jmb Article

Hieranoid: Hierarchical Orthology Inference

Fabian Schreiber[1,2] and Erik L. L. Sonnhammer[1,2,3]

*1 - **Stockholm Bioinformatics Center,** Science for Life Laboratory, Box 1031, SE-17121 Solna, Sweden*
*2 - **Department of Biochemistry and Biophysics,** Stockholm University, SE-10691 Stockholm, Sweden*
*3 - **Swedish e-Science Research Center,** SE-10044 Stockholm, Sweden*

***Correspondence to Fabian Schreiber:** Wellcome Trust Sanger Institute, Wellcome Trust Genome Campus, Hinxton, Cambridgeshire CB10 1SA, UK. Fab.Schreiber@gmail.com*
http://dx.doi.org/10.1016/j.jmb.2013.02.018
Edited by A. Panchenko

Abstract

An accurate inference of orthologs is essential in many research fields such as comparative genomics, molecular evolution, and genome annotation. Existing methods for genome-scale orthology inference are mostly based on all-*versus*-all similarity searches that scale quadratically with the number of species. This limits their application to the increasing number of available large-scale datasets.

Here, we present Hieranoid, a new orthology inference method using a hierarchical approach. Hieranoid performs pairwise orthology analysis using InParanoid at each node in a guide tree as it progresses from its leaves to the root. This concept reduces the total runtime complexity from a quadratic to a linear function of the number of species. The tree hierarchy provides a natural structure in multi-species ortholog groups, and the aggregation of multiple sequences allows for multiple alignment similarity searching techniques, which can yield more accurate ortholog groups.

Using the recently published *orthobench* benchmark, Hieranoid showed the overall best performance. Our progressive approach presents a new way to infer orthologs that combines efficient graph-based methodology with aspects of compute-intensive tree-based methods. The linear scaling with the number of species is a major advantage for large-scale applications and makes Hieranoid well suited to cope with vast amounts of sequenced genomes in the future.

Hieranoid is an open source and can be downloaded at Hieranoid.sbc.su.se.

Introduction

The increasing availability of fully sequenced genomes provides a wealth of evolutionary information.[1] This information is brought to full light by studies ranging from, for example, the annotation of newly sequenced genomes[2] to comparative/functional genomics[3] and phylogenomics.[4] All these studies require the correct identification of homologs. If a pair of homologous proteins diverged as a result of a speciation event, we call them orthologs. If a homolog pair originated from a duplication event within the same species, then we call them paralogs.[5] The latter category can be further divided in relation to a given speciation event: inparalogs that arose from a duplication after the speciation event and outparalogs that arose from a duplication before the speciation.[6,7] The definition of orthology is purely evolutionarily and does not, by itself, include any implications about conserved function. However, the assumption that two orthologs are more likely to be functionally conserved than two outparalogs is commonly used.

Today's approaches for inferring orthology relationships can be roughly divided into graph-based and tree-based methods. Given a set of proteins from the species of interest, graph-based methods start with a similarity search using tools such as BLAST[8] and use bit score or *E*-value as a proxy for the evolutionary distance between protein pairs. From the distances between all protein pairs between complete proteomes, graph-based methods build ortholog groups using a variety of clustering criteria: best reciprocal hits (e.g., OMA,[9]

OrthoInspector,[10] InParanoid[11]), best triangular hit (e.g., COG,[12] eggNOG[13]), or Markov clustering (OrthoMCL[14]). In contrast, tree-based approaches use reconstructed protein family trees to infer orthologs. The inference is performed by a tree reconciliation or mapping of the protein tree to the corresponding species tree,[15,16] which gives a labeling of the internal nodes of the protein trees as either speciation or duplication events.[17] From this labeling, orthology and paralogy relationships are inferred†.

Both approaches have advantages and disadvantages. Tree-based approaches are perhaps a more intuitive way of assigning orthology and make use of a multiple sequence alignment, which is normally more reliable than pairwise alignments. Another advantage is that tree-based methods produce hierarchical ortholog groups in the form of trees. There are graph-based methods that provide hierarchical groups,[9,13,18] but they are the product of a postprocessing step rather than the results of a hierarchical inference algorithm. Unfortunately, the application of tree-based methods is limited by their high computational complexity and reliance on correct multiple sequence alignments and protein trees,[19,20] which makes them unsuitable for large (numbers of) protein families. Graph-based methods are computationally less demanding and easier to automate, and implementations handling large sets of sequences already exist. However, they are not hierarchical and group proteins from different species into single flat group, which is not a natural representation. Furthermore, they have at least N^2 computational complexity, which although much better than most tree-building methods, still poses a problem for hundreds of species.

Recent benchmarks have compared different orthology inference methods (see A[21–24]). In general, graph-based methods yielded lower error rates than tree-based methods. The most recent of such benchmarks is orthobench.[21] It is the most comprehensive reference-tree-based benchmark and the only one available for download. In this benchmark, graph-based methods produced fewer orthology relationships than tree-based but produced a lower number of falsely assigned orthology relationships. The popular InParanoid method that performed among the best in most recent benchmarks[22–24] was not used in the orthobench benchmark evaluation as they included only methods that produce multi-species ortholog groups. A multiple-species version of InParanoid called MultiParanoid exists,[25] but it is not hierarchical and only suited to groups of equally distant species.

We identify the following areas in need of improvement:

1. Scalability: Reducing the computational complexity of similarity searches;
2. Accuracy: More reliable orthology inference from multiple alignments; and
3. Multi-species InParanoid: Extending the InParanoid algorithm to infer hierarchical multi-species ortholog groups.

