METHOD

Recursive dynamic Markov clustering for fine-grained orthogroup classification

Stephen R Bond, Karl E Keat and Andreas D Baxevanis*

Abstract

Background: Blahh

Results: Blahh

Conclusions: Blahh

Keywords: Some; Awesome; Keywords

Background and rationale

When a gene evolves an important physiological role, purifying selection tends to maintain that function through evolutionary time [1, 2, 3]. As a result, orthology (i.e., homology via speciation) has become a widely used predictor of shared gene product function among species, with considerable effort made to develop computational methods for identifying orthologs. The algorithms in current use generally fall into two distinct categories: Tree-based and graph-based clustering methods [4]. Tree-based approaches (e.g., Ensembl Compara [5], LOFT [6], and SYNERGY [7]) broadly rely on estimating a phylogenetic tree for a target gene family, and then reconciling the gene tree with a 'known' species tree to identify orthologous clades. While tree-based methods are very accurate under ideal conditions, estimating the species trees they rely creates a considerable source of uncertainty [8]. Alternatively, pairwise similarity graph clustering methods leverage graph theory to rapidly identify groups of related sequences from genome scale datasets [REF]. Reciprocal best-hit methods were among the earliest developed for this purpose, but were restricted to assessing only two species at a time [REF]. Due to the non-transitive nature of orthology (i.e., paralogs in one species can be orthologous to a single gene in another species), it is more difficult (or impossible) to explicitly assign sequences into groups of pure orthologs [REF]. Instead, the term 'orthogroup' has come to represent a cluster of orthologs that may include closely related paralogs [REF]. InParanoid, EggNOG [9], and OMA [10] are popular tools for assigning sequences to orthogroups using a 'best-hit clique' approach, where closed best-hit sub-graphs are identified in the dataset. While accurate within each sub-graph, these methods tend to be overly strict in their assignment; this causes an under-representation of actual orthologous relationships among many species. Alternatively, Markov clustering (MCL) is very efficient at isolating more inclusive sub-graphs. OrthoMCL is one of the most popular MCL-based ortholog prediction methods, but it is prone to placing too many in-paralogs into orthogroups (i.e., it is less precise). In the current study we have increased the overall resolving power of de novo MCL-based orthogroup assignment with a number of novel enhancements, including refinement of the pairwise similarity metrics, using a supervised heuristic to dynamically select MCL parameters, recursively subdividing orthogroups, and testing putative orthogroups for best-hit cliques to maximize resolution.

BLAST scores (bit or e-value) have a strong length bias when calculating orthogroups [11]. OrthoFinder also uses a static inflation/edge similarity threashold [11]

Methods

- I used MAFFT [12], because it's awesome.

- The COG, KOG, arCOG databases may all be rich sources of data for validation. COGs are 'clusters of orthologous genes', which can includes many individual orthogroups.

- KEGG OCs may work in place of KOGs [13]

- Construct species trees with *BEAST (Heled and Drummond 2010) and BPP (Yang and Rannala 2014;

*Correspondence: andy@mail.nih.gov
Computational and Statistical Genomics Branch, Division of Intramural Research, National Human Genome Research Institute, National Institutes of Health, 50 South Drive, 20892 Bethesda, MD, USA
Full list of author information is available at the end of the article

Rannala and Yang 2016) to do a tree based comparison against RD-MCL.

- Possible sample data: CYP proteins [14]

- Might want to compare results agains OrthologFinder if appropriate [15]

- Construction of an ortholog ontology [16]?

- Try OrthoFinder length-normalized bit scores as similarity metric between sequences [11].

- Use precision and recall as measures of accuracy of simulated data

(https://en.wikipedia.org/wiki/Precision_and_recall and [11])

$Precision = \frac{TP}{TP+FP}$

$Recall = \frac{TP}{TP+FN}$

$F - Score = 2 * \frac{precision*recall}{precision+recall}$

Where TP is True Positive, FP is False Positives, and FN is False Negatives.

- Create raw clusters using MMseqs2.0 (https://github.com/soedinglab/MMseqs2).

- Output orthogroups in plain text and the *Quest for Orthologs* community standard OrthoXML [17]

Results and Discussion

This is a subsection header

Spill some ink regarding in/out paralogs [18, 4].

Conclusions

Can't wait to share my conclusions.

Competing interests

The authors declare that they have no competing interests.

Author's contributions

SRB is the lead developer of RD-MCL and wrote the manuscript, KEK contributed significantly to the code base, and ADB was involved in the design and coordination of the project. All authors read and approved the final manuscript.

Acknowledgements

This research was supported by the Intramural Research Program of the National Human Genome Research Institute, National Institutes of Health.

