif web qui combine pour une famille de gènes donnée (ou un ensemble de gènes) leurs séquences alignées, leur arbre de gène ainsi que le graphe d'orthologie en relation.

Dans cette thèse, je montre que le concept des HOGs est utile à la phylogénétique et qu'un travail considérable a été réalisé dans le but d'améliorer leur inférences de façon robuste et rapide. Un autre point important est que la qualité de nos inférences soit contrôlée en réalisant des études de cas manuellement ou en utilisant le Quest for Orthologs Benchmark qui est une référence dans le contrôle de la qualité de l'orthologie. Dernièrement, une des avancée majeure proposée est la conception et l'implémentation d'outils pour visualiser et manipuler les HOGs. Ces outils s'avèrent déjà utilisés tant pour l'étude des HOGs dans un but d'amélioration de leur qualité que pour leur utilisation dans des analyses biologiques.

Pour conclure, on peut noter que tous les aspects qui semblent avoir évolué en relation avec l'histoire évolutive des gènes ou des génomes ancestraux peuvent être intégrés au concept des HOGs.

Acknowledgements

First, I would like to thank Christophe Dessimoz, my thesis supervisor. He gave me the opportunity to join his lab and to work on interesting research items. I'm grateful for the supervision he provided and the scientific framework he offered me to work with. I loved to work, to laugh and to discuss (or to listen more often) with him for the past 6 years.

Adrian Altenhoff is also a very important lab member for me that I would like to particularly thank here. During the last 6 years of supervision and collaboration, he provides me an incredible support and help for various aspects of my thesis. His patience with me was invaluable as much as his friendship.

Now, I want to thank all my friends and colleagues from the Dessimoz lab and the Salamin lab for the wonderful atmosphere they created and the amazing times we spent together: Anna, Natasha, David, Warwick, Victor Rosier, Marie-Pierre, Laurent, Arnaud, Theo, Sara, Amel, Raphael, Monique for the short list.

At last but not the least, I want to thank all my friends and family that contribute in various and diverses form to my happiness and give the strength to achieve what I have done.

Contents

Abbreviations

GETHOGs: Graph-based **E**fficient **T**echnique for **H**ierarchical **O**rthologous **G**roup**s**

GTM: Graph-**T**ree-**M**ultiple sequence alignment

HOGs: Hierarchical **O**rthologs **G**roup**s**

iHam: interactive **H**OGs **a**nalysis **m**ethod

pyHam: python **H**OGs **a**nalysis **m**ethod

OMA: Orthologous **Ma**trix

Chapter 1: Introduction

Investigating the evolutionary history underlying present day organisms is important to characterise the evolutionary relationships between extant species and to investigate the evolutionary mechanisms that shape genomes and genes resulting in extant organisms. By reconstructing the evolutionary histories of gene families between extant species, we aim to have a better insight of the ancestral states of genomes in terms of gene organisation and to unravel the phylogenetic complexity underlying current biodiversity. Evolutionary relationships have been proven to be a very useful resource for many applications such as identifying taxonomically restricted genes (which are often used for genome barcoding), for identifying genes related to taxon specific characters and functions, predicting the function of unknown genes based on their evolutionary related genes gene ontologies, verifying function conservation across related genes, phylogenetic profiling for groups of genes, identifying gene losses and gene gains among gene families, or computing phylostratigraphy. The applications that rely on evolutionary relationships are numerous and diverse and can be affected by spurious or missing evolutionary pairwise relations (Dalquen and Dessimoz 2013; Moreno-Hagelsieb and Latimer 2008) .

Nevertheless, reconstructing the evolutionary history of genes is not an easy task, many aspects can compromise and make the reconstruction more complicated . First, the intrinsic complexity and nesting of genomic events both at chromosomal level (Dalquen et al. 2013) (inversion, deletion, duplication) and genetic level (varying evolution rates, domain shuffling) can result in huge and complex gene families. Secondly, the quantity of available genomic information has exploded thanks to the major breakthrough of the past decades in sequencing technologies and computational tools, leading to an increase of the scale of genomic set up sizes (up to thousand of genomes with millions of proteins). The quality of the available genomic data is another important aspect. Indeed, sequencing and assembly errors can make the reconstruction of evolutionary history more complicated due to misleading signal. In order to reconstruct accurate gene families with their related evolutionary histories, phylogenetic methods need to be robust and scalable.

Homology

Homology is an important concept in phylogenetics to describe genes that share a common evolutionary history. As described in (Chothia and Lesk 1986), homologous genes tend to conserve similar protein folding although the sequence similarity may be decreasing due to sequence variations (mutation, deletion, insertion) during evolution. Indeed, modifications of the protein sequences is observed along evolution introducing protein structural and/or functional changes. These changes can be either deleterious for the organisms reducing their survival and reproduction by consequence the chance for the mutation to be fixed in a population (negative selection) or can be either beneficial for organisms meaning the individuals fitness increase in the population which increase the probability to fix the protein changes (positive selection) in the population. Nevertheless, mutations does not always have a beneficial or deleterious effect on protein function but rather are neutral which does not affect the survival and reproduction of the individuals but still include sequence variations. Neutral theory (Kimura 1983) states that the fixation of mutant alleles in a population are mostly due to randomness (random genetic drift). While deleterious mutation are obviously not fixed due to their negative effects on survival and reproduction of individuals, most of the other changes in protein sequences are not beneficial but rather neutral. The homologous sequences are composed of variable regions where sequence variations are more likely to happen and common structural cores with the highest percentage of sequence identity. The percentage of residue identity in common core regions is directly correlated with the similarity of the general protein folding, showing that conservation inside the core regions are playing an active role in maintaining a constant protein folding across gene family members. Many dynamic programming and heuristic methods have been proposed in the past decades to infer homology between sequences with their own benefits and drawbacks.

Dynamic programming approaches perform global (Needleman and Wunsch 1970) or local (Smith and Waterman 1981) sequence alignments by using a scoring matrix to score match and mismatch of characters in the alignment, as well as a gap penalty to account for gaps in the alignment. In protein alignments, amino acid matches and mismatches are scored using a substitution matrix. For DNA alignments, the use of a positive score for matches and a negative score for mismatch is commonly used instead of a substitution matrix. In order to reduce the number of gaps in the final alignment, a scoring variant can be used to increase the cost of gap opening compared to the cost of extending the gap itself. Even if dynamic programming approaches return the most optimal alignment, it is highly dependent on the chosen scoring function. Moreover, dynamic programming approaches may be time consuming for long proteins or large numbers of proteins.

Heuristics approaches such as BLAST (Altschul et al. 1990) rely on the fact that two sequences that share more similarities than expected by chance have a common ancestry and did not arise independently (William R. Pearson 2013). Contrary to dynamic programming approaches, heuristics searches are not guaranteed to return the best possible alignment but are faster to compute. Such methods use a statistical estimator to assess the significance of their search and the excess of similarity among amino acids. Tools like BLAST (Altschul et al. 1990), FASTA (Pearson 2000), SSEARCH (W. R. Pearson 2000) use the Expected value or E-value to describe the number of possible hits in the database that can occur by chance. The E-value is directly correlated to the size of the database and decreases exponentially with an increasing alignment score. The bigger the database, the greater the chance of finding a high scoring alignment by chance.

Orthology and Paralogy

Evolutionary relationships between homologs (Fitch 1970), can be subclassified into orthologs or paralogs depending on whether they started diverging by a speciation or a duplication event, respectively. Determining orthology is a fundamental step in many phylogenetic, functional and comparative studies. Indeed orthologs, sometimes denoted as “same genes in different species”, are good candidates to estimate differences and similarities among genomes or genes since organisms have diverged from one another by speciation. Pairwise orthologous relations are well suited to understand small genomic comparisons, e.g estimating the amount of shared orthologs between two genomes, but scale poorly to larger datasets (Gabaldón and Koonin 2013; Sonnhammer et al. 2014). Orthologs and paralogs also diverge regarding their function evolution; orthologs seems to be more conservative while paralogs are more likely to evolve freely (Altenhoff et al. 2012). The orthologue conjecture denotes the fact that orthologs tend to have the same or similar functions. At contrary, paralogs may diverge more regarding gene functions. Indeed, additional copies of a gene provide supplementary materials where sequences changes affecting the gene function may be less harmful than if only one gene copy exists.

Orthology (and paralogy) have been proven to be very useful for a wide range of application as shown in figure 1:

- Phylogenomics. Since orthologs are related through speciation events, they are good candidates to investigate species phylogeny. Marker genes, i.e. sets of orthologous genes highly conserved in a specific clade, can be used as a source of orthology signal for the inference of species trees. Orthology inference software such as OMA standalone (Altenhoff et al. 2019) has been used in phylogenomic studies to reconstruct and elucidate complex phylogenies such as that of centipedes (Fernández et al. 2014), arachnids (Prashant P. Sharma et al. 2014; Fernández and Giribet 2015), assassin flies (Dikow et al. 2017), scorpions (P. P. Sharma et al. 2015), spiders (Garrison et al. 2016), flatworms (Egger et al. 2015; Laumer, Hejnol, and Giribet 2015), tapeworms (Tsai et al. 2013), or Archaea (Williams et al. 2017).

- Predicting gene function. Orthologous genes tends to conserve similar function during evolution (Adrian M. Altenhoff et al. 2012) which is very useful to predict the function of an unknown gene within an orthologous group. Indeed, if genes within the same orthologous group have a similar function then it is likely that other unknown genes in this group have a similar function.
- Elucidating gene loss and duplication and finding taxonomically restricted genes. As shown in the next section, orthologs can be used to reconstruct complete gene families. Gene families contain all the information about gene duplication, gene loss and the apparition of new genes. Indeed, once the delimitation of orthologous groups is made at all levels of a gene family, it is trivial to infer the related gene duplication and gene loss events. In addition, the root of the gene family may be used to determine when the ancestral gene initiating this new family arose.
- Phylostratigraphy. Orthology may also be useful to investigate how and when genes arise. For example, genes ages of human proteins can determined as the age of the last common ancestor for a given orthologous groups (Liebeskind et al. 2016).
- Finding the best models systems. Depending on the physiological problem of interest, specific model systems are more relevant than others. For example, the ferret (Mustela putorius furo) is a better model organism when studying the human respiratory diseases than the animal mouse even though they have diverged earlier. Indeed, the protein divergence for between ferret-human orthologs is smaller than between human-mouse orthologs (Peng et al. 2014). This relies on the assumption that closely related genes may conserved more similar physiological processes.
- Verification of function conservation. In order to investigate on the orthologue conjecture stating that orthologs tend to conserve similar functions Edward Marcotte use orthologs

to design an in vivo experiment where yeast genes were replaced by their human orthologs. Results shown that 43% of 414 essential yeast genes can be replaced by they human orthologs (Kachroo et al. 2015).

- Phylogenetic profiling. Orthology can also be used as a source of signal for phylogenetic profiles. The ideas is to look if there is a pattern in the presence/absence of genes between species (Tran et al. 2018).

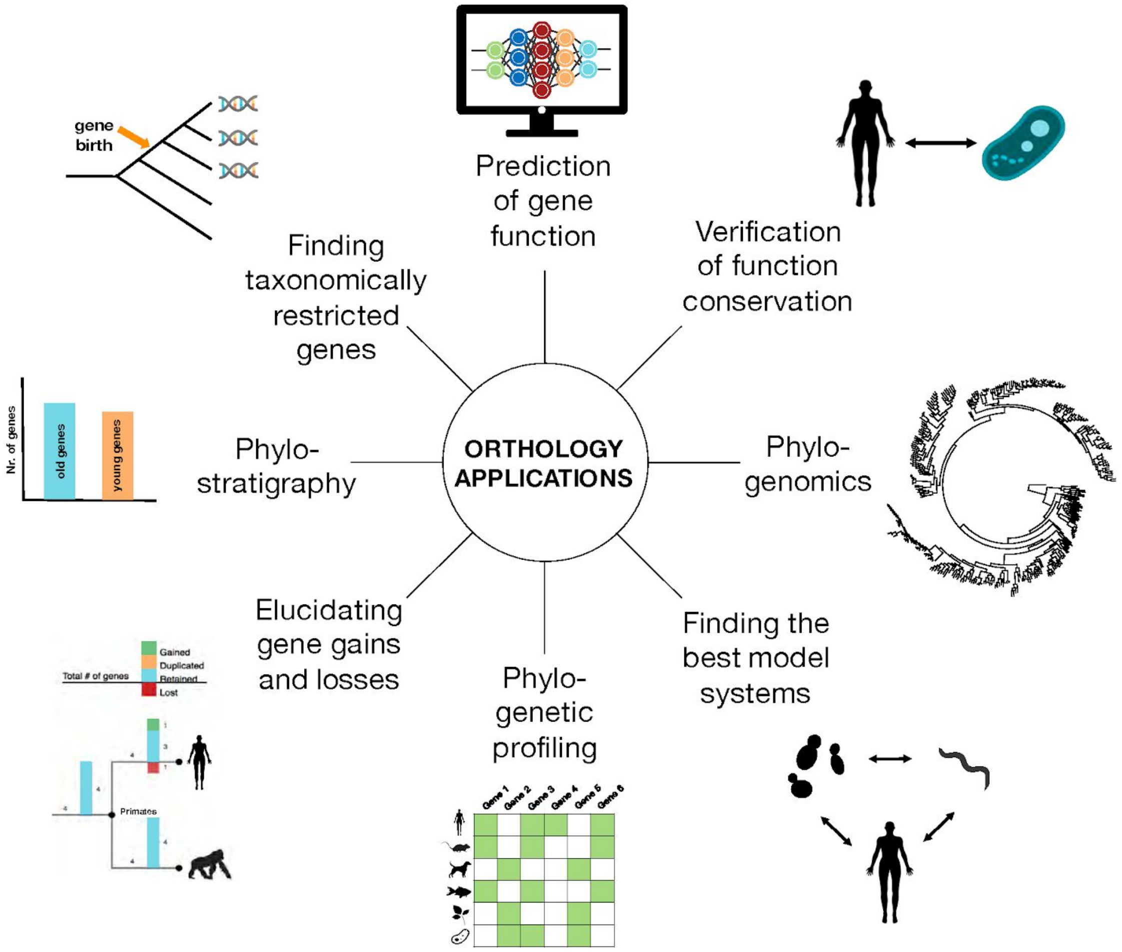
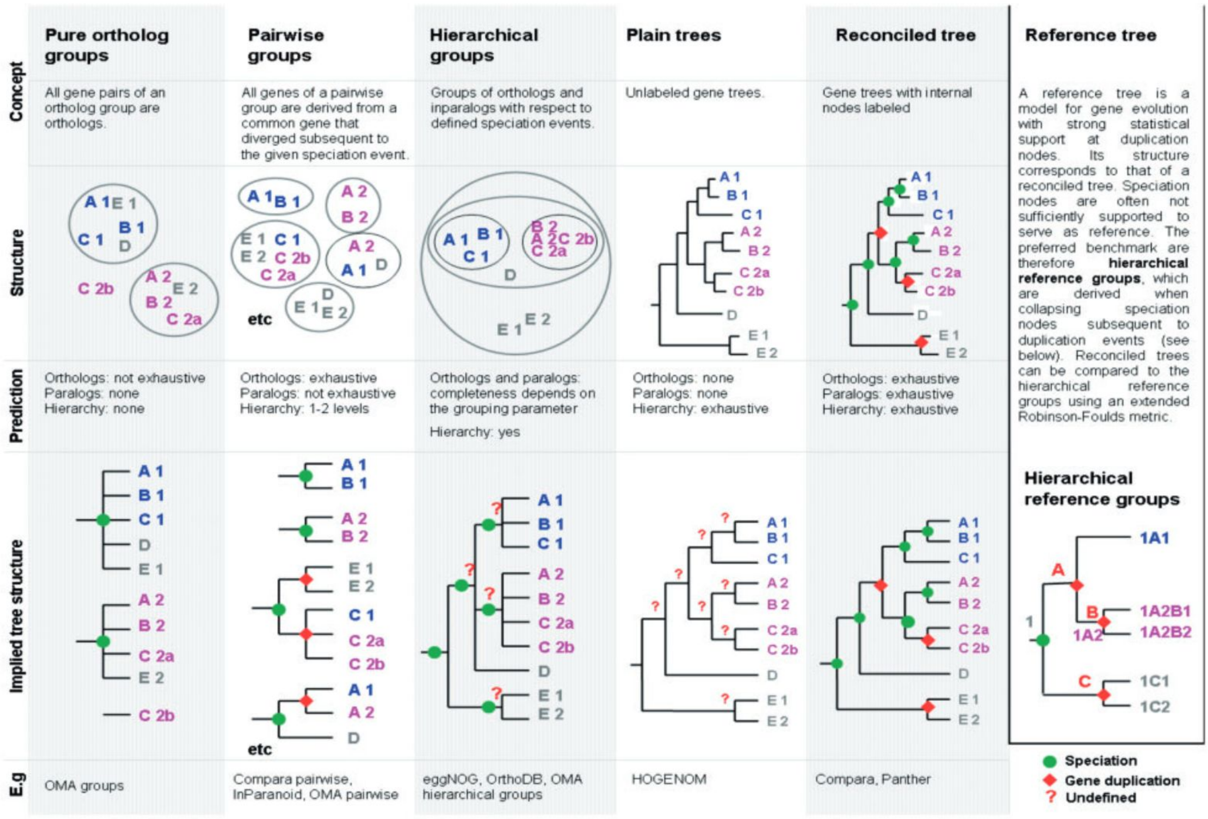


Figure 1: Applications of orthology. Orthology inferences can be used for many applications. Since orthologs shared common evolutionary history, all aspects related to the evolution of genes may be investigated under the prism of orthology. Kindly provided by (Glover et al. 2019).

Groups of Orthologs

In contrast to pairwise orthology, groups of orthologs scale better for multispecies comparative analysis. Such groups concentrate more orthology signal than pairwise orthology by integrating multiple genes across multiple species. Based on the concept used to define them, there are several types of orthologous groups (Boeckmann et al. 2011), each with its own particular structure and implied information.



	Pure ortholog groups	Pairwise groups	Hierarchical groups	Plain trees	Reconciled tree	Reference tree
Concept	All gene pairs of an ortholog group are orthologs.	All genes of a pairwise group are derived from a common gene that diverged subsequent to the given speciation event.	Groups of orthologs and inparalogs with respect to defined speciation events.	Unlabeled gene trees.	Gene trees with internal nodes labeled	A reference tree is a model for gene evolution with strong statistical support at duplication nodes. Its structure corresponds to that of a reconciled tree. Speciation nodes are often not sufficiently supported to serve as reference. The preferred benchmark are therefore **hierarchical reference groups**, which are derived when collapsing speciation nodes subsequent to duplication events (see below). Reconciled trees can be compared to the hierarchical reference groups using an extended Robinson-Foulds metric.
Prediction	Orthologs: not exhaustive Paralogs: none Hierarchy: none	Orthologs: exhaustive Paralogs: not exhaustive Hierarchy: 1-2 levels	Orthologs and paralogs: completeness depends on the grouping parameter Hierarchy: yes	Orthologs: none Paralogs: none Hierarchy: exhaustive	Orthologs: exhaustive Paralogs: exhaustive Hierarchy: exhaustive	
E.g	OMA groups	Compara pairwise, InParanoid, OMA pairwise	eggNOG, OrthoDB, OMA hierarchical groups	HOGENOM	Compara, Panther	

Figure 2: Concepts of selected orthology databases. Rows (from top to bottom) indicate the different database concepts, the structure of orthologous groups, the completeness of predicted gene relationships and the implied tree structures. The latter visualizes the captured phylogenetic information. Re-used with permission from (Boeckmann et al. 2011).

As shown in figure 2, we can catalog 6 types of orthologous groups:

- **Pure orthologous groups** *(described in more detail in the next section)***:** The pure orthologous group is defined as a set of genes in which all genes are orthologous to each other. In terms of orthology graphs, such groups are referred to as a clique, where all nodes (genes) of the clique are connected to each other. Such groups provide a non exhaustive list of orthologs only and paralogy information is absent. Indeed, they lack many to many orthologous relations (due to lineage specific duplications) because they have to choose to conserve only one representative gene per set of inparalogs. In OMA, this type of group is referred to as OMA Groups (Altenhoff et al. 2019).

- **Pairwise groups:** The pairwise group contains all the genes that descended from a single ancestral gene at a specific taxonomic range. In terms of labelled genes tree, it is composed of all the genes within a sub tree rooted by a speciation node of interest. In this group, the list of orthologs is exhaustive and integrates some paralogy information for inparalogs (all duplications that may have occurred after the speciation event of reference).
- **Hierarchical groups** (described in more detail below): These groups are composed of nested sets of genes, each composed of genes descending from the same speciation event. Each sub group represents an ancestral gene at a given taxonomic range. The nested structure of those orthologous groups conveys information about paralogy and duplications. Indeed, if two subgroups have the same taxonomic range for their related speciation event, a duplication event prior to this speciation is implied, meaning that the two sets of genes are paralogous. Hierarchical groups can be found in several publicly available orthology database such as eggNOG (Huerta-Cepas et al. 2016), OrthoDB (Waterhouse et al. 2013) and OMA (Roth et al. 2008).
- **Plain gene tree:** Plain trees (or unlabelled gene trees) only convey the exhaustive topology (or hierarchy) of the gene tree. The lack of duplication and speciation events as labels for internal nodes make the inference of orthology or paralogy impossible from these trees without reconciliation. The HOGENOM database is inferring plain trees for gene family topologies.

- **Reconciled tree**: Reconciled trees (or labeled gene trees) are plain trees with internal nodes labeled as duplication or speciation events. They are the most complete source of information regarding orthology, paralogy and topology. Several databases use them to store gene family histories, such as Ensembl Compara (Cunningham et al. 2015) or Panther (Mi et al. 2016).
- **Reference trees and groups**: Reference trees are reconciled tree like structures with strong statistical support at their duplication nodes. Nevertheless, speciation nodes may not be well supported. To fulfil this lack of confidence about speciation events, hierarchical reference groups where introduced. They correspond to reference trees with speciation nodes collapsed after duplication events. References trees and groups can be used as standard resources for benchmark purposes.

Pure orthologous groups

As described previously, pure orthologous groups are composed of sets of genes, all of which are orthologous to each other. These groups can be used as marker genes for phylogenetic reconstruction since they contain strong orthology signal for a given clade. Since orthologs tend to conserve function, orthologous groups are often used to perform gene orthology enrichment to investigate clade specific variation of gene functions. However, orthologous groups only provide rough information about a gene set without any precision about evolutionary events (e.g. duplications or gene losses) underlying the genes' evolutionary history. Indeed, only the information about presence and absence of genes is present. For example, if we observe two gene copies for two species in a group we cannot state if one ancestral gene duplicated before their speciation or if there were two species-specific duplications. This is restricting the downstream processing of orthologous groups to bulk analysis of genes without taking into account the underlying history the ancestral genes. Furthermore, it is hard or even practically impossible to analyse large gene families composed of up to thousands of members, for example to investigate the number of ancestral genes in mammals for a specific gene family.

Hierarchical Orthologous Groups

In order to tackle this problem, the concept of Hierarchical Orthologous Groups (HOGs) was introduced (van der Heijden et al. 2007; Jensen et al. 2008; Kriventseva et al. 2008). As illustrated in figure 3, HOGs can be defined as a set of genes, all descending from a single common ancestral gene at a given taxonomic range. Represented as a nested structure of orthologous and paralogous groups related to specific taxonomic ranges, each HOG contains the complete evolutionary history of a gene family. Such groups provide detailed information about ancestral gene states (e.g. number of ancestral genes at specific taxonomic ranges) or evolutionary events (e.g. when duplications or gene losses occurred). A one-to-one correspondence exists between HOGs and labelled gene trees; both contain the same information about speciation and duplication events; but they are encoded in different data structures: labeled gene trees are encoded in tree-like structures while HOGs are encoded in nested group structures.

In order to facilitate the storage and processing of HOGs, a standard format called OrthoXML is broadly used across orthology resources. OrthoXML is based on the classic XML format. OrthoXML is composed of two parts: a first mapping section that contains all genes grouped by species with related mapping information (unique OrthoXML ID, external ID, database provenance, etc..) and a second groups section that contains the nested orthologous groups.

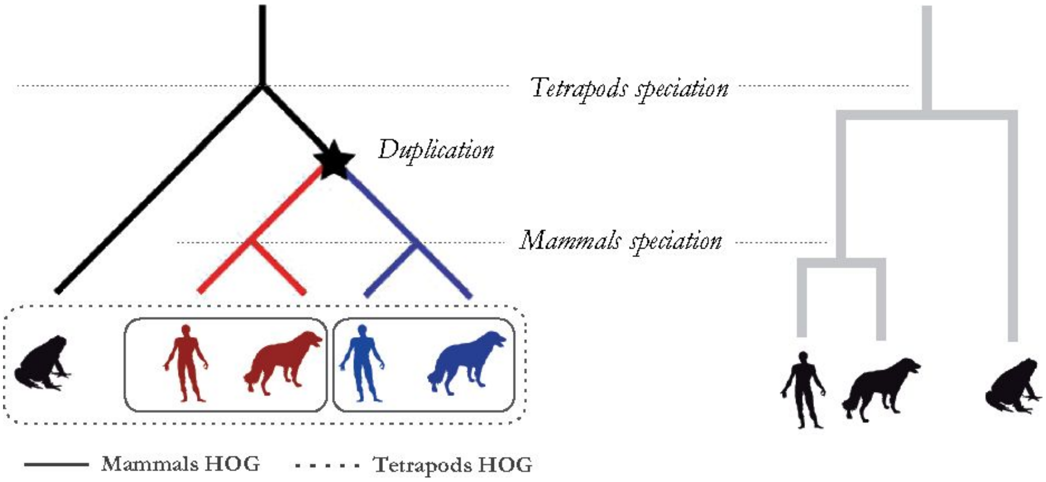


Figure 3: Hierarchical Orthologous Groups. Labeled gene tree (left) and its related species tree (right) illustrating the evolutionary history of five genes that all descended from a single common ancestral gene at the tetrapods level. These five genes called homologs can be classified as orthologs if they start diverging by speciation (human versus dog genes of same color) or as paralogs if they start diverging by duplication (blue versus red genes). We can identify in the example HOGs at two taxonomic levels: one larger HOG at the tetrapods level (dotted-line rectangle) containing all the homologous genes that emerged from the single tetrapod ancestral gene, and two HOGs at the mammalian level (solid-line rectangles), due to a duplication of the tetrapod ancestral gene before the mammals speciation.

From HOGs to ancestral genomes and ancestral synteny

When considered separately, HOGs are representing individuals gene families at a given taxonomic level. If we now consider all HOGs for one specific taxonomic range, we are not anymore dealing with independent gene families but with a sets of HOGs each representing ancestral genes that was all contained in a same ancestral genomes at this taxonomic range. In this thesis, I denote an ancestral genome at a specific taxonomic range by a set of ancestral genes (that can be represented by a single HOG at the related level). Ancestral genomes can be reconstructed at all taxonomic range that the sets of HOGs is covering and may offer a new source of phylogenetic signal to unveil their related underlying evolutionary history. Ancestral genomes can be useful in several types of applications. For example to investigate the evolutionary history of several species, their related ancestral genome may be useful to estimate when gene duplications, gene losses and gene gains occurred or to infer the number of ancestral genes each ancestral genome contained and to count the proportion of evolutionary events occurring between two ancestral genomes.

Ancestral genomes can also be useful for ancestral synteny reconstruction (how ancestral genes were ordered). The idea is to that knowing how extant genomes are arranged in terms of genes order (extant synteny) and how the gene families evolved (duplications, losses, gains) we could infer how the ancestral genes ordering was (ancestral synteny). One example of ancestral synteny application is the reconstruction of genomic rearrangement history. If the ancestral synteny is available for a sets of ancestral genomes, we can infer when and how chromosomal duplications, inventions, insertions and deletions occurred and by consequence reconstructing the whole genomic history of a group of species.

In this thesis, I'll not investigate on the ancestral genomes and ancestral synteny reconstruction to rather focus on inferring accurate HOGs that may serve in the future as robust material for ancestral reconstruction.

Orthology inference

In past decades, the quantity of available genomic data has massively increased due to improvements in sequencing technologies (Sanger, Nicklen, and Coulson 1977; Margulies et al. 2005; Bennett 2004). Nevertheless, even if the amount of available sequenced genomes is constantly increasing, their quality is not necessarily improving. This results in a major constraint for orthology inference (Sonnhammer et al. 2014) and clustering algorithm design: being scalable without losing robustness. To meet this need, many orthology inference methods have been proposed over the last two decades that can be divided into two types of methods: graph-based and tree-based methods (Altenhoff and Dessimoz 2012).

Pairwise orthology

Graph-based methods

Graph-based methods are designed to deal with the need for efficient methods to detect orthology on complete gene sets. Graph-based methods are usually composed of two phases: the graph reconstruction phase, where pairwise relationships are inferred, and the clustering phase where orthologous groups are constructed. The first step connects orthologous genes (nodes in the graph) with their related pairwise orthologous relations (edges in the graph) by inferring orthology considering a species pair at a time. The principle underlying this orthology inference is that orthologs are the least diverging homologs because speciation is the last event to distinguish two genes in two different species. BBH (Bidirectional Best Hits) (Overbeek et al. 1999) rely on this principle to infer one-to-one orthology using sequence similarity scores in an efficient manner (quadratic to the number of genes) and is more robust to gene loss due to bidirectionality check for symmetric orthology. Nevertheless, lineage specific duplications (that occurred after speciation) result in more than only one orthologous counterpart, called in-paralogs or co-orthologs, in the paired species are not identified by the BBH that only retained the best hit. Inparanoid (Remm, Storm, and Sonnhammer 2001) extends the BBH method to return a group of best hits for each species, corresponding to the respective inparalogs. Lineage-specific duplications imply that many pairwise orthologous relations are found between the in-paralogous genes and their counterparts in the other species. If inparalogs are connected to a single gene they are referred to as one-to-many orthology while if they are connected to another group of inparalogs, they are referred to as about many-to-many orthology. Other methods, such as OMA (A. M. Altenhoff et al. 2011) or OrthoDB (Kriventseva et al. 2008), have been designed to use maximum likelihood estimates of the evolutionary distance of sequence pairs to identify closest genes which can be a better estimate than the highest scoring alignment (Koski and Brian Golding 2001). OMA introduced a test to detect paralogs wrongly inferred as orthologs due to asymmetric gene losses (Dessimoz et al. 2006) (only paralogs remain in the homologous cluster and are wrongly inferred as orthologs). Indeed, if gene loss occurred in the two genomes, it may be that only two paralogs remain as the closest pair. To prevent such cases, a third genome where both copies are still present is used as a "witness of non orthology".

As seen before, simple pairwise orthology has its limits and integrating multiple species may help to yield a more powerful signal. COGs (Tatusov 1997) introduced the concept of clusters of orthologs to denote a group of genes orthologous to each other. The principle is to connect together triangles of genes in the graph that share pairwise orthologous relations. OrthoMCL (Li 2003) developed another type of clustering based on a Markov Clustering that uses sequences similarity scores to weight edges and partition the graph in clusters containing orthologs and recent paralogous genes. The OMA (Dessimoz et al. 2005) strategy is to identify fully connected components in the graph as a cluster of orthologous genes. In those clusters, all genes are orthologous to each other and no inparalogs are present.

