|  |
| --- |
| Genome Analysis  PorthoMCL: A parallel implementation of OrthoMCL for the Realm of Massive Genome availability  Ehsan S. Tabari1,\* and Zhengchang Su1  1Department of Bioinformatics and Genomics, University of North Carolina at Charlotte, 9201 University City Blvd, Charlotte, NC 28223.  Received on XXXXX; revised on XXXXX; accepted on XXXXX  Associate Editor: XXXXXXX |

[[1]](#footnote-2)\*abstract

**Motivation:** Finding orthologous genes is a primary step in every comparative genomic study. Orthologous group identification lay the basis for genome annotations. With the unprecedented increase in the number of fully sequenced genomes, conventional methods are becoming less and less feasible.

**Results:** Herein, we represent PorthoMCL, a parallel implementation of OrthoMCL, which will address the problem of finding orthologous genes among large number of genomes.

**Availability:** PorthoMCL (source, binaries, sample dataset and documentation) is available under the MIT license in the github repository: [github.com/etabari/PorthoMCL](https://github.com/etabari/OrthoMCLP). The results of ortholog identification for 2,758 prokaryotic genomes are available to download from: UPLOAD IT SOMEWHERE (10GB compressed [51gb uncompressed]).

**Contact:** sseyedit@uncc.edu.

# introduction

The advancements in speed, efficiency and affordability of whole-genome sequencing methods have offered the field of comparative genomics the potential to analyze individual genomes instead of reference assemblies. Such comparative analyses could be used to discover differences between individuals and the extent to which these differences affect their response to the environment. The rate at which this data is growing is outpacing the conventional comparative genomics methods and algorithms that is used to store, process and analyze. This requires new methods and algorithms as well as adaptation of existing procedures to which scientist are accustomed.

Orthology is a strong indication of functional conservation and, therefore, provides a functional annotation of experimentally undetermined proteins as well as a basis for individual genome comparison (Alexeyenko *et al.*, 2006). Orthologs are genes in different species that derive from a single gene in their last common ancestor and are created by speciation events, whereas paralogs are results of gene duplication within a species with a purpose of functional redundancy. If the duplication happened before speciation, they are called outparalogs, while inparalogs are when the duplication happened after speciation (Sonnhammer *et al.*, 2002).

OrthoMCL is one of the most widely used algorithms for generating orthologous groups across multiple eukaryotic genomes. Similar to many other orthology finding algorithm, it is based on reciprocal best hits in all-against-all BLAST searches of complete proteomes. To distinguish paralogs and orthologs, OrthoMCL identifies recent paralogs to be included in ortholog groups as within-species BLAST hits that are reciprocally better than between-species hits (Li *et al.*, 2003). It then applies the Markov Cluster algorithm (Van Dongen, 2000) it to group (putative) orthologs and paralogs.

# POrthoMCL

OrthoMCL relies on a relational database system to find reciprocal blast hits and assign scores to them. This becomes a computational bottleneck when the number of genomes in a study becomes large. Here, we present PorthoMCL, a parallel implementation of OrthoMCL that is scalable and can be ran for large number of genomes. PorthoMCL is platform independent and can be ran on a wide range of computing clusters. We have supplied a sample dataset and execution pipeline in the accompanying documentation for convenience. PorthoMCL does not require OrthoMCL to work. Options and arguments required at every step is discussed in detail in the documentation that accompanies PorthoMCL.

## Workflow

PorthoMCL’s workflow is very similar to the one of OrthoMCL. We have identified computationally intensive steps of OrthoMCL and parallelized them. Also, because PorthoMCL does not require a database server be present, steps that set up a database, load data to the database and retrieve data from it are missing. Figure 1 shows the steps of PorthoMCL in comparison to the ones of OrthoMCL. PorthoMCL only replaces the most computationally intensive step of OrthoMCL, orthomclFindPairs, with a parallel implementation. The PorthoMCL Find Pairs steps are marked in green, in contrast to OrthoMCL steps that are marked in blue. The darker blue boxes represent extra packages that OrthoMCL requires to run. PorthoMCL specific steps are designed to be executed in parallel on variety of high performance computing (HPC) environments. They are scalable and can exploit the capacity available to the HPC. However, the Find Pairs steps are not independent and they need to be ran in the designed order of execution. Each step builds on top of the previous step. The detail of these steps are:

PairsBestHit: It will keep only the highest scored hits from one genome to any other.

