OrthoXML-tools: a toolkit for manipulating OrthoXML files for orthology data

Ali Yazdizadeh Kharrazi[1*], Adrian M. Altenhoff [2,3], Nikolai Romashchenko[3,4], Christophe Dessimoz[3,4], Sina Majidian[5*]

[1] DataChef, The Hague, The Netherlands.
[2] ETH Zurich, Computer Science, Universitätstr. 6, 8092 Zurich, Switzerland.
[3] SIB Swiss Institute of Bioinformatics, 1015 Lausanne, Switzerland.
[4] Department of Computational Biology, University of Lausanne, 1015 Lausanne, Switzerland.
[5] Department of Computer Science, Johns Hopkins University. 3400 North Charles St., Baltimore, MD 21218. USA.

Abstract

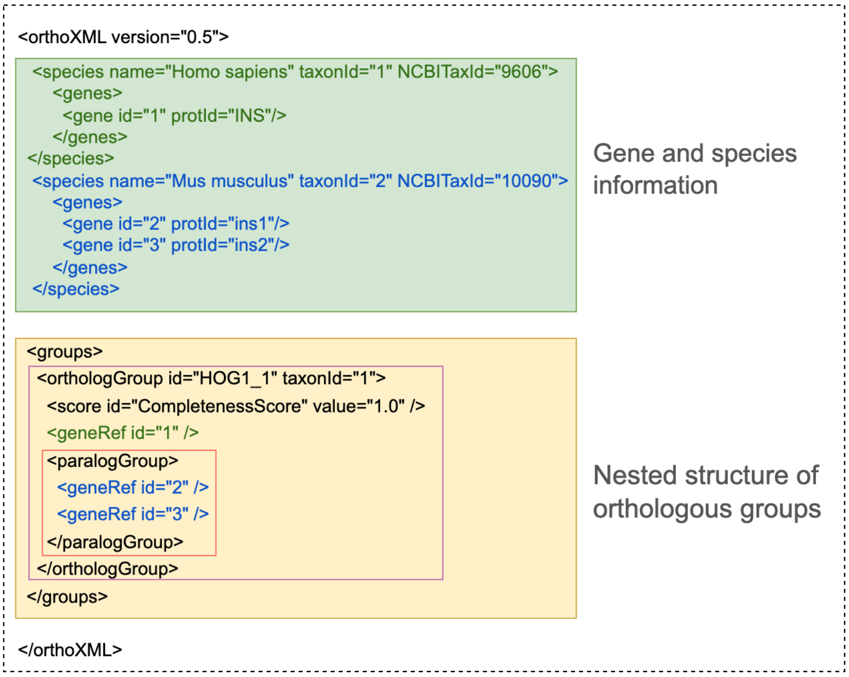
The OrthoXML file is a standard file format for orthology data. It provides a standardized structure for describing orthologous and paralogous relationships while allowing the user to store custom and database-specific data in the same format. Although many orthology databases use it as a way to export data, there is no comprehensive toolkit for working with this format. Here, we introduce the OrthoXML-tools (https://github.com/DessimozLab/orthoxml-tools), a comprehensive toolkit for loading, manipulating, and exporting the OrthoXML files to other formats. We show its capabilities and performance on our benchmarks.

Keywords: Comparative genomics, Orthology, Hierarchical Orthologous Groups, OrthoXML.

Introduction

Comparative genomics has revolutionized our understanding of evolution by enabling systematic comparisons across diverse species. At its core lies the concept of orthology, genes in different species that descended from a single ancestral gene through speciation events and often retain similar functions (Gabaldón and Koonin, 2013). Building on this, Hierarchical Orthologous Groups (HOGs) organize these relationships into multi-taxonomic level frameworks that capture speciation and gene duplication events—both ancient and recent (Altenhoff *et al.*, 2013; Zahn-Zabal, Dessimoz and Glover, 2020; Sarton-Lohéac *et al.*, 2025).