Tree based methods

Tree based methods rely on building labelled gene trees with duplication and speciation events and then identifying the inferred orthologs and paralogs. The principle of the tree based method is to reconcile gene trees with a species tree. The reconciliation is required due to potential differences between gene and species tree topologies due to evolutionary events such as gene losses and duplication, incomplete lineage sorting, long branch attraction or lateral gene transfer. To elucidate which is the best scenario to reconcile the trees, the parsimony principle is applied to select the case where the minimum number of duplication and losses is required.

Several methods have been developed in the last two decades to infer orthology from trees such as PhylomeDB (Huerta-Cepas et al. 2007), LOFT (Levels of Orthology From Trees) (Huerta-Cepas et al. 2007; van der Heijden et al. 2007), Ensembl/TreeBeST (Vilella et al. 2009).

Inferring Hierarchical Orthologous Groups

Several methods have been proposed in the past decade to infer HOGs. Inconsistencies across reconstructed levels or poor scalability are the major drawbacks that concern to the following methods.

EggNOG 4.5

The EggNOG algorithm version 4.5 (Huerta-Cepas et al. 2016) infers nested groups of orthologs across predefined taxonomic levels, each processed independently from each other. Taxonomic ranges of interest are chosen according to their coverage for evolutionary relevant orthologous groups and model organisms. The first step is to fetch genomes and proteomes from public databases and apply a quality control step to remove draft or partial genomes. To ease downstreamed analysis, protein sequences and identifiers are synchronized with the STRING (Szklarczyk et al. 2015) and STITCH (Kuhn et al. 2014) databases. The second step infers pairwise orthology using Smith Waterman alignments with adjustments to remove spurious hits with low complexity sequence regions. All hits with a bit score (the bit score corresponds to a numerical value that described the general quality of an alignment according to a chosen substitution matrix and a gap penalty) the greater than 50 are used for the next steps. The third step aims to build the orthologous groups at the previously selected levels. The algorithm uses as the basis the Cluster of Orthologous Groups from COGs (universal), KOGs (Eukaryotes) and arKOGs (archaea) database. These groups serve as references at each taxonomic range in eggNOG and are extended with the new proteomes input by the users. The goal is to first create in-paralog groups, and then to merge them with single genes to create a cluster of homologs. The cluster of homologs can later be split back if reciprocal best hits are observed with clusters from other lineages. Since levels are computed independently, inconsistencies may occur depending on how duplications are positioned. A post-processing step is then applied to remove such inconsistencies across levels by merging and splitting spurious groups. The algorithm uses a bottom-up traversal to target orthologous groups that have divided at their upper parent level. For each pair of resulting groups, the algorithm determines the species overlap then decides whether to combine these groups or not. Inconsistencies may remain due to real gene fusion events or assemblies errors that harder the reconstruction of hierarchical groups. No information is provided regarding the time performance of the algorithm. The limits of this algorithm are the narrow catalogs of levels reconstructed, the inconsistency between levels and probably the time required to perform the whole clustering pipeline.

OrthoDB

The OrthoDB algorithm (Waterhouse et al. 2013) is one of the first methods to infer hierarchical catalogs of orthologous genes. The HOG inference is performed at every level of interest but independently, in contrast to eggNOG, resulting in clustering that is not consistent across levels and inconsistencies are likely to be found. The principle of the OrthoDB algorithm is to cluster best reciprocal hits between genes of species pairs. It performs Smith-Waterman protein sequence alignments using SWIPE (Rognes 2011) on the longest transcript for each gene and only the longest gene copy with a CD HIT (Fu et al. 2012) identity greater than 97%. Clusters of orthologs are then made iteratively with an e-value threshold of 1e-3 for best reciprocal hit triangulation and of 1e-6 for pair-only best reciprocal hits. A minimum of 30 amino acids overlap is required. Once the clusters of best reciprocal hits are built, they are expanded to include inparalogs by including within-species homologs that are more closely related than the clustered best reciprocal hits. This clustering phase is applied at selected levels of a given phylogeny without any cross-level verifications of orthology clustering inconsistency. Levels are not necessarily nested to each other, depriving downstream analysis of information regarding evolution of particular orthologous groups along branches.

Hieranoid 2

The Hieranoid 2 algorithm (Kaduk and Sonnhammer 2017) is a tree guided method to build hierarchical orthologous groups. It traverses a guide tree to compute sequence similarities and to reconstruct at each level the related 'meta-species' composed of lower orthologous groups in the tree. By performing only relevant proteome pair comparisons, the time complexity is reduced to N-1 with N number of proteomes. The Hieranoid 2 algorithm uses as input a fully bifurcated tree (polytomies need to be expanded) where leaves are composed of complete proteomes (longest protein representative per gene) and internal nodes representing 'meta-species' composed of orthologous groups from lower levels. The algorithm iterates along genome pairs, starting with the closest pair in the tree and computes at each level the following 4 steps: sequence similarity search, orthologous group inferences, multiple sequence alignments and consensus sequence building. The search of sequence similarities between the two proteome pairs is performed using BLAST or USEARCH to yield potential matches. A filtering step is applied to remove matches that do not fulfil the default InParanoid overlap criterion that requires that the distance from the first to the last aligned residue must be at least 50% of either protein and the length of the aligned regions must be at least 25% of the length of either sequences. Orthologous clustering is then performed on those matches using the default Inparanoid algorithm (Remm, Storm, and Sonnhammer 2001). The third step is to build multiple sequence alignments from all sequences of orthologous groups in order to capture the sequence diversity within each group. Once the alignment is built, the consensus sequence is calculated by using the consensus residue with the highest score in the column using the BLOSUM62 substitution matrix. Columns in the alignments with more than 50% gaps are trimmed out. All consensus sequences at a given internal node represent the related pseudo-species (ancestral proteome) with one species per representative orthologous group rooted at this level (ancestral gene). For pseudo-species versus pseudo-species reconstruction the consensus sequence of each group is used and then recalculated with all original sequences of the two orthologous groups. Once the algorithm reaches the root of the tree, all the orthologous clusters at each level are combined to obtain hierarchical orthologous groups.

GETHOGs version 1

The GETHOGs (“Graph-based Efficient Technique for Hierarchical Orthologous Groups”) algorithm uses a reference species tree (including polytomies) and its related orthology graph to build HOGs in a time efficient manner. Considering a reconciled gene tree, HOGs at any level of interest can be easily found by searching for (sub-)gene trees rooted by a speciation at those levels of interest. Nevertheless, this method requires reconciled gene trees to be built, a step which is not time efficient, and can be complex or not scalable to large gene families. The idea of GETHOGs is to use an orthology graph instead of reconciled gene trees; this is computationally less expensive to build and more scalable for large dataset reconstructions. As demonstrated in (Altenhoff et al. 2013), a one-to-one correspondence exists between connected components (set of interconnected nodes) in a perfect—i.e. complete and entirely correct—orthology graph and HOGs. The algorithm relies on this one-to-one correspondence to reconstruct the HOG from the orthology graph. The algorithm uses a top-down traversal of the reference tree and first extracts at each taxonomic range the related sub-orthology graph. Then, it searches for connecting components in this graph to infer HOGs. Spurious orthologous relations in the inputed orthology graph may connect unrelated HOGs. In order to prevent unwanted clustering of HOGs, a Min-Cut algorithm is applied to remove weakly supported edges in the connected components, e.g a single edge connecting two densely interconnected groups of orthologs. The algorithm produces consistent nested HOGs across the whole input graph (each level is computed and represented in the final form of the HOGs).

OMA

In this project we focus on OMA ("Orthologous MAtrix"), a graph based algorithm and database for orthology inferences. The OMA algorithm (Roth, Gonnet, and Dessimoz 2008; Dessimoz et al. 2005) uses protein sequences of multiple genomes to infer pairwise orthologous relations between genes and produce orthologous groups. There exist two types of orthologous groups inferred in OMA: the 'OMA group' is a set of genes all orthologous to each other, and the Hierarchical Orthologous Groups, reconstructed using a hierarchical clustering algorithm called GETHOGs (Altenhoff et al. 2013). The OMA method shows a high precision (low false-positive rate) but low recall (high false-negative rate) compared to other orthology inference methods, as has been shown in several benchmark studies (Altenhoff and Dessimoz 2009; Altenhoff et al. 2016; Boeckmann et al. 2011; Trachana et al. 2011).

Open Challenges

Although building OMA groups can be performed by simply searching for fully-connected components in the orthology graph, reconstructing HOGs is not a trivial task. Indeed, several factors can explain the difficulties of HOG reconstruction: the presence of spurious/missing orthologous relationships that are the building blocks of HOG reconstruction, the complexity in the evolutionary history of genes and genomes, or the size of genomic datasets used which can go up to thousands of species. In addition, the current HOGs clustering algorithm (GETHOGs) in OMA uses a top down approach to reconstruct HOGs that is not scalable to very large datasets (some gene families contain over 100,000 members in OMA). Designing a hierarchical clustering algorithm that produces high confidence HOGs on a large scale dataset is now mandatory to face the current growth of genomic data.

In addition to the difficulty of reconstructing HOGs, assessing their quality is also a challenging task. Indeed, even if several orthology benchmarks exist (Linard et al. 2014; Altenhoff et al. 2016) (and can use the HOGs-induced pairwise orthology relations as a proxy to assess the HOGs quality) no gold standard HOGs reference or quality assessment metrics have been proposed. Establishing methods and metrics that estimate the quality of HOGs inferences is now mandatory to assess the performance of newly created HOGs reconstruction algorithms.

Moreover, since HOGs are relatively recent and are restricted to a specific set of analysis, there are no standard tools available to explore (e.g. retrieving evolutionary-based information) or visualise them (e.g. visually exploring their structure and capturing the main information at a glance).

Aims of the thesis and organisation

The aims of this thesis project are:

1. to improve algorithms to infer HOGs in terms of accuracy, scalability and robustness (chapter 2 & 5),
2. to develop methods and metrics to improve the benchmarking of HOG inference algorithms (chapter 4),
3. to devise tools for the visualisation of HOGs and to facilitate the application of HOGs to downstream analyses (chapter 3).

This PhD thesis is organised into 6 chapters.

In the first chapter, I introduce concepts and paradigms of phylogenetics along with orthology inference and orthology clustering methods with their applications and limits.

In chapter 2, I describe an improvement of orthology inference in the OMA algorithm and a new bottom-up variant of the HOGs clustering algorithm in OMA called GETHOGs 2.0.

In chapter 3, I present a new tool to explore HOGs and to facilitate the extraction of the phylogenetic information they contain. Finally, I will introduce two new visualisation tools to investigate the HOGs from different angles.

In chapter 4, I discuss the limits and the errors of the new GETHOGs algorithm with a benchmarking strategy on simulated data and on a real dataset.

In chapter 5, I propose new alternative heuristics to overcome such limits on HOGs inference algorithms.

Finally, chapter 6 concludes the thesis with a general discussion and perspectives.

In addition, I was involved in several other projects in parallel to the work described in this thesis, which are not included in this manuscript but are published elsewhere: contribution to the conception and implementation of the Orthology Benchmark Service web server (Altenhoff et al. 2016), contribution to the conception and implementation of visualization tools for the OMA Browser (synteny viewer for chromosome pairs, dynamic table with taxonomy-driven filtering) (Altenhoff et al. 2017), HOG-based benchmark of a new algorithm to identify fragments of the same gene in draft-quality assemblies (Piližota et al. 2018).

Chapter 2: OMA Algorithm 2.0

This chapter has been published in **Clément-Marie Train, Natasha M Glover, Gaston H Gonnet, Adrian M Altenhoff and Christophe Dessimoz. Orthologous matrix (OMA) algorithm 2.0: more robust to asymmetric evolutionary rates and more scalable hierarchical orthologous group inference. Bioinformatics, 2017, i1-i8 (ISMB 2017 proceedings)**

In this chapter, we focus on improving the orthology inferences in OMA by accounting for fast evolving duplicated genes and including an additional control to verify evolutionary distance additivity (witness of evolutionary distance congruences). Since orthologs are the fundamental resource to build HOGs, improving ortholog inferences will considerably increase the quality of HOGs. A second part of this chapter focuses on improving the orthology clustering itself. The original hierarchical clustering algorithm in OMA called GETHOGs (Altenhoff et al. 2013) uses a 'top-down' approach. The algorithm starts the HOGs reconstruction at the most ancestral taxonomic ranges (where the largest quantity of information is required and where the quality of information is the lowest due to age) until most recent taxa. In addition, spurious edges and missing relations highly increase the probability of making clustering mistakes that are vertically propagated through the whole clustering procedure, considerably affecting the final results. In this chapter, I introduced a new hierarchical clustering algorithm called GETHOGS 2.0 ('bottom-up') with a better scalability to large datasets and an improved robustness of HOGs inferences. This work was published in 'Orthologous Matrix (OMA) algorithm 2.0: more robust to asymmetric evolutionary rates and more scalable hierarchical orthologous group inference' (Train et al. 2017).

2.1 Abstract

Accurate orthology inference is a fundamental step in many phylogenetics and comparative analysis. Many methods have been proposed, including OMA (Orthologous MAtrix). Yet substantial challenges remain, in particular in coping with fragmented genes or genes evolving at different rates after duplication, and in scaling to large datasets. With more and more genomes available, it is necessary to improve the scalability and robustness of orthology inference methods.

We present improvements to the OMA algorithm: (i) refining the pairwise orthology inference step to account for same-species paralogs evolving at different rates, and (ii) minimizing errors in the pairwise orthology verification step by testing the consistency of pairwise distance estimates, which can be problematic in the presence of fragmented sequences. In addition we introduce a more scalable procedure for hierarchical orthologous group (HOG) clustering, which is several orders of magnitude faster on large datasets. Using the Quest for Orthologs consortium orthology benchmark service, we show that these changes translate into substantial improvements on multiple empirical datasets.

This new OMA 2.0 algorithm is used in the OMA database (http://omabrowser.org) from the March 2017 release onwards, and can be run on custom genomes using OMA standalone version 2.0 and above (http://omabrowser.org/standalone).

2.2 Introduction

Inferring evolutionary relationships between genes lies at the heart of comparative, phylogenetic, and functional analyses. Homologs are genes that share a common ancestry (Fitch, 1970). They can be further classified into: orthologs if they arose by speciation events, or paralogs if they arose by duplication events (Fitch, 1970; Figure 4). These evolutionary relations are all defined among pairs of genes and—except for homology—are not transitive. Many orthology inference methods have been proposed over the years, such as COGs (Tatusov et al., 1997), bidirectional best hits (Overbeek et al., 1999), Inparanoid (Remm et al., 2001), OrthoMCL (Li et al., 2003), Ensembl Compara (Vilella et al., 2008) or OrthoDB (Kriventseva et al., 2008).

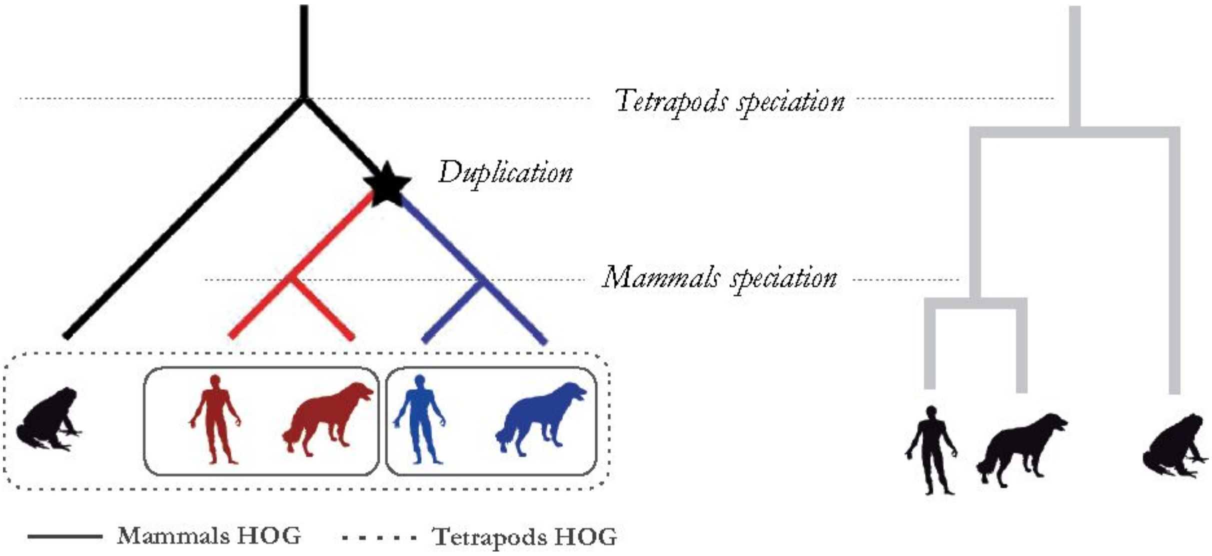


Figure 4: Hierarchical Orthologous Groups. Labeled gene tree (left) and its related species tree (right) illustrating the evolutionary history of five genes all descended from a single common ancestor at the tetrapods level. Those homologs can be classified as orthologs if they start diverging by speciation (human versus dog genes of same color) or as paralogs if they start diverging by duplication (blue versus red genes). We can identify in this example HOGs at two taxonomic levels: one larger HOG at the tetrapods level (dotted-line rectangle) containing all the homologous genes that emerged from the single tetrapod ancestral gene, and two HOGs at the mammalian level (solid-line rectangles), due to a duplication of the tetrapod ancestral gene before the mammals speciation.

The Orthologous Matrix (OMA) algorithm infers orthologous genes among multiple genomes on the basis of protein sequences (Dessimoz et al., 2005; Roth et al., 2008). In addition to inferring such pairwise evolutionary relationships, OMA infers two types of orthologous groups. The first, called 'OMA groups', are sets of genes in which every pair is inferred to be orthologous. The second, introduced more recently and called 'hierarchical orthologous groups' (HOGs), are defined as a set of genes that have all descended from a single common ancestral gene at a specific taxonomic range of interest (Altenhoff et al., 2013; Figure 4).

When compared with most other methods, the OMA algorithm has been shown to have high precision (i.e. low false-positive rate) but low recall (i.e. high false-negative rate) in several benchmark studies (Altenhoff and Dessimoz, 2009; Altenhoff et al., 2016; Boeckmann et al., 2011; Trachana et al., 2011). Even so, predicting correct evolutionary relationships becomes more difficult due to complex mechanisms such as differential gene loss, asymmetric evolutionary rates, gene duplications and poor quality genomes. This can lead to spurious or missing relationships (Dalquen and Dessimoz, 2013).

The final stage of the OMA pipeline infers HOGs from pairwise orthologs (Altenhoff et al., 2013). Such groups are useful for analyzing multiple genomes or genes, but require scalable clustering algorithms due to the complexity in reconstructing them.

Here, we present two new improvements to our orthology inference algorithm in order to better handle rapidly evolving duplicated genes and to improve detection of asymmetric gene loss. In addition, we introduce a 'bottom-up' HOGs clustering algorithm that can scale up to thousands of genomes.

2.3 Materials and methods

We first provide an overview of the OMA algorithm, then present in detail the three refinements introduced in this new version, and finally provide methodological details about the benchmarking.

2.3.1 Overview of the OMA algorithm

The following section provides an overview of the existing OMA algorithm, of which the details are described in (Roth et al., 2008).

The OMA algorithm infers pairs of orthologous genes from complete genomes in a four-step process (Figure 5):

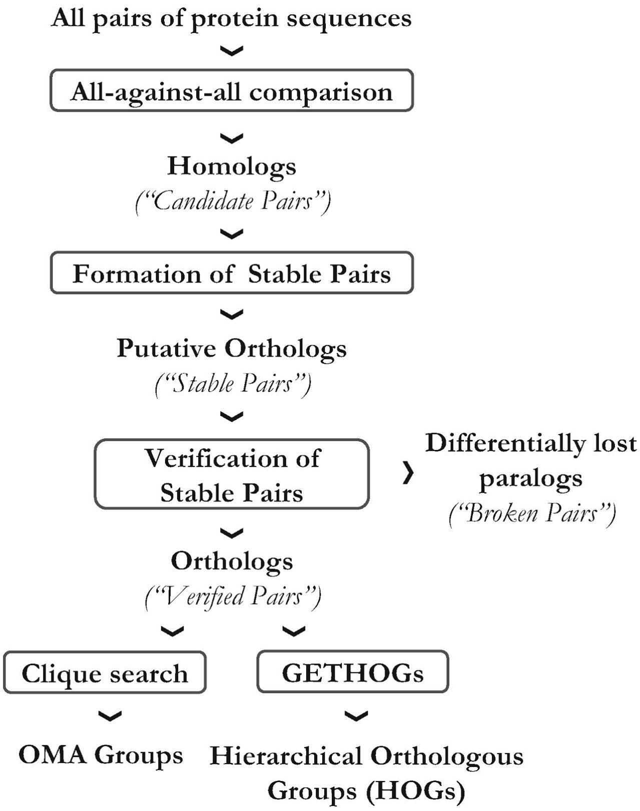

All pairs of protein sequences
All-against-all comparison
Homologs
("Candidate Pairs")
Formation of Stable Pairs
Putative Orthologs
("Stable Pairs")
Verification of Stable Pairs
Differentially lost paralogs
("Broken Pairs")
Orthologs
("Verified Pairs")
Clique search
GETHOGs
OMA Groups
Hierarchical Orthologous Groups (HOGs)

Figure 5: Overview of the OMA pipeline. Boxes denote individual steps in the pipeline, while the text outside boxes denotes the input or output of these processes and their terminology in OMA.

I. **Homology inference:** Alignments are made with all possible pairs of sequences from all genomes using local dynamic programming (Smith and Waterman, 1981), and pairs with sufficient score and overlap are promoted to Candidate Pairs.

II. **Ortholog and co-ortholog inference:** Candidate Pairs that are the mutually evolutionary closest sequences between a pair of genomes are upgraded to Stable Pairs. In order to include many-to-many orthologous relationships, Candidate Pairs found within a confidence interval (corresponding to distance variance) are also upgraded to Stable Pairs.

III. **Witness of non-orthology verification:** At this point, some pairs of paralogs may still be misidentified as orthologs due to differential gene loss (Dessimoz et al., 2006a). To avoid such cases, a verification step is added to assess the orthologous origin of a Stable Pair by using a third genome that retained both orthologous copies, which thus acts as witnesses of non-orthology. Pairs that pass this test are upgraded to Verified Pairs.

IV. **Ortholog clustering:** Once the pairwise orthologs are inferred, a clustering algorithm is applied to group genes descending from a common ancestral gene into HOGs or using a clique search algorithm for OMA Groups.

2.3.2 Algorithmic refinements: taking into account fast-evolving duplicated genes in the orthology inference step

In the current orthology inference step of the OMA algorithm, genes that are mutually the closest pairs of sequences across genomes are considered as putative orthologs. Due to lineage-specific duplications, orthology relationships are however not necessarily one-to-one (e.g. Dalquen and Dessimoz, 2013). Thus, OMA considers a tolerance interval during the mutually closest gene search to allow for inclusion of potential inparalogs.

Specifically, the criterion originally used in OMA was as follows: a Candidate Pair xy between genomes X and Y is upgraded to a Stable Pair if for all genes xi from X and for all genes yj from Y with xi ≠ x and yj ≠ y:

$$d_{xy_j} - d_{xy} > -k \, * \, \mathrm{stdev}\left(d_{xy_j} - d_{xy}\right)$$

and

$$d_{x_iy} - d_{xy} > -k \, * \, \mathrm{stdev}\left(d_{x_iy} - d_{xy}\right)$$

where d is the pairwise maximum likelihood distance estimate, k the tolerance parameter of the standard deviation between the two distances, and where stdev() is the distance standard deviation of the difference (Dessimoz et al., 2006a,b). This means that a Candidate Pair xy is upgraded to a Stable Pair if and only if there are no other pairs xyj or yxi with significantly smaller evolutionary distances.

So far in the orthology inference step, only the distances between genes from different genomes are taken into account. However, if a duplicated gene evolved faster than its related in-paralog, searching for mutually closest genes between genomes can fail to identify it as an ortholog (Figure 6.A). Because of the distance asymmetry, the original algorithm does not detect the fast evolving gene as a co-ortholog, thus wrongly implying an ancestral duplication as the origin of divergence (Figure 6.B).

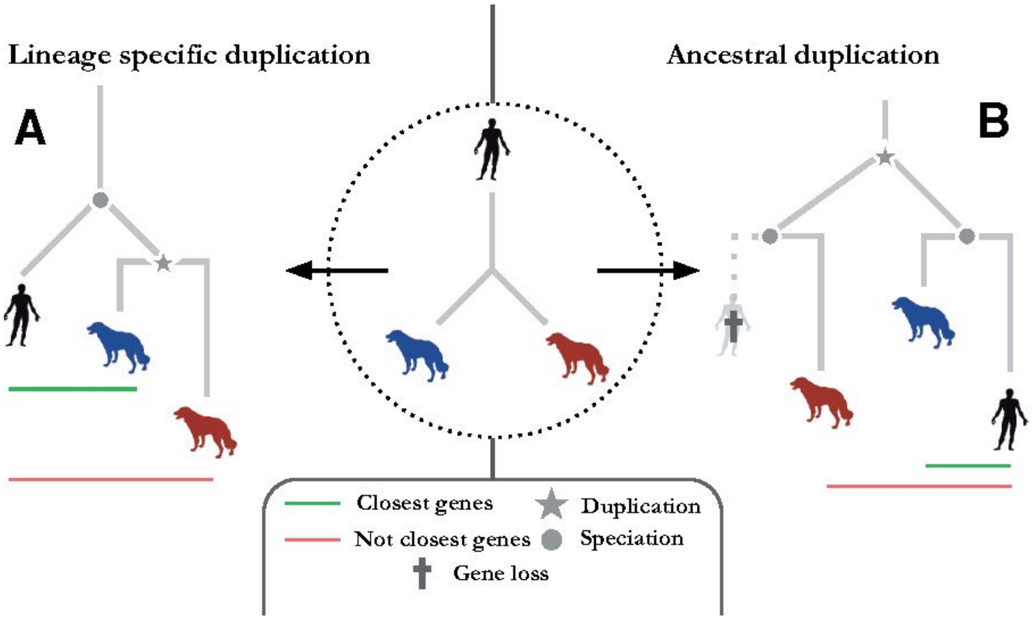


Figure 6: Putative evolutionary scenario for a gene triplet containing 1 human gene and 2 asymmetrically evolving dog genes.

A. Reconciled labeled gene tree for the gene triplet where the red dog gene (orthologous to the human gene) evolved at a faster rate.

B. Reconciled labeled gene tree for the gene triplet where an ancestral duplication gave rise on one side to the blue dog gene and the black human gene and on the other side only to the red dog gene, since the related gray human gene had been lost. The red dog gene is thus paralogous to the black human gene

The refinement introduced here also takes into account the evolutionary distance between inparalogs. Inspired by other orthology algorithms detecting co-orthologs on the basis of alignment scores, such as Inparanoid (Remm et al., 2001) or OrthoInspector (Linard et al., 2011), we added a new check that the distance between the two potential in-paralogous dog genes is significantly smaller than the distance between the closest genes (black and blue genes), as illustrated in the Figure 6.A. More precisely, we retain as Stable Pairs all Candidate Pairs xy between genomes X and Y that were previously discarded during orthology inference if, for any genes yj from Y with yj ≠ y there exists a gene yi that has a distance to y significantly closer than the distance between the Candidate Pair genes x and y2:

$$d_{xy} - d_{yy_j} > -k \; * \; stdev\left(d_{xy} - d_{yy_j}\right)$$

where d is a pairwise maximum likelihood distance estimate, k the inparalogs tolerance parameter of the standard deviation between the two distances and where the distance standard deviation stdev() is computed according to Dessimoz et al. (2006a,b).

2.3.3 Algorithmic refinements: extended witnesses of non-orthology with verification of distance additivity

As mentioned earlier, the verification step of the OMA algorithm aims to detect paralogs resulting from differential gene losses (Figure 7.A). Indeed, paralogs can be the only remaining homologs between two genomes and since they are mutually the closest genes across those genomes they can be wrongly inferred as orthologs. To prevent such cases, OMA searches for each pair of putative orthologs (‘Stable Pairs’) whether there might be a third genome that has retained paralogs that could act as a witness of non-orthology (Dessimoz et al., 2006a,b).

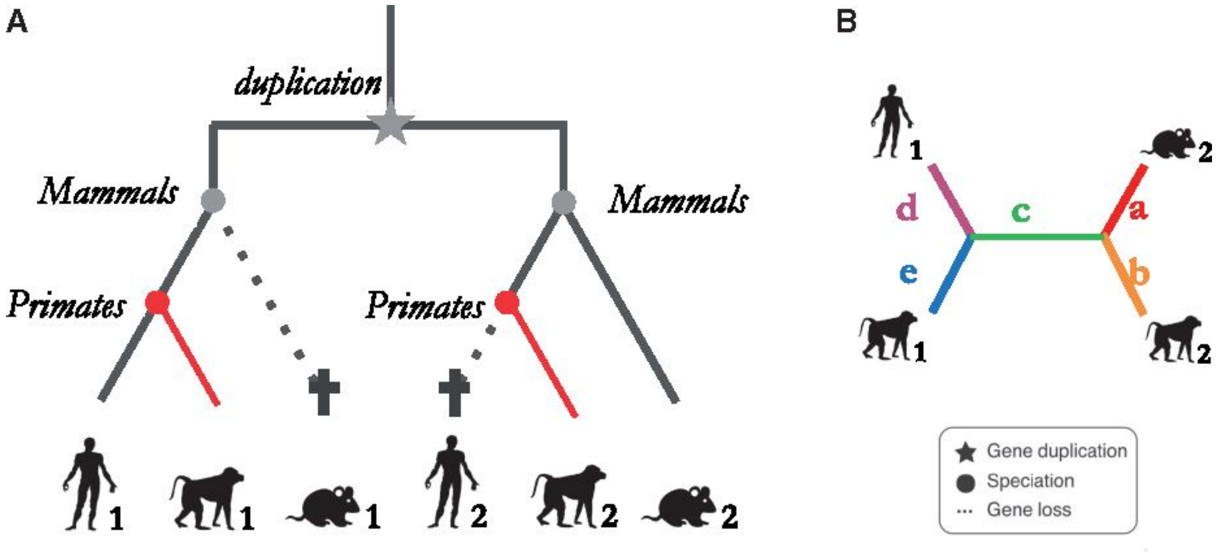


Figure 7: Hidden paralogs example and witness of non-orthology gene quartet.

A. Example of labeled gene tree containing hidden paralogs due to asymmetric gene losses between human and mouse. This can occur when an ancestral duplication is first followed by a speciation then by asymmetric genes losses. The resulting paralogs are wrongly inferred as orthologs because they are the mutually closest pairs between two genomes (Human1, Mouse2 sequences). OMA attempts to identify such cases through the use of a third species (here a monkey) that has retained both copies, which can act as witnesses of non-orthology.

B. The four extant genes form a quartet with branches labeled a–e.

This test is based on pairwise evolutionary distance comparison of the gene quartet, without reconstructing the underlying gene tree (which, given the very large number of quartets of homologous genes across many genomes, would be too time consuming). However, direct comparison of pairwise distances implies that the distances among the four genes are additive, and by consequence, that a phylogenetic tree can be reconstructed from them. We have found cases, particularly in the presence of fragmented sequences, where additivity is far from being met.

