## 

MDSC 507

Written Summary

Abdullah Zubair

UCID: 30113730

February 2023

Orthologous genes or orthologs are a type of homologous genes that occur in different species and evolved due to speciation from a common ancestral gene1. Orthologs are generally assumed to have the same biological functions within different species and as such the identification of orthologs is a crucial aspect in the field of comparative genomics1,2. Within the field of comparative genetics, the prediction of orthologs is frequently utilized as an initial step towards identifying the gene or genes in a novel species that perform a similar function as a gene in another species, whose role has been established3. As such there has been a multitude of software tools developed with the aim of providing a reliable prediction of orthologs with a given a set of either nucleotide or protein gene sequences4-7. These programs, which will be referred to as ‘ortholog inference tools’ have existed for years, however, with more recent advances in technology newer tools have been developed that claim to be both faster and more accurate3. Comparisons of ortholog inference tools are rare and are dated8. As such the advantages and disadvantages of a given tool as compared to other existing tools are currently unknown. The goal of this summary is to identify and analyze four ortholog inference tools and to note not only the mechanism behind the tools but also the potential advantages each tool could possibly provide. With the analysis of the tools, a hypothesis can be developed as to which tool will provide good accuracy whilst maintaining low use of memory and CPU resources and having the shortest runtime. Additionally, a quantitative description of the Culicidae ortholog database, OrthoDB, will also be provided as this database will be used for future testing of the inference tools9.

One method of developing ortholog inference tools involves the use of profile hidden Markov models (pHMMs) which are derived from multiple sequence alignments (MSAs)4. Orthograph is one such tool that builds upon and improves another tool, HaMStR4. Developed as an open-source tool, Orthograph utilizes a best reciprocal hit approach to identify orthologs from either transcriptomic or genomic sequence data4. Orthograph work to determine the orthologous genes by first taking in reference gene sets, as input data4. For the purposes of the future analysis of this tool, OrthoDB will be used to obtain these gene sets9. Orthologous genes are then clustered from the reference gene sets4. To construct pHMMs, MSAs are performed on each orthologous group (OG) by aligning the sequences4. The pHMMs are then used to search for potential orthologs in the sequence of interest, which is also taken as an input, on an amino acid level4. The results of these searches are then stored in a relational SQL database as is referred to as the forward search4. In a reverse search, the target amino acid sequence sections, matching the pHMM from the forward search, are used as a query to search for all genes within the RGS using BLAST4. The results of this reverse search are also stored in the relational database4. Finally, a combination of best reciprocal hit pairs is conducted to assign the sequence to an OG4. Specifically, the forward search results are sorted by descending alignment bit score and each corresponding reverse search result is evaluated in order of descending alignment bit score for the forward search4. If the best reverse search hit is part of the OG that the pHMM for the forward search is based on, the sequence of interest is assigned to the OG and marked as such to prevent further assignment, a step which differs from other tools4. This evaluation process is then repeated for each entry in the database4. Orthograph then also conducted post-processing steps on sequences assigned to OGs4.

Orthograph’s approach for identifying orthologs has some potential advantages that could allow for more accurate orthology prediction as compared to other tools. Firstly, Orthograph combines a hybrid approach utilizing both profile hidden Markov models (pHMMs) as well as BLAST4. This feature differentiates Orthograph from older tools that utilize one or the other and as well allows Orthograph to obtain greater sensitivity and accuracy4,5. Orthograph is also able to utilize both transcriptomes as well as full genomes which is beneficial as in certain cases full genomic sequences may not be available4. In terms of computational resources, Orthograph supports multi-threading4. The creators of Orthograph claim, Orthograph exhibits a 15% to 80% reduction in running time per additional thread when utilizing between 4 and 12 threads4. The utilization of 16 threads also results in a significant decrease in Orthograph running time, with a reduction to approximately 11% compared to running on a single-threaded machine4. Additionally, with regards to runtime, with each OG added, the runtime of Orthograph only increases linearly rather than exponentially meaning extremely large datasets used for large-scale studies can be used4. Another potential benefit of Orthograph is the fact that most of the data is stored in a database on a system’s hard drive, meaning less usage of system memory4. This feature also allows for saved search results to be re-evaluated quickly with different parameters, taking less time than the initial analysis4. Overall, with all the features and optimizations mentioned, Orthograph claims to not require the use of any high-performance computing infrastructure and instead it can be run on a standard workstation computer4. Although specifics on memory and CPU requirements, it can be assumed that Orthograph likely makes limited use of computational resources while keeping runtimes reasonable4. With these factors considered, Orthograph is included in this comparison as it represents one method of ortholog inference involving a graph-based approach utilizing pHMMs and BLAST4.

