Bioinformatics, 33(8), 2017, 1154–1159
doi: 10.1093/bioinformatics/btw774
Advance Access Publication Date: 17 January 2017
Original Paper

OXFORD

Phylogenetics

Improved orthology inference with Hieranoid 2

Mateusz Kaduk[1,2,*] and Erik Sonnhammer[1,2]

[1]Department of Biochemistry and Biophysics, Stockholm University and [2]Science for Life Laboratory (SciLifeLab), Tomtebodavagen 23, Solna, Sweden

*To whom correspondence should be addressed.
Associate Editor: Alfonso Valencia

Received on May 18, 2016; revised on November 4, 2016; editorial decision on November 30, 2016; accepted on December 7, 2016

Abstract

Motivation: The initial step in many orthology inference methods is the computationally demanding establishment of all pairwise protein similarities across all analysed proteomes. The quadratic scaling with proteomes has become a major bottleneck. A remedy is offered by the Hieranoid algorithm which reduces the complexity to linear by hierarchically aggregating ortholog groups from InParanoid along a species tree.
Results: We have further developed the Hieranoid algorithm in many ways. Major improvements have been made to the construction of multiple sequence alignments and consensus sequences. Hieranoid version 2 was evaluated with standard benchmarks that reveal a dramatic increase in the coverage/accuracy tradeoff over version 1, such that it now compares favourably with the best methods. The new parallelized cluster mode allows Hieranoid to be run on large data sets in a much shorter timespan than InParanoid, yet at similar accuracy.
Contact: mateusz.kaduk@scilifelab.se
Availability and Implementation: Perl code freely available at http://hieranoid.sbc.su.se/.
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Orthologs are genes that were duplicated upon speciation. This distinguishes them from paralogs which are genes duplicated within the same species after or before speciation (Jensen, 2001). Paralogs of the type inparalogs duplicated after speciation, and can simultaneously be orthologs with respect to a gene in a different species (O'Brien *et al.*, 2005). Conversely, outparalogs are paralogs that come from a duplication predating the speciation event; for example, HA1 cannot be considered orthologs to WB1. The evolutionary relationship between inparalogs and outparalogs is depicted in Figure 1.

An important and challenging topic in bioinformatics is to distinguish inparalogs from outparalogs in order to infer real orthologs from a vast amount of data. The inference of orthologs is important because many proteins related by orthology often retain similar biological functions which can be preserved over species (Gabaldón and Koonin, 2013; Remm *et al.*, 2001). By definition orthologs were duplicated only upon speciation, so the phylogeny directly reflects their lineage. Paralogs are members of multi-gene families and in the case of missing copies their use in inference of relationships between species may be misleading (Baldauf, 2003). Thus, orthologs are suitable and important candidates for inferring phylogenetic trees.

Recent advances in whole-genome sequencing bring an opportunity to infer orthology between many species. Thus, algorithms that can determine orthologs accurately and quickly are needed to make sense of the growing sequence databases. Orthology methods can be categorized as either tree-based or graph-based (Sonnhammer *et al.*, 2014). Despite algorithmic differences, most methods rely on exhaustive all-versus-all sequence similarity searching. Although these can be parallelized or speeded up algorithmically (Wittwer *et al.*, 2014), the quadratically growing search space is outpacing the growth in computational power. What is needed is an algorithm that scales linearly with the growth of sequence data.

To achieve this, Hieranoid was introduced (Schreiber and Sonnhammer, 2013; Wittwer *et al.*, 2014). It can be seen as a hybrid between tree and graph based approaches. A species tree is utilized as a guide tree which reduces the number of proteome comparisons to N-1 for N proteomes. This also naturally infers groups of orthologs in a hierarchical structure. However, Hieranoid version 1 was only benchmarked with Orthobench (Trachana *et al.*, 2011) but this