To meet these needs, we here present Hieranoid, a new method that infers orthologs between multiple species by progressively applying the pairwise InParanoid method.[11] The progressive idea takes its cue from the "progressive alignment" approach.[26] Orthology relationships are inferred at the nodes of a bifurcating guide tree, the species tree. Using a hierarchical progressive approach, Hieranoid combines the advantages of graph-based methods in that it is computationally less expensive and of tree-based methods in that it produces tree-structured hierarchical groups.

This progressive approach results in a linear computational complexity and exploits valuable evolutionary information contained in the guide tree. Results on the orthobench benchmark show that Hieranoid yields lower total levels of false and missing orthology assignments than other methods. The reduced computational complexity makes Hieranoid attractive for the analysis of very large datasets, which is timely given that thousands of genomes are currently being sequenced.

Results

We have adopted the concept of progressive sequence alignment to infer hierarchical ortholog groups along a guide tree. In order to assess the capability of our new method Hieranoid, we examined its performance in two different scenarios, comparing it to InParanoid and to other orthology inference methods. The default mode of Hieranoid is to use consensus sequences; if not specified, this is the mode used. We focused on answering the following questions:

1 How much do inferred orthology relationships from Hieranoid overlap with those from InParanoid? (*Overlap dataset*)
2 What is the runtime complexity of Hieranoid compared to InParanoid? (*Runtime dataset*)
3 What is the accuracy of Hieranoid compared to other orthology inference methods? (*orthobench benchmark*)

Comparison to InParanoid

Although Hieranoid uses the InParanoid method for inferring orthologs, the two methods differ in which underlying similarity search tools they use by default and the fact that Hieranoid performs a

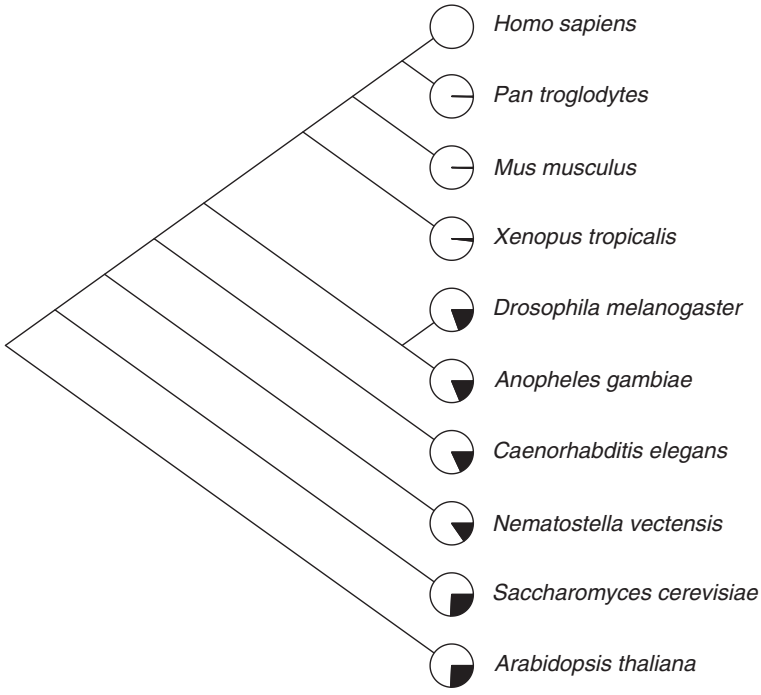

Fig. 1. Comparison of ortholog inferences from InParanoid and Hieranoid consensus. The pie charts along the guide tree represent the agreement of inferred orthologs for human *versus* other species comparisons. The whiter the pie charts, the larger this agreement. The tree shown in the picture was used as the guide tree for the Hieranoid analysis.

progressive analysis. To see how much these two factors influence orthology inference, we used the 10 species *overlap dataset*.‡ A comparison between the InParanoid and Hieranoid results is shown in Fig. 1. The species are ordered by their evolutionary distance to human. For each human *versus* other species comparison, we counted the fraction of matching pairwise orthology assignments between Hieranoid and InParanoid relative to the union of all their orthology assignments. This percentage is depicted as white and black parts, respectively, of the pie charts at every leaf of the tree.

The overlap for human–chimpanzee orthologs is 99.7% and that for human–mouse is 99.5%. For distantly related yeast and thale cress, the overlap is 74.2% and 74.2%, respectively. The human–chimpanzee comparison results in nearly identical groups. This is because the Hieranoid and InParanoid are identical for a pairwise comparison given that the two species are sister taxa in the Hieranoid guide tree. The small difference is due to the use of USEARCH instead of BLAST and comes with at large decrease in runtime. For all other comparisons, most of the difference can be explained by the way Hieranoid and InParanoid infer orthologs. While InParanoid compares distant species directly, Hieranoid does so indirectly via the ancestral pseudospecies inferred at the internal nodes of the guide tree.