OrthoXML (Schmitt *et al.*, 2011) is a widely adopted standard for representing orthology data, facilitating the storage and comparison of such information across data from various databases. It provides an XML schema that defines a structured format for describing orthology and paralogy relationships between genes across different species, while also supporting the inclusion of custom and database-specific data in the same format. The OrthoXML schema specifies detailed elements, including species and gene identifiers, database links, taxonomic hierarchies (supporting both custom trees and NCBI taxonomy), score definitions, and group structures such as ortholog and paralog groups (**Figure 1**). Despite its widespread use, there has been a lack of comprehensive tools for efficiently handling OrthoXML files.



```
<orthoXML version="0.5">
 <species name="Homo sapiens" taxonId="1" NCBITaxId="9606">
    <genes>
      <gene id="1" protId="INS"/>
    </genes>
</species>
 <species name="Mus musculus" taxonId="2" NCBITaxId="10090">
    <genes>
      <gene id="2" protId="ins1"/>
      <gene id="3" protId="ins2"/>
    </genes>
 </species>

<groups>
 <orthologGroup id="HOG1_1" taxonId="1">
   <score id="CompletenessScore" value="1.0" />
   <geneRef id="1" />
   <paralogGroup>
     <geneRef id="2" />
     <geneRef id="3" />
   </paralogGroup>
 </orthologGroup>
</groups>

</orthoXML>
```

Gene and species information
Nested structure of orthologous groups

Figure 1. Overview of the OrthoXML File Format (simplified). A schematic representation of an OrthoXML file, a standardized XML-based format for representing orthology data. OrthoXML follows a hierarchical structure where elements are enclosed within opening `<tag>` and closing `</tag>` tags. `<orthoXML>` is the root element enclosing other elements. The `<species>` element contains information of genes. An OrthoXML file can include a `<taxonomy>` element, which specifies the species tree used to generate the file. Additionally, the `<groups>` element encapsulates the orthology and paralogy relationships among genes.

Many resources and tools have adopted OrthoXML as a standard format for orthology data exchange. For instance, the Quest for Orthologs (QfO) consortium extensively uses OrthoXML for benchmarking and data sharing (A. Altenhoff *et al.*, 2024; Langschied *et al.*, 2024; Majidian, Hadziahmetovic, *et al.*, 2025). Several orthology inference tools and comparative genomics databases, including Ensembl Compara (Harrison *et al.*, 2024), OMA (A. M. Altenhoff *et al.*, 2024), FastOMA (Majidian, Nevers, *et al.*, 2025), InParanoid (Sonnhammer and Östlund, 2015), OrthoInspector (Nevers *et al.*, 2019), PANTHER (Thomas *et al.*, 2022), PhylomeDB (Fuentes *et al.*, 2022), OrthoFinder (Emms *et al.*, 2025) and HieranoiDB (Kaduk *et al.*, 2017) support OrthoXML export, which enhances compatibility and reuse of their orthology predictions. This widespread use highlights the need for a unified and flexible toolkit to manipulate, analyze, and convert these files.

Existing solutions offer partial functionalities; for instance, the ETE Toolkit (Huerta-Cepas, Serra and Bork, 2016) provides the `etree2orthoxml` script, which converts phylogenetic trees in Newick format to OrthoXML. Additionally, pyHam facilitates the analysis of HOGs encoded in OrthoXML (Train *et al.*, 2019) and infers evolutionary events like gene loss, duplication and gain at user-defined taxonomic levels. While both tools are useful, they are primarily designed for phylogenetic analysis and HOG exploration, rather than for comprehensive manipulation of OrthoXML files.

In this context, we introduce OrthoXML-tools, a Python package designed to provide a robust and versatile toolkit for loading, manipulating, and exporting OrthoXML files. Our package aims to streamline workflows involving orthology data by offering a flexible solution that addresses the limitations of existing tools. We demonstrate its capabilities and performance using data from various sources on our benchmarks, highlighting its potential to become an essential resource for researchers working with orthology data.

By simplifying the handling of OrthoXML files, we anticipate that OrthoXML-tools will encourage broader adoption of this standard format within the scientific community, thereby enhancing data interoperability and integration across various platforms.