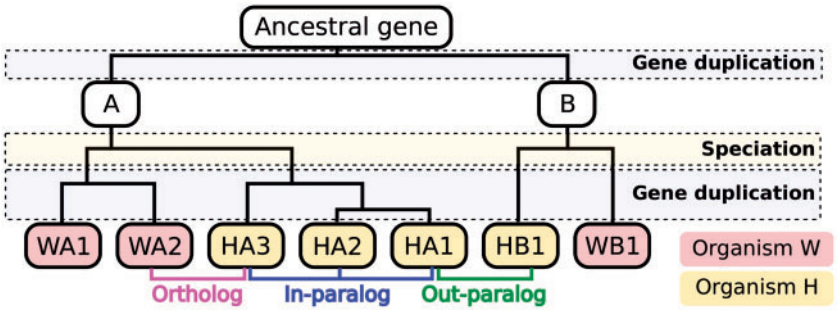

Fig. 1. Evolutionary relationships between orthologs, inparalogs and outparalogs

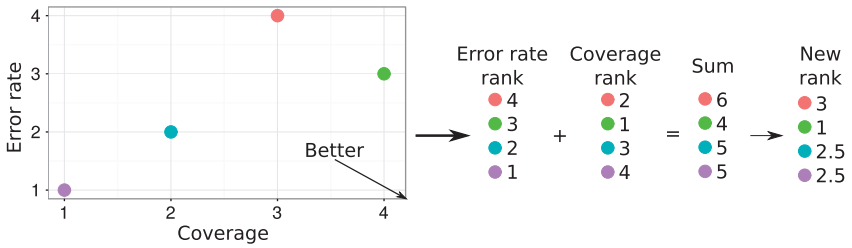

Fig. 2. Total rank is the rank based on both the horizontal and vertical axes (Color version of this figure is available at *Bioinformatics* online.)

was later found to not be very specific as it does not penalize erroneous orthology assignments such as outparalogs in recently diverged species. It also was not possible to run the algorithm in parallel on a compute cluster which severely limited its throughput.

Proper benchmarking of Hieranoid with the Quest for Orthologs benchmarks revealed that both its coverage and accuracy were far worse than InParanoid and other top-performing methods. We therefore analysed its algorithmic components and identified a number of shortcomings. These have been mended in Hieranoid 2, which now yields reproducible results with coverage and accuracy similar to InParanoid. Here we present the novel aspects of the Hieranoid 2 algorithm and assess its quality by standardized benchmarks for orthology inference methods.

2 Used software

To approximate an evolutionary distance between protein sequences, BLAST version 2.2.18 (Camacho *et al.*, 2009) was used. Although it is not as accurate as BLAST, it is possible to use USEARCH as an alternative to save time (Edgar, 2010). MUSCLE version 3.8.31 (Edgar, 2004) was used to build multiple alignments. The Neighbour-Joining tree in Supplementary Figure S1 was built with Belvu (Sonnhammer and Hollich, 2005) using Scoredist distances. Ortholog pairs were extracted from protein trees using tree reconciliation in the ETE3 (Huerta-Cepas *et al.*, 2016) package.

3 Benchmark

Results were evaluated based on standard orthology benchmarks at http://orthology.benchmarkservice.org established by Quest for Orthologs community (Altenhoff *et al.*, 2016; Dessimoz *et al.*, 2012). It is based on the 66 reference proteomes QfO_2011-04 [http://www.ebi.ac.uk/reference_proteomes]. This benchmark allows us to directly compare the performance of Hieranoid 2 to other methods, as well as analysing the impact of algorithmic variants. Since the Hieranoid output is a tree with ortholog assignments at many hierarchical levels, pairs of orthologs were obtained by parsing out each pair of species separately and finding their last common ancestor (LCA) to resolve all their orthologs below that point for entry into the benchmark service which requires flat pairwise orthologs.

3.1 Species discordance test

The species discordance test (Altenhoff *et al.*, 2016) used in the benchmark is based on the premise that the topology of the species tree should agree with the tree of orthologs. Given a species tree, orthologs are sampled for all species from predicted orthologs, after which a multiple alignment is built and a phylogenetic tree is inferred. This is attempted 50 000 times. The number of successfully sampled trees can be seen as a proxy of coverage; alternatively the number of predicted orthologs can be used. To estimate accuracy, the trees are compared with a curated species tree to calculate the average Robinson–Foulds distance (avgRF) or the average fraction incorrect trees.

3.2 Ranking

Total rank, as explained in Figure 2 was introduced as a combined performance indicator of orthology methods. The individual axes may represent coverage (horizontal) and error rate (vertical). The axis values are ranked first; these ranks are summed up and ranked again to yield the new rank, which gives equal weight to both axes. If several methods share the same final rank, they are all assigned the average rank of the span.