Runtime complexity of Hieranoid and InParanoid

One of the biggest advantages of Hieranoid is that it scales linearly in computational complexity with the number of species, in contrast to other methods that perform all-*versus*-all searches and scale quadratically (Fig. 2). To test this in practice, we bench-

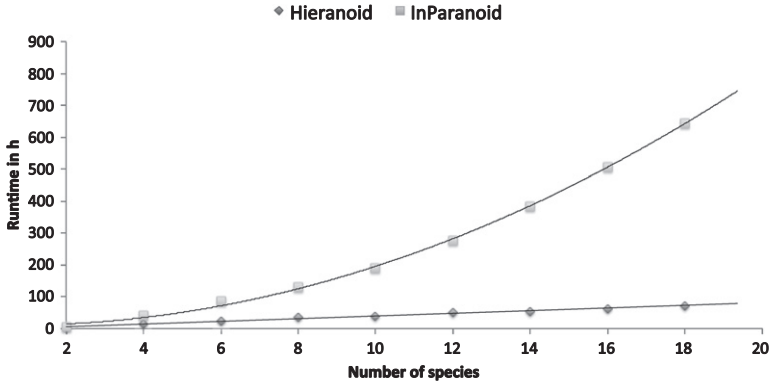

Fig. 2. Hieranoid *versus* InParanoid runtime comparison. Runtimes for Hieranoid consensus and InParanoid plotted for 2–20 input species, using the most diverse species for subsets. An exponential trendline and a linear trendline were fitted to the data for InParanoid and Hieranoid, respectively. While the runtime is similar for a small number of species, it increases linearly for Hieranoid but quadratically for InParanoid. As Hieranoid uses a guide tree, it performs $n-1$ pairwise comparisons, with n being the number of species. In contrast, InParanoid performs $n(n-1)/2$ comparisons. The larger the number of input species becomes, the more of a problem this will be for quadratically scaling methods like InParanoid.

marked the runtime of Hieranoid and InParanoid—as a representative of methods using all-*versus*-all similarity searches—on the *runtime dataset*. We used varying numbers of input species to see how the increase in runtime is related to an increase in input size.

Performance comparison with other orthology inference methods

Taking *orthobench* as the reference benchmark, we compared Hieranoid to six other ortholog databases§. As orthology is a property of protein pairs,[5] we based our performance evaluation on protein pairs.

Our main interest is to see how well the methods correctly infer orthologous relationships and detect missing orthologs. We used the following counting scheme:

1. For each true pair in an *orthobench* group, we count how often this pair has not been inferred by one of the methods (false negative), given that both sequences were included in the input data of the database.
2. For each ortholog group in a database, we count how often a protein pair is inferred as being orthologous, but is not orthologous in the benchmark dataset (false positive), given that both sequences are in included in the orthobench input data.

Furthermore, we were interested to see the effects of different approaches within the Hieranoid method, that is, using profiles instead of consensus sequences and including an outgroup species. To this end, we included four versions of Hieranoid.

Figure 3 shows percentages of false positive and negative orthology assignments for each of the tested methods. The methods' general performance is in line with what was reported in the *orthobench* benchmark paper. There is one group of methods with a low level of false negatives but a high level of false positives (OrthoMCL, TreeFam) and another group with the reverse trend (Hieranoid, InParanoid, OMA, OrthoDB). eggNOG had about equal levels of both types of errors. As the authors have noted, this benchmark dataset is focusing on orthology relationships that are difficult to infer. This leads to a poor performance of all tools.

Looking at our tool Hieranoid, it misses more orthology relationships than eggNOG, OrthoMCL, and TreeFam and makes more false positives than OMA. However, as can be seen from the stacked error bars in Fig. 3, Hieranoid shows the overall lowest error rate.

Hieranoid with profile versus consensus sequences

In contrast to consensus sequences, where a multiple alignment is represented as a single sequence, profile HMMs (*h*idden *M*arkov *m*odel) capture an alignment's whole evolutionary information. This should have an advantage in cases where the underlying alignments are diverse, as in the case of Hieranoid. This is supported by our results. Hieranoid shows a slightly better performance when using profile HMM searches as compared to consensus sequences. The fact that the difference between the two is relatively small can be explained by the use of a relatively small and taxonomically narrow dataset (only metazoans).

Hieranoid using outgroup species or not

When Hieranoid infers orthologs, it builds ortholog groups from reciprocally best hits. If, however, one of those two hits has a better hit in an outgroup species,

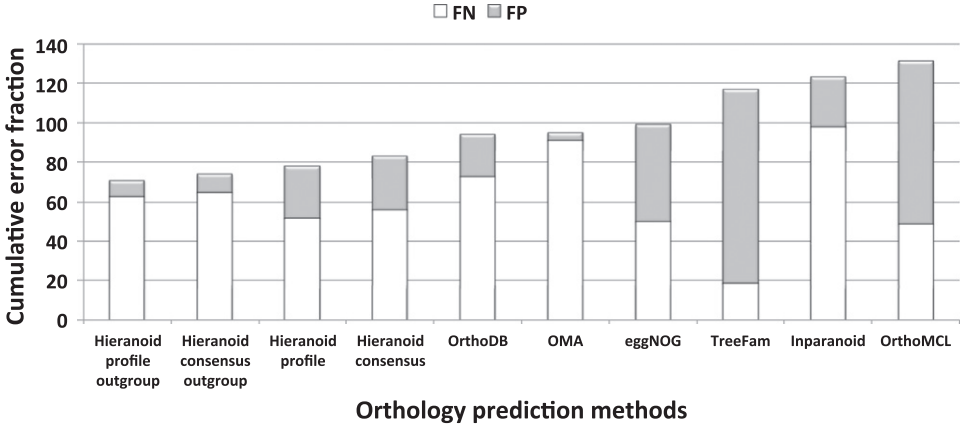

Fig. 3. Results from the orthobench benchmark. The percentages of the two types of errors [FP, false positives (gray); FN, false negatives (white)] are presented as stacked bars for the different orthology databases. The lower the stacks, the better the method.

this ortholog group will be discarded. This allows us to detect gene losses in one of the two species. Although it increases the runtime (additional similarity searches for species A-C and B-C), it drastically reduces the number of false positive assignments in the benchmark (see Fig. 3). The level of false negatives is also increased but to a lesser extent, giving a lower overall error rate.