With regards to different approaches taken when developing tools for ortholog inference, tools utilizing a tree-based approach, where orthology is determined using a reconstructed phylogeny, often require more computational resources making them less efficient than tools utilizing a graph-based approach such as Orthograph4,5. OrthoReD (Orthology predictions from Reduced Datasets), however, is an open-source tree-based orthology prediction tool specifically designed to run on a standard workstation computer5. This tool claims to have been developed to utilize minimal system resources while maintaining accuracy comparable to other tree-based tools5. OrthoReD uses two main strategies to reduce computational resources required for ortholog inference: Firstly, the tool creates a subset of the dataset solely for each gene under analysis and secondly, it then restricts the number of genes involved in the computationally intensive task of tree-building5. More specifically, OrthoReD works by conducting a one-at-a-time approach to predict orthologs for each gene of interest, unlike other programs that group all genes into orthologous sets at once5. Similar to Orthograph, OrthoReD takes in an initial dataset of genes4,5. By default, OrthoReD then uses BLASTP for conducting similarity searches for each gene of interest, and the parameters used in the process are specifically optimized for orthology predictions5. BLASTP hits are then used to generate a reduced dataset5. A pairwise similarity search within the reduced dataset is then conducted to generate a pairwise similarity score matrix5. Using MCL, any outliers within the reduced dataset are then removed based on the pairwise similarity score5. Also using MCL, the genes in the reduced dataset are then separated into clusters, and each cluster is aligned to create a multiple sequence alignment (MSA) using MAFFT5. Using each MSA as input, RAxML is employed to reconstruct a maximum-likelihood tree5. Depending on if genes of an outgroup are present, the tree is rooted differently5. With the tree created, the final step uses a part of another tool, OrthologID, to determine whether each node in the gene tree represents a speciation event or a gene duplication event, or more specifically, this step attempts to differentiate between orthologs and paralogs5. This process involves checking for species overlap in tree nodes and assigning gene duplication events to nodes with overlap, except single-species clades as those treated as a single gene with multiple isoforms5. Within the tree, speciation events are then converted to gene duplication events, genes from the most recent duplication event including the gene of interest are then output and are the final orthologs found by the tool5.

The main feature of OrthoReD and its biggest advantage is the fact that due to its limited use of computational resources it can be run on more common hardware that is more accessible5. The tool claims to be able to conduct similar analysis and provide similar results as other tools while using fewer resources making it more efficient5. In addition, the tool provides enhanced accessibility by virtue of its utilization of a Perl script framework, which facilitates functionality across different operating systems, including Windows5. This stands in contrast to Orthograph, which is specifically designed for Unix-based machines4. OrthoReD also provides adjustments allowing for increased accuracy with the trade-off being greater runtime5. One such adjustment is that OrthoReD can utilize AB BLAST or SWIPE instead of BLASTP similar searches5. This option allows for greater control over search parameters and sensitivity5. OrthoReD claims to provide results of high similarity to OrthologID, another common inference tool5. Furthermore, when tested with a dataset of 444,382 sequences on a Mac Pro with a 3.33GHz Intel Xeon processor and 2 cores allocated, the median total CPU time for OrthoReD was under 15 minutes5. Using the same system, various datasets ranging from 194,469 to 532,305 were also evaluated, and it was observed that the median actual runtime, or wall-clock real time, consistently remained under 12 minutes5. Although CPU runtime was measured, memory usage was not meaning the exact computational load this tool puts onto a system will have to be tested5. Altogether, OrthoReD is included in this comparison as it has potential benefits in terms of efficiency which may be useful for smaller-scale applications with low amounts of data, applications where accuracy is slightly less relevant, or situations where high-performance compute clusters are not available.

