BIOINFORMATICS APPLICATIONS NOTE Vol. 31 no. 3 2015, pages 447–448
doi:10.1093/bioinformatics/btu642

Databases and ontologies Advance Access publication October 1, 2014

OrthoInspector 2.0: Software and database updates

Benjamin Linard[1,2], Alexis Allot[1], Raphaël Schneider[1], Can Morel[1], Raymond Ripp[1], Marc Bigler[1], Julie D. Thompson[1], Olivier Poch[1] and Odile Lecompte[1,*]

[1]LBGI, Computer Science Department, ICube, UMR 7357, University of Strasbourg, CNRS, Fédération de médecine translationnelle, 4 rue Kirschleger 67085 Strasbourg, France and [2]Department of Life Sciences, Natural History Museum, Cromwell Road, London SW7 5BD

Associate Editor: Jonathan Wren

ABSTRACT

Summary: We previously developed OrthoInspector, a package incorporating an original algorithm for the detection of orthology and inparalogy relations between different species. We have added new functionalities to the package. While its original algorithm was not modified, performing similar orthology predictions, we facilitated the prediction of very large databases (thousands of proteomes), refurbished its graphical interface, added new visualization tools for comparative genomics/protein family analysis and facilitated its deployment in a network environment. Finally, we have released three online databases of precomputed orthology relationships.

Availability: Package and databases are freely available at http://lbgi.fr/orthoinspector with all major browsers supported.

Contact: odile.lecompte@unistra.fr

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on July 1, 2014; revised on September 4, 2014; accepted on September 21, 2014

1 INTRODUCTION

High throughput comparative analyses, functional annotations, or evolutionary studies involve massive transfers of information between organisms using orthology inference. As defined by Fitch (1970), orthologs are homologous genes that diverged from an ancestral speciation event, while paralogs emerged from a duplication event. Today, it is widely accepted that orthologs generally share similar functions, whereas paralogs can potentially evolve new functions. Numerous algorithms based on the results of Blast searches were developed to infer orthology relations (see Kristensen *et al*., 2011; Altenhoff and Dessimoz, 2012 for reviews). We previously developed an orthology inference algorithm also based on Blast and implemented it in the OrthoInspector (OI) package (Linard et al., 2011). Our focus was to maintain a balance between sensitivity and specificity (Linard *et al*., 2011; Dalquen *et al*., 2013). Contrary to most other packages, OI is not limited to predictions and provides tools for comprehensive mining of large orthology databases, nonspecialist use through a desktop graphical interface and can easily be deployed in a network environment. Here, we describe the main improvements implemented in the second version of the OrthoInspector package.

2 DISTINCTIVE FEATURES

2.1 Requirements

OI 2.0 requires the NCBI blast+ toolkit (Camacho *et al*., 2009), a Java compatible operating system and a local or remote access to a SQL database. Any SQL engine is compatible as long as it allows Java connections via a JDBC driver. OI provides an extended support for Postgresql and MySQL engines, for which more operations are automated. To predict new orthologs, OI requires as inputs one proteome in FASTA format per species of interest and a blast all-against-all constructed from the same proteomes.

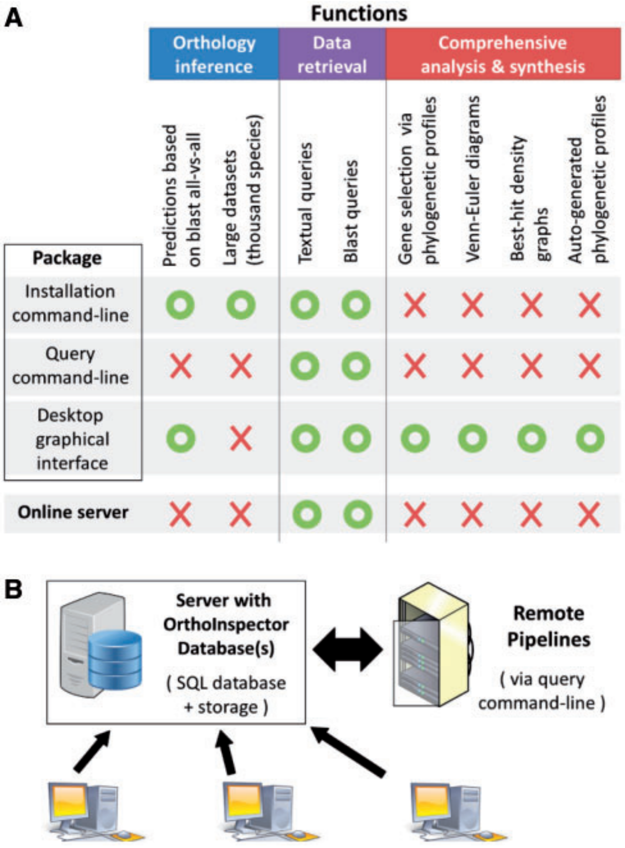
2.2 Implementation and network exploitation

The package is based on database/client interactions and separated into three clients (Fig. 1A): a command-line for initial orthology predictions and database installation, a query command-line to retrieve precomputed predictions and a graphical interface designed for desktop querying and data visualization. Limited computational resources are required as the clients delegate many operations to the SQL engines and their optimized dataset manipulation capabilities. Management of large databases (thousands of species) is, however, facilitated by several options from the installation command-line. The components of the graphical interface make easier the mining and the visualization of complex orthology relationships for nonspecialists. A small 6six species database dump can be downloaded from the website to rapidly test these tools. The three clients can be used on a single desktop computer but can also be deployed in a network. Orthology databases can then be stored in one server while several users/pipelines exploit the clients for various purposes (Fig. 1B). Then, OI responsiveness will mainly depend on server and network speed.