Discussion

We have adapted the concept of progressive sequence alignment to the area of orthology inference. Our new method Hieranoid performs pairwise orthology inferences using the well-established InParanoid method along a guide tree. We addressed the following three shortcomings mentioned in Introduction and provided reasonable solutions:

1. Scalability: Reducing the computational complexity of similarity searches

The use of a guide tree allows Hieranoid to perform a significantly smaller number of comparisons/inference steps compared to previous methods. Instead of the usual all-*versus*-all similarity searches that require $n(n-1)/2$ proteome comparisons, Hieranoid just uses $n-1$ comparisons that is equal to the number of inner nodes in the guide tree. Our test using the runtime dataset confirmed this in an experimental setup.

2. Accuracy: More fine-grained orthology inference by building hierarchical groups

In contrast to most other graph-based methods, Hieranoid infers hierarchical ortholog groups. The advantage of hierarchical groups over non-hierarchical or flat groups is—besides the information that all proteins from different species are orthologs—that they include evolutionary information as to which proteins are more closely related to each other. Hierarchical groups can be seen as trees with inparalogs represented as multifurcations. They are a more accurate representation of how a group of orthologs evolved. While orthologs evolved from a common ancestor to leaves of a species tree, Hieranoid takes this tree and performs an analysis in the opposite direction, from the leaves to the root. Our results on the recently published orthobench benchmark show that this idea shows improved accuracy in ortholog inference. Given that the orthobench benchmark consists of orthology relationships that are hard to infer, all tested methods have problems inferring them. Our tool Hieranoid performs best in this benchmark. The use of profiles over consensus sequences leads to slightly more accurate results. The difference in accuracy between the two is likely to become bigger when looking at a higher number of or more diverse species. The use of an outgroup species, however, is independent of the number of used species. Our orthobench results show that its use leads to more accurate ortholog inferences.

3. Multi-species InParanoid: Extending the InParanoid algorithm to infer multi-species ortholog groups

The Hieranoid algorithm progresses iteratively along the guide tree. Applying the InParanoid algorithm to an initial species–species comparison results in pairwise ortholog groups. Moving along the guide tree toward the root, these groups will be expanded every time additional orthologs in another single species or pseudospecies can be found. This way, we extend the previously reported good performance of InParanoid to infer multiple-species ortholog groups. A previous multi-species framework for InParanoid called MultiParanoid[25] has the drawback that it does not aggregate the ortholog groups hierarchically and, therefore, only gives reasonable groups for species that are approximately equally distant from each other.

The Hieranoid approach assumes that the actual gene histories are reasonably approximated by the used species tree. This is a reasonable assumption for multicellular eukaryotes, where lateral gene transfer events are essentially non-existent. However, this is not the case for prokaryotes and some unicellular eukaryotes. As a consequence, the improved accuracy observed for Hieranoid is likely to be more pronounced for eukaryotic species.

InParanoid, and therefore also Hieranoid, computes the orthology graph from whole-length protein alignment and, by default, requires a minimum length overlap of 50%. This means that orthologs with extensive domain rearrangements may be missed. Future plans include the use of domain information for improved treatment of domain orthology.

Further advantages are the modular implementation of Hieranoid and its availability as an open source tool.‖ We developed Hieranoid in a modular way so that one can easily replace all components of the system, for example, using another similarity search tool or changing the orthology inference algorithm. This way, Hieranoid can function as a framework for future developments of orthology inference tools.

Non-comparability of bit scores from different similarity search tools

When Hieranoid predicts orthologs for the inner node HMW (picture 1), it has to do a profile–profile (HM–HM), profile–sequence (HM–W and W–HM), and sequence–sequence (W–W) searches. The

naive idea to simply mix profile–profile searches for HM–HM with profile–sequence searches for HM–W and W–HM and sequence–sequence searches for W–W fails, as different tools for each of the comparisons report scores that are different and therefore not directly comparable (unpublished data). To guarantee that scores returned by all similarity searches for a given inner node are comparable, we decided to implement both an accurate search based on HMM comparisons and a faster search using consensus sequences.

Profile–profile comparisons

The heart of our new algorithm is the use of profile HMMs to extend the pairwise InParanoid to a multi-species version. We tested different methods for doing profile–profile comparisons (e.g., Refs. 27 and 28). Most methods were discarded because their computational complexity was too high. This left us with HHSearch and HHBlits, an extended HHSearch version that manifests its strength when it comes to large databases of HMMs. However, both of these are too slow to handle the volumes of HMM searching in Hieranoid. Therefore, we use BLAST of consensus sequences as a prefilter for HMM–HMM comparisons, which effectively leaves us with very small HMM libraries. For those small libraries, HHBlits is actually slower due to the overhead to build indices for fast searching of the HMM library, making HHSearch our choice of HMM–HMM alignment tool.

Why is Hieranoid more accurate than InParanoid?

Both Hieranoid and InParanoid use the alignment bit score as a proxy for evolutionary distance. This is a reasonably accurate proxy for the comparison of closely related species, but for distantly related species, the bit score is less accurate. For instance, when inferring orthology relationships between human and yeast, InParanoid uses the bit score of human and yeast proteins as their evolutionary distance. In contrast, Hieranoid uses the bit score of a yeast protein and a protein present in the last common metazoan ancestor as their evolutionary distance (see Fig. 4). Using intermediate ancestral sequences for scoring distant proteins likely better estimates the true evolutionary distance and can partly explain why Hieranoid is more accurate than InParanoid.