Materials and methods

OrthoXML-tools is implemented in Python using the **lxml** library (Behnel, Faassen and Bicking, 2005), which is a binding for the C libraries **libxml2** and **libxslt**. The toolkit is designed with performance in mind. Given that OrthoXML files can be extremely large, we employed a streaming approach for parsing, leveraging the **iterparse** method provided by **lxml**, combined with techniques to promptly release unneeded memory and unused allocations, whenever possible. This design ensures a minimal memory footprint, even when processing large-scale datasets.
OrthoXML-tools provides a command-line interface (CLI) that supports a variety of operations. It can extract general statistics from OrthoXML files and provides conversion functionalities, including loading and exporting phylogenetic trees. It also enables the export of information such as orthologous and paralogous pairs, as well as orthologous groups. Furthermore, it offers manipulation features such as splitting files based on root hierarchical orthologous groups (rootHOGs) and filtering orthologous groups according to pre-defined scores (e.g. group completeness).

OrthoXML-tools is available as a Python package and a command line tool. It can be installed from PyPI (https://pypi.org/project/orthoxml-tools). The documentation of OrthoXML-tools is provided on its GitHub repository (https://github.com/DessimozLab/orthoxml-tools) and **Supplementary Note 1**.

Results

Functionality Overview

OrthoXML-tools provides a wide range of commands for analyzing, converting and manipulating orthology data in Hierarchical Orthologous Groups (HOGs). The software offers several key functionalities, beginning with basic statistics. For instance, the command *orthoxml-tools stats* reports general statistics such as the number of genes and species, while *orthoxml-tools gene-stats* computes the distribution of genes in groups per taxonomic level (‘taxonId’). Format validation is also integrated into the workflow; using *orthoxml-tools validate* ensures that OrthoXML files conform to the schema version specified in the file header.

In terms of orthologous gene identification, the command *orthoxml-tools export-pairs ortho* extracts all orthologous gene pairs from the OrthoXML file, and *orthoxml-tools export-pairs para* extracts paralogous relationships. The command *orthoxml-tools export-ogs* reports a special type of orthologous groups consisting of sets of genes that are all orthologous to each other (Fernández, Gabaldon and Dessimoz, 2020).

The extraction of gene families is performed using *orthoxml-tools split* command, which splits the OrthoXML file into smaller files corresponding to root HOGs. Additionally, conversion of annotated gene trees (NHX files) to the OrthoXML format is handled by the *orthoxml-tools from-nhx* command while the conversion from OrthoXML file into annotated gene tree is done by *orthoxml-tools to-nhx* command.

Finally, the filtering and selection process is facilitated by the command *orthoxml-tools filter*, which filters gene groups based on a specified score, such as completeness score. This completeness score is defined as the fraction of expected species represented in a group. OrthoXML-tools implements two strategies for filtering by this score. If a group does not satisfy the required score threshold, the group is removed along with all its descendant subgroups (the 'cascade-remove' strategy). Alternatively, all subgroups that satisfy the threshold themselves become independent root-level groups (the 'extract' strategy).

These commands are accessible through the command-line interface (CLI). Examples of each feature are provided in **Supplementary Note 1**. **Table 1** summarizes the features of OrthoXML-tools in comparison with the ETE Toolkit (Huerta-Cepas, Serra and Bork, 2016), and pyHam (Train *et al.*, 2019), two existing tools that offer OrthoXML support. Notably, ETE is well-suited for constructing and manipulating phylogenetic trees, while pyHam is designed to infer evolutionary events. In contrast, OrthoXML-tools is specifically designed to provide insight into orthology information stored in an OrthoXML file, offering methods to manipulate the data, extract relevant information, and convert it to and from other formats, such as phylogenetic trees.

Table 1. Features of the OrthoXML-tools compared with ETE4 and PyHam to process OrthoXML files.

Feature	**OrthoXML-tools**	**ETE4**	**PyHam**
Report statistics (e.g. number of genes in groups per taxon)	✅	❌	✅[1]
Find orthologous genes	✅	✅[2]	❌
Find paralogous genes	✅	✅[2]	❌
Find orthologous group	✅	❌	❌
Extract gene families	✅	❌	✅[1]
Export labeled gene trees (NHX)	✅	✅	❌
Convert phylogenetic tree (NHX) to OrthoXML	✅	✅	❌
Convert Orthologous Groups (CSV) to basic OrthoXML[3]	✅	❌	❌
Validate the OrthoXML format	✅	❌	❌
Filter OrthoXML (e.g. completeness score)	✅	❌	❌
Infer evolutionary events (e.g. duplications and losses)	❌	✅	✅
Ancestral genome reconstruction	❌	❌	✅
Command line tools	✅	✅	❌

[1] PyHam parses the OrthoXML as a Python object, with which a user can write Python scripts to extract the statistics and gene families (see https://zoo.cs.ucl.ac.uk/tutorials/tutorial_pyHam_get_started.html).