4 The Hieranoid algorithm

The algorithm requires a fully bifurcated species tree. Each leaf represents a proteome corresponding to all proteins of a species and should only include the longest protein sequence for each gene.

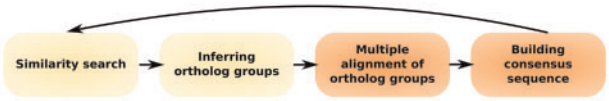

Fig. 3. Workflow of the Hieranoid algorithm. Important improvements in version 2 have been made to the last two stages

Internal nodes in the species tree represent higher-level 'meta-species' and will aggregate all the ortholog groups in the nodes below.

Hieranoid iterates orthology inference hierarchically across the entire species tree (Fig. 3). It starts by searching for sequence similarities between the two closest leaves in the tree. BLAST or USEARCH can then be used to collect sequence similarities. The second stage infers ortholog groups using InParanoid, and the third builds multiple sequence alignments (MSAs) for each ortholog group. The fourth stage then builds consensus sequences or profiles from each MSA. After this the algorithm loops back to the beginning and applies the four stages to the next closest pair in the species tree. If this involves an internal node, the previously obtained consensus sequences or profiles will be used.

4.1 Similarity search

The homology data are converted to the InParanoid input format from the used homology detection tools such as BLAST or USEARCH which search for matches between parts of the sequence and may assign high scores for only small regions of the whole sequence. Highly similar fragments might be related to conserved domains, but such similarity does not necessarily reflect evolutionary similarity of the whole protein. Therefore, the default InParanoid overlap criteria were used to prevent the algorithm from taking into account results based on too short matches. The following criteria need to be fulfilled for the match to be accepted. First, the distance from the first to the last aligned residue must be at least 50% of either sequence. Second, the length of the aligned regions must be at least 25% of the length of either sequence. Finally, if multiple maximal segment pairs (MSPs) are present, they need to be in the same order on both sequences, and the overlap between each consecutive MSP must not exceed 5%.

4.2 Inferring ortholog groups

Once the initial similarities are established, either with BLAST or USEARCH, results in the BLAST format are passed over to the InParanoid module, where the clustering is performed according to the original InParanoid algorithm (Remm *et al.*, 2001). Inferred clusters will henceforth be referred as ortholog groups, and in such groups only pairs of proteins from different species may be considered as orthologs.

4.3 Multiple sequence alignment of ortholog groups

To capture the sequence diversity in an ortholog group, a multiple sequence alignment (MSA) is built. The MSA attempts to align all the sequences simultaneously, so that each column of multiple sequences results is in the optimal arrangement of residues. The MSA in Hieranoid 1 was not built from all sequences that belong to the original species, but Hieranoid 2 parts from this behaviour. The MSA is rebuilt on each branch when the ortholog group is extended by new proteins using all putative sequences from the updated ortholog groups (rather than relying on consensus sequences). This guarantees that any decision on representative residue within a MSA column may be corrected if more sequences are added in the process of adding new orthologs to ortholog group.

4.4 Building consensus sequences

Once the sequences for each ortholog group are aligned, a consensus sequence is calculated. One can optionally use profiles instead, but searching these against each other is prohibitively computationally demanding. The consensus building was originally taken from BioPerl 1.6.9 (Stajich *et al.*, 2002) but is now replaced by custom routine. In Hieranoid 1, a consensus residue was only produced if an amino acid had a frequency of 50% or higher, leading to shrinkage of the consensus sequence. In Hieranoid 2, only positions with more than 50% gaps are removed. For other positions, the consensus residue is picked that has the highest score against all residues in the column, using the BLOSUM62 substitution matrix (Pearson, 2002). If several amino acids are tied, the first in alphabetical order is picked.

4.5 Further iterations

Internal nodes are used as the algorithm proceeds, rather than leaves, to infer orthologs. The set of all sequences at an internal node represents a pseudo-species for all species below that point in the species tree. When two pseudo-species meet at a node, searching is performed with consensus sequences versus consensus-sequences from corresponding pseudo-species. However, once the results are clustered, the algorithm returns to the previously described stages of building MSA (updating it in this case) and building new consensus sequences from all original sequences. For this stage, consensus sequences are unfolded into original sequences. When the algorithm finishes at the root node, all intermediate results are combined into the final hierarchical groups of orthologs.