As a final remark, our benchmark results show that our tree-guided graph-based approach overall outperforms other methods that are classical graph-based or tree-based methods. Some methods have a lower false positive rate and some have a lower false

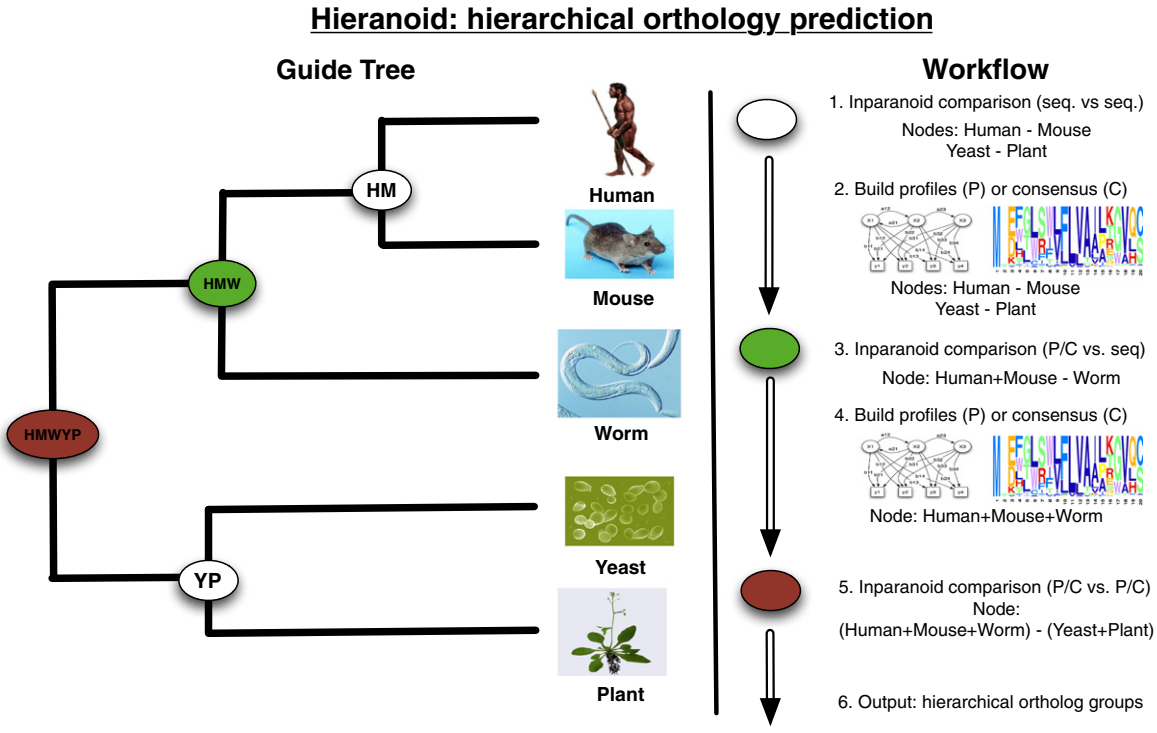

Fig. 4. Example tree and workflow of Hieranoid analysis. The guide tree has five species [human (H), mouse (M), worm (W), yeast (Y), and plant (P)] and four inner nodes (HM, HMW, YP, and HMWYP). Traversing the tree from the leaves to the root gives the order in which Hieranoid infers orthologs. Each tree node is colored as the corresponding step in the workflow.

negative rate than Hieranoid, at the expense of a much higher error rate of the other type. Our "hybrid" tree/graph method strikes a better compromise between these two types of errors. This suggests that there might be scope to improve other graph-based tools as well using a species guide tree.

Materials and Methods

The progressive alignment approach has been used extensively for building multiple sequence[26] and gene order alignments.[29] This approach divides the computationally complex problem of aligning multiple sequences into a set of pairwise alignment problems. The sequence of pairwise comparisons is given by a guide tree that connects leaf sequences progressively until the root node. It starts at the leaves with sequence-to-sequence alignments and switches to sequence-to-profile and profile-to-profile alignments for aligning groups of sequences as it approaches the root of the tree.

We have adopted this popular and efficient approach to the problem of orthology inference between multiple species. The pairwise alignment step was replaced by a pairwise orthology inference step using the InParanoid algorithm. The order of comparisons is given by a user-provided guide tree, which should represent the known species tree.

Sequences and a tree as input

As input, our method requires a set of proteome sequences from the species under study in either FASTA or SeqXML[30] format and a guide tree connecting the species. This tree either can be derived from, for example, the NCBI (*N*ational *C*enter for *B*iotechnology *I*nformation) taxonomy[31] or can be user defined. The guide tree should be in Newick format, with leaves labeled with the same string as the proteome file names.

Progressive orthology inference strategy

The order of pairwise comparisons is determined by a given guide tree. This tree is traversed from the leaves to the root with the leaves being the species under study and inner nodes being hypothetical ancestors or pseudospecies. Each such inner node represents the results of the pairwise orthology inference of the two daughter nodes, that is, a set of inferred ortholog groups. The possible pairwise comparisons are as follows:

1. a pair of species,
2. a species and a pseudospecies, or
3. two pseudospecies.

To clarify how Hieranoid works, we provide an example in Fig. 4.