[2] ETE4 cannot readily report orthologous genes as a command-line tool; however, its Python interface could be used in custom scripts to extract orthologous groups via ETE4 functions (see https://github.com/etetoolkit/ete4/blob/ete4/ete4/orthoxml/_orthoxml.py#L1071).

[3] The input CSV file should be structured such that each row represents an orthogroup (OG), each column corresponds to a species, and each element is a gene name. Note that since the CSV does not contain the full information required to represent the hierarchical structure of HOGs, the output OrthoXML file contains only root-level groups.

We evaluate the performance of the OrthoXML-tools on two datasets (Table 2). The first dataset is from the QfO Reference Proteomes (Nevers *et al.*, 2022; Dyer *et al.*, 2025) (www.ebi.ac.uk/reference_proteomes), comprising 78 species and 988,778 genes. We used three OrthoXML files generated by FastOMA (Majidian, Nevers, *et al.*, 2025), Ensembl Compara (Herrero *et al.*, 2016), and OrthoFFGC (Marilia, 2024). We downloaded these from the QfO webservice at https://orthology.benchmarkservice.org. See **Data Availability** for the full link to each OrthoXML file. The second dataset is the whole OMA database, which covers 2,927 species and 24.7 million genes (A. M. Altenhoff *et al.*, 2024). The OrthoXML file was generated by OMA standalone (Altenhoff *et al.*, 2013), and is available on the OMA browser at https://omabrowser.org/oma/current/. The OMA OrthoXML contains about 96 billion orthologous pairs. In both cases, OrthoXML-tools demonstrated faster parsing performance and significantly lower memory usage compared to ETE4.

We conducted the benchmarking on a 64-bit Linux system with 48 Intel CPU cores of Xeon(R) Gold 3.00GHz. We used the command line GNU *time* to measure the run time and peak memory.

Table 2. Description of OrthoXML files used in the benchmarking

Proteome dataset	**OrthoXML source**	**Number of species**	**File size** (Megabyte)	**Number of genes**	**Number of orthologous genes**	**Number of orthologous groups**
QfO	FastOMA	78	171.2	988,778	10,896,873	62551
QfO	Ensembl Compara	78	4,515.8	719,680	27,803,099	NA[1]
QfO	OrthoFFGC	78	63.5	988,778	3,641,988	71,933
OMA database	OMA	2,927	3,298.8	24,790,217	95,969,542,540	10,647,945

[1] The QfO Ensembl Compara only includes orthologous pairs in OrthoXML format.

Table 3. Performance Comparison of OrthoXML compared to ETE4 on OrthoXML files.

Proteome dataset	**OrthoXML source**	**Tool**	**Reporting statistics**		**Finding orthologous genes**		**Finding orthologous group**	
			Time (s)	Peak Memory (MiB)	Time (s)	Peak Memory (MiB)	Time (s)	Peak Memory (MiB)
QfO	FastOMA	ETE4	33.5	2,593	NA	NA	NA	NA
		OrthoXML-tools	12.8	69.5	13.8	751.3	15.5	170.5
	Ensembl Compara	ETE4	1469.2	91450.7	NA	NA	NA	NA
		OrthoXML-tools	255.8	64.9	326.4	557	NA	146.2
	OrthoFFGC	ETE4	17.52	1,425	NA	NA	NA	NA
		OrthoXML-tools	12	69.7	11.8	749.9	13.8	169.4
OMA database	OMA	ETE4	806.6	52468.6	NA	NA	NA	NA
		OrthoXML-tools	476.7	382	30300	17,921	598.2	2,869