4.6 Parallel mode

This new feature of Hieranoid 2 adds the ability of Hieranoid to be run in parallel. New cluster mode or fixed multi-core mode combined with the hierarchical approach further strengthens Hieranoid position in applications that demand solving large scale multi-species orthology inference problems. For the 66 proteomes in study we ran Hieranoid 2 on an Intel Xeon E5-2623 3.0 GHz processor with 48 cores. This took 2.5 real clock core days and 2.3 user core days.

5 Results

5.1 Benchmark performance

Hieranoid 2 was run on the 66 reference proteomes that the Quest for Orthologs benchmarks are based on. It produced a total of 40 700 ortholog group trees, with 7.8 species and 12.3 sequences per ortholog tree on average. The largest ortholog tree had 6973 sequences from 14 species; 2 ortholog trees included all 66 species, and 3096 ortholog trees included at least half of all species. The ortholog trees were parsed into 6 349 666 ortholog pairs, which were uploaded to the orthology benchmark service. For comparison, we also ran the code with the old Hieranoid 1 algorithmic components but with patches for efficient computation, as the 1.0 release was not parallelized. This only inferred 4 817 251 ortholog pairs.

The benchmark service contains several benchmarks. The 'generalized species tree discordance benchmark' tests whether orthologs sampled for a set of species produce a tree that agrees with the known species tree (Altenhoff and Dessimoz, 2009). We show this benchmark for Eukaryota as an example in Figure 5 because it separates the methods well, but other benchmarks give a similar picture, see Supplementary Materials. The complete results for Hieranoid 2 are available at the benchmark service for public viewing. We note that Hieranoid 2 is at a similar level of accuracy as

InParanoid but has lower coverage in Figure 5. Hieranoid 2 has a total rank of 3 for coverage measured as sampled trees and 2.5 for number of orthologs, after InParanoid, RBH/BBH, and OMA GETHOGs. All methods show a tradeoff between accuracy and coverage. The goal is to maximize both, i.e. to come as close as possible to the bottom right corner. Hieranoid 2 is not dominated or shadowed by any other method in this respect, i.e. it lies on the 'Pareto frontier'. This suggests that it strikes a unique and competitive balance between accuracy and coverage.

Hieranoid 1 which is also marked in Figure 5 is placed among the poorer performing methods. The transition from rank 9.5 to 3 (Fig. 5A) and 4.5 to 2.5 (Fig. 5B) shows that Hieranoid 2 is a substantial improvement compared to version 1. Hieranoid 1 using USEARCH, the version that was previously published, is also marked and shows considerably lower coverage than the Blast-using option.

The 'generalized species tree discordance benchmark' comprises four taxonomic ranges, each having four plots combining two accuracy and two coverage measures (Supplementary Materials). As these tests give different total rankings we wanted to measure how the methods perform overall. All 16 plots were summarized by a median total rank across all tests for each method. The best median rank of 2 across all plots was obtained by InParanoid, the second best by phylomeDB with 4.75, and the third best by Hieranoid 2 (Blast) with 5.5 (Supplementary Table S1). In many tests, Hieranoid's performance is not much worse than InParanoid's, and in fact Hieranoid achieves much better total rank in all *Vertebrata* tests.

5.2 Benchmarking individual improvements

The following results reflect only the difference between changes introduced to the algorithm. The benchmark results from InParanoid are used here as a reference. The benchmark performance of Hieranoid, which does not require all one-to-one comparisons, should ideally approach the results of the all one-to-one method InParanoid.

Figure 4 shows the effect on benchmark performance of two major Hieranoid algorithm improvements: the new consensus sequence building (CS2) and the new routine for building multiple sequence alignments (MSA2). In combination they achieve the performance that comes closest to InParanoid's. The CS2 code appears to make a bigger difference in coverage than the MSA2 code. It can also be observed that the original code (CS1–MSA1) performs better than the single modification CS1–MSA2. This is actually not surprising, as the flaws of CS1 should become more deleterious when using the full alignment. Also, the improvement of CS2 only shows in combination with MSA2, as the MSA1 flaws affect CS1 and CS2 about equally.