To ensure that the evolutionary distances do not depart excessively from additivity, in the verification of Stable Pair x1,y2 using potential witnesses of non-orthology z1,z2, we test a 'soft' variant of the four-point condition (Buneman, 1974), which allows for distance estimation uncertainty. We check that the sum of the distances d(x1,z2) and d(y2, z1) is approximately equal to the sum of the distances d(x1, y2) and d(z1, z2). Indeed, considering the branch labels defined in Figure 7.B, under the model and assuming no error, the following equality holds:

$$(d+c+b)+(a+c+e)=(d+c+a)+(e+c+b)$$

Taking inference uncertainty into account, we test the equality as follows:

$$\left| d_{x_1 z_2} + d_{y_2 z_1} - d_{x_1 y_2} - d_{z_1 z_2} \right| < 2 * \sqrt{\mathrm{var}\left(d_{x_1 z_2} \right) + \mathrm{var}\left(d_{y_2 z_1} \right) + \mathrm{var}\left(d_{x_1 y_2} \right) + \mathrm{var}\left(d_{z_1 z_2} \right)}$$

where x1 and y2 are the Stable Pair genes from genomes X and Y, z1 and z2 are the witnesses of non-orthology in the third genome Z, d is a pairwise maximum likelihood distance estimate, and var(d(x,y)) is the variance of the distance estimate between sequences x and y. If the test fails, z1 and z2 are not used as witnesses of non-orthology.

2.3.4 Algorithmic refinements: bottom-up HOG inference

In this section, we present improvements to the hierarchical orthologous group (HOG) clustering phase (Altenhoff et al., 2013). The work established a one-to-one correspondence between the connected components of a perfect orthology graph—i.e. containing no false positives or negatives— and HOGs. Based on this, but allowing for a noisy input, we introduced a heuristic called GETHOGs (‘Graph-based Efficient Technique for Hierarchical Orthologous Groups’), which used the min-cut algorithm to break down spurious orthologous relationships before identifying HOGs as the connected components. This was performed for each taxonomic range of a reference phylogeny, starting from the root and walking down the tree to the most specific clades, in a ‘top-down’ fashion.

Nevertheless, inconsistencies in the orthology graph due to spurious inferences or missing relations increase the probability of making errors during the clustering. Such mistakes in grouping are then propagated through the entire clustering procedure due to the greedy nature of the algorithm, and can affect the final result. Furthermore, the original GETHOGs algorithm started at the root of the reference phylogeny, where the graph is largest (since it contains pairs of orthologs between all species instead of subsets of them) and most uncertain (since it also contains orthologous relationships among the most distant species).

Here, we introduce a ‘bottom-up’ variant of GETHOGs, which infers HOGs starting with the most specific taxonomy and incrementally merges them toward the root (Figure 9). More specifically, the new approach reconstructs HOGs by applying the following procedure with each speciation node of the species tree as reference, from the leaves to the root:

I. Build inter-HOG orthology graph (Figure 8 BuildInterGraph, Figure 9.D left): Define a graph in which the nodes are the HOGs inferred at the level of each child of the reference speciation. If a child is a leaf of the species tree (i.e. a child is an extant species), the HOGs defined at this level are simply the individual sequences of that species. The edges of the graph represent one or more pairwise orthology relationships between members of the HOGs, with the number of such relationships recorded as weights.

II. Remove spurious edges (Figure 8 BuildInterGraph line 7–9, Figure 9.D middle): Once the orthology graph is built, we next assess whether each edge is well supported or not. For each edge, the algorithm computes the ratio of the number of pairwise orthologous relations (edge weight) to the maximum number of possible pairwise orthologous relations (equal to the product of the size of the two HOGs connected by the edge). If the input orthology graph is perfect (i.e. correct and complete), this ratio is one. A cutoff α (set to 0.8 throughout this article and by default) is then used to remove all edges with insufficient connections.

III. Search for connected components (Figure 8 GETHOGSBottomUp line 10–12, Figure 9.D right): The final step searches for connected components inside the graph and clusters them together as a single HOG at the level of the speciation of reference.

The asymptotic complexity is determined by the complexity of the species tree traversal and the complexity for the HOG inference at each internal node of the species tree (i.e. inference for each taxonomic level). Tree traversal has a runtime complexity of O(n) where n is the number of species, because there are n-1 internal nodes. The runtime of the HOG inference at each level (steps 1–3 above) primarily depends on the number of pairwise orthology relationships. The total number of sequences is O(n) because we can expect a natural limit on the size of each proteomes. Thus, the total number of pairwise relationships is O(n2). Using Union-Find data structures, finding connected components in a graph of m edges is O(m) (Cormen, 2009). There are potentially O(n2) edges in each inter-HOG orthology graph, but since each orthology relationship only needs to be considered once in the entire traversal (at the speciation node which induces them), the amortized complexity at each internal node is O(n) resulting in a total complexity of bottom-up GETHOGs of O(n2). This compares favorably to the top-down GETHOG algorithm, which has complexity O(n3·log4n) (Altenhoff et al., 2013).

```
Input: Rooted species tree $T$, a set of tuples of pairwise orthologs $R$ and cutoff $0 < \alpha \leq 1$
function GETHOGSBottomUp($T, R, \alpha$)
    $OG \leftarrow \emptyset$
    if $T$ is not a leaf then
        $children \leftarrow$ GetChildren($T$)
        for all $child$ in $children$ do
            $OG \leftarrow OG \cup$ GETHOGSBottomUp($child$)
        end for
        $SubHogs \leftarrow \{\forall g \in OG \mid \text{TaxRange}(g) \in children\}$        ▷ direct children HOGs
        $HogGraph \leftarrow$ BuildInterHogGraph($SubHogs, R, \alpha$)
        for all $CC$ in ConnectedComponents($HogGraph$) do
            $OG \leftarrow OG \cup (T, CC)$
        end for
    end if
    return $OG$
end function
Output: Set of tuples of orthologs groups with their related taxonomic range
```

```
Input: A set of HOGs $H$, a set of tuples of pairwise orthologs $R$ and cutoff $0 < \alpha \leq 1$
function BuildInterHogGraph($H, R, \alpha$)
    $Edges \leftarrow \emptyset$
    for $h_1, h_2$ in $\binom{H}{2}$ do
        $g_1 \leftarrow$ ExtantGenes($h_1$)        ▷ Set of extant gene in HOG $h_x$
        $g_2 \leftarrow$ ExtantGenes($h_2$)
        $r \leftarrow$ FilterOrthologsBetweenGeneSets($R, g_1, g_2$)
        if $\frac{2|r|}{|g_1||g_2|} > \alpha$ then
            $Edges \leftarrow Edges \cup (h_1, h_2)$
        end if
    end for
    return Graph($H$, $Edges$)
end function
Output: Graph composed of HOGs as nodes with edges among them if orthologous at current
    taxonomic level.
```

Figure 8: Pseudocode of bottom-up GETHOGs algorithm.

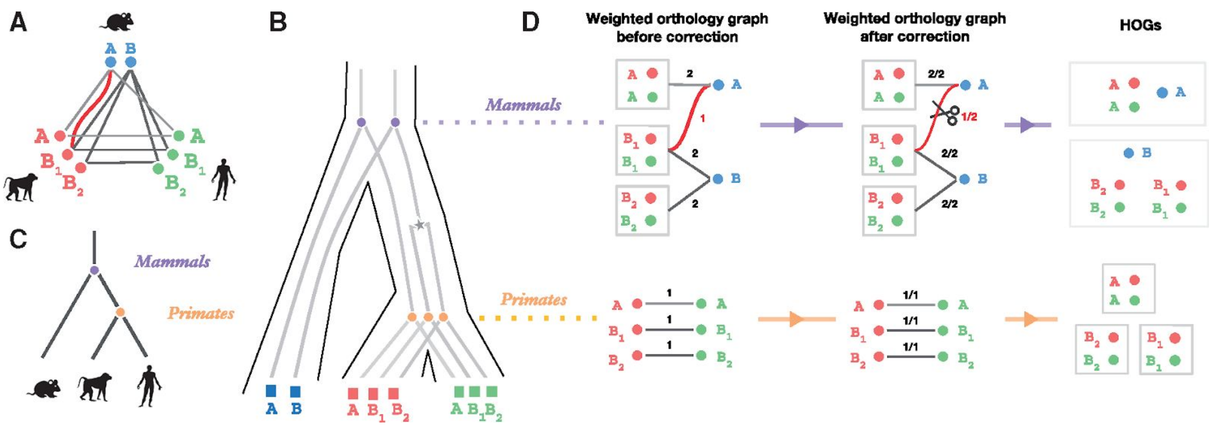


Figure 9: Bottom-up GETHOGs reconstruction example.

A. Orthology graph, where circles represent extant genes with a species-specific color and edges represent pairwise orthologous relations between genes. The red edge represents a spurious orthologous relation between the mouse gene A and the monkey gene B1.

B. Reconciled gene trees corresponding to the orthology graph in (A). Extant genes are represented by squares, speciation events by circles and duplication events by stars.

C. Corresponding species tree.
D. HOGs reconstruction using bottom-up GETHOGs with a minimal edges removal threshold of 0.8. The algorithm starts by reconstructing HOGs at the level of the primates and finishes at the level of mammals. The left panel displays the sub-orthology graph composed of HOGs (or extant genes) as nodes connected by weighted edges according to the number of existing orthologous relations between HOG genes. In the middle panel, to identify spurious edges, GETHOGs computes the fraction of orthologous pairs over the maximal number of possible pairs. The algorithm removes the red edge because the score is smaller than the minimal edge removal threshold. The right panel depicts the HOGs reconstructed from the connected component of the corrected graph.

2.3.5 Validation and benchmarking

We used the Quest for Orthologs (QfO) reference proteomes dataset (Altenhoff et al., 2016) to benchmark our method and to analyze case studies. It consists of 66 (40 eukaryotes, 20 bacteria, 6 archaea) proteomes, and contains more than 750 000 non-redundant protein sequences. It includes a broad selection of genomes covering the tree of life, including model organisms of interest and those important in biomedical or phylogeny research. In addition, as a reference tree we used a manually curated species tree for the 66 organisms contained in the QfO reference proteomes (Boeckmann et al., 2015).

The orthology benchmarking service (http://orthology.benchmarkservice.org) is an automated web-based tool for orthology inference quality assessment (Altenhoff et al., 2016). This service takes ortholog relations inferred on the QfO reference dataset as input, and after running a broad range of tests, it summarizes and plots the results. We focused on the generalized species tree discordance test for our benchmark analysis, as it is a robust way to assess the quality of orthology predictions.

The generalized species tree discordance test estimates the agreement between orthology predictions and a reference species tree. Since orthologs originate by speciation, comparing the similarity of a tree reconstructed using pairwise orthology relations to a reference species tree is a way to assess the quality of the orthology predictions. We applied this procedure to a subset of the QfO references proteomes, covering different taxonomic ranges (Last Universal Common Ancestor, Eukaryotes, Vertebrates and Fungi). The main results provided by this test are the 'error rate' (average Robinson-Foulds distance between the reconstructed gene tree and reference species tree), the 'number of complete trees sampled' (number of trees fully reconstructed out of 50 k trials), and the 'number of predicted orthologs'.

In the context of HOGs benchmarking, the generalized species tree discordance test is a valuable metrics to assess two types of quality aspects of the HOGs reconstruction: the completeness of the HOGs (how much the HOGs are complete and dense) using the recall as a proxy measure and the quality of the internal genes clustering of each HOGs by estimating the error rate between the reconstructed gene tree topology and reference gene tree topology.

3 Results

Before presenting aggregate benchmarking results, we first present detailed examples of improvements obtained by the refinements described in the previous section. We begin with a case study of a family containing fast-evolving genes, where we recover orthologous relations and correct the orthology graph. We then present an example of the kind of improvement obtained by the new additivity test.

3.1 Fast-evolving duplicated genes case study: the haptoglobin family

The first orthology inference refinement we present aims to include fast evolving duplicated genes in orthology predictions by not only looking at evolutionary distances between genomes but also within genomes.

In order to investigate the performance of this refinement, we used the haptoglobin gene family as an example, which duplicated in the primates (Figure 10.A). One branch of the primate paralogs evolved at a higher rate than its sister branch, leading to asymmetry in the distance between the paralogs. As a result, although there is a one-to-many relationship between rodent haptoglobin and primate haptoglobin, the original OMA algorithm only uncovers the most conserved ortholog pairs (Figure 10.B). By taking into account the relatively short distance between the in-paralogous copies (see section 2), the updated OMA algorithm now recovers both copies as co-orthologs to their rodent counterparts (Figure 10.C).

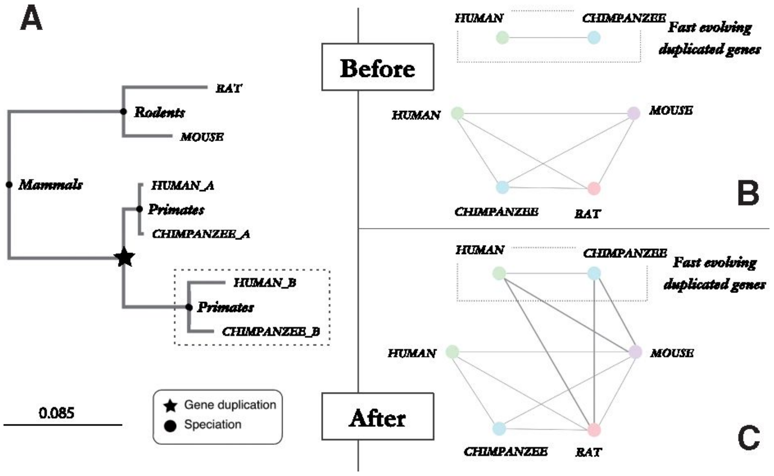


Figure 10: Analysis of haptoglobin gene family in mammals.

A. Phylogenetic labeled gene tree of the haptoglobin family built using 6 proteins sequences from 4 mammals (rat, mouse, human, chimpanzee). The dotted rectangle highlights the fast evolving primate paralogous genes.

B,C. Orthology graph of the haptoglobin gene family shown in A. Nodes represent extant genes denoted by a species-specific color and their identifier meanwhile the edges represent pairwise orthologous relations between genes. The orthology graph in B, relies on the pairwise orthologous relations inferred using the classic OMA algorithm, while the orthology graph in C is built using the orthology relations including the refinement for paralogs evolving at different rates. (UniProt IDs of the sequences involved Mouse→Q16646, Rat→A0A0H2UHM3, Human_a→HOY300, Chimpanzee_a→H2RAT6, Human_b→P00739, Chimpanzee_b→H2RB63).

3.2 Additivity of distances in witnesses of non-orthology step

As previously discussed in the section 2, the OMA algorithm attempts to uncover hidden paralogs (pairs of paralogs resulting from differential gene losses, thus each lacking an ortholog in the other species). This step compares evolutionary distances among quartets of genes without explicitly reconstructing their underlying phylogenetic gene tree (for performance reasons), under the assumption of near additivity of these distances.

However, in some cases—typically in the presence of one or more fragmented sequences—the assumption of additivity is strongly violated. Figure 10 shows an example of a quartet of genes with non-additive distances, where a Stable Pair between two mammal genes is erroneously discarded using two arabidopsi genes as witnesses of non-orthology. The underlying phylogenetic gene tree (Figure 11.A) indicates that the arabidopsis gene are in fact the result of a duplication within plants and not an ancestral duplication shared with the mammals in question. Without resorting to tree inference on a multiple sequence alignment (which would be prohibitively costly considering the number of quartets needed to verify every putative ortholog), the non-additivity of the pairwise distances in this quartet (Figure 11.B) can be detected by applying the new condition (see section 2), which in this case is violated:

$$|191 + 192 - 62 - 169| \overset{?}{<} 2 * \sqrt{169 + 193 + 120 + 121}$$
$$152 \nless 2 * 24.55$$

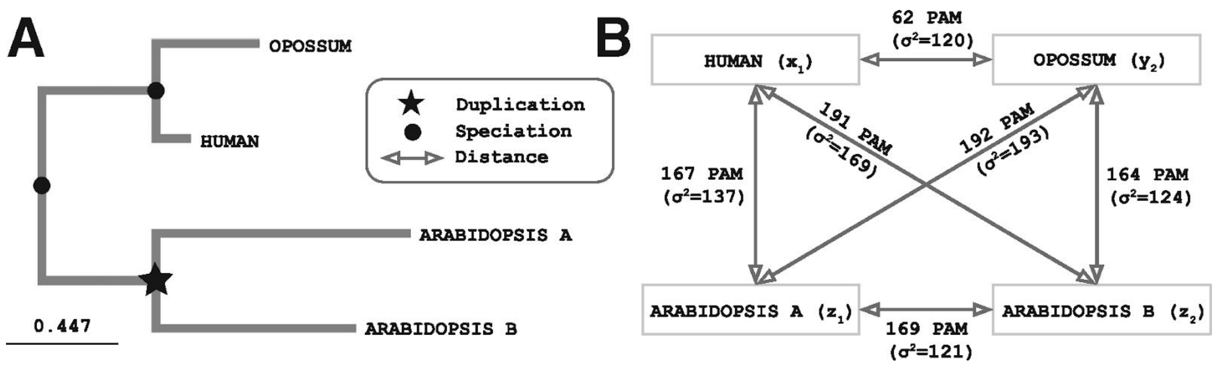


Figure 11: Example of non additivity among gene quartet distances.

A. The two arabidopsi genes arose from a duplication within the plants, which can be inferred from a tree inferred using a multiple sequence alignment.

B. However, if we consider pairwise distances estimated from independent pairwise alignments, one arabidopsi gene appears to be closer to the human sequence, while the other appears to be closer to the opossum gene. In the original OMA algorithm, this would result in these arabidopsi genes being erroneously used as witnesses of non-orthology; in the new algorithm, the non additivity of these distances (in Point Accepted Mutation units, with estimator variance in parentheses) is detected and the Arabidopsis genes are not used. (UniProt IDs of sequence involved: Human → Q16874, Opossum → F7FI80, arabidopsi a → Q93ZB2, arabidopsi b → Q9LNJ4)

The equation does not hold, thus we cannot rely on this pair of arabidopsis gene as witnesses of non-orthology.

To understand how such non-additivity arises, consider that the evolutionary distances are computed independently during the all-against-all phase. As a result, the pairs of residues aligned (thus inferred to be homologous) can be inconsistent across the different sequences and some inconsistencies can appear within the pairwise alignments (non-conservation of homologous sites Figure 12). In our example, the additivity test will fail; thus the Arabidopsis genes will not be used as witnesses of non-orthology, and the orthology inferred between the human and opossum sequence will stand (unless of course a different pair of witnesses, with additive distances this time, is found).

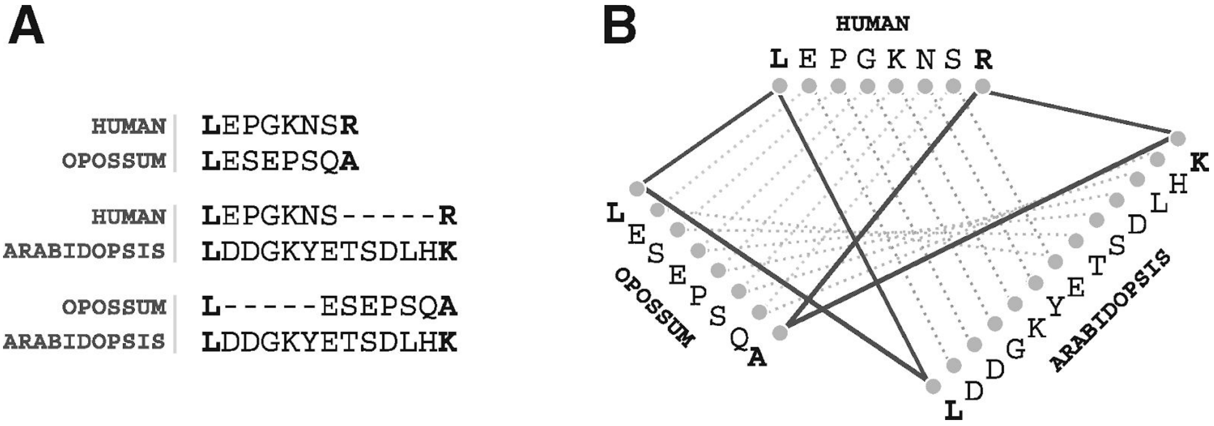


Figure 12: Example of non conservation of homologous sites across independent pairwise alignments.

A. Excerpts of three pairwise alignments between three sequences.

B. Graph-representation of the three alignments, where lines connect aligned residues. The lines are depicted as full lines if the characters are aligned consistently—thus forming closed triangles—and as dotted lines if they are aligned inconsistently—thus forming open triangles. (Sequence mapping to Uniprot Id: Human → H. sapiens|Q16874, Opossum → M. domestica|F7FI80, Arabidopsis → A. thaliana|Q93ZB2.)

3.3 QfO benchmarking results

To quantitatively assess the impact of the changes in the OMA algorithm, we submitted results obtained with them—individually and in combination—to the QfO orthology benchmark service (Altenhoff et al., 2016).

We first consider the results at the level of pairwise orthology ('OMA Pairs'). Applying the new handling of asymmetrically evolving paralogs and the additivity test separately, we observe a significant increase in the number of predicted orthologs while maintaining a similar or even slightly better precision (Figure 13). Here precision is measured in terms of average topological distance between the reference species tree and the gene tree reconstructed from the inferred orthologs (the lower the better). When the two refinements are combined, there is an even higher increase in the number of predicted orthologs compared with the current OMA predictions, while maintaining further the quality of the inferences. Consistent results are obtained for the different resolutions provided by the QfO benchmark service, though the increase in the number of inferred pairs is more modest in the fungal dataset (http://orthology.benchmarkservice.org/cgi-bin/gateway.pl?f=CheckResults&p1=25fe02429dc60c51f81da2de).

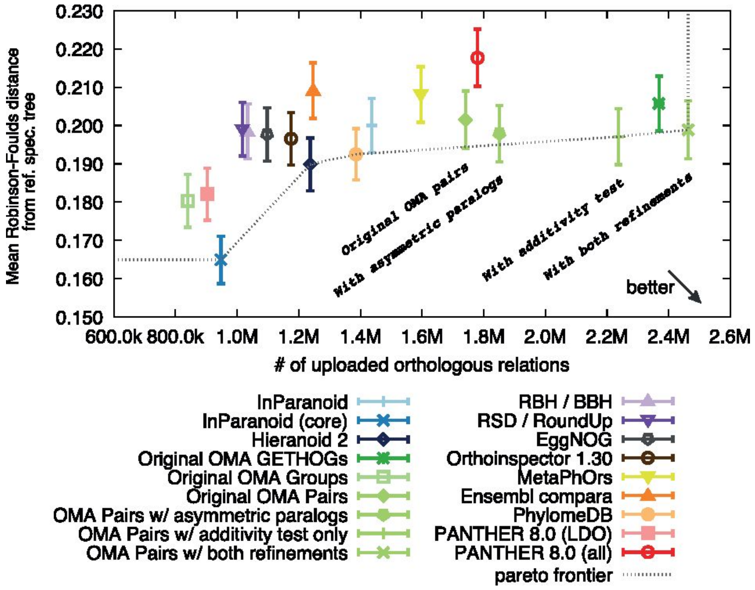


Figure 13: Effect of the refinements on pairwise orthology relationships (OMA Pairs) in the generalized species tree discordance test at vertebrate level. The asymmetric paralogs denotes the change in the OMA algorithm aiming to include fast evolving duplicated genes during orthology inferences. The additivity test denotes the new quartet consistency test added to the witness of non-orthology step. Error bars denote the 95% CI of the mean.

Next, we turn to the improvements in HOG inference. As described in more detail in section 2, the new HOG inference approach ('bottom-up GETHOGs') implements several modifications compared with the original version (Altenhoff et al., 2013): (i) The taxonomy is no longer traversed top-down but from the bottom-up, in a postfix traversal of the species tree; (ii) In the inter-HOG orthology graph considered for each clade, the nodes now represent HOGs instead of single genes, thereby considerably reducing the complexity of these graphs; (iii) The edges are weighted according to the number of orthology relations between two clusters of genes; (iv) Instead of removing spurious edges in the orthologous graph using a minimum cut algorithm, the bottom-up HOG inference enables us to assess the support of orthologous relationships between HOGs in terms of the total number of orthologous relationships that would be expected given perfect input pairwise orthologs.

To assess the impact of the change, we first compared the top-down and bottom-up variants on the QfO ortholog benchmark service on the original OMA pairs as input (i.e. without new asymmetric paralogy and additivity tests). The bottom-up algorithm resulted in a substantial increase in the number of predicted orthologs, indicating higher recall (Figure 14). On the Eukaryotic, Vertebrate, and Fungal datasets, the error rate is also markedly lower, while on the universal dataset (including bacteria, archaea and eukaryotes), the error rate is about the same (http://orthology.benchmarkservice.org/cgi-bin/gateway.pl?f=CheckResults&p1=98f077d9d00d3ab0375be957).

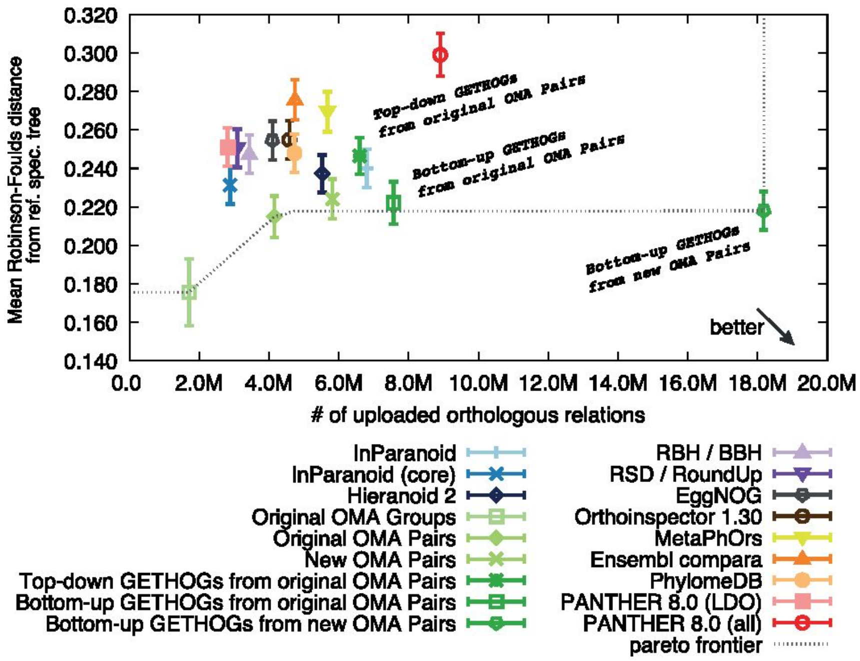


Figure 14: Assessment of HOG inference on the generalized species tree discordance test (eukaryotic dataset). Error bars denote the 95% CI of the mean. The data points with ‘original OMA’ refer to the algorithm used before this study and ‘new OMA’ refer to the predictions produced by the refinements introduced in section 2.3.

Combining the new OMA pair inference with bottom-up HOG inference results in the largest increase in predicted orthologs. On the Eukaryotic dataset, the number of predicted orthologs almost triples without negatively affecting precision (Figure 14).

In terms of time requirements, consistent with the asymptotic time complexity analysis (see section 2), the bottom-up approach is vastly more efficient and scalable (Figure 15). With 100 genomes as input, the bottom up variant is already two orders of magnitude faster. In contrast to top-down GETHOGs, which is prohibitively expensive on very large protein families (Altenhoff et al., 2013), bottom-up GETHOGs can process the entire public OMA database of 2024 genomes and 10.5M sequences in 9 CPU hours.

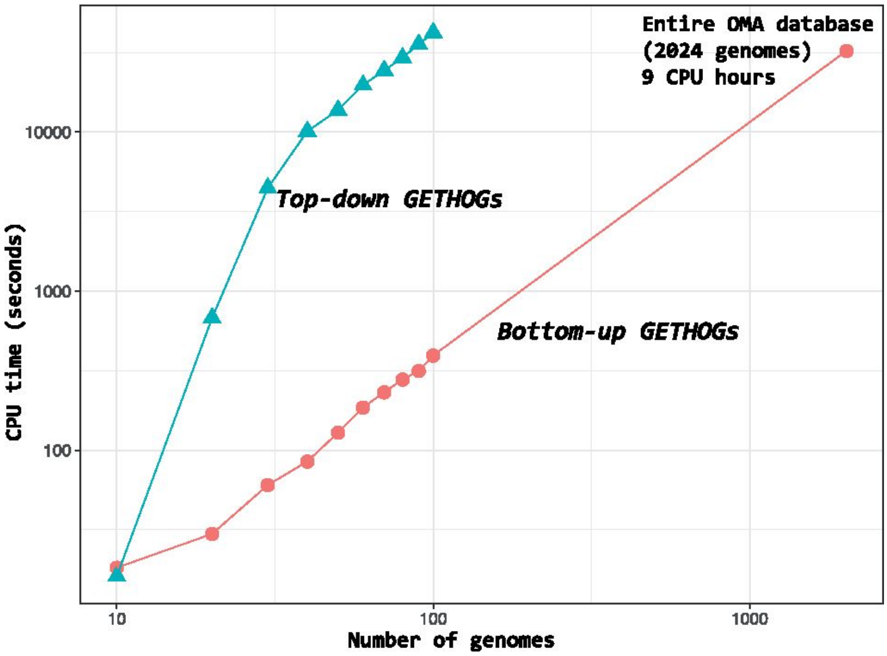


Figure 15: Time performance of GETHOGs algorithm. CPU time to compute the HOGs reconstruction on datasets of different sizes. The timing is recorded on a single instance running on a Intel(R) Xeon(R) CPU E5540 2.53GHz

4 Discussion and conclusion

When compared with other methods, the OMA algorithm has often been reported to be stringent, yielding highly reliable inferences, but suffering from low recall (Altenhoff et al., 2016;

Ballesteros and Hormiga, 2016; Trachana et al., 2011). This is certainly true of the ‘OMA groups”, which require fully connected subgraphs of orthologs. For pairs and HOGs, however, we show with this new version that recall can be considerably improved without negatively affecting precision.

Indeed, we introduced multiple improvements to the OMA algorithm, both in the inference of pairwise orthologs and in the inference of HOGs. At the pairwise level, the asymmetric paralogy test increases the number of one-to-many and many-to-many ortholog relationships recovered when the paralogous copies evolve at different rates. Furthermore, the new additivity test reduces errors due to inconsistent distance computations in quartets of sequences (used to infer differential gene losses in the OMA algorithm). These inconsistent distances often arise due to fragmented sequences, typical of draft-quality genomes.

The improvements in pairwise orthology are not only useful in and of themselves—they directly translate into better HOG inference. Combined with the more scalable and accurate bottom-up GETHOGs, the HOGs inferred by OMA are much more complete, with no or even positive impact on precision.

Some of the ideas underlying these improvements are not new. Methods such as Inparanoid (Remm et al., 2001) or OrthoInspector (Linard et al., 2011) have long been exploiting distances between inparalogs—albeit using alignment score as a proxy—to increase the robustness of one-to-many or many-to-many orthology inference. Likewise, Hieranoid (Schreiber and Sonnhammer, 2013) also infers HOGs in a bottom-up fashion.

However, the distinctive feature of the OMA algorithm has been—and continues to be with this new version—its modular approach, with well-defined and testable objectives at each step of the pipeline (e.g. inference of pairwise orthologs, detection of differential gene losses, inference of HOGs from pairwise orthologs). OMA's modular approach makes it possible to test and optimize each step in isolation, and to expect an overall improvement when these are combined—as the empirical benchmarks reported above clearly support. In contrast, ad hoc methods can prove difficult to maintain and improve over time, with changes in one part of the pipeline affecting other parts in unexpected ways.