2.3 Eukaryote and prokaryote databases

We have constructed three orthology databases with OI. The first database, named “Prokaryotes,” contains orthologs between 120 Archaea and 1568 Bacteria proteomes. The “Eukaryotes” dataset contains 259 complete proteomes and covers all main eukaryotic phyla, from unicellular organisms to plants, fungi, and metazoan. The last dataset, “Quest For Orthologs” (QFO), combines bacteria, archaea and eukaryote proteomes and corresponds to the latest version of the orthology benchmark released by the QFO Consortium (Dessimoz *et al*., 2012).

*To whom correspondence should be addressed.



	Functions							
	Orthology inference		Data retrieval		Comprehensive analysis & synthesis			
Package	Predictions based on blast all-vs-all	Large datasets (thousand species)	Textual queries	Blast queries	Gene selection via phylogenetic profiles	Venn-Euler diagrams	Best-hit density graphs	Auto-generated phylogenetic profiles
Installation command-line	○	○	○	○	✗	✗	✗	✗
Query command-line	✗	✗	○	○	✗	✗	✗	✗
Desktop graphical interface	○	✗	○	○	○	○	○	○
Online server	✗	✗	○	○	✗	✗	✗	✗

A
B
Server with OrthoInspector Database(s)
(SQL database + storage)
Remote Pipelines
(via query command-line)

Fig. 1. Package organization and main functionalities. (A) Command-lines are used for initial orthology inference, database querying and to handle large datasets. The graphical interface is used for all other tasks. (B) Typical deployment of the package on a network

Supplementary File S1 lists all the species included in these databases and their taxonomy.

3 MAIN ADDITIONS

3.1 Large-scale phylogenetic profiles

Several analyses are now supported by an interactive tree of life in the graphical interface, facilitating in particular the establishment of “phylogenetic profile” queries. A selection of presence/absence criteria at different levels in the tree, allows the extraction of large-scale sets of genes that respect the profile through the orthology criteria. For instance, one can retrieve all Microsporidia sequences with orthologs in Basidiomycota but not in Ascomycota (Supplementary Fig. S2).

3.2 Best-hit density graph and Euler diagrams

Two new visualization tools are now part of the graphical interface. First, the “best-hit density graph” is designed to analyze the orthologous relationships linking genes in a particular family and to reveal potential subfamilies. Through a dynamic graph representation of BLAST best hits linking a protein family, the user can explore conservation patterns within the set by modifying the BLAST score or *E*-value thresholds on the fly (Supplementary File S3). This tool can be used to adapt the delineation of subfamilies to the evolutionary rate of the family under consideration or to a given phylogenetic scope. Second, Venn diagrams (3 organisms), but also more complex Euler diagrams (>3 organisms), can be generated. When more than 3 organisms are considered, diagram overlaps are based on the VennEuler library (Wilkinson, 2012), which provides a statistical framework to estimate the best possible circle-based representation.

3.3 Web server

Our precomputed datasets (Eukaryote, Prokaryote, and QFO) can be accessed via a web server allowing ortholog retrieval by textual or Blastp searches. A list of organisms can be selected with an interactive species tree. Orthology relationships corresponding to the query sequence are compiled in a table format with phylum color codes to facilitate the analysis of phylum specific orthology distributions and produce a user-friendly and intuitive overview of the revealed evolutionary history (see Supplementary File S4 for a case study). All datasets can be downloaded in CSV and OrthoXML formats (Schmitt *et al.*, 2011).

4 CONCLUSION

OI is a package dedicated to the efficient calculation and analysis of orthology data and allows a rapid and intuitive analysis of relationships associated with large clades. The OI web server allows the retrieval of precomputed orthology data from thousands of eukaryote and prokaryote proteomes.

ACKNOWLEDGEMENTS

Funding: This work was supported by the ANR [grant ANR-10-INSB-05-01 FRISBI, grant ANR-10-BINF-03-02 BIPBIP] and Institute funds from the CNRS, the Faculté de Médecine de Strasbourg and the Université de Strasbourg.

Conflict of interest: none declared.

REFERENCES

Altenhoff,A.M. and Dessimoz,C. (2012) Inferring orthology and paralogy. *Methods Mol. Biol.*, **855**, 259–279.

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

Dalquen,D.A. *et al.* (2013) The impact of gene duplication, insertion, deletion, lateral gene transfer and sequencing error on orthology inference: a simulation study. *PLoS One*, **8**, e56925.

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

Fitch,W.M. (1970) Distinguishing homologous from analogous proteins. *Syst. Zool.*, **19**, 99–113.

Kristensen,D.M. *et al.* (2011) Computational methods for Gene Orthology inference. *Brief Bioinform.*, **12**, 379–391.

Linard,B. *et al.* (2011) OrthoInspector: comprehensive orthology analysis and visual exploration. *BMC Bioinformatics*, **12**, 11.

Schmitt,T. *et al.* (2011) Letter to the editor: SeqXML and OrthoXML: standards for sequence and orthology information. *Brief Bioinform.*, **12**, 485–488.

Wilkinson,L. (2012) Exact and approximate area-proportional circular Venn and Euler diagrams. *IEEE Trans. Vis. Comput. Graph.*, **18**, 321–331.