We note that the performance improvement of the combined MSA2 and CS2 modifications when using BLAST, i.e. between 'Hieranoid 1 Blast' and 'Hieranoid 2 Blast' in Figure 5, is roughly the same as the improvement from USEARCH to BLAST, i.e. between 'Hieranoid 1 Usearch' and 'Hieranoid 1 Blast' in Figure 5. This suggests that although USEARCH may seem attractive because it is much faster than BLAST, its considerably lower coverage makes it less suitable for orthology inference.

5.3 Comparison to InParanoid

InParanoid has previously been shown to keep both false positives and false negatives at low rates (Figure 5; Altenhoff and Dessimoz, 2009; Chen *et al.*, 2007; Hulsen *et al.*, 2006) and is widely used for inferring orthologs between pairs of species. As Hieranoid employs InParanoid during its hierarchical process, it is of interest to compare the results of both approaches. To get an overview, the relative increase/decrease of orthologs between them is plotted for each species pair in the 66-species benchmark dataset in Figure 6.

In general, Hieranoid 2 finds a lower amount of orthologs per species pair than InParanoid. The most extreme example of this is the species pair closest to the top left corner of Figure 6, corresponding to *Danio rerio* versus mouse (*Mus musculus*) where Inparanoid infers 3.5 times more orthologs than Hieranoid. It is mainly due to a huge ortholog group of olfactory receptors that in InParanoid has 50 *D. rerio* and 864 mouse proteins, generating 43 200 ortholog pairs. This is the highest number of ortholog pairs from one ortholog group in the entire InParanoid dataset. The corresponding ortholog tree for the same species pair in Hieranoid contains 66 *D. rerio* and 1095 mouse proteins, of which 32 and 863 (98%) are the same as in the InParanoid ortholog group, respectively. However, Hieranoid has in this case used the context of other species to infer that many duplications happened before the *D. rerio*/mouse split, and all proteins arising this way are counted as outparalogs, i.e. do not produce any ortholog pairs. Because of this, Hieranoid's ortholog tree, although much larger than InParanoid's ortholog group, yielded only 2234 ortholog pairs in 5 'ingroup' clades after *D. rerio*/mouse speciation nodes. The main reasons for the much lower number of ortholog pairs in Hieranoid are that 28 *D. rerio* proteins are placed as outparalogs in the tree, and that the ingroup clade with most mouse proteins (649) contained only 1 *D. rerio* protein.

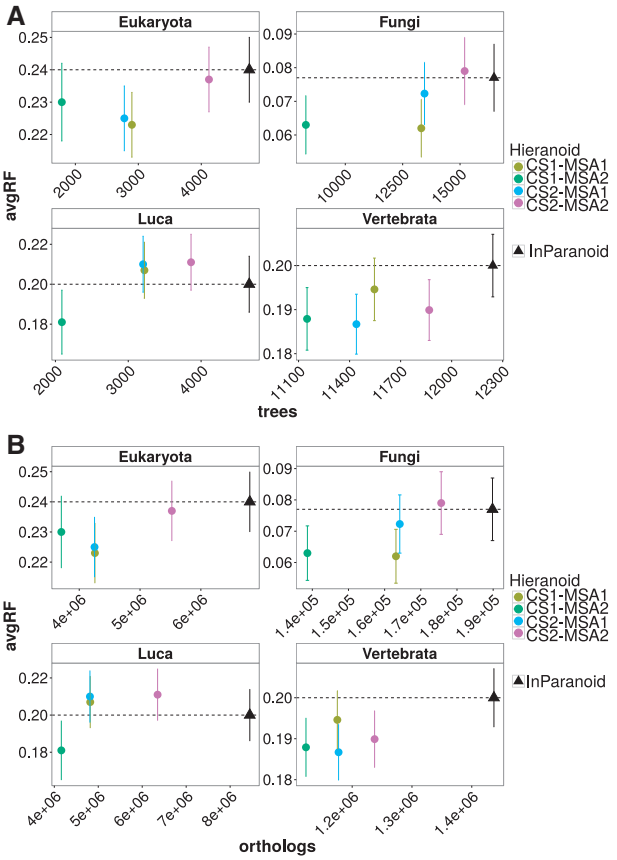

Fig. 4. Generalized species tree discordance benchmark for eukaryota, revealing the performance of Hieranoid 2 and 1 as well as a range of other orthology inference methods. avgRF: average Robinson–Foulds distance between inferred and true species tree (proxy for accuracy). The number of sampled trees or uploaded orthologs on the *x*-axis is a proxy for coverage (Color version of this figure is available at *Bioinformatics* online.)