Looking ahead, we see further opportunities for improvement. Unlike pairs and groups in OMA, inference of HOGs strongly relies on knowledge of the species tree. However, many parts of the tree of life remain either poorly resolved or even misleading for some gene families due to incomplete lineage sorting, horizontal gene transfer or hybridization (Philippe et al., 2011). Currently, we collapse branches that are uncertain—however this means that gene duplication occurring within such multi-furcations (i.e. polytomies) confound the HOG inference. Approaches taking a more flexible reading of species phylogeny, such as NOTUNG (Durand et al., 2006) or PHYLDOG (Boussau et al., 2012), may provide a better way forward. We also see considerable potential in exploiting the paralogy graph to further improve HOG inference (Lafond and El-Mabrouk, 2014).

Meanwhile, this OMA 2.0 algorithm is used in the public OMA database from the March 2017 release onwards (Altenhoff et al., 2015; http://omabrowser.org), and can be applied to custom genomes using the open source OMA standalone software version 2.0 (http://omabrowser.org/standalone).

Chapter 3: Visualisation & Data Exploration

Visual analysis of reconciled gene trees is a cornerstone of gene family evolutionary history investigation. By pinpointing duplications and speciations in reconciled gene trees, we can reconstruct the ancestral states of gene families and determine the genomic evolution underlying homologs. Such investigations can be performed using web based resources such as Ensembl (Herrero et al. 2016), EggNog (Huerta-Cepas et al. 2016), PhylomeDB (Huerta-Cepas et al. 2016) or tools such as ETE (Huerta-Cepas, Dopazo, and Gabaldón 2010) or SylvX (Chevenet et al. 2016). Nevertheless, due to the large genomic setups used or complex evolutionary histories, hierarchical orthologous groups can be inferred in large quantity and can be complex to analyse. In a 100 species dataset, there can be approximately 25,000 HOGs where some can contain up to 100,000 members. Programmatic exploration of such large scale data is mandatory. I introduce in this chapter two tools I devised to meet this need: 'pyHam' (Train et al. 2018) a python library to explore and extract phylogenetic information from OrthoXML bundled with two HOG based interactive visualisation tools, and 'GTM' (Graph-Tree-Multiple sequence alignment) a visualization tool combining an orthology graph with its related multiple sequence alignment and gene tree.

3.1 Pyham & iHam

This chapter was published in **Clément-Marie Train, Miguel Pignatelli, Adrian Altenhoff, Christophe Dessimoz, iHam and pyHam: visualizing and processing hierarchical orthologous groups, *Bioinformatics* (2019) 35:14, pp. 2504–2506.**

The evolutionary history of gene families can be complex due to duplications and losses. This complexity is compounded by the large number of species simultaneously considered in contemporary comparative genomic analyses. As provided by several orthology databases, hierarchical orthologous groups (HOGs) are sets of genes that are inferred to have descended

from a common ancestral gene within a species clade. This implies that the set of HOGs defined for a particular clade correspond to the ancestral genes found in its last common ancestor. Furthermore, by keeping track of HOG composition along the species tree, it is possible to infer the emergence, duplications and losses of genes within a gene family of interest. However, the lack of tools to manipulate and analyse HOGs has made it difficult to extract, display and interpret this type of information. To address this, I introduce interactive HOG analysis method, an interactive JavaScript widget to visualize and explore gene family history encoded in HOGs and python HOG analysis method, a python library for programmatic processing of genes families. These complementary open source tools greatly ease adoption of HOGs as a scalable and interpretable concept to relate genes across multiple species.
iHam's code is available at https://github.com/DessimozLab/iHam or can be loaded dynamically. pyHam's code is available at https://github.com/DessimozLab/pyHam and or via the pip package 'pyham'.

Background

The evolution of a gene family describes the history of all the genes that shared a common ancestral gene. Those genes called homologs can be distinguished into orthologs if they start diverging by speciation and paralogs if they start diverging by duplication (Fitch, 1970). In comparative genomics, gene families are a fundamental resource since they tend to represent the links between several organisms from a gene centric perspective and allow us to understand how genes and genomes have evolved over time. In other words, gene families contain the evolutionary history underlying present day genes and genomes.

The evolutionary history of gene families can be studied by visualizing reconciled gene trees, using web-based resources such as Ensembl (Herrero et al., 2016), HOGENOM/HOVERGEN

(Dufayard et al., 2005), EggNOG (Huerta-Cepas et al., 2016), PhylomeDB (Huerta-Cepas et al., 2014) or tools such as ETE (Huerta-Cepas et al., 2010) and SylvX (Chevenet et al., 2016). However, when considering large families across many species, reconciled gene trees can become prohibitively complex to infer and interpret.

As a scalable alternative to reconciled gene trees, the concept of Hierarchical Orthologous Groups (HOGs) is increasingly adopted. HOGs generalize Fitch's definition of orthology to more than two species, by grouping sequences that have descended from a common ancestral gene within a clade of interest. Thus, the set of all HOGs defined for a given clade corresponds to the set of ancestral genes in the common ancestor of that clade. Furthermore, if HOGs are available for nested clades (e.g. vertebrates versus mammals), the difference between their HOG repertoires imply gene duplication and loss events on the branch separating them: a HOG split implies a duplication, while a HOG disappearance implies a loss.

HOGs are inferred by several leading orthology databases such as OrthoDB (Zdobnov et al., 2017), EggNOG (Huerta-Cepas et al., 2016), HieranoidDB (Kaduk et al., 2017) or OMA (Altenhoff et al., 2018). In OMA, for instance, some HOGs connect large gene families of over 100 000 members across 1000's of genomes. Because of this complexity, manual exploration of gene families encoded in HOGs can be challenging. Currently, there is a lack of tools for visualizing, exploring and processing HOGs to tackle specific biological questions.

In this application note, we introduce two tools to facilitate the visualization and analysis of HOGs: interactive HOG analysis method (iHam) for web-based interactive visualization and exploration of individual HOGs and python HOG analysis method (pyHam) to perform aggregate analyses.

iHam

iHam is an interactive JavaScript tool to visualize the evolutionary history of a specific gene family encoded in HOGs. The viewer is composed of two panels (Figure 16.A): a species tree which lets the user select a node to focus on a particular taxonomic range of interest, and a matrix that organizes extant genes according to their membership in species (rows) and HOGs (columns). The tree-guided matrix representation of HOGs facilitates: (i) delineation of orthologous groups at given taxonomic ranges, (ii) inference of duplication and loss events in the species tree, (iii) gauging the cumulative effect of duplications and losses on gene repertoires and (iv) identification of potential mistakes in genome assembly, annotation or orthology inference (e.g. if losses are concentrated on terminal branches—suggestive of incomplete genomes; or if the species coverage within a HOG looks implausible—suggestive of orthology inference error).

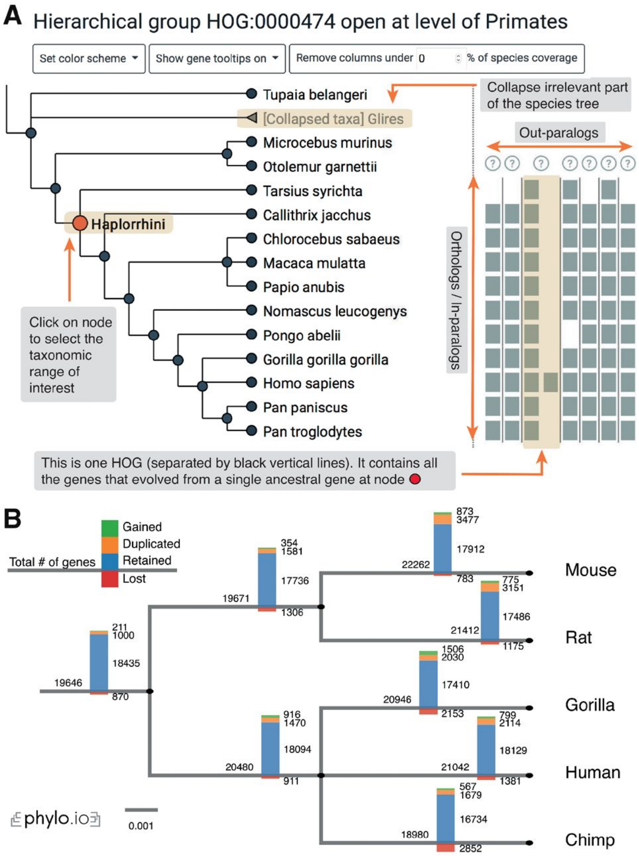
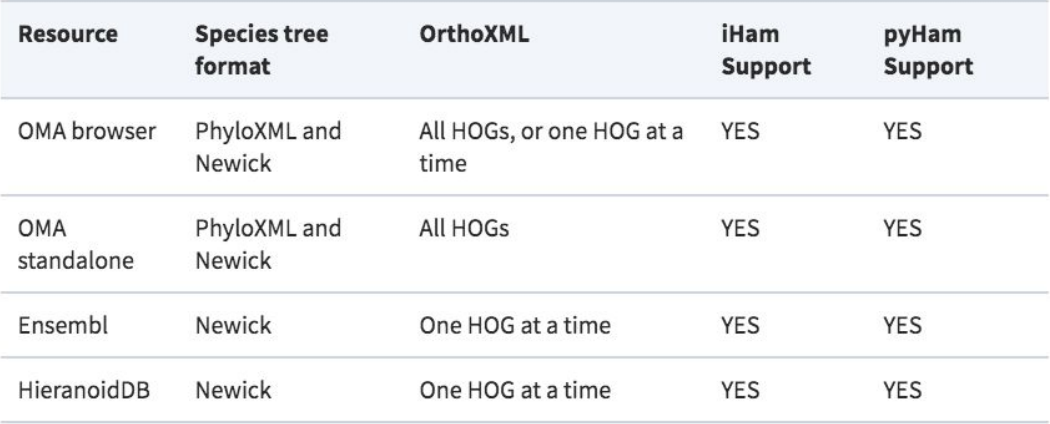


Figure 16: iHam and pyHam visualization tools.
A. An iHam excerpt of the Tetraspanin family at the Haplorhini level: the tree depicts relationships between species, squares depict genes and HOGs are delineated by vertical bars.
B. pyHam can be used to map gene losses, duplications or new appearances ('gained') onto species trees (here, using the NCBI taxonomy tree).

Users can customize the view in different ways. They can color genes according to protein length or GC-content. Low-confidence HOGs can be masked. Irrelevant species clades can be collapsed. iHam is a reusable web widget that can be easily embedded into a website; for instance, it is used to display HOGs in OMA (http://omabrowser.org; Altenhoff et al., 2018). Implemented as a JavaScript library using the TnT framework (Pignatelli, 2016), iHam merely requires as input HOGs in the standard OrthoXML format (Schmitt et al., 2011) and the underlying species tree in newick or PhyloXML format (supported resources listed in Table 1).



Resource	Species tree format	OrthoXML	iHam Support	pyHam Support
OMA browser	PhyloXML and Newick	All HOGs, or one HOG at a time	YES	YES
OMA standalone	PhyloXML and Newick	All HOGs	YES	YES
Ensembl	Newick	One HOG at a time	YES	YES
HieranoidDB	Newick	One HOG at a time	YES	YES

Table 1: Support for iHam and pyHam by various HOG inference resources

pyHam

pyHam makes it possible to extract useful information from HOGs encoded in standard OrthoXML format. It is available both as a python library and as a set of command-line scripts. Input HOGs in OrthoXML format are available from multiple bioinformatics resources, including OMA, Ensembl and HieranoidDB (Table 1).

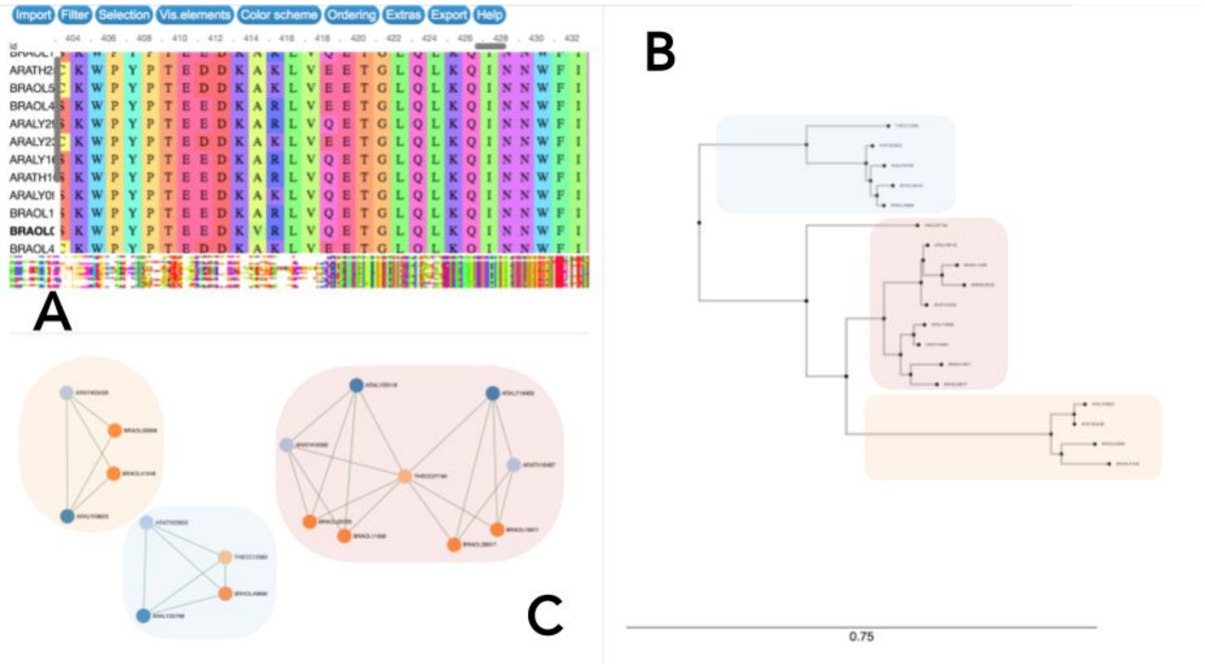
The main features of pyHam are: (i) given a clade of interest, extract all the relevant HOGs, each of which ideally corresponds to a distinct ancestral gene in the last common ancestor of the clade; (ii) given a branch on the species tree, report the HOGs that duplicated on the branch, were lost on the branch, first appeared on that branch or were simply retained; (iii) repeat the previous point along the entire species tree and plot an overview of the gene evolutionary dynamics along the tree (Figure 16.B) and (iv) given a set of nested HOGs for a specific gene family of interest, generate a local iHam web page to visualize its evolutionary history.

Conclusion

pyHam and iHam are two complementary tools providing a solution to ease the in depth exploration and visualisation of large gene families. The combination of iHam and pyHam enable users to unlock the full potential of HOGs.

3.2 GTM

The analysis of a gene family requires a meticulous investigation of several key taxonomic ranges to understand the evolutionary history underlying extant genes. I develop a visualisation tool to facilitate the analysis of a set of genes called GTM (for ‘Graph-Tree-Multiple sequence alignment’) that combines 3 types of phylogenetic information: an orthology graph, a multiple sequence alignment and its related phylogenetic gene tree. I developed GTM as an interactive javascript tool that combines several existing libraries: MSAViewer was developed by (Yachdav et al. 2016) and the phylo.io was developed by (Robinson et al. 2016)). As illustrated in figure 17, GTM allows visualisation of the underlying phylogenetic landscape of the genes of interest for a given gene family (KNOX2) at a specific taxonomic range (Malvids). Indeed, we can easily hypothesize about the presence of 3 ancestral genes in this family at the Malvids level (grouped in colored boxs in figure 17.B and 17.C).



```
id       . 404 . 406 . 408 . 410 . 412 . 414 . 416 . 418 . 420 . 422 . 424 . 426 . 428 . 430 . 432
ARATH2   C K W P Y P T E D D K A K L V E E T G L Q L K Q I N N W F I
BRAOL5   C K W P Y P T E D D K A K L V E E T G L Q L K Q I N N W F I
BRAOL4   S K W P Y P T E E D K A R L V E E T G L Q L K Q I N N W F I
ARALY2   S K W P Y P T E E D K A R L V Q E T G L Q L K Q I N N W F I
ARALY2   C K W P Y P T E D D K A K L V E E T G L Q L K Q I N N W F I
ARALY1   S K W P Y P T E E D K A R L V Q E T G L Q L K Q I N N W F I
ARATH1   S K W P Y P T E E D K A R L V Q E T G L Q L K Q I N N W F I
ARALY0   S K W P Y P T E E D K A K L V Q E T G L Q L K Q I N N W F I
BRAOL1   S K W P Y P T E E D K A R L V Q E T G L Q L K Q I N N W F I
BRAOL(   S K W P Y P T E E D K V R L V Q E T G L Q L K Q I N N W F I
BRAOL4   C K W P Y P T E D D K A K L V E E T G L Q L K Q I N N W F I
```

Figure 17: GTM of the KNOX 2 family at Malvids.

A. Multiple sequence alignment panel, using the MSA viewer from github.com/wilzbach/msa.

B. Phylogenetic genes tree, using the phylo.io javascript library from phylo.io.

C. Orthology graph, extant genes are denoted by circles colored by species while lines denote orthologous relations. Each colored boxe in B,C represent ancestral genes at Malvids levels.

The presented version of GTM (figure 17) has been developed and will be integrated into the OMA browser in future releases. Several aspects are still under development, such as improving the interoperability among the different panels (e.g. selected elements in the graph and highlighting them in the tree and sequence alignment), facilitating the integration of this tool in websites or custom analysis, and other user interface refinements.

Chapter 4: Towards a better understanding of HOG inference mistakes

Despite the substantial improvements achieved in Chapter 2, HOG inference is far from being perfect. The goal of this chapter is to gain insights into the types of errors the GETHOGs algorithm makes. For this, we use a two-fold strategy: a benchmark on a simulated dataset and detailed case studies on real data from the Quest for Orthologs dataset.

Simulation study using ALF

The first part of the strategy aims to assess the potential limits of our new GETHOGs 2.0 algorithm to infer HOGs by using simulated data. Indeed, to be able to identify mistakes and to characterise the proportion of correct assignments in our HOG inferences, we need to know the true evolutionary history of the gene families inferred. Previous work has been performed by Dalquen and Dessimoz (Dalquen et al. 2013) using simulated data by the Artificial Life Framework (ALF) (Dalquen et al. 2012) to evaluate the advantages and limitations of BBH for pairwise orthology inference under various evolutionary scenarios. However, that work was limited to pairwise orthology benchmarking and not oriented towards assessing the quality of gene family reconstruction. Fortunately, ALF provides the following information when simulating gene family evolution: the reference species phylogeny, the true gene trees, the perfect orthologous relations and other resources that can be used as references for benchmarking.

In our strategy to benchmark the performance of GETHOGs on simulated data, we simulated several genomic setups with various parameters to mimic different evolutionary processes: a first dataset is simulated only with duplications and losses as evolutionary events for each gene family, while a second dataset uses the same parameters with an additional probability to have gene fusion/fission occurring after a duplication event. In order to assess the quality of our

HOGs reconstruction for each of these different simulated datasets, we ran GETHOGs inputting either the perfect orthology graph provided by ALF or the orthology inferred by running OMA on the simulated proteomes. We designed 3 measures to assess the quality of our inferences: the quality of the input orthology graph (Measure A), the completeness of the reconstructed genes families (Measure B) and the quality of the HOG clustering (Measure C).

Methods

Genome wide simulation

We used the ALF web interface (Dalquen et al. 2012) to build our two simulated genomic datasets. They both use the same general parameters except one variant contains gene fusion and fission. The simulation uses as ancestor an ancestral genome of 1000 genes with minimum 50 amino acids per gene with the following default globin family settings : a gene duplication rate of 0.001, a gene loss rate of 0.001 and, for the fusion and fission simulation variant, a fission rate for duplicated genes of 0.1 and a fusion rate for duplicated genes of 0.1. Using these settings, ALF simulates 36 genomes with the related 1000 genes families. Each genome is represented by 2 FASTA files with the amino or the nucleic acid sequences of all genes. In addition, ALF outputs the perfect pairwise orthologous relations and the related true gene trees and multiple sequence alignment for all gene families. For this benchmark, we used the proteomes, the reference species trees, the true gene trees and the perfect pairwise orthology. The first simulation with default parameters was denoted as 'default dataset' (composed of 993 gene families), while the second dataset variant with fusion and fission was denoted as 'fusion-fission dataset' (composed of 1000 gene families with potential fragmented sequences in order to simulate poor-quality input data).

Orthology inferences

In real conditions, the orthology calling is not perfect due to orthology inference method limitations meaning that they can infer spurious (“false-positive”) relations or miss some true relations (“false-negative”). In this benchmark, we wanted to both assess the quality of the orthology inference and of the HOG inferred using these orthology relations. We thus used the amino acid sequences to infer pairwise orthology using OMA standalone version 2.3.1 Adrian M. Altenhoff et al.) with default parameters on the two simulated datasets.

Measure A: quality assessment of pairwise orthology

The first measure aims to estimate the amount of spurious and missing orthology relations in the orthology graph inferred by OMA and later used to reconstruct HOGs. Indeed, GETHOGs uses the pairwise orthologous relations as core data for its HOG inferences, making its performance highly dependent on the accuracy of the orthology graph. In our analysis, we have two types of orthology graph per simulation: the ‘perfect orthology graph’ which is directly provided by ALF based on the true evolutionary history simulated and the ‘inferred orthology graph’ which is inferred by using OMA standalone. While the perfect orthology graph contains no spurious or missing pairwise orthology, the inferred orthology graph may contain spurious orthology relations or lack some expected orthology due to the imperfect nature of orthology inference algorithms to deal with edge case scenarios (fast evolving genes, fragmented sequences, domain shuffling, etc...). In order to estimate the percentage of mistakes in the inferred orthology graph, the first measure takes as input the perfect orthology graph provided by the simulated framework and the orthology graph inferred using OMA on the simulated proteomes. Then, a simple pairwise comparison is performed on the two graph edges to detect the edges only present in the perfect graph (missing orthology) and the ones only present in the inferred graph (spurious orthology). This measure indicates the percentage of missing (false negative) and spurious (false positive) pairwise orthology in the inferred orthology graph.

Measure B: family-level delineation (Broad HOG delineation)

The second measure estimates the completeness of the reconstructed gene families in terms of gene membership. During the reconstruction of HOGs, the aggregation of orthologous groups is subject to errors due to the imperfect nature of the orthology graph as discussed in the previous section. This may result in split gene families if two orthologous groups are not assigned to the same HOG due to missing orthologous relations between their member genes or, on the contrary, in orthologous groups wrongly clustered together due to spurious orthology relations. The idea here is to look for each gene in a gene family to which HOGs it belongs to. This will help to estimate in how many HOGs each gene family is split. The lower this number is, the better the reconstruction have been. We can report a few types of scenarios: (i) a true gene family from the simulation that overlaps with one or more inferred HOGs, (ii) one HOG spanning over two or more true gene families, or (iii) a combination of (i) and (ii) where several gene families covered by multiple HOGs due to inference errors.

This measure is calculated using as input the true gene trees from the simulation as reference and the reconstructed HOGs. The goal is to create clusters of connected HOGs and gene trees according to their gene membership overlap. This is done by building a graph where nodes are either gene trees or HOGs and edges represent an overlap of one or more genes between the two nodes. A simple connected component search retrieves the previously described clusters of gene trees/HOGs.

The measure outputs the amount of true single gene trees that spanned over one or several HOGs and inversely the number of true genes trees that are connected through HOGs genes membership.

Measure C: Accuracy of implied gene tree (Fine HOG delineation)

While the second measure evaluates the completeness of each gene family in terms of gene membership, it doesn't bring any information as to how those genes are structured inside the HOGs. We introduce a third measure to assess the accuracy of the internal structure of the HOGs and to ensure that nesting of orthologous groups along with their related duplications are as correct as possible. The principle is to first select gene trees from the second measure that can be fully sampled with only one HOG at the root level. Then, the idea is to compute the Robinson-Foulds (RF) distance between the true gene tree provided by the simulation framework and the gene tree induced by the HOG clustering; as explained in the introduction there is a one-to-one correspondence between HOG and gene tree. This will provide an estimator to evaluate how accurate the clustering is in terms of orthologous group delineation and duplication placement. Since several duplications may occur in between two speciation events and GETHOGs can only create one between two taxonomic levels, the consecutive duplicates between two levels in the true gene trees were collapsed and treated as polytomies.

Results

As described in the genome-wide simulation section above, we simulate 2 datasets of proteomes: the *default dataset* where no gene fusion and gene fission events are observed and the *fusion-fission dataset* where these events are likely to occur after the gene duplication.

HOGs reconstruction performance using perfect orthology graph

The primary aspect we benchmark in this simulation study is the performance of the GETHOGs algorithm using perfect orthology graph. To proceed, we calculate the family level delineation (measure B) and accuracy of the implied gene trees (measure C) on the two datasets to assess the completeness and the accuracy of the HOGs reconstructed. We obtained the following results for the two datasets (illustrated in figure 18):

- **Family level delineation (measure B):** This shows that 99.4% / 99% (default dataset / fusion-fission dataset) of the gene families have a one-to-one correspondence between the true gene tree and HOGs. For the rest, we observed that 0.6% / 0.5% (default dataset / fusion-fission dataset) correspond to gene trees which are split into two HOGs. Such cases occurred because these gene families start by a duplication event in the reference gene tree. These scenarios will necessarily be split into different HOGs because by definition, the deepest event in a HOG is a speciation event. The remaining 0.5% for the fusion-fission dataset represent many-to-many tree-HOGs connections, i.e trees that are covered by multiple HOGs due to inference errors.

- **Accuracy of the implied gene trees (measure C):** this measure shows that 100% / 100% (defaults dataset / fusion-fission dataset) of the gene families are perfectly reconstructed with a RF distance of 0 between the true gene trees and the HOGs.

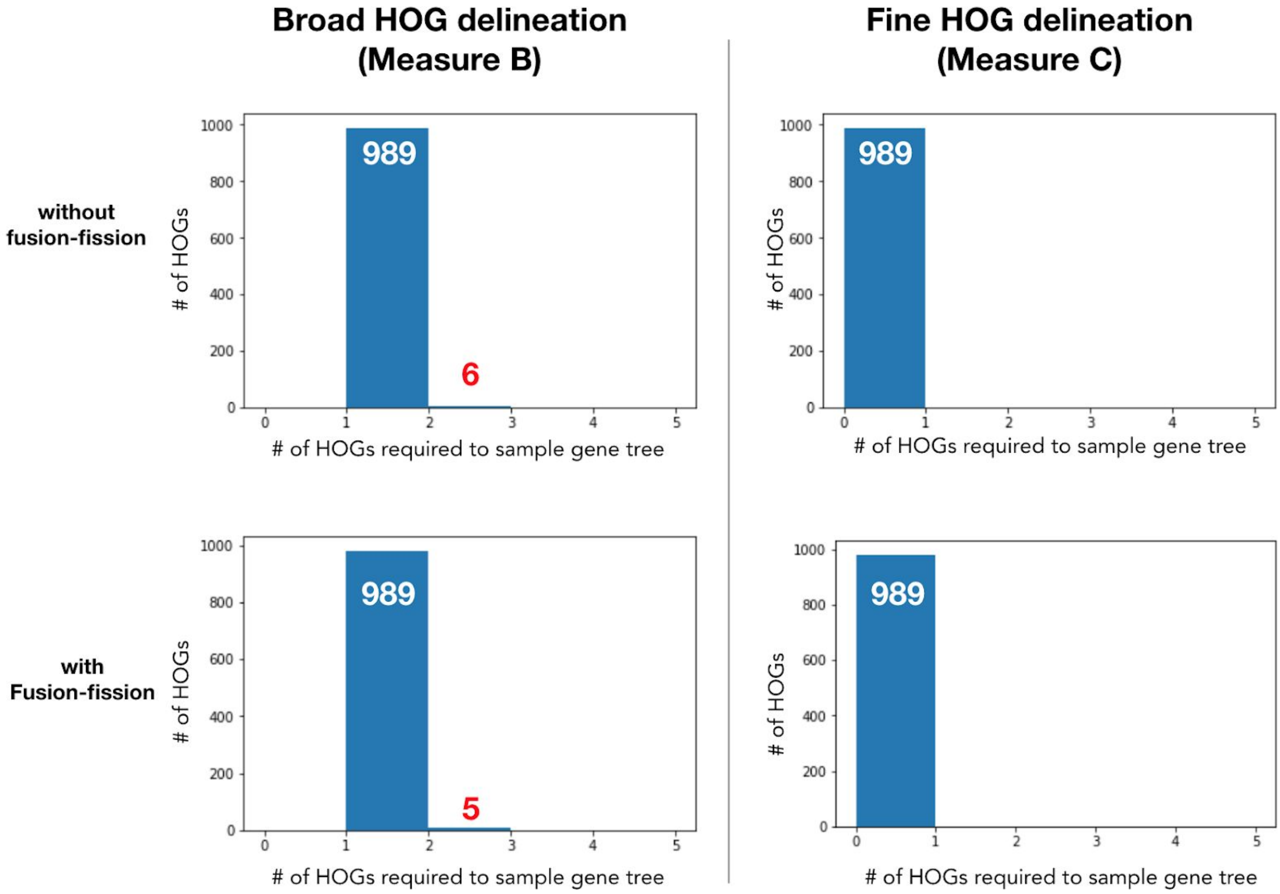


Figure 18: Broad HOG delineation and fine HOG delineation using perfect orthology graph.

Such results are expected from the theory and in agreement with the results of the original GETHOGs paper (Altenhoff et al. 2013) showing that perfect input data will produce perfect HOGs due to the absence of mistakes and uncertainty in the orthology graph. This first segment of the benchmark shows that our implementation of the GETHOGs algorithm is correct.

GETHOGs performance using inferred orthology graph from simple simulation context

The second aspect of the benchmark strategy is to assess how well the GETHOGs algorithm performs when we introduce spurious and missing data in the orthology graph in a simple evolutionary setup (only gene duplications and gene losses are simulated). To proceed, we inferred the orthology on the default dataset using OMA standalone version 2.3.1 and we ran the triplet of measures A, B and C to assess the number of mistakes in the orthology graph, the completeness of the HOG reconstruction and the accuracy of the gene families, respectively, when introducing mistakes in the orthology graph. We obtain the following results for the default dataset on the 3 measures (illustrated in figure 19):

- **Quality of the pairwise orthology (measure A):** The benchmark shows that 97.6% of the expected orthologous relations are correctly inferred with 2.4% of the expected orthology missing. In addition, the test reports that 6.1% of the inferred pairwise orthologous relations are considered as spurious.
- **Family level delineation (measure B):** The benchmark shows that 98% of the gene families have a one-to-one correspondence between true gene tree and HOGs. As described previously, we see that 1.8% of HOGs correspond to gene trees which are split in two HOGs due to a duplication at the root level; such cases are impossible to solve for GETHOGs since paralogous groups can only be created between two existing taxonomic ranges. The remaining 0.2% represents many-to-many gene tree/HOGs sampling, i.e. several gene trees that were covered by multiple HOGs due to inference errors.
- **Accuracy of the implied gene trees (measure C):** the benchmark shows that 68.4% of the gene families are perfectly reconstructed with an RF distance of 0 between the true gene trees and the HOGs, and that the remaining 31.6% of the HOGs have an observed RF ranging from 2 to 101. Such discordance between the true gene trees and the reconstructed HOGs is due to mistakes in the orthology graph that are wrongly orienting the placement of gene duplications and ortholog clustering by GETHOGs (as illustrated in figure 20).