Hieranoid performs the following pairwise comparisons:

1. inner node HM→orthology inference: human–mouse
2. inner node YP→orthology inference: yeast–plant
3. inner node HMW→orthology inference: HM–worm
4. inner node HMWYP→orthology inference: HMW–YP

The Hieranoid analysis finishes once orthologs between pseudospecies HMW and YP have been inferred for the root node HMWYP. Hieranoid outputs the results as hierarchical multiple-species ortholog groups. These are stored in Newick tree format, but we are using the branch length field to store the inparalog score. However, the branch lengths do not reflect evolutionary distance but instead the confidence of being an inparalog. The groups can be viewed using a tree viewer (e.g., FigTree¶)

We will now take a closer look at how orthologs are inferred at each inner node.

Each pairwise orthology inference consists of the following two parts:

1. building an initial set of homologs using similarity search
2. the actual orthology inference

While the second step is the same for all types of inner nodes, they differ in how the initial set of homologs is calculated.

Building an initial set of homologs

The process of building an initial set of homologs is adapted from InParanoid. First, InParanoid performs a set of all-*versus*-all BLAST searches to estimate the evolutionary distance between all pairs of proteins of the two species ("human-mouse" and "mouse-human") and within each species ("human-human" and "mouse-mouse"). The latter is required to infer inparalogs based on the assumption that the distance between inparalogs within each species should be smaller than that between orthologs in different species, within an ortholog group. InParanoid uses the BLAST bit score as a proxy for evolutionary distance. The most recent version of InParanoid[11] introduced a more stringent set of filters for fragmentary matches during the sequence comparison step and adaptations to reduce false positives matches due to low complexity regions.[32] These postprocessing filters applied to the similarity search results lead to high-quality orthologs with very few false positives.

Hieranoid also uses bit scores as an evolutionary proxy and the set of postprocessing filters. However, it may replace the underlying similarity search tool depending on the type of comparison. The types that can occur at the inner nodes of the guide tree during the Hieranoid analysis are as follows: a comparison of species *versus* species (sequence *versus* sequence), species *versus* pseudospecies (sequence *versus* profile), and pseudospecies *versus* pseudospecies (profile *versus* profile).

Species *versus* species

For the similarity search between two species, that is, two sets of sequences from "human-mouse" or "yeast-plant", Hieranoid performs a regular InParanoid comparison but by default replaces BLAST by the less time-consuming USEARCH method.[33] USEARCH has been

shown to be orders of magnitudes faster than BLAST with equal sensitivity. The output is a list of ortholog groups that satisfy the default InParanoid conditions.

Species *versus* pseudospecies and pseudospecies *versus* pseudospecies

The next step is the inference of orthologs between the pseudospecies HM and worm (W). The pseudospecies HM is the total set of inferred ortholog groups between human and mouse, plus all orphan genes from H and M. In this case, the similarity search of HM against worm estimates the similarity between an ortholog group, that is, a group of sequences (HM), and single sequences (W). It involves the following comparisons: HM–W, W–HM, HM–HM, and W–W.

Hieranoid offers two approaches that differ in speed and accuracy: One is based on consensus sequences and the other one is based on profile HMMs.

Consensus sequences

Instead of using the whole sequence information, a consensus sequence is calculated for each ortholog group. Here, the consensus is the sequence of residues with the highest occurrence frequency for each column in the ortholog group alignment. In case that the occurrence frequency is equal for different residues, one residue is selected at random in the current implementation. The benefit of using a consensus sequence is that sequence–sequence comparisons using USEARCH can be used. This results in immensely reduced computational complexity of the similarity search as compared to the profile-based search. A drawback is that, by selecting a single residue for each column, potentially valuable evolutionary information is lost. This might lead to a higher number of false negatives in the similarity search, which is aggravated for bigger and less similar ortholog groups.

Profile HMMs

The most accurate way to compare two ortholog groups is to use profiles built from the alignments of ortholog groups and use a profile–profile search for each of the four proteome comparisons. Such profile HMMs use a position-specific system to capture information about the frequency of nucleotides or amino acids at each column in a multiple sequence alignment.[8,34] Using these alignments, Hieranoid builds HMMs using hmmbuild with default parameters from the HMMER package[35] and HHSearch[36] to perform profile–profile searches. Note that, due to the modular implementation of Hieranoid, alternative methods can be easily plugged in. As efficient implementations are lacking, the use of profile–profile comparison methods is limited to datasets with small numbers of sequences. Hieranoid reduces the number of required profile–profile searches by performing an initial sequence–sequence search using consensus sequences to get a list of potential hits. A second search then is a profile–profile search between the query and the top hits. As profile–profile searching is only used to find the best cross-species match, we found it sufficient to search the top 10 hits.

Orthology inference

Once an initial set of putative homologs is built, Hieranoid infers orthologs and inparalogs using the existing InParanoid algorithm. The clustering step of the orthology inference is independent of the method for inner node comparison, as both the consensus and the profile HMMs approaches result in a list of pairwise protein distances. The basic idea of the orthology inference in InParanoid is that proteins that are each other's best reciprocal hits form the seeds ortholog clusters. For example, protein H1 from human has the reciprocally highest bit score to protein M1 from mouse. H1 and M1 will be the seed orthologs of an ortholog group. Inparalog sequences are added if their distance to the seed ortholog from the same species is shorter than to the seed ortholog in the other species. If there is a protein H2 with a higher bit score to H1 than to M1, then H2 will be added as an inparalog to ortholog group H1M1 (see Ref. 11 for more details).

InParanoid comparison and orthobench benchmark

We compiled two different datasets called the overlap dataset and the runtime dataset to compare Hieranoid with InParanoid in terms of agreement of inferred orthology relationship and runtime over varying sizes of input species. Additionally, we used the recently published orthobench benchmark dataset[21] to compare Hieranoid with other orthology inference methods.