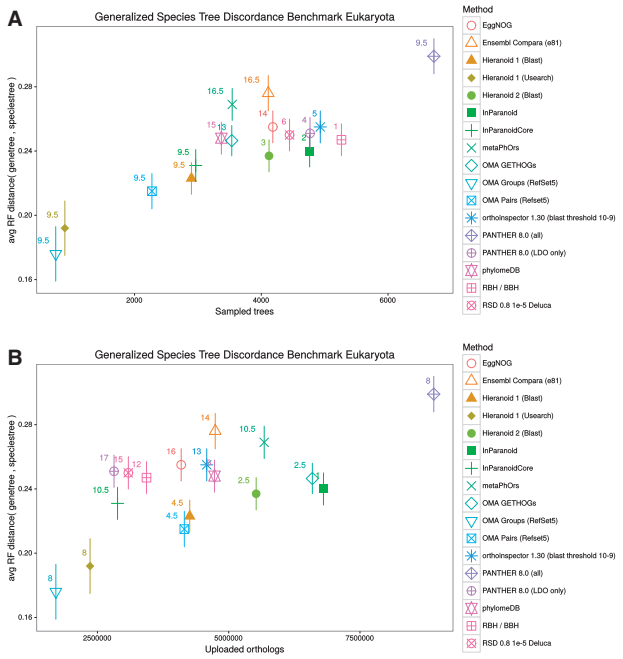

Fig. 5. Generalized species tree discordance benchmark of individual algorithmic modifications in Hieranoid. The lowest average Robinson–Foulds distance to (avgRF) is the best accuracy, while the number of sampled trees (A) or orthologs (B) represents the coverage

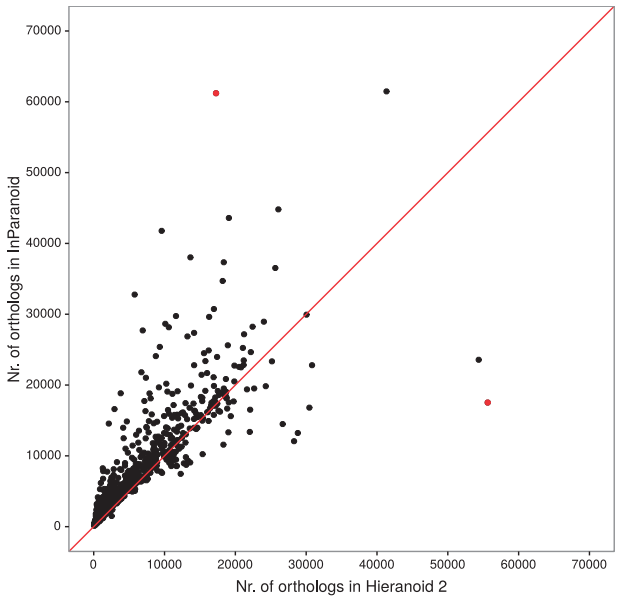

Fig. 6. Total number of ortholog pairs per species pair found by Hieranoid 2 and InParanoid. The Spearman correlation coefficient is 0.95. The line indicates the diagonal with the same number of orthologs for both methods. The two most extreme outliers analysed in the text are the points in top left and bottom right corners (Color version of this figure is available at *Bioinformatics* online.)

The evolution of hugely duplicated families such as this one is often difficult to reconstruct confidently. With hundreds of similarly diverged proteins, assignment of duplication nodes relative to speciation nodes becomes rather imprecise, and orthology assignments will vary substantially between methods. For comparison we built a Neighbour-Joining tree of all 6973 sequences in the Hieranoid ortholog tree. Overall, this gave a similar pattern as Hieranoid with many duplications preceding the *D. rerio*/mouse split, which yielded 1001 ortholog pairs between them in three ingroup clades after *D. rerio*/mouse speciation nodes. The tree, restricted to these two species, is provided as Supplementary Figure S1. The main reasons for the relatively low number of ortholog pairs are that 50 *D. rerio* proteins were placed as outparalogs, and that the ingroup clade with most mouse proteins (967) only contained 1 *D. rerio* protein. This suggests that the number of ortholog pairs between *D. rerio* and mouse was severely overestimated by InParanoid for this family, and that Hieranoid gives a more realistic picture.