1. PairsOrthologs: It will look for reciprocal hits between different genomes and if passed a customizable threshold lists them as orthologs. A normalized score will be calculated for each orthology relationship. This step requires data generated in step (1).
2. PairsInParalogs: It will look for reciprocal hits in a genome and if passed a customizable threshold and better than all the Orthologs, lists them as inParalogs. A normalized score will be calculated for each relationship. This step requires data generated in steps (1) and (2).
3. PairsCoOrthologs: Finds all pairs of proteins across two genomes that are connected through ortholog and inParolog relationships. This step requires data generated in steps (1), (2) and (3).

The output of steps (2), (3) and (4) will be sent to the Markov Clustering to find orthologous, paralogous and co-orthologous groups respectively. This can be done in when their results becomes available.

## High performance computing support

Calculating orthology among a magnitude of individual genomes is a computationally intensive process. Therefore, PorthoMCL is designed to exploit high performance computing environments such as computing clusters or cloud computing. We have included a TORQUE script with the package that can be used for such environments. However, PorthoMCL like its counterpart OrthoMCL still runs on desktop and server computers without the requirement of having a database server present.

# RESULTS

We have calculated orthologs for all sequenced bacterial genomes using PorthoMCL to illustrate the power of PorthoMCL. We downloaded all the annotated genes for all the 2,758 prokaryotic organisms from GenBank which have a total of 8,661,583 protein sequences with the median length of 270 amino acids. They serve both as the query and the database for the all-against-all BLAST search. After splitting the query into smaller files each containing 10,000 sequences, we used PorthoMCL’s parallelizing script to run BLAST searches (e-value cutoff: 1e-5; database size: 1e8). The combined output of the BLAST contained 2,957,375,578 hits. The total runtime of the BLAST searches added up to 549 days, which were parallelized in 11 days using a computing cluster with 60 computing nodes (each nodes has 12 cores and 36GBs of RAM). At the next step, PorthoMCL searched for reciprocal best hits and identified 850,273,323 ortholog gene pairs that fell into 208,530 ortholog groups. While OrthoMCL never finished this step after 35 days on a database server (the server has 40 cores and 1TBs of RAM), PorthoMCL managed to finish the identification in 8 days.

The compressed ortholog pairs (file size: 10GBs) and ortholog groups (file size: 51MBs) are available for download at UPLOAD IT SOME WHERE.

acknowledgements

We thank Jonathan Halter for his support and valuable contributions.

Find Pairs

orthomcl  
AdjustFasta

orthomcl  
FilterFasta

All-v-All

BLAST

orthomcl

BlastParser

porthomcl  
PairsBestHit

porthomcl  
PairsOrthologs

porthomcl  
PairsInParalogs

porthomcl  
PairsCoOrthologs

porthomcl

DumpPairFiles

MCL

orthomcl

MclToGroups

**Fig. 1.** Workflow of PorthoMCL. Blue boxes are original OrthoMCL steps while PorthoMCL steps are colored green. Dark blue boxes are the external applications that OrthoMCL requires.

*Funding*: This work was funded by GRANT INFORMATION.

*Conflict of Interest*: none declared.

References

Alexeyenko, A., Lindberg, J., Pérez-Bercoff, A., & Sonnhammer, E. L. L. (2006). Overview and comparison of ortholog databases. *Drug Discovery Today. Technologies*, 3(2), 137–43. doi:10.1016/j.ddtec.2006.06.002

Li, L., Stoeckert, C. J., & Roos, D. S. (2003). OrthoMCL: identification of ortholog groups for eukaryotic genomes. *Genome Research*, *13*(9), 2178–89. doi:10.1101/gr.1224503

Sonnhammer, E. L. ., & Koonin, E. V. (2002). Orthology, paralogy and proposed classification for paralog subtypes. *Trends in Genetics*, 18(12), 619–620. doi:10.1016/S0168-9525(02)02793-2

Van Dongen, S. (2000). Graph Clustering by Flow Simulation.

PhD thesis, University of Utrecht, Netherlands.

1. \*To whom correspondence should be addressed. [↑](#footnote-ref-2)