Overlap dataset

The overlap dataset consists of 10 species from version 5 (2011_04) of the Reference Proteome Project[37] with a taxonomic distribution ranging from *Homo sapiens* to *Saccharomyces cerevisiae*. The corresponding species tree was extracted from the NCBI taxonomy and used as the guide tree for the Hieranoid analysis. Both Hieranoid and InParanoid were run with default settings. For Hieranoid, we used the consensus sequence option. Pre-computed results were downloaded from the InParanoid Website[a]. Hieranoid was run on a two-quadcore Intel Harpertown 2.66-GHz central processing unit with 8-GB random access memory.

Runtime dataset

We extended the 10 species dataset to 20 species by selecting a set of additional species with a similar taxonomic distribution. We kept the program settings unchanged. To allow a fair comparison, we run both Hieranoid and InParanoid using USEARCH on a single core of the same hardware as above on even subsets between 2 and 20 species. Runtime was measured as total user time.

Orthobench benchmark

The orthobench dataset consists of 70 manually curated protein families from 12 metazoan species (see Ref. 21 for the full list of species). The families were selected to represent a wide spectrum of biological complexity and to serve as good examples of potential error sources for orthology

inference tools. The families differ in size, rate of evolution, alignment quality, and domain architecture complexity. The authors compared their in-house tool eggNOG to four other methods (OrthoDB,[18] OMA,[9] OrthoMCL,[14] and TreeFam[38]).

For the Hieranoid analysis, we downloaded the proteomes of the 12 species from Ensembl v60[39] and used the corresponding NCBI taxonomy tree as a guide tree.

We downloaded the most recent orthology inferences and corresponding input data from the other databases (eggNOG: meNOGs, version 2.0; OMA: May 2011; OrthoDB: version 4; TreeFam: release 7.0; OrthoMCL: version 5; InParanoid: version 7). We mapped all transcript or protein IDs used by the orthology inference methods to the corresponding protein of the orthobench dataset using a mapping file provided by.[21] IDs that could not be mapped to proteins in the orthobench dataset were not counted as false positives in order to allow a fair comparison. Likewise, proteins in orthobench not mappable to the input data of a database were not counted as false negatives if absent from an orthobench group.

Acknowledgements

We thank Kalliopi Trachana for help with the *orthobench* benchmark dataset and Michael Remmert for help with HHSearch. This work was supported by the Wenner-Gren Foundations.

Received 6 December 2012;
Received in revised form 13 February 2013;
Accepted 16 February 2013
Available online 26 February 2013

Keywords:
Orthology inference;
inParanoid;
guide tree;
profile-profile comparison

† For a full list of orthology methods, see http://questfororthologs.org/orthology_databases

‡ The Hieranoid results for this dataset are available at http://sonnhammer.sbc.su.se/download/Hieranoid/

§ The orthobench results can be downloaded at http://sonnhammer.sbc.su.se/download/Hieranoid/

‖ While most other tested methods are not available for download, Hieranoid can be downloaded for free at https://github.com/fabsta/Hieranoid and http://software.sbc.su.se/cgi-bin/request.cgi?project=hieranoid

¶http://tree.bio.ed.ac.uk/software/figtree/

[a] http://InParanoid.sbc.su.se/download/Reference_Proteomes/

References

1. Koonin, E. & Galperin, M. (2003). *Sequence–Evolution–Function.* Kluwer Academic Publishers, Boston, MA.
2. Eisen, J. (1998). Phylogenomics: improving functional predictions for uncharacterized genes by evolutionary analysis. *Genome Res.* **8**, 163.
3. Kuzniar, A., Vanham, R., Pongor, S. & Leunissen, J. (2008). The quest for orthologs: finding the corresponding gene across genomes. *Trends Genet.* **24**, 539–551.
4. Eisen, J. A. & Fraser, C. M. (2003). Phylogenomics: intersection of evolution and genomics. *Science*, **300**, 1706–1707.
5. Fitch, W. M. (1970). Distinguishing homologous from analogous proteins. *Syst. Zool.* **19**, 99–113.
6. Koonin, E. (2005). Orthologs, paralogs, and evolutionary genomics. *Annu. Rev. Genet.* **39**, 309–338.
7. Sonnhammer, E. L. L. & Koonin, E. V. (2002). Orthology, paralogy and proposed classification for paralog subtypes. *Trends Genet.* **18**, 619–620.
8. Altschul, S., Madden, T., Schaffer, A. & Zhang, J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.* **25**, 3389–3402.
9. Altenhoff, A. M., Schneider, A., Gonnet, G. H. & Dessimoz, C. (2011). OMA 2011: orthology inference among 1000 complete genomes. *Nucleic Acids Res.* **39**, D289–D294.
10. Linard, B., Thompson, J., Poch, O. & Lecompte, O. (2011). OrthoInspector: comprehensive orthology analysis and visual exploration. *BMC Bioinformatics*, **12**, 11.
11. Ostlund, G., Schmitt, T., Forslund, K., Köstler, T., Messina, D. N., Roopra, S. *et al.* (2010). InParanoid 7: new algorithms and tools for eukaryotic orthology analysis. *Nucleic Acids Res.* **38**, D196–D203.
12. Tatusov, R. L., Fedorova, N. D., Jackson, J. D., Jacobs, A. R., Kiryutin, B., Koonin, E. V. *et al.* (2003). The COG database: an updated version includes eukaryotes. *BMC Bioinformatics*, **4**, 41.
13. Powell, S., Szklarczyk, D., Trachana, K., Roth, A., Kuhn, M., Muller, J. *et al.* (2012). eggNOG v3.0: orthologous groups covering 1133 organisms at 41 different taxonomic ranges. *Nucleic Acids Res.* **40**, D284–D289.
14. Chen, F., Mackey, A., Stoeckert, C., Jr. & Roos, D. (2006). OrthoMCL-DB: querying a comprehensive multi-species collection of ortholog groups. *Nucleic Acids Res.* **34**, 363–368.
15. Mirkin, B., Muchnik, I. & Smith, T. F. (1995). A biologically consistent model for comparing molecular phylogenies. *J. Comput. Biol.* **2**, 493–507.
16. Page, R. D. & Charleston, M. A. (1997). From gene to organismal phylogeny: reconciled trees and the gene tree/species tree problem. *Mol. Phylogenet. Evol.* **7**, 231–240.
17. Zmasek, C. M. & Eddy, S. R. (2001). A simple algorithm to infer gene duplication and speciation events on a gene tree. *Bioinformatics*, **17**, 821–828.
18. Waterhouse, R. M., Tegenfeldt, F., Li, J., Zdobnov, E. M. & Kriventseva, E. V. (2013). OrthoDB: a hierarchical catalog of animal, fungal and bacterial orthologs. *Nucleic Acids Res.* **41**, D358–D365.
19. Liu, K., Raghavan, S., Nelesen, S., Linder, C. R. & Warnow, T. (2009). Rapid and accurate large-scale