On the other hand, Hieranoid 2 found more orthologs than InParanoid for 5% of the species pairs. The most extreme case of this corresponds to the species pair closest to the bottom right corner of Figure 6, *Branchiostoma floridae* versus *Nematostella vectensis*, where Hieranoid found about three times more orthologs than InParanoid. The average number of proteins per ortholog group for this pair in Hieranoid 2 is 3.95, while in InParanoid it is 2.79. We found that the main source of Hieranoid ortholog pairs was an ortholog group with 718 proteins from B. *floridae* and 61 from *N. vectensis*. The biggest overlapping ortholog group in InParanoid has only 74 members. None of the proteins in this group has a known function, yet many contain zinc finger domains (Sonnhammer *et al.*, 1997). Many members of this family are unlikely to pass the coverage thresholds in InParanoid because they are fragments (36% of the sequences) or have large deletions and insertions in a large disordered region. However, because Hieranoid builds up multiple alignments iteratively, it can compensate for these lost regions and still infer them as orthologs.

6 Discussion

The second version of Hieranoid bundles a lot of improvements compared to version 1. We have improved multiple alignnment and consensus sequence building to make Hieranoid achieve substantially higher coverage of ortholog detection. Thanks to the new cluster mode, Hieranoid can now be parallelized on compute cluster. Algorithmic changes were motivated by the inspection of sequence alignments and consensus sequences, and validated by standardized orthology benchmarking. We show that two major changes in the core algorithm are responsible for best overall improvement. The combination of CS2 and MSA2 increases the coverage of ortholog detection to a level that approaches InParanoid while maintaining a similar accuracy. The new version of Hieranoid now ranks among the top performing methods in the benchmark. We have also made dozens of minor code improvements, e.g. related to thread synchronization in multithreaded mode, data parsing and handling of large alignments.

A major benefit of Hieranoid is its linearly scaling compute time which makes it an excellent tool for applications with large multispecies sets of proteomes. In general, *ab initio* orthology inference methods have a computational complexity that grows quadratically with the number of proteomes analyses, and with the availability of thousands of proteomes this has become a major stumbling block. This means that orthology inference can be considered a Big Data problem (Sonnhammer *et al.*, 2014), and that Hieranoid will be an important contribution to keep up with future growth of proteome data.

Another benefit is that Hieranoid places predicted orthologs in a phylogenetic context by providing trees with hierarchical groups of orthologs. Although these trees are not regular evolutionary trees as produced by other methods, e.g. in PylomeDB, they do reflect inferred orthology/paralogy relations. Because the ortholog trees may contain thousands of sequences it is often not feasible to make a full phylogenetic tree reconstruction of the members. They are topological trees, and instead of branch lengths representing sequence evolution, orthology confidence values are stored. Hieranoid 2 can use profiles instead of consensus sequences. Profiles can encode more information than a consensus sequence, hence they are expected to yield better accuracy and coverage (Patthy, 1987; Sonnhammer and Kahn, 1994). However, they are vastly slower, which in practise precludes them from being used instead of consensus sequences for all-versus-all similarity searching. Although profile searching against sequences has been accelerated (Wheeler and Eddy, 2013), this is not yet available for profile-profile searching needed by Hieranoid.

It is possible in Hieranoid to use profiles not for searching but just for scoring. Here candidate homologs are first found using the consensus sequence, but profiles are used to score multiple alignments against them. However, as this decreases the number of false positives and false negatives by just a small fraction (Schreiber and Sonnhammer, 2013), this mode was excluded from this evaluation.