Conclusion and Discussion

Here, we introduced OrthoXML-tools, a comprehensive and efficient Python-based toolkit for working with OrthoXML files and extracting useful evolutionary information. The OrthoXML format has gained traction within the comparative genomics community and beyond, in part due to its structured representation of hierarchical orthologous groups (HOGs). Its adoption by major resources and software attests to its utility. However, the complexity and verbosity of XML pose practical challenges, particularly for developers and users working with large datasets. These issues include complexity in format definition, large file sizes, high memory requirements for parsing, and the lack of widely available downstream tools for analysis. Alternative formats such as YAML or TOML might offer improved human readability and reduced overhead. However, these alternatives currently lack the schema richness and community support needed to represent complex orthology structures, design new standards, and develop downstream tools. In this work, we focused on supporting OrthoXML due to its widespread use and formal specification, which we believe remains a useful standard for representing orthology relationships.

Given the growing volume of sequenced genomes (Lewin *et al.*, 2022; Chikhi *et al.*, 2024), the need for scalable, memory-efficient tools in comparative genomics has become increasingly pressing. Our work addresses this gap. In our latest release, we reduced memory usage for key operations, such as computing statistics, from 48 GB to around 380 MB for the OMA OrthoXML file, including around 3000 species. This was achieved by reengineering the software to process each rootHOG independently, thereby avoiding full in-memory loading. We anticipate further improvements in performance and scalability, potentially by integrating components written in Rust or C++. Looking ahead, we plan to expand OrthoXML-tools with new features, such as reporting orthologous pairs for user-specified genes and enabling selective removal of species from OrthoXML files.

In summary, OrthoXML-tools provides rapid performance with efficient memory usage while providing a broader set of features including querying, validation, filtering, and format conversion. We anticipate it will further accelerate research and foster broader adoption of the OrthoXML format in comparative genomics, by making it more accessible and practical for large-scale analyses. By addressing the practical downstream applications associated with OrthoXML, our toolkit enables broader access to the rich evolutionary information encoded in this format.

Code Availability

The OrthoXML-tools is available under GNU General Public License v3.0 at https://github.com/DessimozLab/orthoxml-tools.

Data Availability

The QfO reference proteomes are from https://www.ebi.ac.uk/reference_proteomes/, with the direct link https://ftp.ebi.ac.uk/pub/databases/reference_proteomes/previous_releases/qfo_release-2022_02/QfO_release_2022_02.tar.gz. The OrthoXML files are available for FastOMA at https://doi.org/10.23728/b2share.028e02c9c38f4698ba074a7f0ae158d6, for OrthoFFGC at https://doi.org/10.23728/b2share.994214e4a3404bb2bc3150d0e505700f, and for EnsemblCompara at https://doi.org/10.23728/b2share.b6336c505b9d431ea3d76251edb7fc64. All links are available in the section “Public Projects”, under the tab “QfO Benchmark release **2022**” of the QfO benchmark server https://orthology.benchmarkservice.org/proxy/projects/2022/. The OMA database can be accessed through https://omabrowser.org/oma/current/, and the OMA OrthoXML file was downloaded using https://omabrowser.org/All/oma-hogs.orthoXML.gz.

Acknowledgment

We thank Irene Julca, Stefano Pascarelli and the Comparative Genomics Lab for their feedback on the software and the manuscript.

References

Altenhoff, A. *et al.* (2024) 'New developments for the Quest for Orthologs benchmark service', *NAR genomics and bioinformatics*, 6(4), p. lqae167.

Altenhoff, A.M. *et al.* (2013) 'Inferring hierarchical orthologous groups from orthologous gene pairs', *PloS one*, 8(1), p. e53786.

Altenhoff, A.M. *et al.* (2024) 'OMA orthology in 2024: improved prokaryote coverage, ancestral and extant GO enrichment, a revamped synteny viewer and more in the OMA Ecosystem', *Nucleic acids research*, 52(D1), pp. D513–D521.

Behnel, S., Faassen, M. and Bicking, I. (2005) *lxml: XML and HTML with Python*. Available at: https://lxml.de/ (Accessed: 2025).

Chikhi, R. *et al.* (2024) 'Logan: Planetary-scale genome assembly surveys life's diversity', *bioRxiv*. Available at: https://doi.org/10.1101/2024.07.30.605881.