coestimation of sequence alignments and phylogenetic trees. *Science*, **324**, 1561–1564.
20. Thompson, J. D., Linard, B., Lecompte, O. & Poch, O. (2011). A comprehensive benchmark study of multiple sequence alignment methods: current challenges and future perspectives. *PLoS One*, **6**, e18093.
21. Trachana, K., Larsson, T. A., Powell, S., Chen, W.-H., Doerks, T., Muller, J. & Bork, P. (2011). Orthology prediction methods: a quality assessment using curated protein families. *BioEssays*, **33**, 769–780.
22. Altenhoff, A. & Dessimoz, C. (2009). Phylogenetic and functional assessment of orthologs inference projects and methods. *PLoS Comput. Biol.* **5**, e1000262.
23. Alexeyenko, A., Lindberg, J., Pérez-Bercoff, Ĺ. & Sonnhammer, E. (2006). Overview and comparison of ortholog databases. *Drug Discovery Today: Technol.* **3**, 137–143.
24. Chen, F., Mackey, A., Vermunt, J. & Roos, D. (2007). Assessing performance of orthology detection strategies applied to eukaryotic genomes. *PLoS One*, **4**, 3383.
25. Alexeyenko, A., Tamas, I., Liu, G. & Sonnhammer, E. L. L. (2006). Automatic clustering of orthologs and inparalogs shared by multiple proteomes. *Bioinformatics*, **22**, e9–e15.
26. Feng, D. F. & Doolittle, R. F. (1987). Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J. Mol. Evol.* **25**, 351–360.
27. Sadreyev, R. I., Tang, M., Kim, B.-H. & Grishin, N. V. (2009). COMPASS server for homology detection: improved statistical accuracy, speed and functionality. *Nucleic Acids Res.* **1**, 1–5.
28. Madera, M. (2008). Profile Comparer: a program for scoring and aligning profile hidden Markov models. *Bioinformatics*, **24**, 2630–2631.
29. Rödelsperger, C. & Dieterich, C. (2010). CYNTENATOR: progressive gene order alignment of 17 vertebrate genomes. *PLoS One*, **5**, e8861.
30. Schmitt, T., Messina, D. N., Schreiber, F. & Sonnhammer, E. L. L. (2011). Letter to the editor: SeqXML and OrthoXML: standards for sequence and orthology information. *Brief. Bioinform.* **12**, 485–488.
31. Sayers, E. W., Barrett, T., Benson, D. A., Bolton, E., Bryant, S. H., Canese, K. *et al.* (2010). Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.* **38**, D5–D16.
32. Forslund, K. & Sonnhammer, E. L. L. (2009). Benchmarking homology detection procedures with low complexity filters. *Bioinformatics*, **25**, 2500–2505.
33. Edgar, R. C. (2010). Search and clustering orders of magnitude faster than BLAST. *Bioinformatics*, **26**, 2460–2461.
34. Eddy, S. R. (2009). A new generation of homology search tools based on probabilistic inference. *Genome Inform.* **23**, 205–211.
35. Finn, R. D., Clements, J. & Eddy, S. R. (2011). HMMER web server: interactive sequence similarity searching. *Nucleic Acids Res.* **39**, W29–W37.
36. Söding, J. (2005). Protein homology detection by HMM–HMM comparison. *Bioinformatics*, **21**, 951–960.
37. Gabaldón, T., Dessimoz, C., Huxley-Jones, J., Vilella, A., Sonnhammer, E. & Lewis, S. (2009). Joining forces in the quest for orthologs. *Genome Biol.* **10**, 403.
38. Ruan, J., Li, H., Chen, Z., Coghlan, A., Coin, L. J. M., Guo, Y. *et al.* (2008). TreeFam: 2008 Update. *Nucleic Acids Res.* **36**, D735–D740.
39. Flicek, P., Amode, M. R., Barrell, D., Beal, K., Brent, S., Chen, Y. *et al.* (2010). Ensembl 2011. *Nucleic Acids Res.* **39**, D800–D806.