The Quest for Orthologs benchmark service is a valuable resource for the orthology community, and is likely to become the standard to assess accuracy of new algorithms. One concern might be that developers are over-optimising their methods to these benchmarks. However, it is probably not possible to do this for all benchmarks at the same time. In the case of Hieranoid, we have not used the benchmarks to optimize any parameters, only to detect problem areas which were then improved based on sound principles and logic. We foresee that the benchmarks will also be useful for spotting problems with other algorithms in a similar way.

Funding

This work was funded by Swedish Research Council.

Conflict of Interest: none declared.

References

Altenhoff,A.M., and Dessimoz,C. (2009) Phylogenetic and functional assessment of orthologs inference projects and methods. *PLoS Comput. Biol.*, **5**, e1000262.

Altenhoff,A.M. *et al.* (2016) Standardized benchmarking in the quest for orthologs. *Nat. Methods*, **13**, 425–430.

Baldauf,S.L. (2003) Phylogeny for the faint of heart: a tutorial. *Trends Genet.*, **19**, 345–351.

Camacho,C. *et al.* (2009) BLAST+: architecture and applications. *BMC Bioinformatics*, **10**, 421.

Chen,F. *et al.* (2007) Assessing performance of orthology detection strategies applied to eukaryotic genomes. *PLoS One*, **2**, e383.

Dessimoz,C. *et al.* (2012) Toward community standards in the quest for orthologs. *Bioinformatics*, **28**, 900–904.

Edgar,R.C. (2004) MUSCLE: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics*, **5**, 113.

Edgar,R.C. (2010) Search and clustering orders of magnitude faster than BLAST. *Bioinformatics*, **26**, 2460–2461.

Gabaldón,T., and Koonin,E.V. (2013) Functional and evolutionary implications of gene orthology. *Nat. Rev. Genet.*, **14**, 360–366.

Huerta-Cepas,J. *et al.* (2016) ETE 3: Reconstruction, analysis and visualization of phylogenomic data. *Mol. Biol. Evol.*, **33**, 1635–1638.

Hulsen,T. *et al.* (2006) Benchmarking ortholog identification methods using functional genomics data. *Genome Biol.*, **7**, R31.

Jensen,R.A. (2001) Orthologs and paralogs - we need to get it right. *Genome Biol.*, **2**, 1002.1–1002.3.

O'Brien,K.P. *et al.* (2005) Inparanoid: a comprehensive database of eukaryotic orthologs. *Nucleic Acids Res.*, **33**, D476–D480.

Patthy,L. (1987) Detecting homology of distantly related proteins with consensus sequences. *J. Mol. Biol.*, **198**, 567–577.

Pearson,W.R. (2002). *Selecting the Right Similarity-Scoring Matrix*. John Wiley and Sons, Inc., West Sussex.

Remm,M. *et al.* (2001) Automatic clustering of orthologs and in-paralogs from pairwise species comparisons. *J. Mol. Biol.*, **314**, 1041–1052.

Schreiber,F., and Sonnhammer,E.L.L. (2013) Hieranoid: hierarchical orthology inference. *J. Mol. Biol.*, **425**, 2072–2081.

Sonnhammer,E.L., and Kahn,D. (1994) Modular arrangement of proteins as inferred from analysis of homology. *Protein Sci.*, **3**, 482–492.

Sonnhammer,E.L.L., and Hollich,V. (2005). A simple and robust protein sequence distance estimator. *BMC Bioinformatics*, **6**, 108.

Sonnhammer,E.L.L. *et al.* (1997) Pfam: a comprehensive database of protein domain families based on seed alignments. *Proteins Struct. Funct. Genet.*, **28**, 405–420.

Sonnhammer,E.L.L. *et al.* (2014) Big data and other challenges in the quest for orthologs. *Bioinformatics*, **30**, 2993–2998.

Stajich,J.E. *et al.* (2002) The bioperl toolkit: Perl modules for the life sciences. *Genome Res.*, **12**, 1611–1618.

Trachana,K. *et al.* (2011) Orthology prediction methods: a quality assessment using curated protein families. *Bioessays*, **33**, 769–780.

Wheeler,T.J., and Eddy,S.R. (2013) nhmmer: DNA homology search with profile HMMs. *Bioinformatics*, **29**, 2487–2489.

Wittwer,L.D. *et al.* (2014) Speeding up all-against-all protein comparisons while maintaining sensitivity by considering subsequence-level homology. *PeerJ.*, **2**, e607.