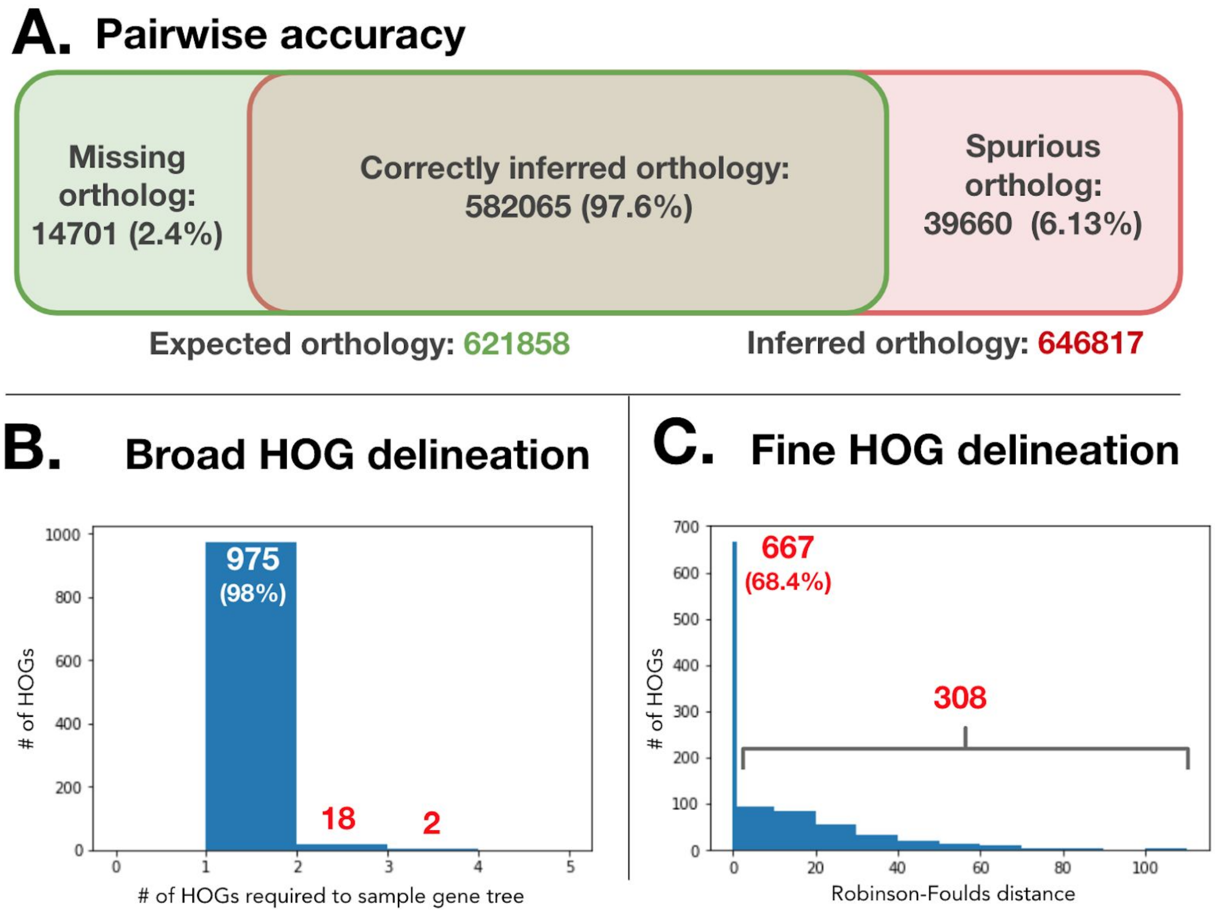


Figure 19: Benchmark of the pairwise accuracy, broad HOG delineation and fine HOG delineation using OMA inferred orthology graph.

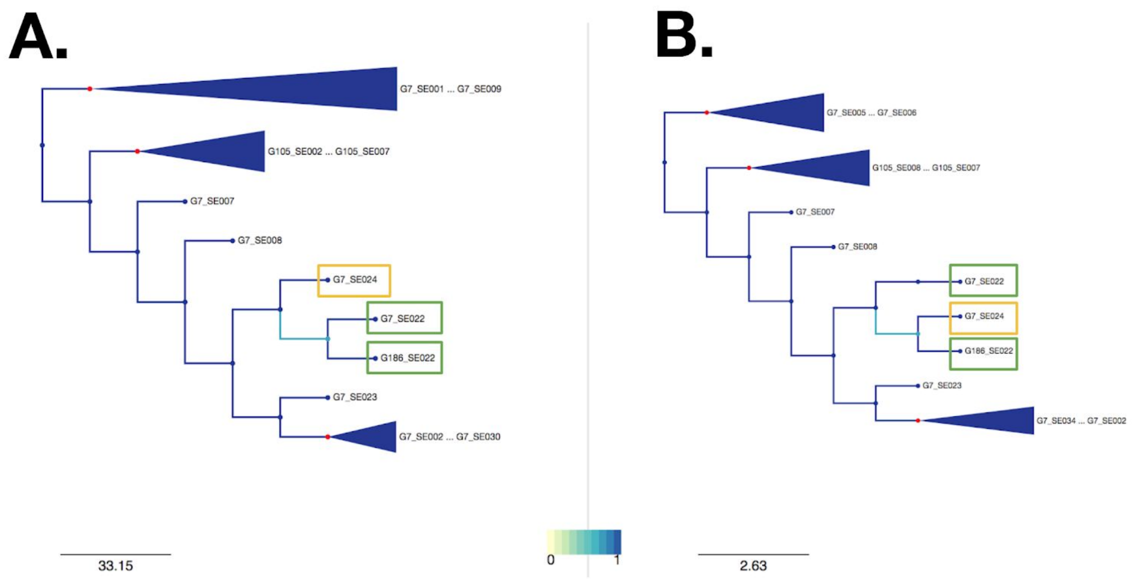


Figure 20: Example of discordance between perfect gene tree and reconstructed HOG with a Robinson-Foulds distance of 4. The tree comparison is performed by phylo.io (Robinson, Dylus, and Dessimoz 2016) where the true gene tree **(A)** is compared to the gene tree predicted by the inferred HOG **(B)**. Correct parts of the tree (without discordance) are collapsed. The three border colored boxes represent the 3 genes involved in a topological difference between the two trees (green is gene 7 and gene 186 in species 22, yellow is gene 7 in species 24). In the true gene tree (A), we observe that the orange gene (species 24) and the green genes (species 22) diverge by a speciation event with a later species-specific duplication. In the reconstructed HOGs using an imperfect orthology graph, we observed that the green genes are separate. This can be explained by the fact that a missing orthologous relation between one copy of the green gene and the yellow force the algorithm to cluster the two connected genes together and later incorporate the second green gene as paralog.

These benchmarks show that, even if the orthology graph contained few mistakes (less than 5%) due to orthology inference methods, GETHOGs still gives good quality reconstruction with a completeness of 98% and an accuracy of 68.4% perfect reconstruction, and 31.6% of HOGs correctly sampled but with clustering errors leading to some discordance between perfect gene trees and HOGs.

HOGs reconstruction performance using inferred orthology graphs from realistic simulation contexts

The last aspect of this benchmark strategy is to assess the quality of the GETHOGs inferences when more mistakes are introduced in the orthology due to complex evolutionary scenarios, such as gene fusion or fission. As in the previous section, we first inferred the orthology on the fusion-fission dataset using OMA standalone version 2.3.1 and we ran the triplet of measures A, B and C on the reconstructed HOGs in order to assess the amount of mistakes in the orthology graph, the completeness of the HOG reconstruction and the accuracy of the gene families, respectively. We obtain the following results for the default dataset for the three tests (illustrated in figure 21):

- **Quality of the pairwise orthology (Measure A):** The benchmark shows that 96.9% of the expected orthologous relations are correctly inferred with 3.1% of the expected orthology missing. In addition, the test reports that 4.8% of the inferred pairwise orthologous relations are considered spurious.
- **Family level delineation (measure B):** The benchmark shows that 95.3% of the gene families have a one-to-one correspondence between true gene trees and HOGs. The 4.7% remaining HOGs correspond to gene trees that require several HOGs for a complete sampling. The number of required HOGs ranges from 2 to 36. In addition, 4 cases where several gene trees were covered by multiple HOGs are reported.
- **Accuracy of the implied gene trees (measure C):** the benchmark shows that 68% of the gene families are perfectly reconstructed with an RF distance of 0 between the true gene trees and the HOGs, and that the remaining 32% of the HOGs have an observed RF ranging from 2 to 117. Such discordance between the true gene trees and the reconstructed HOGs is due to mistakes in the orthology graph that are wrongly placing the gene duplication and interfering with the orthologous clustering of GETHOGs (as previously illustrated in figure 20).

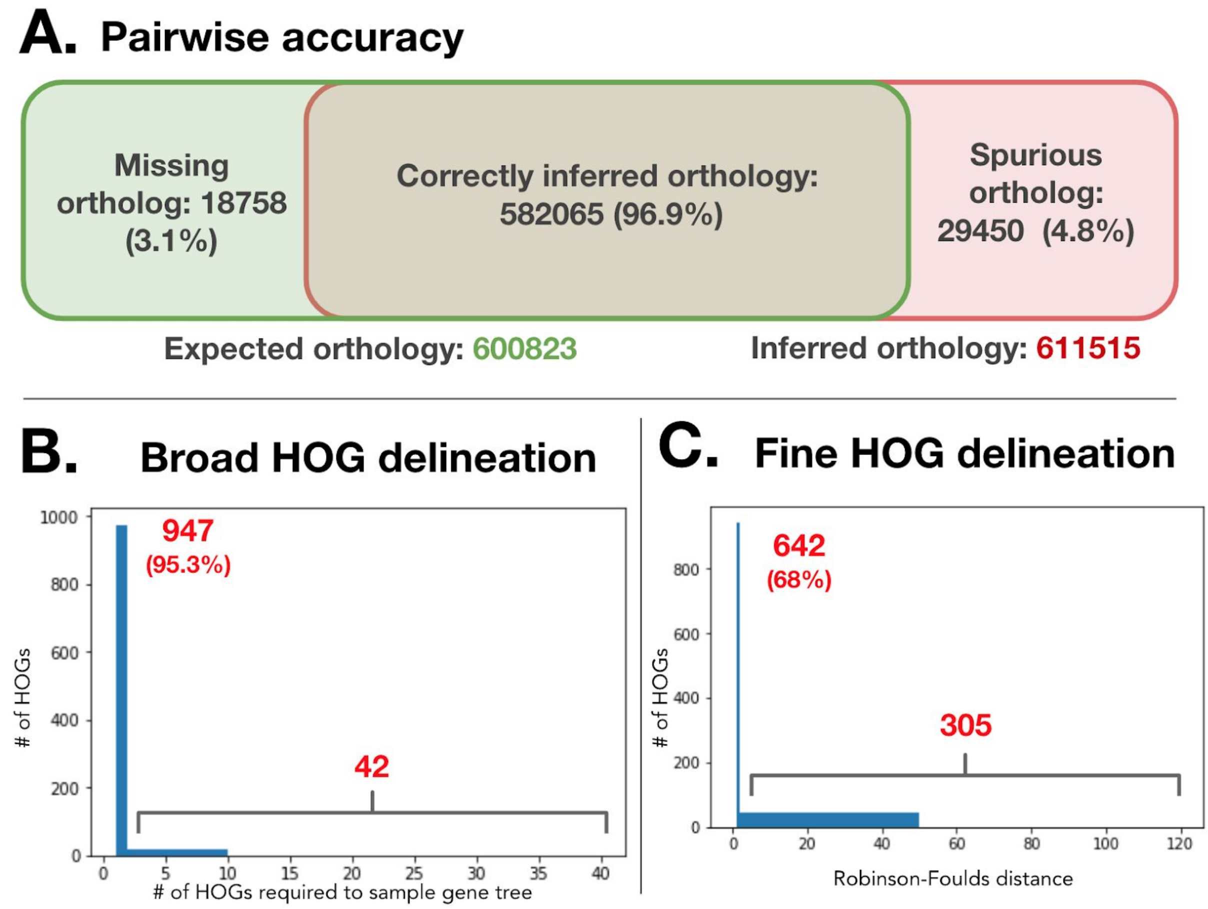


Figure 21: Benchmark of the pairwise accuracy, broad HOG delineation and fine HOG delineation using OMA inferred orthology graphs on complex simulated evolutionary scenario.

Conclusion

With the first benchmark we observe that perfect orthology relations result in perfect HOGs when using GETHOGs, in agreement with previously published results (Altenhoff et al. 2013). A small proportion of mistakes (< 0.2%) are observed (Figure 18) due to families that start with a duplication event, which is conceptually not possible to reconstruct based on HOGs definition and OrthoXML format specifications.

If we introduce a small proportion of mistakes in the pairwise relations due to orthology inference methods (2.4% of missing relations and 6.1% of spurious relations), we observe that

98% of the family delineation is correct, with the remaining 2% of trees fully sampled with 2 or 3 HOGs. For the implied gene trees quality, we observe that 68.4% of families are perfectly reconstructed and that the remaining 31.6% have an RF distance ranging between 2 and 101.

If we use a more realistic evolutionary scenario with gene fusion and gene fission with inferred orthology relations using OMA (3.1% of missing relations and 4.8% of spurious relations), we observe that 95.3% of the family delineation is correct with the remaining 4.7% trees fully sampled with the number of HOGs ranging from 2 to 36. For the implied gene trees quality, we observe that 68% of families are perfectly reconstructed and that the remaining 32% have an RF distance ranging between 2 and 117. In comparison with the results in the previous section, we see that adding more realistic constraints to the simulation strongly affects the family level delineation. Indeed, without gene fusion/fission, the numbers of HOGs required to cover a single gene family spans from 1 to 3. Here, with gene fusion/fission, the maximum number required increased to 36. Even though the proportion of mistakes in the orthology graph is very similar, these changes greatly affect the reconstruction.

Case studies

To complement the simulation-based analysis, we also performed individual case studies of gene families, on real data, of HOG reconstructed using GETHOGs 2.0. To proceed, we use the reference proteomes of the *Quest for Orthologs* initiative (Adrian M. Altenhoff et al. 2016), along with their related phylogeny (both version of August 2018) as input dataset. We infer pairwise orthology and HOGs using OMA standalone version 2.3.0 with default parameters. We process and explore the HOGs using pyHam, visualise HOG structure using iHam and produce visualization graphs for each family at different levels (orthology graph, gene tree and multiple sequences alignment) using GTM (see chapter 3). The multiple sequence alignment is performed using MAFFT version 7.221 with default parameters and the tree building is performed using Fasttree version 2.1.1 using the default parameters.

Example HOG #1: YIPF protein and fragmented genes

Description of reference gene family evolutionary history

As illustrated in Figure 22, the YIPF gene family is ubiquitous in the whole Eukaryotic clade but for the sake of this case study, we will focus our investigation on the Eumetazoa clade. We can infer from the gene tree topology that a single duplication occurred in this clade at the level of Euteleostomi (sub mammalian group) resulting in 2 gene copies for all the Euteleostomi species.

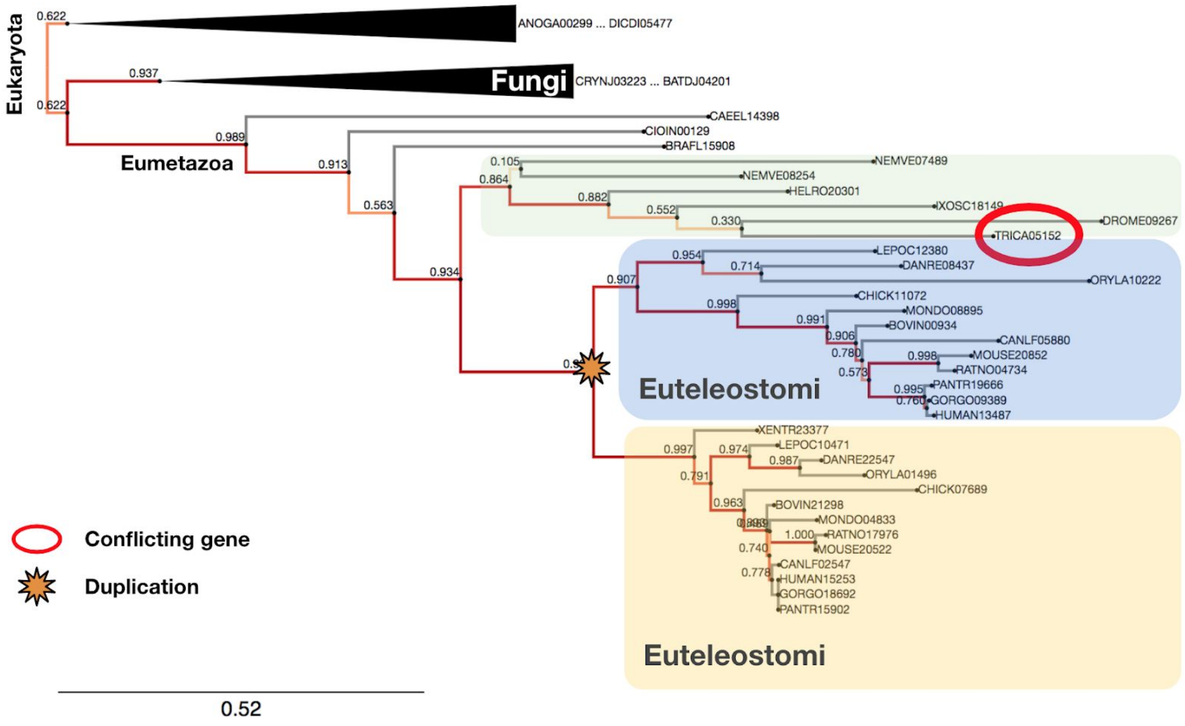


Figure 22: Gene tree of the YIPF protein family.

Characterisation of the HOG reconstruction errors

Nevertheless, the reconstructed HOG observed in figure 23 is not in agreement with the evolutionary history described previously and supported by the gene tree topology (Figure 22). We see in panels A and B of figure 23 that the gene family remains single copy until the level of Euteleostomi where a duplication occurred leading to two gene copies per species, in agreement with the tree topology. If we now consider panels C and D of figure 23, we can see that a duplication has been inferred between the Protostomia and Ecdysozoa level, implying the existence of two HOGs, albeit with a suspicious complementary pattern. The complementarity here refers to the non overlapping species coverage inside each orthologous group between the single tribolium castaneum gene HOG and the other HOG in which only the tribolium castaneum gene is not represented. The consequence of this duplication placement is that it creates a spurious duplication along with two paralogous groups where there should be only one HOG and, importantly, a large number of independent genes losses are wrongly implied. Using the iHam visualisation, we can see that 4 independent gene losses are required inside the Ecdysozoa clade to explain such clustering. In addition to the fact that this clustering is not in agreement with the supported gene tree topology shown in Figure 22, this evolutionary scenario is not likely to happen due to the large number of independent gene loss events following the spurious duplication. The most likely scenario would be that these two 'complementary' Ecdysozoa HOGs should be combined, firstly removing the wrong duplication at this level, as well as all the spurious genes losses.

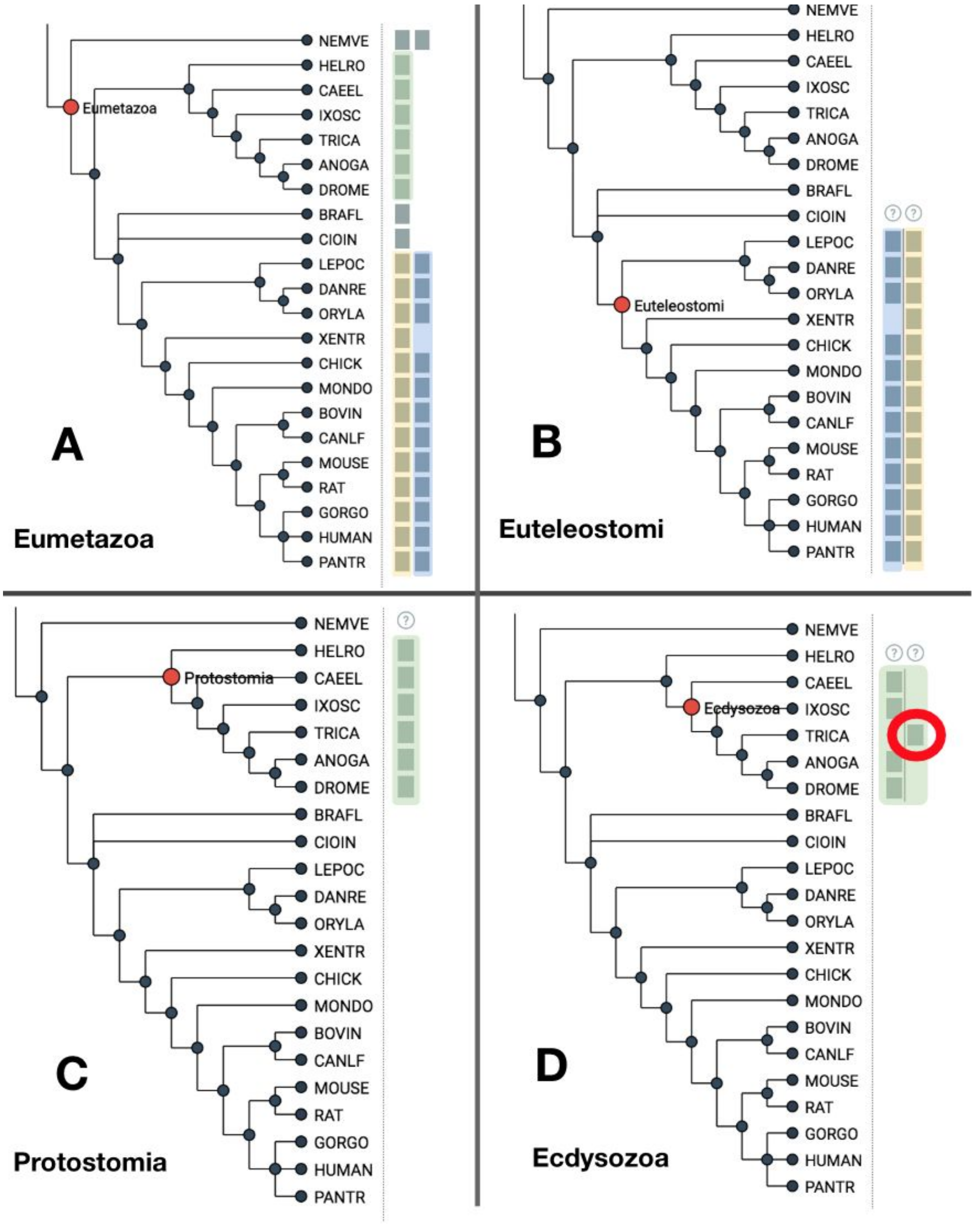


Figure 23: iHam visualisation of the HOG clustering of the YIPF protein family. A,B,C,D panels focus the visualisation on the Eumetazoa, Euteleostomi, Protostomia and Ecdysozoa clade respectively. The semi transparent colored rectangles denote the same genes colored in figure 22.

Investigation of the orthology graph

In order to understand what happened in such a case, we need to investigate the reconstruction itself by first establishing if errors arise at the level of pairwise orthology inference or later, during the HOG reconstruction. To proceed, we plot the related orthology graph at the level of the conflict, i.e. that of Protostomia. It is important here to mention that the orthology graph shown is produced by looking at the pairwise orthology inferred by OMA later used as input by GETHOGs, and not constructed by the pairwise orthology induced by the HOG clustering. This is an important point to clarify here because differences may be observed between the input pairwise orthology inferred by OMA here and the pairwise orthology implied by the HOG clustering. Indeed, whenever the GETHOGs algorithm clusters or splits groups of genes due to incomplete orthology (this is where the merge threshold of GETHOGs has an impact) an orthology discruptcy is created.

As shown in figure 23, the conflicting tribolium castaneum gene highlighted in red is not connected through orthology to any other gene at the level of Ecdysozoa (the whole graph except for the *Helobdella robusta* (HELRO) gene). At the upper level of Protostomia, we see that the Tribolium castaneum gene is now connected to its counterpart Protostomia. The orthology graph inferred using OMA is clearly in disagreement with the supported gene tree topology. Based on this orthology graph, we can understand where the reconstruction is making a mistake by splitting the supposedly single orthologous group at Ecdysozoa level into two paralogous groups due to the lack of intra-HOG pairwise orthology.

If we now look at the multiple sequence alignment in Figure 24.A, we can see that the *Tribolium castaneum* gene is fragmented. Nearly 40 percent of the aligned sequence is composed of gaps. This is affecting the orthology calling and results in a lack of orthologous relations with its closely related sibling Ecdysozoa genes, but when including more distant genes such as the Protostomia genes, the orthology is finally recovered and the orthology clustering can continue without problems.

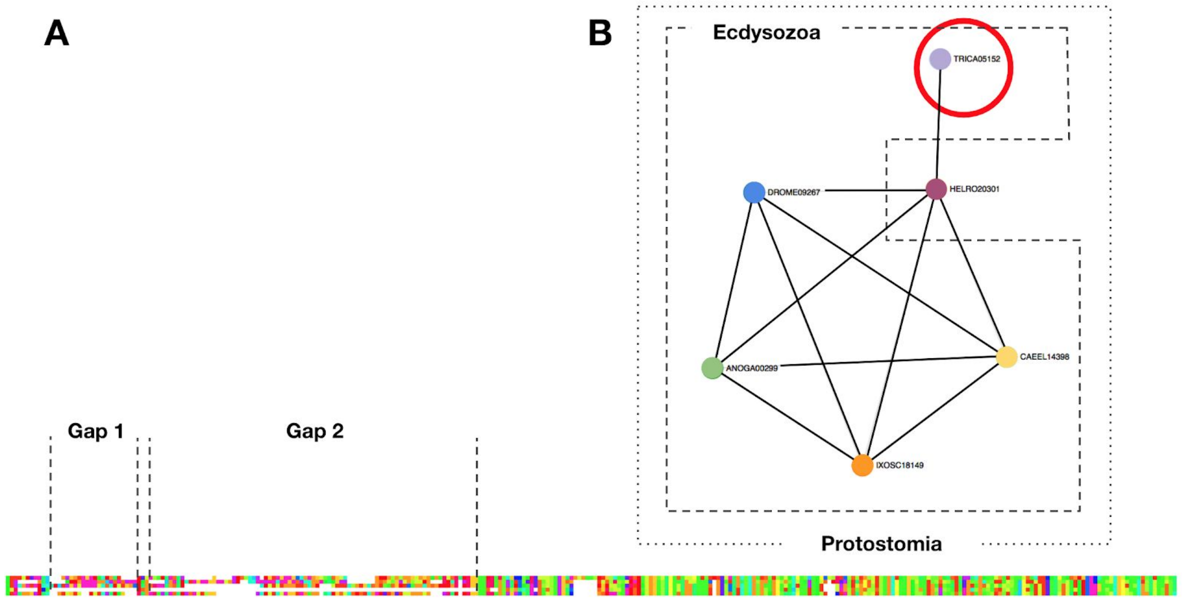


Figure 24: Orthology graph and multiple sequence alignment of the YIPF protein family at Protostomia level.

A. Multiple sequence alignment of ecdysozoan proteins. Gap1 and Gap2 denotes the gap regions specific to the *Tribolium castaneum* gene. **B.** Orthology graph at Protostomia with conflicting *Tribolium castaneum* gene circled in red.

Conclusion

To summarise, the single *Tribolium castaneum* gene is fragmented and the orthology calling failed with the nearest Ecdysozoa genes. Later during the reconstruction the orthology is made with more distance genes. The lack of orthology results in the clustering algorithm making the wrong decision to separate the single gene from the orthologous group it belongs to. Once the orthology is established again with more distant genes, the single *Tribolium castaneum* gene is wrongly split from its related orthologs.

Example HOG #2: forkhead box protein with accelerated rate of evolution

Description of reference gene family evolutionary history

The gene tree reconstructed from the forkhead protein sequences at the level of Tetrapoda shows that no duplication occurred during this evolutionary time frame for these 10 species. This tree topology supports a single gene per species for the whole clade.

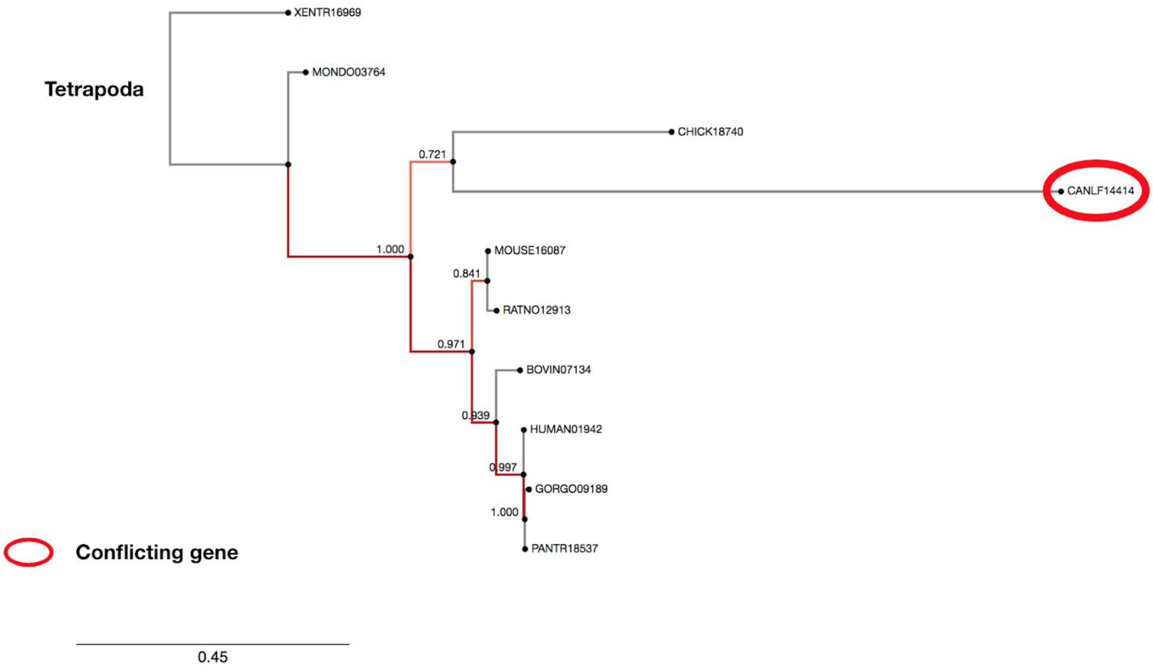


Figure 25: Gene tree of the forkhead protein family.

Characterisation of the HOG reconstruction error

However, the reconstructed HOG pictured in figure 26 is not in agreement with the evolutionary scenario shown in figure 25. We see in panels A and B of figure 26 that a duplication is inferred between the Amniota and Theria level, implying two HOGs with a suspicious complementary pattern with the single dog gene (CANLF - canis lupus familiari). This spurious duplication creates two paralogous groups where there should be only one HOG, as well as a substantial number of spurious independent gene losses. We can count using the iHam visualisation that 4 independent gene losses are required within the Theria clade to explain such clustering. The most likely scenario here is that these two ‘complementary’ Therian HOGs should be combined, removing the wrong duplication event with all the induced spurious genes losses.