Dyer, S.C. *et al.* (2025) 'Ensembl 2025', *Nucleic acids research*, 53(D1), pp. D948–D957.

Emms, D.M. *et al.* (2025) 'OrthoFinder: scalable phylogenetic orthology inference for comparative genomics', *bioRxiv*. Available at: https://doi.org/10.1101/2025.07.15.664860.

Fernández, R., Gabaldon, T. and Dessimoz, C. (2020) 'Orthology: definitions, prediction, and impact on species phylogeny inference', *Phylogenetics in the genomic era*, pp. 2–4.

Fuentes, D. *et al.* (2022) 'PhylomeDB V5: an expanding repository for genome-wide catalogues of annotated gene phylogenies', *Nucleic acids research*, 50(D1), pp. D1062–D1068.

Gabaldón, T. and Koonin, E.V. (2013) 'Functional and evolutionary implications of gene orthology', *Nature reviews. Genetics*, 14(5), pp. 360–366.

Harrison, P.W. *et al.* (2024) 'Ensembl 2024', *Nucleic acids research*, 52(D1), pp. D891–D899.

Herrero, J. *et al.* (2016) 'Ensembl comparative genomics resources', *Database: the journal of biological databases and curation*, 2016. Available at: https://doi.org/10.1093/database/bav096.

Huerta-Cepas, J., Serra, F. and Bork, P. (2016) 'ETE 3: Reconstruction, Analysis, and Visualization of Phylogenomic Data', *Molecular biology and evolution*, 33(6), pp. 1635–1638.

Kaduk, M. *et al.* (2017) 'HieranoiDB: a database of orthologs inferred by Hieranoid', *Nucleic acids research*, 45(D1), pp. D687–D690.

Langschied, F. *et al.* (2024) 'Quest for orthologs in the era of biodiversity genomics', *Genome biology and evolution*, 16(10), p. evae224.

Lewin, H.A. *et al.* (2022) 'The Earth BioGenome Project 2020: Starting the clock', *Proceedings of the National Academy of Sciences of the United States of America*, 119(4), p. e2115635118.

Majidian, S., Nevers, Y., *et al.* (2025) 'Orthology inference at scale with FastOMA', *Nature Methods*, 22(2), pp. 269–272.

Majidian, S., Hadziahmetovic, A., *et al.* (2025) 'Quest for orthologs in the era of data deluge and AI: Challenges and innovations in orthology prediction and data integration', *Submitted*.

Marilia, D.V. (2024) 'Family-Free Genome Comparison', in *Comparative Genomics: Methods and Protocols*. New York, NY: Springer US, pp. 57–72.

Nevers, Y. *et al.* (2019) ‘OrthoInspector 3.0: open portal for comparative genomics’, *Nucleic acids research*, 47(D1), pp. D411–D418.

Nevers, Y. *et al.* (2022) ‘The Quest for Orthologs orthology benchmark service in 2022’, *Nucleic acids research*, 50(W1), pp. W623–W632.

Sarton-Lohéac, G. *et al.* (2025) ‘Reconstructing evolutionary histories with hierarchical orthologous groups’, *Submitted*.

Schmitt, T. *et al.* (2011) ‘Letter to the editor: SeqXML and OrthoXML: standards for sequence and orthology information’, *Briefings in bioinformatics*, 12(5), pp. 485–488.

Sonnhammer, E.L.L. and Östlund, G. (2015) ‘InParanoid 8: orthology analysis between 273 proteomes, mostly eukaryotic’, *Nucleic acids research*, 43(Database issue), pp. D234–9.

Thomas, P.D. *et al.* (2022) ‘PANTHER: Making genome-scale phylogenetics accessible to all’, *Protein science : a publication of the Protein Society*, 31(1), pp. 8–22.

Train, C.-M. *et al.* (2019) ‘iHam and pyHam: visualizing and processing hierarchical orthologous groups’, *Bioinformatics (Oxford, England)*, 35(14), pp. 2504–2506.

Zahn-Zabal, M., Dessimoz, C. and Glover, N.M. (2020) ‘Identifying orthologs with OMA: A primer’, *F1000Research*, 9(27), p. 27.