A significant number of ortholog inference tools including the ones mentioned utilize sequence alignments in some form4,5. With newer machine learning technology, however, newer tools have been developed which are alignment-free6. One such tool is DeepNOG which is a tool created for assigning orthology based on deep convolutional networks that does not utilize sequence alignments6. Deep learning algorithms utilize neural networks, which are a category of machine learning techniques10. DeepNOG utilizes deep learning as it improves upon an existing tool, DeepFam, in a few key aspects6. Specifically, DeepNOG allows for the use of large-scale orthology databases other than COG and it can handle much larger datasets6. Additionally, DeepNOG is developed using PyTorch in contrast to DeepFam which is implemented in TensorFlow6. With the type of deep learning technology that is utilized by this tool, it is the case that the tool first must be trained before it is able to accurately perform6. Although DeepNOG is database agnostic, it is pre-trained with certain portions of the eggNOG 5 database, and if other databases are to be used with this tool, custom models must be trained which the tool does have built-in features to allow for6. Once the learning model is trained, DeepNOG works by first utilizing an ‘encoding layer’ that transforms each input protein sequence into a set of numbers that capture the similarities between the amino acids that make up that sequence6. This transformed data is then passed through a filter that extracts different sequence patterns of various sizes, which is then used to identify which predicted orthologous group for the input sequence6.

DeepNOG’s main advantages stem from the factors that differentiate it from the tool that it is based on, DeepFam6. Specifically, DeepNOG is designed for efficiency when it comes to large datasets at the cost of accuracy6. While alignment-free methods like DeepNOG provide valuable computational efficiency, the highest accuracy for assignment is still obtained by alignment-based tools6. DeepNOG not only demonstrates higher computational efficiency on CPUs but also benefits from GPU acceleration, leading to even greater throughput during analysis6. In the case of the COG database, DeepNOG demonstrates slightly greater speed in comparison to other alignment-free methods and notably faster performance than alignment-based methods6. With these factors considered, DeepNOG has the potential to be significantly more efficient in terms of time and resource utilization at a marginal expense of precision and recall relative to the other tools in this comparison6.

Another recently developed tool, SHOOT, also claims to provide fast and accurate ortholog inference utilizing a pre-existing database of genetic sequences similar to Orthograph4,7. As mentioned earlier, certain tools infer orthology through a process that involves the assembly of a group of gene sequences that share similarities, followed by phylogenetic tree inference on this group to infer the relationships between those genes5. This process, however, can be computationally intensive and time-consuming, especially when analyzing large numbers of sequences7. SHOOT attempts to make such a phylogenetic analysis more efficient by incorporating a two-stage approach7. Firstly, a SHOOT phylogenetic database is constructed through steps involving the dataset of homologous sequences provided, including homology group inference, multiple sequence alignment, as well as phylogenetic tree inference7. These steps utilize existing algorithms and tools such as MAFFT for constructing MSAs, IQ-TREE for the creation of maximum likelihood phylogenetic trees, as well as OrthoFinder for clustering genes into groups of homologs7. The phylogenetic relationships between all genes in the database are established ahead of time, so subsequent SHOOT searches can exploit this pre-computed information to enable the use of accurate phylogenetic methods for the placement of query genes within pre-computed phylogenetic trees7. This is where the second step comes in and that is the placement of the query sequence within one of the groups7. This step involves DIAMOND which creates profiles based on each of the groups of sequences and then searches the query sequence against the profiles7. Once assigned, the query sequence is added to the MSA for that group as well as the phylogenetic tree7. This second step involves minimal extra computation7. In the end, the output of SHOOT is a phylogenetic tree with the query sequence corrected placed which allows for “...the phylogenetic history of the query sequence and its orthologs can be immediately visualized, interpreted, and retrieved”7. This approach provides a highly efficient means of analyzing large numbers of gene sequences for their evolutionary relationships and claims to be highly accurate and reliable in comparison to other existing methods7.

The tool SHOOT has some clear potential benefits due to its unique software pipeline that utilizes multiple algorithms7. SHOOT offers a potentially highly efficient approach by first executing computationally expensive tasks and subsequently adding each query sequence, which requires minimal computational resources7. This method is claimed to work effectively with large datasets7. When comparing accuracy and precision to just DIAMOND and BLAST, SHOOT does appear to come out on top, however, that is not a compelling argument as several tools are also able to do the same7,11. Additionally, when tested using the UniProt Reference Proteomes database and 1000 query sequences, the runtime for the program was a few seconds when conducted on a 16 cores Intel Xeon E5-2683 CPU7. These claimed results provide some insights into the possible efficiency and speed of this tool which is why it will also be included in this comparison7. Additionally, this comparison includes OrthoReD which is a tree-based tool meaning the claims of SHOOT being more efficient than tools that develop phylogenies after grouping genes can be validated7.