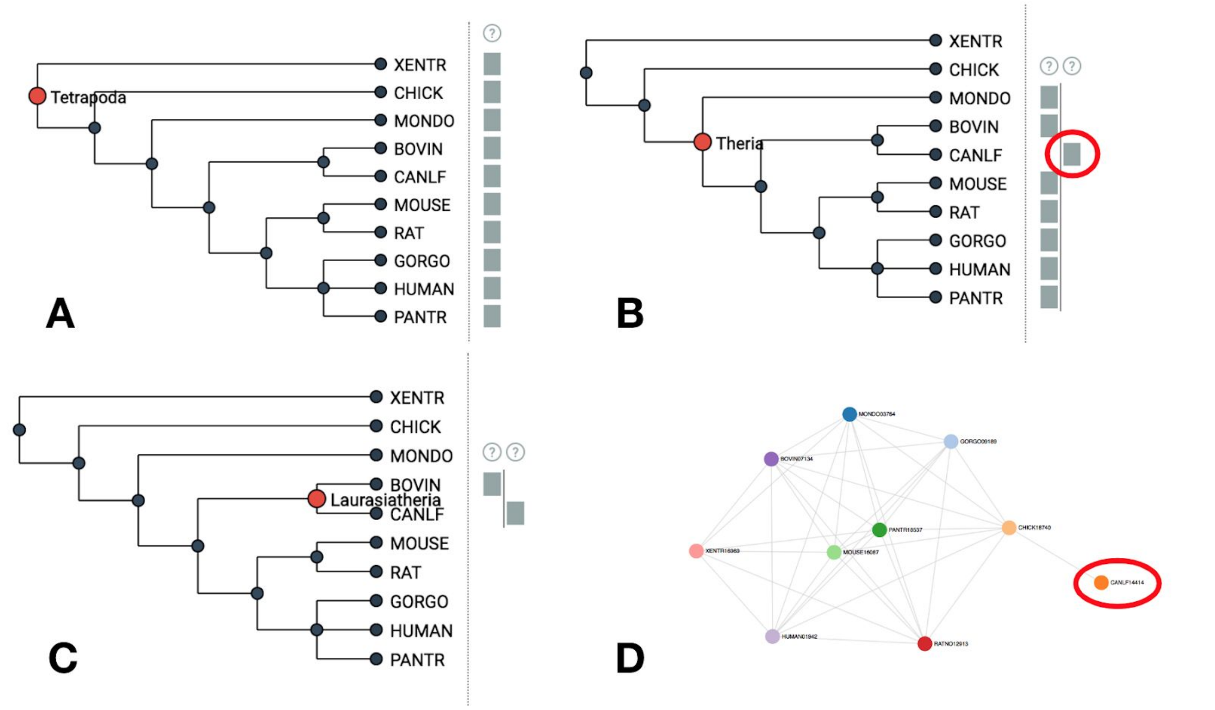


Figure 26: iHam visualisation of the HOG clustering of the forkhead protein family along with orthology graph. A,B,C panels focus the visualisation on the Tetrapoda, Theria and Laurasiatheria clades respectively. **D.** Orthology graph inferred using OMA on the reference proteome of QfO 2018. The red circles in **B** and **D** highlight the conflicting gene.

Investigation of the orthology graph

To understand why GETHOGs is wrongly clustering this simple gene family, we will look at the input orthology graph used as orthology reference. We can see in the orthology graph shown in Figure 26.D a “hairball” of genes all connected to each other, in agreement with the single gene per species topology supported by the gene tree in figure 25, showing a single dog gene (circled in red) which is only connected to the chicken gene. This lack of orthology between the dog gene and the rest of the family forces the algorithm to separate them during the Theria HOG reconstruction. Then, when orthology is finally established (here with the frog gene) they are reunited inside a paralogous group and the upper reconstruction is not affected. The problem for this case is the lack of orthology. By looking at the gene tree topology in figure 25, we see that the conflicting dog gene evolved considerably faster than the rest of the family. This acceleration in the rate of evolution makes the pairwise orthology calling fail except for the chicken, resulting in missing pairwise orthology.

Conclusion

To conclude, the isolated dog gene has evolved much more quickly than its sibling genes, resulting in the failure to call the correct orthology. Later, the orthology is made with more distant species. The lack of orthology makes the clustering algorithm take the wrong decision to cluster the single gene on its own. Once the orthology is established, the single gene is clustered in the group it should belong to as paralogous and the rest of the clustering is not affected.

Example HOG #3: gene PLAGL2 family and misplaced gene duplication

Description of reference gene family evolutionary history

The gene tree of the PLAGL2 gene family shown in figure 27 is covered by the whole Euteleostomi clade with a Homininae specific duplication. We see that Non-Hominidae species are represented by a single gene copy where two genes can be found in the Hominidae clade (except for the human gene where a gene loss seems to have occurred). We can see in the Euarchontoglires clade that the genes evolved slowly and that the branch lengths are too short to clearly distinguish the underlying topology. Figure 27.B presents the subgene tree of Euarchontoglires genes with fixed branch lengths in order to simplify the visualisation and focus only on the topology.

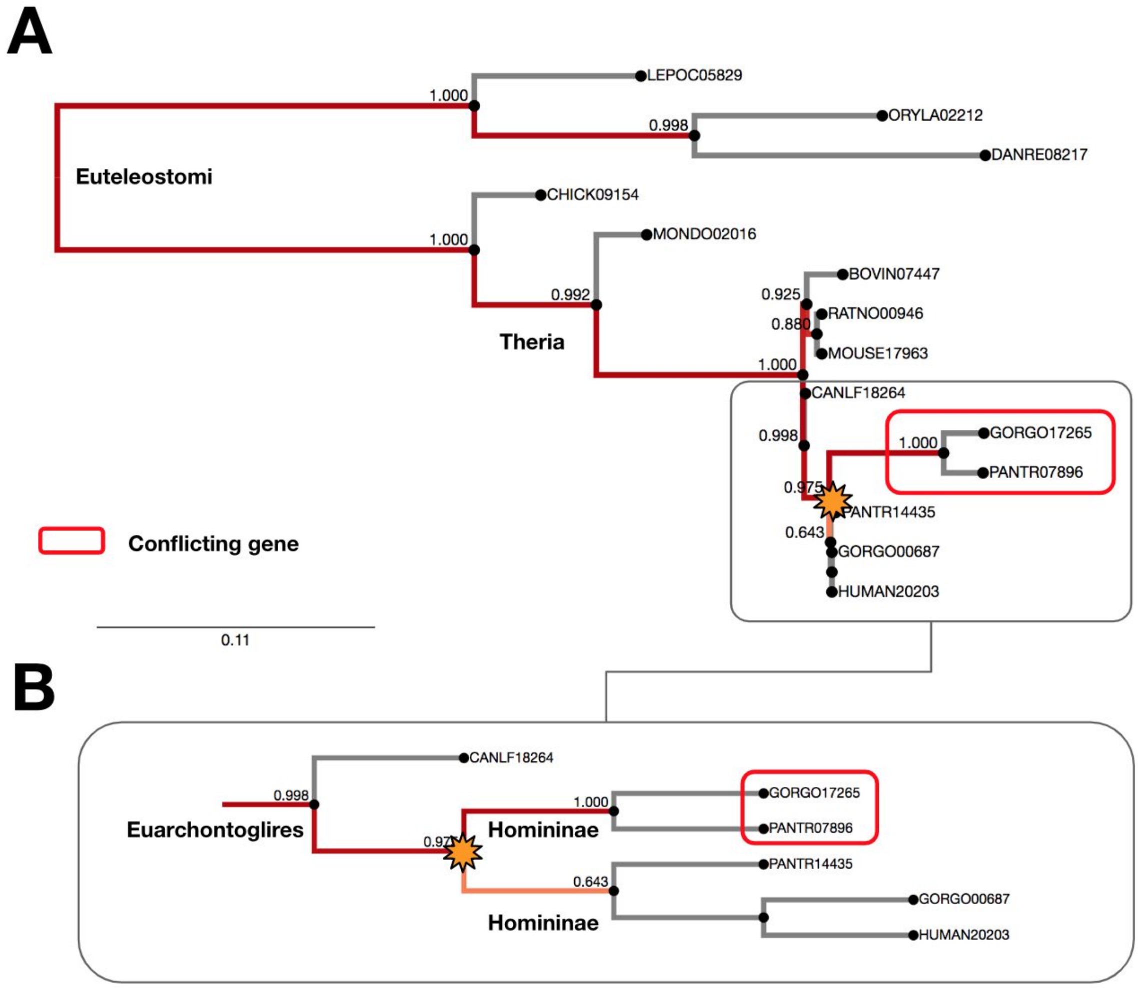


Figure 27: Gene tree of the PLAGL2 protein family.

Characterisation of the HOG reconstruction error

If we consider now the gene family reconstructed by GETHOGs in figure 28, we observe a disagreement with the evolutionary history supported by the gene tree topology (Figure 27). We see from panels D and C of figure 28 that a duplication occurs between the Amniota and Theria levels. One of the paralogous groups is fully formed and covers the whole clade, while the other is only composed of the chimpanzee and gorilla genes. This clustering contains a spurious duplication, as well as a few spurious gene losses, which are wrongly inferred. This evolutionary scenario is not likely to happen due to the high number of independent gene losses following the spurious duplication. The most likely scenario would be to shift the duplication from the Theria to the Homininae.

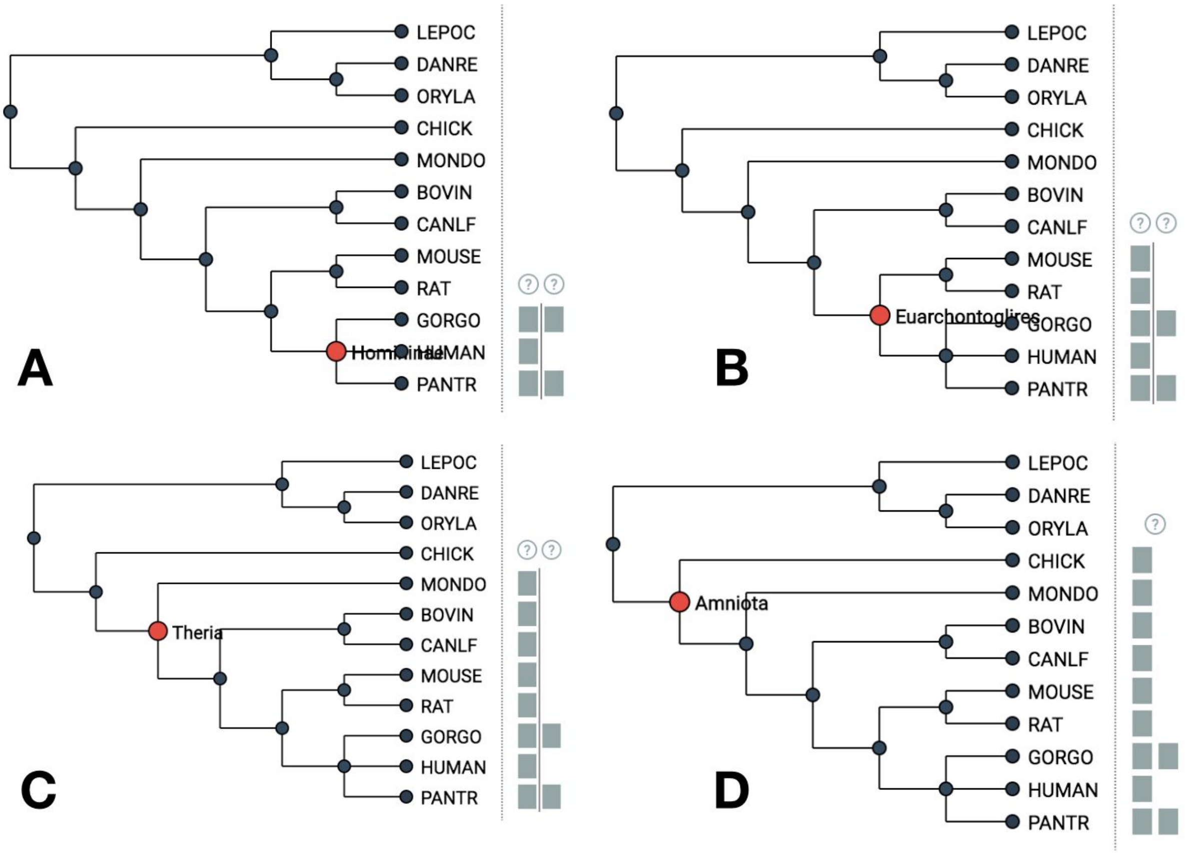


Figure 28: iHam visualisation of the HOG clustering of the PLAGL2 protein family. A,B,C,D panels focus the visualisation on the Homininae, Euarchontoglires, Theria and Amniota clade respectively.

Investigation of the orthology graph

Let us consider the orthology graph plot in figure 29.B to understand why the algorithm failed to reconstruct the correct gene family clustering. We see that the right Homininae gene cluster in the orthology graph (Figure 29.B right blue rectangle) is fully connected to the rest of the therian genes, while the left Hominiae gene cluster (Figure 29.B left blue rectangle) is not connected to it; we spot a single edge between the two clusters which is not significant compared to the 10 expected edges in a correct orthology scenario. This explains why this group of Homininae genes are isolated all the way up to their level.

If we look higher up in the tree and integrate the paraphyletic non-Amniota Euteleostomi genes (figure 29.B purple rectangle), we observe that the cluster is connected to all the rest of the graph. This explains why the duplication is positioned in the branch leading to Theria: the connection is finally made with the conflicting Homininae cluster and integrated into the HOG. Nevertheless, the only way to incorporate it is to create the spurious duplication described in detail previously.

As already pictured in case study #2, the conflicting genes are affected by an acceleration of their rate of evolution after the Homininae duplication, resulting in a failure in the pairwise orthology calling.

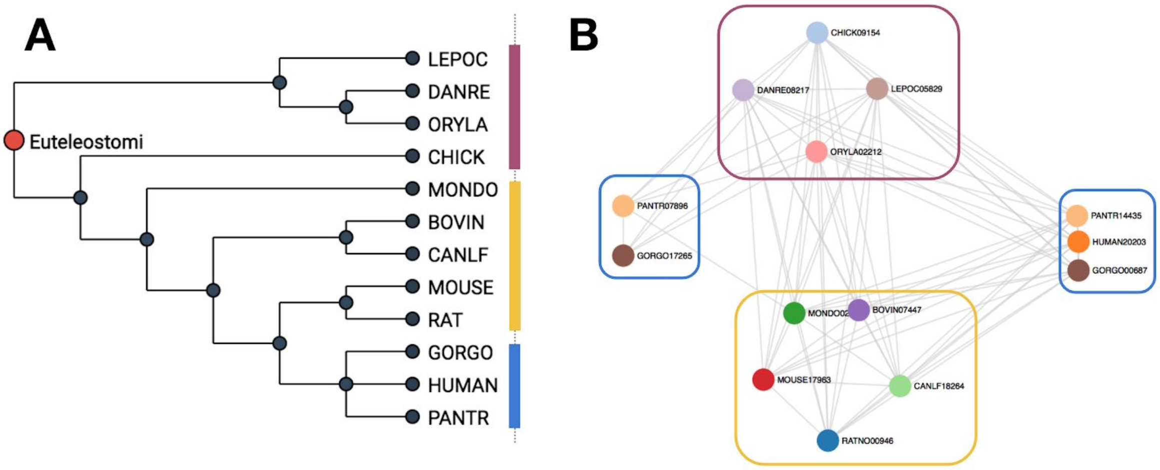


Figure 29: Orthology graph of the PLAGL2 protein family level and reference species tree.
A. Reference species tree of the QfO 2018 proteome at Euteleostomi. **B.** Orthology graph at Euteleostomi. The paraphyletic clades are highlighted and are circled genes are colored to show the correspondence.

Conclusion

To conclude, one of the two Homininae paralogous groups evolved faster than its sibling paralogs, resulting in a lack of pairwise orthology with a part of the rest of the family. Later the orthology is established with more distantly-related genes. The lack of orthology makes the clustering algorithm decide erroneously to segregate one of the paralogous Homininae groups for several taxons and wrongly places the Hominiae duplication at the Theria level.

Discussion/conclusions on the case studies

From these case studies, we observe that several recurring abnormal patterns can be observed in HOGs clustering due to errors in the orthology calling. Indeed, poor-quality genomes or complex evolutionary scenarios, such as fast evolving genes, impact the orthology inferences. In our investigation of abnormal HOG reconstruction, we catalog many cases of fragmented genes (poor genome assembly, sequencing errors, gene annotation errors) and fast evolving genes, where lack of orthology makes the reconstruction harder. In this chapter, we combine several visualisation tools (iHam, GTM, phylo.io) to carefully inspect each case from several different angles to first estimate what is the ground truth regarding the true gene family evolutionary history, and then to diagnose why the reconstruction by GETHOGs is failing. Such cases are easy to spot by using the adapted visualisation tools but require additional work on the algorithm to be overcome. Indeed the greedy nature of the algorithm cannot resolve such cases because 'locally' the decision leading to spurious clustering is the most optimal one.

To prevent such cases, we need to integrate more information than just the level-related orthology between clusters. The idea would be to make decisions on HOG delineation at each internal node of the species tree based on pairwise orthology as is currently the case, but also to integrate information about the internal structure of the subHOGs: (i) before creating a duplication that may lead to numerous gene loss events, maybe considering placing the duplication at a lower level will increase the likelihood of the family history, (ii) if two complementary HOGs (based on species coverage) are clustered as paralogous groups, maybe merging them into one single HOG is the most likely evolutionary scenario.

Discussion and implications

In this chapter, we observed that the quality of the reconstructed HOGs depends on the quality of the input orthology graph. In the first part, we see that, as the complexity of the simulated evolutionary scenario increases, the quality of pairwise orthology inferences decreases, and so does the HOGs quality. In the second part, the case studies show that abnormal HOG clustering on real data are caused by missing orthology in the orthology inferences due to genomes of poor quality or complex evolutionary scenarios. We see that the current greedy nature of the algorithm which takes decisions at each internal node by simply looking at the percentage of existing pairwise orthology between gene clusters cannot solve such cases. In order to solve such cases we also need to consider the general HOG structure and implied evolutionary scenario, in terms of gene losses and duplications, in the decision process. Indeed, we see in our case studies that the problems are mainly caused by 'local' abnormalities in the orthology graph which are then diluted when including more distantly related genes. The idea would be to rely on this distant information when it is integrated in the reconstruction process to go back to conflicting parts and solve then with a more comprehensive view of the problem.

Chapter 5: Heuristics to overcome split HOGs (and the limits of the generalised species discordance test)

In the previous chapter, we illustrated the limitations of the GETHOGs algorithm performance with a catalog of case studies for different types of reconstruction errors. The algorithm faces difficulties to correctly infer HOGs when input orthology data is incomplete. These pairwise orthologous relations missed by the orthology inference methods are due mainly to either sequence-centric errors (sequencing, assembling errors) or the evolutionary complexity of the gene families (fast evolving genes). These missing orthologous relations result in a 'split-hog' pattern, where parts of the gene family are wrongly inferred as paralogous due to the lack of orthology. Nevertheless, the current algorithm cannot resolve such cases because its greedy nature only considers a local optimisation at each level, restraining the information scope to the related level-wise information (amount of pairwise orthology between two sets of genes). Indeed, the algorithm may infer wrong paralogous groups in some part of the family where pairwise orthology is missing, as shown in chapter 4, but when more distant species are introduced the orthology calling is performed correctly and the algorithm continues the clustering normally.

In order to overcome these problems, the idea would be to target such clustering patterns in a first step, and then to apply a post fix strategy to find a better solution. We observe two types of abnormal clustering: the 'complementary' pattern, where there is no species overlap between paralogous groups, and the 'unmerge' pattern when a HOG is not merged for several consecutive levels, after which it is finally merged, implying a much deeper duplication than in reality. In this chapter we present and test two variants of the GETHOGs algorithm to solve these two types of conflicting HOGs. The main idea behind the two refinements is that resolving the conflicting clustering is impossible at the levels where missing orthology are reported, but when integrating more information at higher levels, a solution can be found to restructure the erroneous HOG clustering.

In order to evaluate the performance of the two different GETHOGs variants, we used a set of custom tests and the orthology benchmark service.

Methods

Heuristic #1: Complementarity

The first GETHOGs variant aims to resolve HOGs with an abnormal clustering pattern denoted as complementary and shown in chapter 4 with case studies 1 and 2. In this section, we will first characterise what is the complementary pattern and then propose an algorithmic solution to resolve such HOG clustering.

Characterisation of complementary HOGs

In order to better understand what is the complementary hogs pattern, let us consider the iHam visualisation for the YIPF gene family of figure 23.D. Graphically, we can see that each species (row in the matrix) is only represented in one and only one paralogous group (column). It means that the two paralogous groups have a species coverage that are not overlapping. In other words, the intersection of species sets represented in each paralogous group is empty. Such cases may not be the most likely to happen phylogenetically, because it implies a lot of independent gene losses and a spurious duplication, which can be evolutionarily less likely. The most probable scenario in these cases is to have only one orthologous group composed with genes from the two complementary hogs.

Detection and resolution of complementary HOGs

The refinement of the GETHOGs algorithm is composed of two steps: identifying clusters of complementary paralogous HOGs in connected component formation (see Chapter 2 for connected components description) and resolving these clusters by finding the optimal new combination of complementary paralogs.

- **Identification of complementary paralogous groups:** The goal of this step is to identify paralogous groups with complementary patterns during the HOGs reconstruction. As presented in Chapter 2, GETHOGs extracts connected components at each taxonomic range from the related sub-orthology graph. Then, it clusters together the paralogous groups (all members that belong to the same lower level genome will be part of a same paralogous groups and will initiate from the same duplication event) and combines all connected components into a single HOG. The search of complementary HOGs (and their potential resolution) is performed on connected components before the paralogy clustering is performed in order to fix potential spurious paralogous groups (figure 30.D). The complementary criterion is attributed to all pairs of paralogous groups within a connected component, where no overlap in terms of species coverage is observed. It results in clusters of paralogous groups connected by their complementary relations. Nevertheless, complementarity is not transitive inside these clusters and many scenarios of paralogous 'combination' may be possible. In order to have the optimal scenario(s), each possible scenario is considered and scored according to its phylogenetic likeliness.

- **Resolving each paralogous cluster:** The second step takes each cluster of paralogous groups, connected through their complementary relations, and generates for each one all possible scenarios of paralogous combinations. To proceed, GETHOGs will generate all possible combinations by computing all partitions of paralogs in the cluster (Figure 30.E). The following partitioning criteria are mandatory: a partition can contain one or more paralogous groups, a paralogous group can stay alone in a partition or can be added to a partition if, and only if, it is complementary to all the other partition members. Once the partitioning is carried out, each of the partitions is scored according to a scoring function. The scoring function aims to capture the most likely phylogenetic scenario, where the number of gene duplications and gene losses are minimized. For each partition, a score is attributed based on the number of gene losses and duplications induced by the new combination of paralogs. Finally, we select the best combination of paralogs out of all the potential scenarios. To proceed, we have two variants for the scenario selection method for this heuristic #1:

- The 'safe' variants will be limited to select a scenario without introducing any stochasticity. This means that only the partitions that contain one, and only one, scenario with the lowest score are going to be selected. Indeed, when only one best scenario is present the algorithm will always produce the same result out of two runs and hence not be considered as stochastic.
- The 'ambitious' variant aims to resolve more complementary cases by including during its scenario selection method the partition with co-optimal scenarios. The choice between the co-optimal scenarios is made randomly.

This refinement aims to resolve only the case of total complementarity between paralogous groups. The only possible operation is the combination of two HOGs where no species overlap is observed. A whole set of paralogous groups can be converted into a single HOG composed of all the paralogs, thereby removing the duplication that initiated them. Furthermore, no additional duplication is introduced in this process.

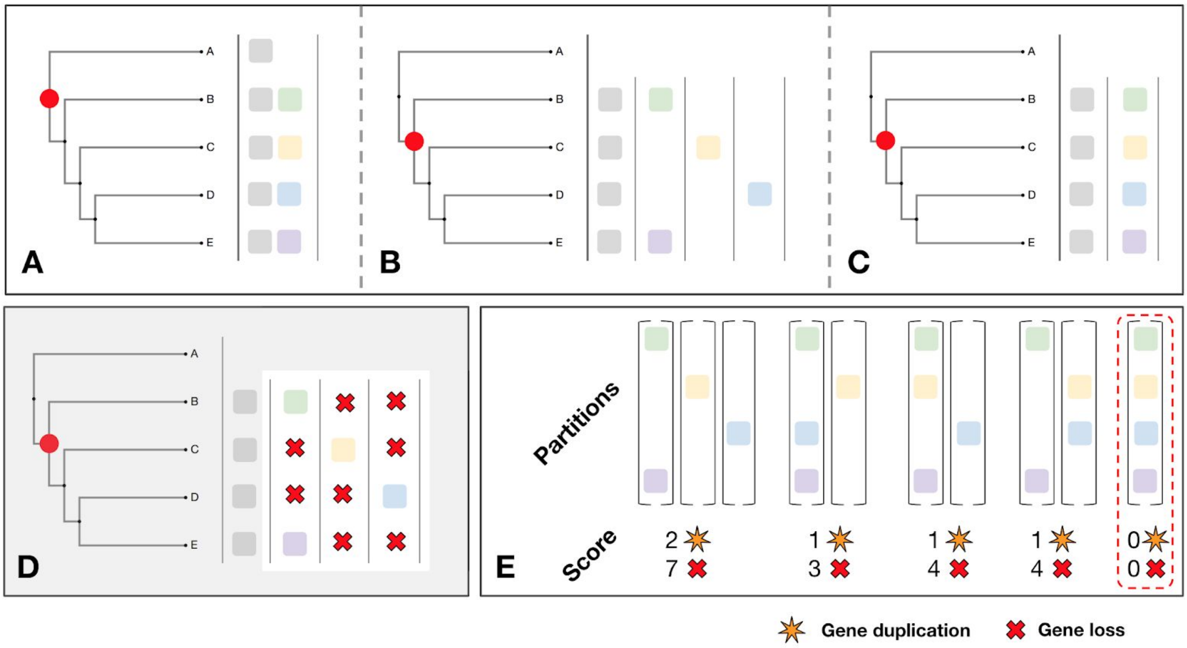


Figure 30: Workflow of the complementary refinement on a toy example.
A. iHam visualisation of the example HOG at root level.
B. iHam visualisation opened at level of interest with complementary paralogs.
C. iHam visualisation opened at same level as in (B) with resolved complementary HOGs.
D. Identification of the complementary paralogs.
E. All possible complementary partitions with their associated number of gene duplications and losses. The highlighted partition is the optimal combination scenario.

Heuristic #2: Unmerged

The second GETHOGs variant aims to resolve HOGs with an abnormal clustering pattern denoted as unmerged as illustrated in Chapter 4 with the case study #3. In this section, we will first characterise what are the ‘unmerged’ HOGs and then propose an algorithmic solution to resolve this type of HOG clustering problem.

Characterisation of unmerged HOGs

The second GETHOGs variant aims to resolve HOGs, denoted as ‘unmerged’ HOGs, that are not merged for several consecutive levels due to a lack of orthology relation with any other HOGs until one orthologous counterpart HOG is found and they re-enter in the HOG merging process. Two types of scenarios are then possible: (i) the ‘unmerged’ HOG is clustered with another HOG but does not end up in a paralogous group or (ii) it goes into a paralogous group. In case (i), the evolutionary history implied by the HOG clustering can be explained by several consecutive independent gene losses for each unmerged level. In case (ii), the evolutionary history implied by the HOG clustering is a scenario that is less likely than case (i). The evolutionary history of (ii) would be explained by a gene duplication with one of its paralogous groups followed by several consecutive independent gene losses for each of its unmerged levels. The case study #3 on PLAGL2 gene family (Chapter 4) shows a typical case (ii) example, where a duplication is placed too high in the gene tree compared to where it should be due to missing orthology in a lower level, leading to spurious genes losses in one of its paralogous group and a misplaced duplications event.

Detect and resolve unmerged HOGs

The refinement of the GETHOGs algorithm is composed of two steps: identifying unmerged HOGs in paralogous clusters inside connected components (see Chapter 2) and resolving these conflicting paralogous clusters by finding the optimal scenario where the number of spurious gene duplications and gene losses are minimized.

1. **Identification of unmerged paralogous groups:** The goal of this step is to identify, in a paralogous group, the HOGs that have not been merged for one or several levels. To proceed, the algorithm keeps track of the number of consecutive levels where a HOG fails to be merged with an orthologous counterpart HOG (Figure 31.A). This will inform us that maybe the duplication occurred in lower levels, but a lack of orthology failed to correctly cluster them. For a single paralogous group, several HOGs can be marked as unmerged. The rest of the algorithm now needs to decide how to resolve these cases.

- **Resolving each paralogous cluster with unmerge HOGs:** The second step takes each paralogous group with unmerged HOGs and tries to find an optimal scenario where a maximum number of gene losses (unmerged levels) are removed and where duplications are the most correctly located (Figure 31). The idea here is to restructure unmerged HOGs by removing all spurious, unmerged levels and drag down the duplication to the level where the unmerge process initiates. All unmerged HOGs can be potentially inserted in a HOG fully composed by reducing the number of paralogous members of the paralogous groups. To some extent, the paralogous group itself can be removed if only one HOG without the unmerged pattern is observed and all the other unmerged HOGs are incorporated into it. To proceed, the algorithm will generate all possible partitions of the paralogous group members where the following condition is mandatory: one partition may contain one and only one normal HOG without a limit to the number of unmerged HOGs. Since each partition will correspond to a single paralog in the final paralogous group, we cannot put two normal HOGs together. For each partitioning scenario, the number of implied gene losses and gene duplications is calculated. In order to select the best scenario, we apply the strategy described in the next paragraph. Similarly to the heuristic variant #1, we have two strategies here:
    - The “safe” selection that only works if one, and only one, scenario is the best scenario score-wise.
    - The “ambitious” scenario that selects one scenario out of all the co-optimal scenarios.

  Once a scenario is selected, the algorithm proceeds with the restructuring according to the partitioning scenario by combining each HOG of a partition into a single HOG. The HOG combination is made by first pruning all the unmerged levels in each unmerged HOG. Then, these pruned HOGs are incorporated at the corresponding level into the unmerged HOG.

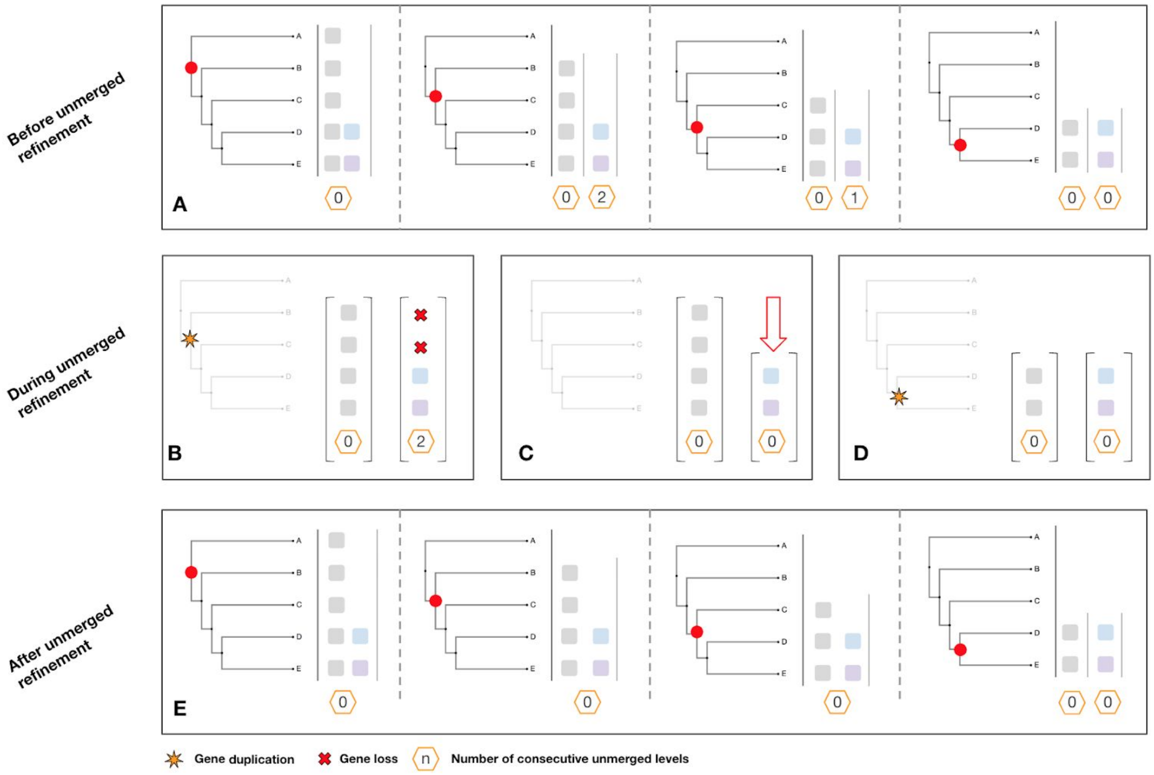

Before unmerged refinement
During unmerged refinement
After unmerged refinement
A
B
C
D
E
Gene duplication
Gene loss
n
Number of consecutive unmerged levels

Figure 31: Workflow of the unmerged refinement on a toy example.
A. iHam visualisation of the example HOG opened at several levels.
B. Identification of the unmerged HOG in the paralogous group (here with two consecutive levels)
C. Trimming of the conflicting unmerged HOG to remove all unmerge levels
D. Shifting of the duplication at the starting level of unmerged process.
E. iHam visualisation of the HOG example once resolve by the unmerged refinements.

This refinement aims to resolve cases where a paralogous group member has not been merged after the related duplication event for at least one level. In such cases, the algorithm will try to remove as much as possible the unmerged level in this unmerged HOG and plug it into another HOG at a level that minimises the total number of gene duplication and gene losses. Creating/moving duplications is tolerated in this refinement.

Benchmarking using Quest for Orthologs 2018 dataset and generalised species discordance test

In order to investigate the impact of the two heuristic refinements of the GETHOGs algorithm on the HOG inferences we used the Quest for Orthologs Proteomes of 2018 as reference datasets to perform our quality control analysis. This dataset is composed of 78 complete proteomes from species gathered from various public databases (UniProtKB, Ensembl and Ensembl Genomes). The proteomes dataset is provided with its associated, manually curated reference species tree. The pairwise orthology inference was performed by OMA Standalone version 2.3.1. The inferred pairwise orthologs and the references species tree were provided as input to GETHOGS to perform three types of inferences, depending on the algorithm version: (i) the "default HOG inferences" produced using the normal GETHOGs algorithm, (ii) the "complementary HOG inferences" produced by using the GETHOGs algorithm with the complementary fix variant (Heuristic #1), (iii) the "unmerged HOG inferences" produced by using the GETHOGs algorithm with the unmerged fix variant (Heuristic #2).

We benchmark the performance of the two refinements with a two-fold strategy on real data: (i) first we investigate how the conflicting cases in the default inferences are treated in each of the two heuristic variants, and then (ii) we use the orthology benchmark service to assess the quality of our two refinements, as described in Chapter 2.

In order to investigate how the problematic cases are dealt with, we apply the following strategy for each heuristic refinement to estimate the fraction of resolved cases and characterise unsolved HOGs. First, we fetch all conflicting HOGs in the default HOGs dataset. Depending on the heuristic method benchmarked, the selection criterion is not the same. For the complementary refinement, the HOGs of interest have at least one duplication with two complementary paralogous members inside. For the unmerged refinements, the HOGs of interest have at least one paralogous group not merged in the level above its duplication. Second, we estimate the total fraction of spurious HOGs for both refinements, and then estimate the fraction of the affected HOGs that are resolved by the refinement. We show a list of refined HOGs to illustrate the different kinds of resolution obtained by the algorithm variant. Third, we catalog the different major types of unresolved HOGs and explain why we can not deal with them.

The second part of the strategy is to use the orthology benchmark service proposed by the Quest for Orthologs Consortium to benchmark the quality of the refinements, taking into account different aspects. As in Chapter 2, we focus on the generalised discordance species tree to estimate the recall and precision of orthology clustering through the related inferred pairwise orthology.

Results

Heuristic #1: Complementarity

For this algorithmic variant, we count 3747 HOGs [6.16%] that have triggered the complementary criterion out of the 60870 HOGs in the default dataset. However, these 6% of complementary HOGs encompass 31.6% of the total number of inputted proteins of the proteomes dataset. If we now consider the HOGs inferred using the heuristic variant #1 to resolve complementarity, we observe only 1162 HOGs [1.9%] containing 18.95% of the total number of proteins for the 'safe' variant and 304 additional HOGs [0.5%] containing an additional 11.45% of the total number of proteins for the 'ambitious' variant. The percentage of proteins contained in HOGs affected by one or several complementary patterns only reflects the size of the whole families, and not the precise number of proteins inside these genes families that are actually involved in abnormal clustering. The safe variant resolved 2585 complementary HOGs where one, and only one, optimal solution was present. In addition to these 2585 resolved cases, the ambitious variant resolved a further 858 complementary HOGs where more than one optimal solution is observed.

For the safe variant, we obtain a large reduction in the number of complementary HOGs, from 6% of the total number of HOGs to only <2% remaining. The proportion of proteins contained in these HOGs also drops from 32% to 19% of the total proteins in the dataset. We present in figure 32 a few cases, which show the same complementary pattern as presented in the cases studies of Chapter 4, and which are correctly resolved. These 3 cases are representing the panel of different HOGs present in the 2585 resolved complementary HOGs by the safe variant. The heuristic #1 safe variant shows improvement regarding complementary cases and provides a clear solution.

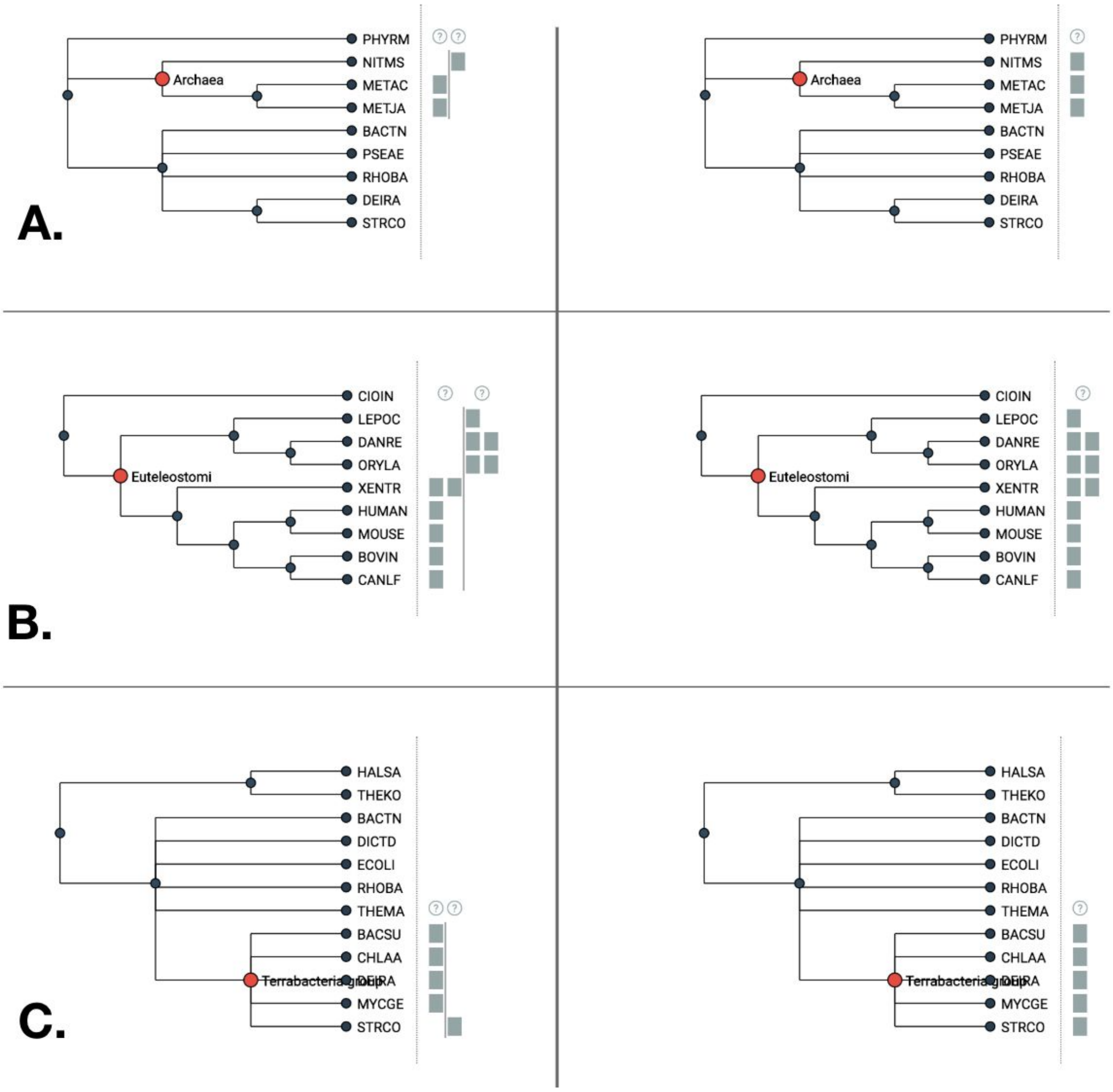


Figure 32: Complementary HOGs resolved by the GETHOGs algorithm using the heuristic 'complementary' variant in safe mode. For each case (A,B,C), the inferences using the default GETHOGs variant (left) and the corresponding inferences using the heuristic variant (right) are shown.

For the ambitious variant, the number of complementary HOGs is further reduced from 2% of the total number of HOGs in the safe variant to only 0.5%. The number of proteins contained in these HOGs also drop from 19% to 11.5% of the total proteins in the dataset. We present in figure 33 a few cases that are treated by the ambitious variant. These 3 cases are a selection representing the panel of different HOGs present in the 858 resolved complementary HOGs by the ambitious variant. We observed that the changes made during the clustering by the heuristic #1 ambitious variant are globally improving the quality of HOGs by reducing the number of misplaced duplications and spurious gene losses. Nevertheless, we observed that around 900 HOGs have a co-optimal solution meaning that we are inferring at minima 450 HOGs with a wrong restructuring. Indeed, in case there are 2 co-optimal scenarios we randomly pick one meaning that half of our changes will be spurious. Since there could have more than 2 co-optimal scenario, this number of 450 HOGs may be even bigger. This can be explained by the stochastic nature of the tie-breaking approach (see Methods). Still, we did not expect such large variation, and consider that the approach fails to meet the 'robustness' requirements we have established for GETHOGs. The stochasticity observed does not encourage us to use the ambitious variant to improves the HOGs clustering.

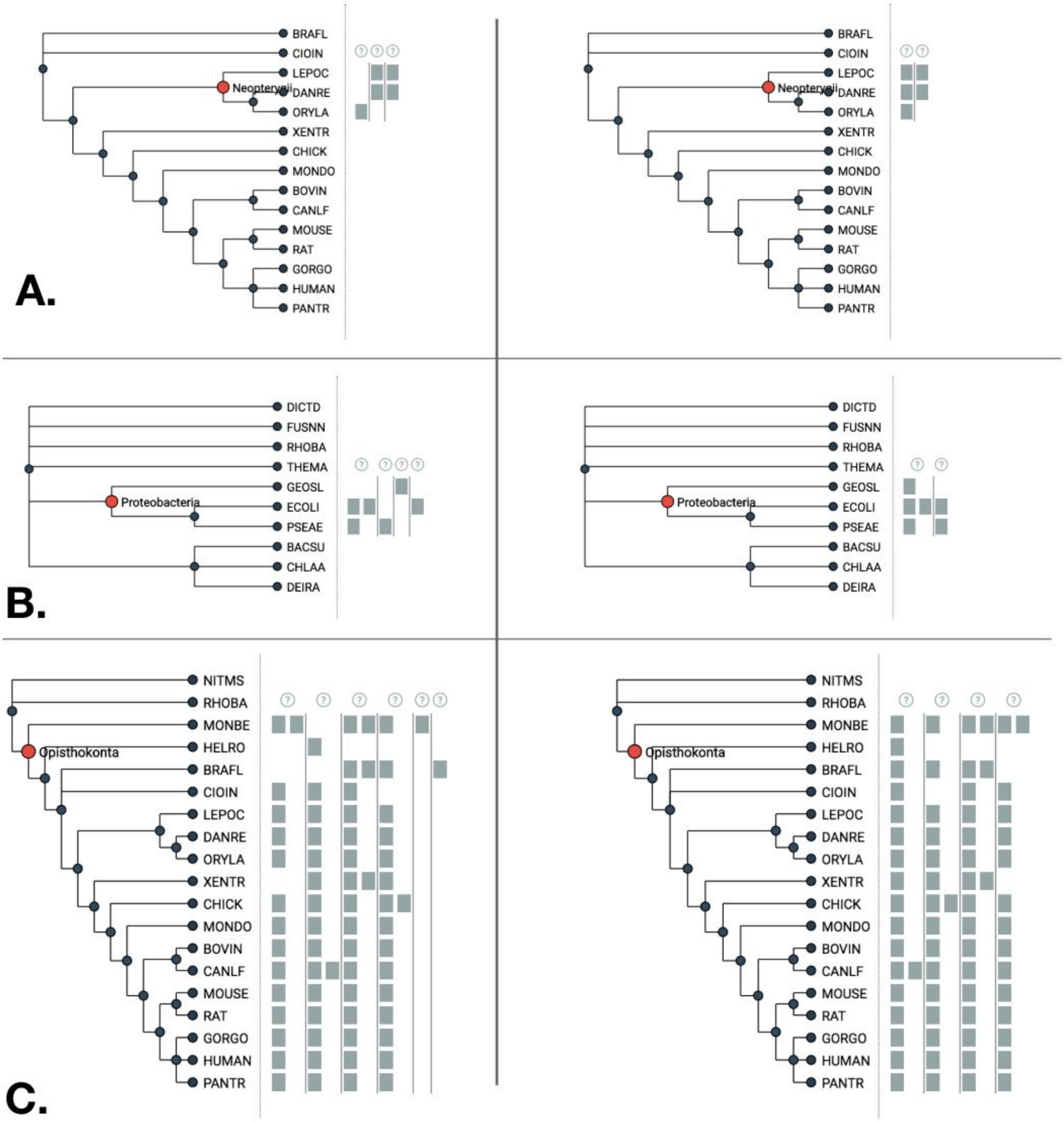


Figure 33: Complementary HOGs restructured by the GETHOGs algorithm using the heuristic 'complementary' variant in ambitious mode. For each case (A,B,C), the inferences using the default GETHOGs variant (left) and the corresponding inference using this heuristic variant (right) are shown. The restructuring is performed by changing randomly one of the co-optimal scenarios which is not guaranteed to be the most phylogenetically correct one.

Finally, we observed that around 300 HOGs representing roughly 0.5% of the HOGs are not treated by the safe nor the ambitious variant. This is due to the intrinsic limitation of the heuristic #1 variant. In order to conserve good scalability, the partitioning is limited to paralogous groups with fewer than 10 members. Indeed, the partitioning time grows exponentially with the size of paralogous groups. Furthermore, a few families are too complex to be resolved by the current implementation of the algorithmic variant and fall into edge cases that are not possible to solve.

Heuristic #2: Unmerged

For the unmerged algorithmic variant, we count 12,578 HOGs [20.66%] that have triggered the unmerged criterion out of the 60,870 HOGs in the default dataset. These 20.66% of unmerged HOGs represent 61.44% of the total number of proteins of the proteomes dataset. If now we look at the HOGs inferred using the heuristic variant #1 to resolve unmerged cases, we now count for the 'safe' strategy 1859 HOGs [3.05%] containing 17.6% of the total number of proteins and for the 'ambitious' strategy 813 additional HOGs [1.34%] containing an additional 11.55% of the total number of proteins. Recall that the percentage of proteins contained in HOGs shown to be affected by one or several unmerged sub-HOG only reflect the size of the whole families and not the precise subset of proteins inside these gene families that are involved in abnormal clustering. We count that the safe variant resolved 10,719 unmerged cases where one, and only one, optimal solution was present. In addition, the ambitious variant resolved a further 1046 unmerged HOGs where more than one optimal solution is observed.

For the safe variant, we obtain a consequent reduction in the number of unmerged HOGs from 20% of the total number of HOGs to only 6%. The number of proteins contained in these HOGs also drops from 61% to 18% of the total proteins in the dataset. We present in figure 34 a few cases that are correctly resolved, similar to the case studies of Chapter 4 that introduce the unmerged HOG problem. These 3 cases are an excerpt representing the panel of different HOGs present in the 10,719 resolved complementary HOGs obtained with the safe variant. The heuristic #2 safe variant shows improvement regarding unmerged cases with clear solutions.

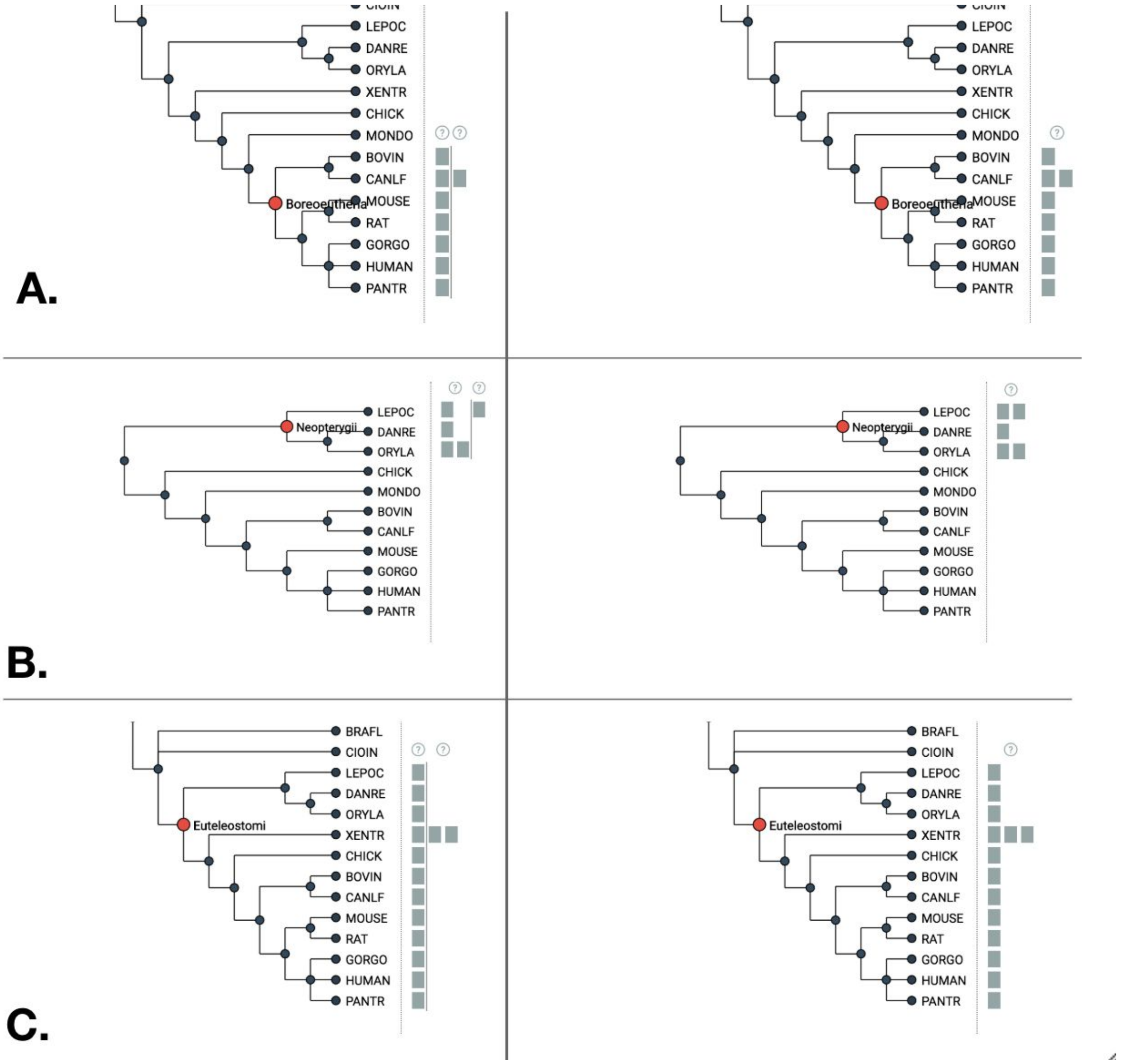


Figure 34: Unmerged HOGs resolved by the GETHOGs algorithm using the heuristic unmerged variant in safe mode. For each case (A,B,C), the inferences using the default GETHOGs variant (left) and the corresponding inference using the heuristic variant (right) are shown.

For the ambitious variant, we obtain an additional reduction of the number of unmerged HOGs from 3% of the total number of HOGs in the safe variant to only 1.34%. The number of proteins contained in these HOGs also drops from 18% to 11.5% of the total proteins in the dataset. We present in figure 35 a few cases that are treated by the ambitious variant. These 3 cases are an excerpt representing the panel of different HOGs present in the 1046 resolved unmerged HOGs by the ambitious variant. We observed that the changes made during the clustering by the heuristic #2 ambitious variant globally improve the quality of HOGs by reducing the number of misplaced duplications and spurious gene losses. Nevertheless, running twice the algorithm does not guarantee to observed twice the same HOG reclustering. Indeed, like the heuristic variant #1 since only one co-optimal solution is chosen among multiple possible solutions we introduce stochasticity to the HOGs resolution. With only pairwise orthology as the clustering signal, which in these unmerged cases is incomplete, we cannot decide in all these co-optimal scenarios which is the correct one. Again, these results are thus not in agreement with our 'robust' quality policy. Even though the result seems better in general, the stochasticity observed does not encourage us to use the ambitious variant to improve the HOGs clustering.

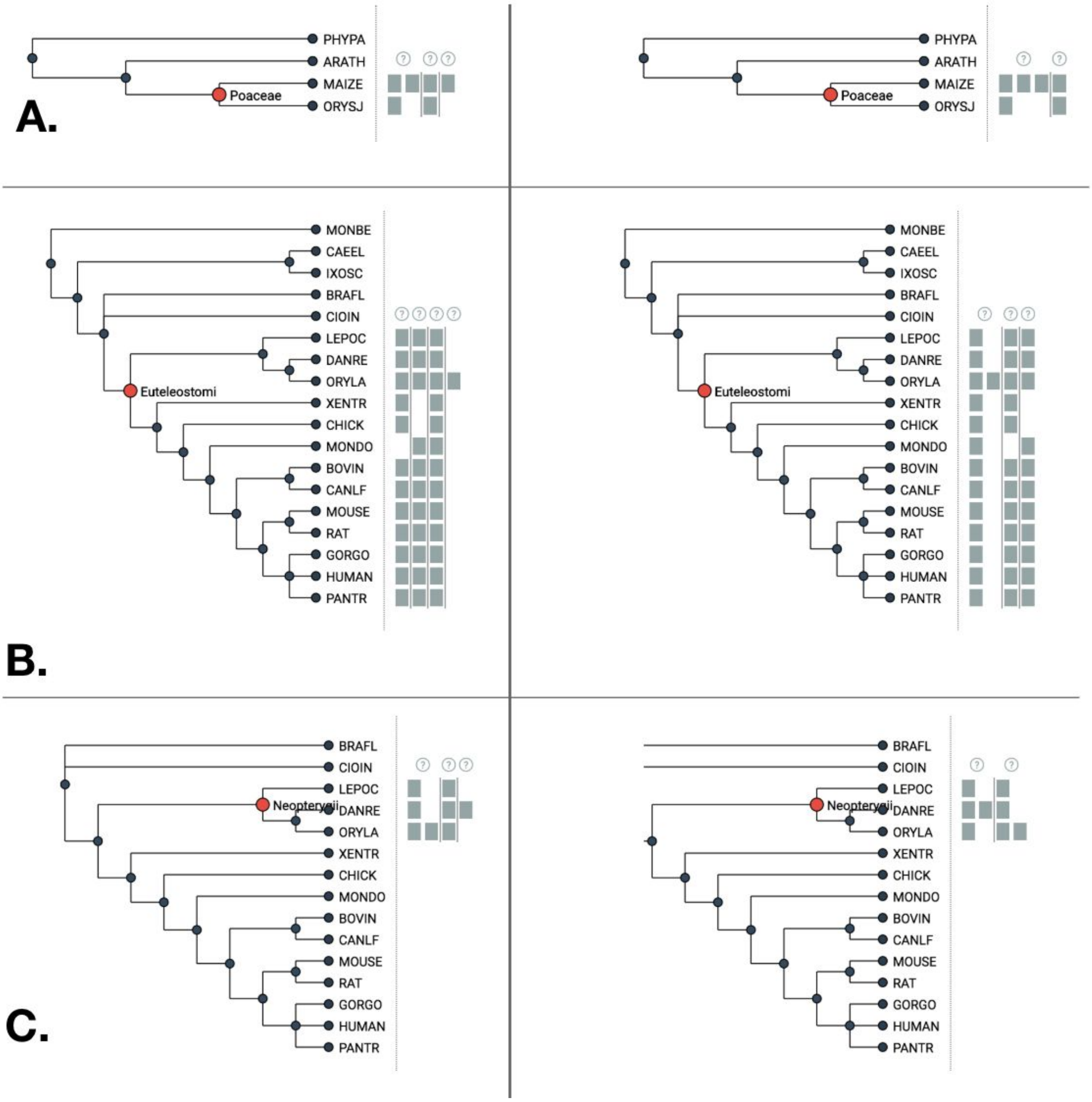


Figure 35: Unmerged HOGs resolved by the GETHOGs algorithm using the heuristic 'Unmerged' variant in ambitious mode. For each case (A,B,C), the inferences using the default GETHOGs variant (left) and the corresponding inference using this heuristic variant (right) are shown. The restructuring is performed by changing randomly one of the co-optimal scenarios which is not guaranteed to be the most phylogenetically correct one.

Finally, we observed that 813 HOGs representing roughly 1.33% of the HOGs are not treated by the safe nor the ambitious variant. This is due to the intrinsic limitation of the heuristic #2 variant. In order to conserve good scalability, the partitioning is limited to paralogous groups with fewer than 10 members (as in heuristic #1). Indeed, the partitioning time grows exponentially with the size of paralogous groups. In addition, a few families are either too big or too complex to be resolved by the current implementation of the algorithmic variant and fall into unresolvable edge cases.

Orthology benchmark service

In order to assess the impact of our refinements on the orthology clustering, we use the orthology benchmark service, as in Chapter 2, to evaluate the quality of the pairwise orthology induced by the HOGs. Using the orthology benchmark web service, we upload the HOGs inferred using (i) the default GETHOGs, (ii) the GETHOGs variant with complementarity refinement in safe mode, (iii) the GETHOGs variant with complementarity refinement in ambitious mode, (iv) the GETHOGs variant with unmerged refinement in safe mode and (v) the GETHOGs variant with unmerged refinement in ambitious mode. We mainly focus, as discussed in Chapter 2, on the generalised species tree discordance test to evaluate the recall and the precision of our uploaded pairwise orthology and, by reflection, our orthology clustering. Results are summarised in figure 36.

If we first look at the difference observed for each variant between its safe mode and its ambitious mode, we can see that no difference is observed in the recall while the precision of the inference may differ. Indeed, the fraction of complete trees sampled is similar between safe and ambitious variant for both heuristic refinements, whereas the average Robinson Foulds distance is decreased in the safe mode variant.

If we now look at the difference between the two heuristic variants without considering the safe/ambitious effects, we see that the two refinements increase the number of complete trees sampled (recall) but lower the quality of the reconstructed gene trees (precision). Scrutinizing the results carefully, we see that this trend is proportional to the impact of the refinements on orthology clustering. Indeed, the larger the refinement effect, the greater this trend will be. The unmerge process has more impact on orthology clustering—the refinement affected 20% of the total HOGs which represent 50% of the whole protein dataset compared to 12% of HOGs representing 30% for the complementary variant. The unmerge process has the biggest recall but seemingly the worse precision among the three GETHOGs inferences.

This can be explained by the following properties of the refinements and a shortcoming of the generalized species tree discordance test which I identified in the course of this work. To recapitulate, the generalized species tree discordance test samples gene families by randomly selecting a gene and then looking for its ortholog in the sister branch. A valid gene family sample should have all the species represented. The recall reflects how many full gene families are sampled, i.e. how many families are complete and well formed. The second part reconstructs from these sampled genes related gene trees and calculates the Robinson Foulds distance with the reference QfO species tree. Since orthologs arise from speciation, the topology of the reconstructed gene tree should be in agreement with the species tree. The precision is calculated on the average error between reconstructed gene trees and reference phylogeny. Regarding the two GETHOGs variants, they target 'abnormal' genes that have either a malformed sequence (fragmentation e.g. Chapter 4 case studies) or a complex evolutionary history (fast evolving gene e.g. Chapter 4 cases studies). Then, it re-clusters with their related orthologous genes these conflicting genes that were out-clustered due to missing orthology.

All of this can explain why the results shown in the benchmark are not in agreement with the quality improvement in the HOGs shown previously. In order to understand what happened we need to decompose the problem into two parts:

- **Recall and complete gene tree sampling**: we observe that the more the refinement processes conflicting HOGs, the greater the recall. This is explained by the fact that the refined GETHOGs variant incorporates conflicting genes inside the ortholog groups they belong to so the chance of being able to completely sample the gene family is increased. Indeed, the less the HOGs are sparse (gene losses) the greater the chance to find orthology.
- **Precision and gene tree quality**: Nevertheless, these conflicting genes may carry conflicting orthology signal that make the orthology calling fail and, consequently, can make the tree building complicated. By incorporating them into the sampled gene family, we add conflicting signal that alters the quality of the reconstructed gene tree and, consequently, the average RF distance between gene trees and the reference phylogeny.