Once pipelines for each of the ortholog inference tools have been developed, each tool will be tested and various factors will be observed such as runtime, the number of sequences found for each orthologous gene within the dataset given (recall), as well as accuracy in assigning genes to the correct ortholog group (precision). For each of the tools, a set of orthologous genes will be used, and as such OrthoDB will be used to obtain these datasets9. OrthoDB is a database developed by the Swiss Institute of Bioinformatics which stores orthologous protein-coding genes9. The database amasses an extensive collection of approximately 1952 species of eukaryotes and over 100 million genes, sourced from numerous databases including Ensembl, UniProt, and NCBI9. Over 3000 complete genomes are also included as part of the database9. Culicidae genomes will be used to test the tools of which OrthoDB contains genomes of 22 species, 18 of which are Anopheles9. With no filtering done, there are over 18,000 Culicidae genes9. A filtered dataset will be used, however, for input within the selected tools. After filtering there are nearly 6000 Culicidae genes that span >90% of Culicidae species and are single copy in >90% of Culicidae species9. These parameters were chosen to filter as the OrthoDB database contains mainly Anopheles genomes meaning genes from this genus are greatly overrepresented9. Additionally, a dataset that is large enough to be used for both testing and training certain tools as well as being used for creating query sequences is needed meaning parameters that are too stringent would result in a dataset that is inadequate in size.

Upon careful selection and evaluation of all available tools, the one that emerges as the most advantageous is OrthoRed. Based on its features and capabilities, OrthoRed outperforms other tools in terms of efficiency in inferring orthologs whilst maintaining accuracy meaning that OrthoRed could be the most suitable tool for performing orthology analysis in a variety of biological applications5. Specifically, OrthoReD appears to be developed from the ground up to run on standard workstation machines rather than compute clusters meaning in order to facilitate this requirement the tool must be as efficient as possible5. In addition to this, OrthoReD appears to have accuracy close to that of other commonly used inference tools meaning efficiency does not necessarily mean sacrificing accuracy5. To summarize, the comparison of these ortholog inference tools will provide greater insight into the advantages and disadvantages of each tool allowing for the right tool to be selected based on the use case while also identifying the one tool which can be the go-to as it provides good performance, runtime, and accuracy.

Reference

1. Gabaldón T, Koonin EV. Functional and evolutionary implications of Gene Orthology. Nature Reviews Genetics. 2013;14(5):360–6.
2. Chen X, Zhang J. The ortholog conjecture is untestable by the current gene ontology but is supported by RNA sequencing data. PLoS Computational Biology. 2012 Nov 29;8(11):e1002784.
3. Sutphin GL, Mahoney JM, Sheppard K, Walton DO, Korstanje R. Wormhole: Novel least diverged ortholog prediction through machine learning. PLOS Computational Biology. 2016;12(11).
4. Petersen M, Meusemann K, Donath A, Dowling D, Liu S, Peters RS, et al. Orthograph: A versatile tool for mapping coding nucleotide sequences to clusters of orthologous genes. BMC Bioinformatics. 2017;18(1).
5. Battenberg K, Lee EK, Chiu JC, Berry AM, Potter D. Orthored: A rapid and accurate orthology prediction tool with low computational requirement. BMC Bioinformatics. 2017;18(1).
6. Feldbauer R, Gosch L, Lüftinger L, Hyden P, Flexer A, Rattei T. DeepNOG: Fast and accurate protein orthologous group assignment. Bioinformatics. 2020;36(22-23):5304–12.
7. Emms DM, Kelly S. Shoot: Phylogenetic gene search and ortholog inference. Genome Biology. 2022;23(1).
8. Nichio BT, Marchaukoski JN, Raittz RT. New tools in Orthology Analysis: A brief review of promising perspectives. Frontiers in Genetics. 2017;8.
9. Waterhouse RM, Tegenfeldt F, Li J, Zdobnov EM, Kriventseva EV. OrthoDB: A hierarchical catalog of animal, fungal and bacterial orthologs. Nucleic Acids Research. 2012;41(D1).
10. What are convolutional neural networks? [Internet]. IBM. [cited 2023 Feb 12]. Available from: https://www.ibm.com/topics/convolutional-neural-networks
11. Deutekom ES, Snel B, van Dam TJ. Benchmarking orthology methods using phylogenetic patterns defined at the base of Eukaryotes. Briefings in Bioinformatics. 2020;22(3).