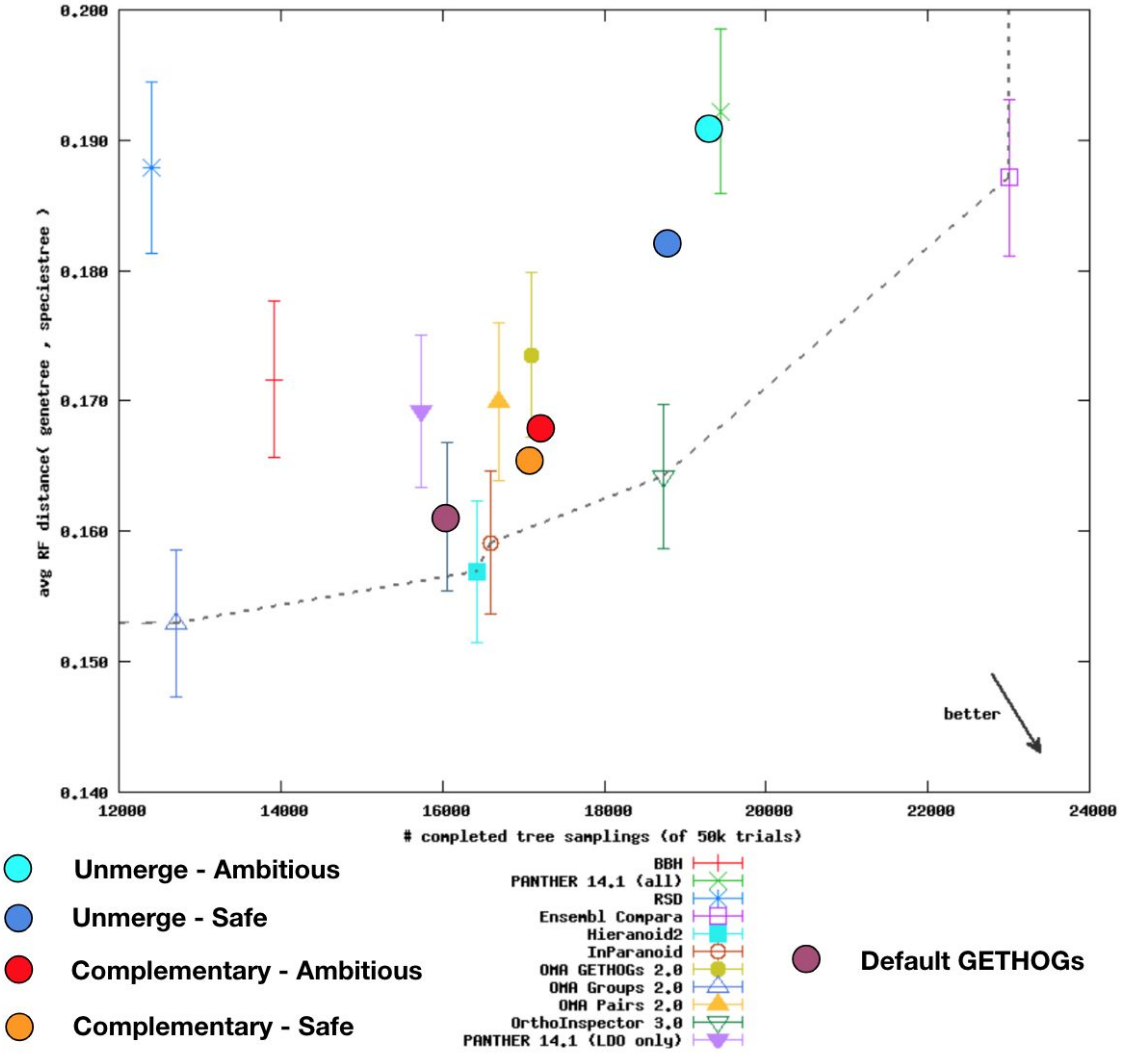


Figure 36: Effect of the two refinements on pairwise orthology relationships in the generalized species tree discordance test at vertebrate level.

To test our hypothesis on benchmark limitations, I assessed the impact of these conflicting genes on the generalized species tree discordance test results using simulated genomes. In order to proceed, I first simulated using AFL an evolutionary scenario with fast evolving genes (default globin family simulation with default genomic events “max. 5 genes, translocation, fusion/fission, rate changes”). Then using the true gene trees provided by ALF, I built two datasets of pairwise orthologs: (i) a first one containing all the orthology induced by the labelled gene trees (5350140 pairwise orthologous relations) and (ii) a second one where orthology was inferred from the labelled gene trees where branches with observed acceleration of the rate of evolution were removed (1821353 pairwise orthologous relations). To proceed, we removed all branches after duplication where the mean distance between the duplication node and their leaves were 1.5 greater than their related sister duplication branches. Then, I ran a custom instance of the Orthology Benchmark Service to evaluate the performance of the two ortholog datasets on the species tree discordance test. As shown in Figure 37, we observed that the ortholog dataset with perfect orthology induced from true gene trees have a better recall (number of complete sampled gene trees) than the ortholog dataset with fast evolving genes discarded with around 37 000 against 27 000 completely sampled trees respectively. But it has a worse precision in the benchmark (average Robinson-Foulds distance) than the fast evolving free dataset that is reduced from around 0.4 to 0.2 respectively. Since only correct orthologs, known from the simulation, were fed, this illustrates that including more pairs can lead to worse precision results in the benchmark. This is in agreement with the results observed in this chapter for the two heuristic refinements and our previous hypothesis (higher recall, but lower precision when including additional orthologs among fast evolving genes). Therefore, this illustrates the limitation of assessing HOGs inference quality using the generalised species discordance test.

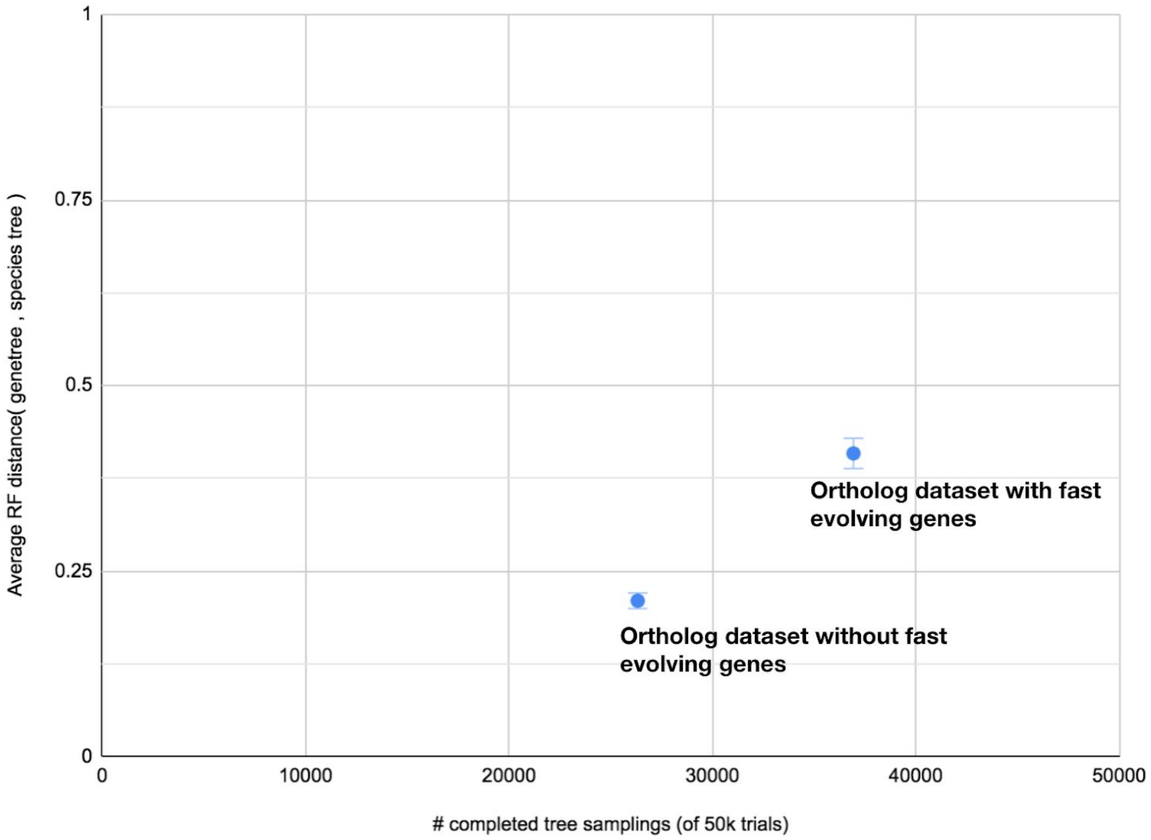


Figure 37: Effect of including fast evolving genes in the generalized species tree discordance test using simulated genomes.

Conclusions

To conclude, we observe that the two new heuristic GETHOGs variants improve the quality of the HOGs inference when using an incomplete orthology graph. The number of abnormal HOGs shown in Chapter 4 is considerably reduced with both refinements. The safe variant for both refinements produces a robust and reliable improvement in the orthology clustering that can be included in a future version of orthology clustering in OMA. On the contrary, the ambitious modes do not produce stable results and the stochasticity introduced is not in agreement with the ‘robustness and scalable’ policy of our inference quality. Even though there are improvements globally, the ambitious mode will not be used as a standard in our HOGs inferences pipeline. In addition, a few cases are not solved by our refinements due to their complexity - the lack of orthology signal makes the reconstruction impossible without an additional source of information (synteny, multiple sequence alignments) - or to their size - in order to conserve scalability we cannot perform the reconstruction on too large a gene family (i.e. paralogous groups with over 10 paralogs).

We also highlight the limitations of the generalised species discordance test of the orthology benchmark service. Indeed, our refinements aim to improve the clustering of genes that have conflicting orthology signal (due to accelerated evolution or fragmentation of their sequences) which is not captured by the pairwise orthology benchmark. On the contrary, the newly added genes make the precision of the reconstruction worse in the benchmark because these conflicting genes are affecting the quality of gene trees reconstructed from the gene families (that is used as a proxy to assess HOG quality). We test this hypothesis by simulating a genomic dataset with fast evolving genes and running a custom generalised species tree discordance test on a full ortholog dataset and on a dataset with fast evolving genes removed. We observed, in agreement with the rest of the chapter results, that these conflicting fast evolving genes improve the recall but have a negative impact on the precision.

Future work may focus first on designing a benchmark test specific to orthologous groups that both assesses the quality of the clustering in terms of membership and in terms of the nested structure. Secondly, the use of dynamic programming approaches may aid the reconstruction of HOGs by considering more potential scenarios at each internal node reconstructed. This could help the algorithm to better select one of the co-optimal scenarios when a conflicting case is found by integrating more information.

Finally, integrating additional sources of evolutionary signal may greatly help the reconstruction. Synteny (and ancestral synteny) is a valuable source of information that can be used when orthology is impossible to call. Furthermore, investigating the multiple sequence alignment of a family during the clustering may bring additional information and correct mistakes introduced by the fact that orthology inference is performed using pairwise protein alignments.

Chapter 6: Conclusion and new directions

Investigating the evolution of modern-day organisms and understanding their associated biodiversity through their gene evolutionary histories have proven to be useful in many research domains. Orthology is a cornerstone of such phylogenetic analysis and has been proven as a strong source of evolutionary signal. The recent breakthrough in sequencing technologies provides complete genome sequences with a better resolution of evolutionary signal, but more available data means dealing with both a large quantity of data (scalability) and potential quality issues (sequencing errors, wrong assemblies, etc..). To address such challenges, many efforts have been made in the past decade to develop robust methods for orthology inferences and HOGs were introduced to meet the need of large scale data structures for phylogenomic analysis. Nevertheless, inferring robust HOGs on a large scale remains a complex task and their downstream analysis can be complicated due to their size and complexity.

In this thesis, we addressed the need to develop robust and scalable tools to infer and process HOGs. My research in this thesis encompasses three subjects: orthology inferences, orthology clustering, orthology visualisation and processing. First, I presented a refined version of the OMA algorithm to include fast evolving genes in orthology inferences which improved the recall and the precision of the method. In addition, I introduced a new version of the OMA HOG inference algorithm which improves both the robustness and scalability of the algorithm. In comparison with the previous GETHOGs top down algorithm, the new bottom up version shows an improved quality of the clusters, in terms of family coverage and orthology inference from the HOGs reconstruction. In addition to the improved robustness, the scalability of the new methods have been reduced from a cubic to quadratic complexity which allows the reconstruction of HOGs with larger datasets in less time (9 hours for 2 thousand genomes with a single process). I then presented two additional heuristic refinements of the GETHOGs algorithm to deal with missing orthology during hierarchical orthology. These new refinements improved the quality of orthology clustering regarding ‘unmerged’ and ‘complementary’ HOGs, reducing the amounts of spurious gene losses and misplaced gene duplications.

In addition to all these algorithm refinements, I introduced several new tools to facilitate the processing and visualisation of HOGs. The lack of existing tools to easily extract, process and visualise information encoded in HOGs motivated us to develop pyHam and iHam. pyHam has been shown to ease the programmatic exploration of HOGs by offering many built-in functionalities to perform not only phylogenetic analysis, but also providing an easy to use programmatic interface to let users customise their analysis and utilisation of pyHam. iHam is complementary in that it provides an interface to visualise and explore HOGs in an intuitive and interactive manner. iHam allows users to fine tune their analysis of a single gene family and to easily synthesize the related evolutionary history. In addition to these two tools, I developed GTM, a web based visualisation tool to explore sequence alignments, the phylogenetic tree and the orthology graph for a given gene family. This tool provides a way to verify that a hypothesis related to a single gene families evolutionary history can be validated by a phylogenetic signal from the sequence alignment or orthology graph.

Despite all this progress, the orthology inferences and clustering are not perfect and require additional improvements. Orthology inference methods are still very sensitive to fragmentary sequences and fast evolving genes. Further improvements to detect and resolve such complicated cases would greatly improve the downstream orthology clustering.

Correspondingly, the orthology clustering quality is highly dependent on the quality of the pairwise orthology used. I show in this thesis that missing orthology is the main source of clustering errors. The development of new orthology clustering methods to efficiently deal with incomplete input data using a dynamic programmatic approach to consider several optimal clustering scenarios might improve HOG reconstruction. Another alternative would be to include other sources of evolutionary signal, such as synteny. Indeed if orthology may have failed in some extent, synteny is a reliable back up source of evidence for evolutionary history reconstruction. The idea is that genes that failed to be inferred as orthologs with traditional sequences-based method due to complex evolutionary history (fast evolving genes, domain shuffling) or due to poor quality assemblies may be saved by their synteny. Indeed, a pair of genes sharing a conserved synteny (meaning that the neighboring genes are orthologous and ordered in the same way) can be inferred as orthologous.

Complementary to improving inferences, there is room for improvement in visualisation and exploration tools of HOGs. For example, integrating more information like Gene Ontology, synteny or secondary structure could help the investigation of HOGs.

The work presented in this thesis on the reconstruction of the evolutionary histories of gene families—a problem much more complex than the already challenging problem of reconstructing species phylogenies—is a resolute step forward toward full ancestral genome reconstruction.

To go even further, I foresee the possibility of including extant synteny information in the HOG framework, so as to be able to infer the synteny of ancestral genomes (which would give a genomic order of HOGs at each level). This ancestral synteny would not only provide additional information for the HOG inference process itself—as orthologs may be more likely to have kept their genomic context than paralogs—but also reveal important genome events occurring over the course of evolution (deletion, insertion, inversion, duplication). Ultimately, the HOG framework is amenable to integration of all aspects which can reasonably be expected to have evolved along the history of genes, such as ancestral alternative splicing, ancestral gene expression, ancestral molecular function, ancestral protein-protein interaction—bringing us ever closer to a comprehensive and accurate reconstruction of the molecular history of life in its full and glorious diversity.

Reference

Altenhoff, Adrian M., Brigitte Boeckmann, Salvador Capella-Gutierrez, Daniel A. Dalquen, Todd DeLuca, Kristoffer Forslund, Jaime Huerta-Cepas, et al. 2016. “Standardized Benchmarking in the Quest for Orthologs.” *Nature Methods* 13 (5): 425–30.

Altenhoff, Adrian M., and Christophe Dessimoz. 2009. “Phylogenetic and Functional Assessment of Orthologs Inference Projects and Methods.” *PLoS Computational Biology* 5 (1): e1000262.

———. 2012. “Inferring Orthology and Paralogy.” In *Evolutionary Genomics*, edited by Maria Anisimova, 855:259–79. Methods in Molecular Biology. Humana Press.

Altenhoff, Adrian M., Manuel Gil, Gaston H. Gonnet, and Christophe Dessimoz. 2013. “Inferring Hierarchical Orthologous Groups from Orthologous Gene Pairs.” *PloS One* 8 (1): e53786.

Altenhoff, Adrian M., Natasha M. Glover, Clément-Marie Train, Klara Kaleb, Alex Warwick Vesztrocy, David Dylus, Tarcisio M. de Farias, et al. 2017. “The OMA Orthology Database in 2018: Retrieving Evolutionary Relationships among All Domains of Life through Richer Web and Programmatic Interfaces.” *Nucleic Acids Research*, November. https://doi.org/10.1093/nar/gkx1019.

Altenhoff, Adrian M., Jeremy Levy, Magdalena Zarowiecki, Bartłomiej Tomiczek, Alex Warwick Vesztrocy, Daniel Dalquen, Steven Müller, et al. n.d. “OMA Standalone: Orthology Inference among Public and Custom Genomes and Transcriptomes.” https://doi.org/10.1101/397752.

Altenhoff, Adrian M., Romain A. Studer, Marc Robinson-Rechavi, and Christophe Dessimoz. 2012. “Resolving the Ortholog Conjecture: Orthologs Tend to Be Weakly, but Significantly, More Similar in Function than Paralogs.” *PLoS Computational Biology* 8 (5): e1002514.

Altenhoff, A. M., A. Schneider, G. H. Gonnet, and C. Dessimoz. 2011. “OMA 2011: Orthology Inference among 1000 Complete Genomes.” *Nucleic Acids Research*. https://doi.org/10.1093/nar/gkq1238.

Altschul, S. F., W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. 1990. “Basic Local Alignment Search Tool.” *Journal of Molecular Biology* 215 (3): 403–10.

Bennett, Simon. 2004. “Solexa Ltd.” *Pharmacogenomics* 5 (4): 433–38.

Boeckmann, Brigitte, Marc Robinson-Rechavi, Ioannis Xenarios, and Christophe Dessimoz.

2011. “Conceptual Framework and Pilot Study to Benchmark Phylogenomic Databases Based on Reference Gene Trees.” *Briefings in Bioinformatics* 12 (5): 423–35.

Chevenet, François, Jean-Philippe Doyon, Celine Scornavacca, Edwin Jacox, Emmanuelle Jousselin, and Vincent Berry. 2016. “SylvX: A Viewer for Phylogenetic Tree Reconciliations.” *Bioinformatics*. https://doi.org/10.1093/bioinformatics/btv625.

Chothia, C., and A. M. Lesk. 1986. “The Relation between the Divergence of Sequence and Structure in Proteins.” *The EMBO Journal*. https://doi.org/10.1002/j.1460-2075.1986.tb04288.x.

Dalquen, Daniel A., Adrian M. Altenhoff, Gaston H. Gonnet, and Christophe Dessimoz. 2013. “The Impact of Gene Duplication, Insertion, Deletion, Lateral Gene Transfer and Sequencing Error on Orthology Inference: A Simulation Study.” *PloS One* 8 (2): e56925.

Dalquen, Daniel A., Maria Anisimova, Gaston H. Gonnet, and Christophe Dessimoz. 2012. “ALF--a Simulation Framework for Genome Evolution.” *Molecular Biology and Evolution* 29 (4): 1115–23.

Dessimoz, Christophe, Brigitte Boeckmann, Alexander C. J. Roth, and Gaston H. Gonnet. 2006. “Detecting Non-Orthology in the COGs Database and Other Approaches Grouping Orthologs Using Genome-Specific Best Hits.” *Nucleic Acids Research* 34 (11): 3309–16.

Dessimoz, Christophe, Gina Cannarozzi, Manuel Gil, Daniel Margadant, Alexander Roth, Adrian Schneider, and Gaston H. Gonnet. 2005. “OMA, A Comprehensive, Automated Project for the Identification of Orthologs from Complete Genome Data: Introduction and First Achievements.” In *Lecture Notes in Computer Science*, 61–72.

Dikow, Rebecca B., Paul B. Frandsen, Mauren Turcatel, and Torsten Dikow. 2017. “Genomic and Transcriptomic Resources for Assassin Flies Including the Complete Genome Sequence of Proctacanthus Coquilletti (Insecta: Diptera: Asilidae) and 16 Representative Transcriptomes.” *PeerJ* 5 (January): e2951.

Egger, Bernhard, François Lapraz, Bartłomiej Tomiczek, Steven Müller, Christophe Dessimoz, Johannes Girstmair, Nives Škunca, et al. 2015. “A Transcriptomic-Phylogenomic Analysis of the Evolutionary Relationships of Flatworms.” *Current Biology: CB* 0 (0). https://doi.org/10.1016/j.cub.2015.03.034.

Fernández, Rosa, and Gonzalo Giribet. 2015. “Unnoticed in the Tropics: Phylogenomic Resolution of the Poorly Known Arachnid Order Ricinulei (Arachnida).” *Royal Society Open Science* 2 (6): 150065.

Fernández, Rosa, Christopher E. Laumer, Varpu Vahtera, Silvia Libro, Stefan Kaluziak, Prashant P. Sharma, Alicia R. Pérez-Porro, Gregory D. Edgecombe, and Gonzalo Giribet. 2014. “Evaluating Topological Conflict in Centipede Phylogeny Using Transcriptomic Data Sets.” *Molecular Biology and Evolution* 31 (6): 1500–1513.

Fitch, W. M. 1970. “Distinguishing Homologous from Analogous Proteins.” *Systematic Zoology* 19 (2): 99–113.

Gabaldón, Toni, and Eugene V. Koonin. 2013. “Functional and Evolutionary Implications of Gene Orthology.” *Nature Reviews. Genetics* 14 (5): 360–66.

Garrison, Nicole L., Juanita Rodriguez, Ingi Agnarsson, Jonathan A. Coddington, Charles E. Griswold, Christopher A. Hamilton, Marshal Hedin, Kevin M. Kocot, Joel M. Ledford, and Jason E. Bond. 2016. “Spider Phylogenomics: Untangling the Spider Tree of Life.” *PeerJ* 4 (February): e1719.

Glover, Natasha, Christophe Dessimoz, Ingo Ebersberger, Sofia K. Forslund, Toni Gabaldón, Jaime Huerta-Cepas, Maria-Jesus Martin, et al. 2019. “Advances and Applications in the Quest for Orthologs.” *Molecular Biology and Evolution*, June. https://doi.org/10.1093/molbev/msz150.

Heijden, René T. J. M. van der, Berend Snel, Vera van Noort, and Martijn A. Huynen. 2007. “Orthology Prediction at Scalable Resolution by Phylogenetic Tree Analysis.” *BMC Bioinformatics* 8 (March): 83.

Herrero, Javier, Matthieu Muffato, Kathryn Beal, Stephen Fitzgerald, Leo Gordon, Miguel Pignatelli, Albert J. Vilella, et al. 2016. “Ensembl Comparative Genomics Resources.” *Database*. https://doi.org/10.1093/database/bav096.

Huerta-Cepas, Jaime, Hernán Dopazo, Joaquín Dopazo, and Toni Gabaldón. 2007. “The Human Phylome.” *Genome Biology* 8 (6): R109.

Huerta-Cepas, Jaime, Joaquín Dopazo, and Toni Gabaldón. 2010. “ETE: A Python Environment for Tree Exploration.” *BMC Bioinformatics* 11 (January): 24.

Huerta-Cepas, Jaime, Damian Szklarczyk, Kristoffer Forslund, Helen Cook, Davide Heller, Mathias C. Walter, Thomas Rattei, et al. 2016. “eggNOG 4.5: A Hierarchical Orthology Framework with Improved Functional Annotations for Eukaryotic, Prokaryotic and Viral Sequences.” *Nucleic Acids Research*. https://doi.org/10.1093/nar/gkv1248.

Jensen, Lars Juhl, Philippe Julien, Michael Kuhn, Christian von Mering, Jean Muller, Tobias Doerks, and Peer Bork. 2008. “eggNOG: Automated Construction and Annotation of Orthologous Groups of Genes.” *Nucleic Acids Research* 36 (Database issue): D250–54.

Kaduk, Mateusz, and Erik Sonnhammer. 2017. “Improved Orthology Inference with Hieranoid 2.” *Bioinformatics* 33 (8): 1154–59.

Koski, Liisa B., and G. Brian Golding. 2001. “The Closest BLAST Hit Is Often Not the Nearest Neighbor.” *Journal of Molecular Evolution*. https://doi.org/10.1007/s002390010184.

Kriventseva, Evgenia V., Nazim Rahman, Octavio Espinosa, and Evgeny M. Zdobnov. 2008. “OrthoDB: The Hierarchical Catalog of Eukaryotic Orthologs.” *Nucleic Acids Research* 36 (Database issue): D271–75.

Kuhn, Michael, Damian Szklarczyk, Sune Pletscher-Frankild, Thomas H. Blicher, Christian von Mering, Lars J. Jensen, and Peer Bork. 2014. “STITCH 4: Integration of Protein–chemical Interactions with User Data.” *Nucleic Acids Research* 42 (D1): D401–7.

Laumer, Christopher E., Andreas Hejnol, and Gonzalo Giribet. 2015. “Nuclear Genomic Signals of the ‘Microturbellarian’ Roots of Platyhelminth Evolutionary Innovation.” *eLife* 4 (March). https://doi.org/10.7554/eLife.05503.

Li, L. 2003. “OrthoMCL: Identification of Ortholog Groups for Eukaryotic Genomes.” *Genome Research* 13 (9): 2178–89.

Linard, Benjamin, Alexis Allot, Raphaël Schneider, Can Morel, Raymond Ripp, Marc Bigler, Julie D. Thompson, Olivier Poch, and Odile Lecompte. 2014. “OrthoInspector 2.0: Software and Database Updates.” *Bioinformatics* 31 (3): 447–48.

Margulies, Marcel, Michael Egholm, William E. Altman, Said Attiya, Joel S. Bader, Lisa A. Bemben, Jan Berka, et al. 2005. “Genome Sequencing in Microfabricated High-Density Picolitre Reactors.” *Nature* 437 (7057): 376–80.

Needleman, S. B., and C. D. Wunsch. 1970. “A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins.” *Journal of Molecular Biology* 48 (3): 443–53.

Overbeek, R., M. Fonstein, M. D’Souza, G. D. Pusch, and N. Maltsev. 1999. “The Use of Gene Clusters to Infer Functional Coupling.” *Proceedings of the National Academy of Sciences of the United States of America* 96 (6): 2896–2901.

Pearson, William R. 2013. “An Introduction to Sequence Similarity (‘Homology’) Searching.” *Current Protocols in Bioinformatics*. https://doi.org/10.1002/0471250953.bi0301s42.

Pearson, W. R. 2000. “Flexible Sequence Similarity Searching with the FASTA3 Program Package.” *Methods in Molecular Biology* 132: 185–219.

Piližota, Ivana, Clément-Marie Train, Adrian Altenhoff, Henning Redestig, and Christophe Dessimoz. 2018. “Phylogenetic Approaches to Identifying Fragments of the Same Gene, with Application to the Wheat Genome.” *Bioinformatics* , September. https://doi.org/10.1093/bioinformatics/bty772.

Remm, Maido, Christian E. V. Storm, and Erik L. L. Sonnhammer. 2001. “Automatic Clustering of Orthologs and in-Paralogs from Pairwise Species Comparisons.” *Journal of Molecular*

Biology 314 (5): 1041–52.

Robinson, Oscar, David Dylus, and Christophe Dessimoz. 2016. “Phylo.io: Interactive Viewing and Comparison of Large Phylogenetic Trees on the Web.” *Molecular Biology and Evolution* 33 (8): 2163–66.

Rognes, Torbjørn. 2011. “Faster Smith-Waterman Database Searches with Inter-Sequence SIMD Parallelisation.” *BMC Bioinformatics* 12 (June): 221.

Roth, Alexander C. J., Gaston H. Gonnet, and Christophe Dessimoz. 2008. “Algorithm of OMA for Large-Scale Orthology Inference.” *BMC Bioinformatics* 9 (December): 518.

Sanger, F., S. Nicklen, and A. R. Coulson. 1977. “DNA Sequencing with Chain-Terminating Inhibitors.” *Proceedings of the National Academy of Sciences* 74 (12): 5463–67.

Sharma, P. P., R. Fernandez, Gonzalez R. Santillan, and L. Monod. 2015. “Phylogenomic Resolution of Scorpions Reveals Discordance with Morphological Phylogenetic Signal.” In *INTEGRATIVE AND COMPARATIVE BIOLOGY*, 55:E165–E165. OXFORD UNIV PRESS INC JOURNALS DEPT, 2001 EVANS RD, CARY, NC 27513 USA.

Sharma, Prashant P., Stefan T. Kaluziak, Alicia R. Pérez-Porro, Vanessa L. González, Gustavo Hormiga, Ward C. Wheeler, and Gonzalo Giribet. 2014. “Phylogenomic Interrogation of Arachnida Reveals Systemic Conflicts in Phylogenetic Signal.” *Molecular Biology and Evolution* 31 (11): 2963–84.

Smith, T. F., and M. S. Waterman. 1981. “Identification of Common Molecular Subsequences.” *Journal of Molecular Biology*. https://doi.org/10.1016/0022-2836(81)90087-5.

Sonnhammer, Erik L. L., Toni Gabaldón, Alan W. Sousa da Silva, Maria Martin, Marc Robinson-Rechavi, Brigitte Boeckmann, Paul D. Thomas, Christophe Dessimoz, and Quest for Orthologs consortium. 2014. “Big Data and Other Challenges in the Quest for Orthologs.” *Bioinformatics* 30 (21): 2993–98.

Szklarczyk, Damian, Andrea Franceschini, Stefan Wyder, Kristoffer Forslund, Davide Heller, Jaime Huerta-Cepas, Milan Simonovic, et al. 2015. “STRING v10: Protein–protein Interaction Networks, Integrated over the Tree of Life.” *Nucleic Acids Research* 43 (D1): D447–52.

Tatusov, R. L. 1997. “A Genomic Perspective on Protein Families.” *Science* 278 (5338): 631–37.

Trachana, Kalliopi, Tomas A. Larsson, Sean Powell, Wei-Hua Chen, Tobias Doerks, Jean Muller, and Peer Bork. 2011. “Orthology Prediction Methods: A Quality Assessment Using Curated Protein Families.” *BioEssays: News and Reviews in Molecular, Cellular and Developmental Biology* 33 (10): 769–80.

Train, Clément-Marie, Natasha M. Glover, Gaston H. Gonnet, Adrian M. Altenhoff, and Christophe Dessimoz. 2017. “Orthologous Matrix (OMA) Algorithm 2.0: More Robust to Asymmetric Evolutionary Rates and More Scalable Hierarchical Orthologous Group Inference.” *Bioinformatics* 33 (14): i75–82.

Train, Clément-Marie, Miguel Pignatelli, Adrian Altenhoff, and Christophe Dessimoz. 2018. “iHam & pyHam: Visualizing and Processing Hierarchical Orthologous Groups.” *Bioinformatics* , December. https://doi.org/10.1093/bioinformatics/bty994.

Tsai, Isheng J., Magdalena Zarowiecki, Nancy Holroyd, Alejandro Garciarrubio, Alejandro Sanchez-Flores, Karen L. Brooks, Alan Tracey, et al. 2013. “The Genomes of Four Tapeworm Species Reveal Adaptations to Parasitism.” *Nature* 496 (7443): 57–63.

Vilella, Albert J., Jessica Severin, Abel Ureta-Vidal, Li Heng, Richard Durbin, and Ewan Birney. 2009. “EnsemblCompara GeneTrees: Complete, Duplication-Aware Phylogenetic Trees in Vertebrates.” *Genome Research* 19 (2): 327–35.

Waterhouse, Robert M., Fredrik Tegenfeldt, Jia Li, Evgeny M. Zdobnov, and Evgenia V. Kriventseva. 2013. “OrthoDB: A Hierarchical Catalog of Animal, Fungal and Bacterial Orthologs.” *Nucleic Acids Research* 41 (Database issue): D358–65.

Williams, Tom A., Gergely J. Szöllősi, Anja Spang, Peter G. Foster, Sarah E. Heaps, Bastien

Boussau, Thijs J. G. Ettema, and T. Martin Embley. 2017. “Integrative Modeling of Gene and Genome Evolution Roots the Archaeal Tree of Life.” *Proceedings of the National Academy of Sciences of the United States of America* 114 (23): E4602–11.