References

1. Altenhoff, A.M., Studer, R.A., Robinson-Rechavi, M., Dessimoz, C.: Resolving the Ortholog Conjecture: Orthologs Tend to Be Weakly, but Significantly, More Similar in Function than Paralogs. PLoS computational biology **8**(5), 1002514 (2012)
2. Rogozin, I.B., Managadze, D., Shabalina, S.A., Koonin, E.V.: Gene family level comparative analysis of gene expression in mammals validates the ortholog conjecture. Genome biology and evolution **6**(4), 754–762 (2014)
3. Kryuchkova-Mostacci, N., Robinson-Rechavi, M.: Tissue-Specificity of Gene Expression Diverges Slowly between Orthologs, and Rapidly between Paralogs. PLoS computational biology **12**(12), 1005274 (2016)
4. Tekaia, F.: Inferring Orthologs: Open Questions and Perspectives. Genomics insights **9**, 17–28 (2016)
5. Vilella, A.J., Severin, J., Ureta-Vidal, A., Heng, L., Durbin, R., Birney, E.: EnsemblCompara GeneTrees: Complete, duplication-aware phylogenetic trees in vertebrates. Genome research **19**(2), 327–335 (2009)
6. van der Heijden, R.T.J.M., Snel, B., van Noort, V., Huynen, M.A.: Orthology prediction at scalable resolution by phylogenetic tree analysis. BMC bioinformatics **8**, 83 (2007)
7. Wapinski, I., Pfeffer, A., Friedman, N., Regev, A.: Automatic genome-wide reconstruction of phylogenetic gene trees. Bioinformatics **23**(13), 549–58 (2007)
8. Xu, B., Yang, Z.: Challenges in Species Tree Estimation Under the Multispecies Coalescent Model. Genetics **204**(4), 1353–1368 (2016)
9. Jensen, L.J., Julien, P., Kuhn, M., von Mering, C., Muller, J., Doerks, T., Bork, P.: eggNOG: automated construction and annotation of orthologous groups of genes. Nucleic acids research **36**(Database), 250–254 (2007)
10. Roth, A.C., Gonnet, G.H., Dessimoz, C.: Algorithm of OMA for large-scale orthology inference. BMC bioinformatics **10**(1), 220 (2009)
11. Emms, D.M., Kelly, S.: OrthoFinder: solving fundamental biases in whole genome comparisons dramatically improves orthogroup inference accuracy. Genome biology **16**(1), 157 (2015)
12. Katoh, K., Standley, D.M.: MAFFT multiple sequence alignment software version 7: improvements in performance and usability. Molecular biology and evolution **30**(4), 772–780 (2013)
13. Nakaya, A., Katayama, T., Itoh, M., Hiranuka, K., Kawashima, S., Moriya, Y., Okuda, S., Tanaka, M., Tokimatsu, T., Yamanishi, Y., Yoshizawa, A.C., Kanehisa, M., Goto, S.: KEGG OC: a large-scale automatic construction of taxonomy-based ortholog clusters. Nucleic acids research **41**(Database issue), 353–7 (2013)
14. Pan, S.-T., Xue, D., Li, Z.-L., Zhou, Z.-W., He, Z.-X., Yang, Y., Yang, T., Qiu, J.-X., Zhou, S.-F.: Computational Identification of the Paralogs and Orthologs of Human Cytochrome P450 Superfamily and the Implication in Drug Discovery. International journal of molecular sciences **17**(7), 1020 (2016)
15. Horiike, T., Minai, R., Miyata, D., Nakamura, Y., Tateno, Y.: Ortholog-Finder: A Tool for Constructing an Ortholog Data Set. Genome biology and evolution **8**(2), 446–457 (2016)
16. Chiba, H., Nishide, H., Uchiyama, I.: Construction of an ortholog database using the semantic web technology for integrative analysis of genomic data. PloS one **10**(4), 0122802 (2015)
17. Dessimoz, C., Gabaldón, T., Roos, D.S., Sonnhammer, E.L.L., Herrero, J., Quest for Orthologs consortium: Toward community standards in the quest for orthologs. Bioinformatics **28**(6), 900–904 (2012)
18. Sonnhammer, E.L.L., Koonin, E.V.: Orthology, paralogy and proposed classification for paralog subtypes. Trends in genetics **18**(12), 619–620 (2002)

Figures

Figure 1 Sample figure title. A short description of the figure content should go here.

Figure 2 Sample figure title. Figure legend text.

Tables

Additional Files

Additional file 1 — Sample additional file title
Additional file descriptions text (including details of how to view the file, if it is in a non-standard format or the file extension). This might refer to a multi-page table or a figure.

Additional file 2 — Sample additional file title
Additional file descriptions text.

Table 1 List of optional third party software that BuddySuite programs can interact with. BuddySuite performs all necessary format conversion to call any of these tools and, where appropriate, returns the result in the same format as the input. This is particularly useful when creating multiple sequence alignments from annotated sequences in GenBank or EMBL format.

BuddySuite program	Third-party program	Reference
SeqBuddy	BLAST	[?]
AlignBuddy	Clustal Omega	[?]
	ClustalW2	[?]
	MAFFT	[12]
	MUSCLE	[?]
	PAGAN	[?]
	PRANK	[?]
PhyloBuddy	FastTree	[?]
	RAxML	[?]
	PhyML	[?]

Table 2 File format support for reading (R) and writing (W) provided by each BuddySuite module.

Format	SeqBuddy	AlignBuddy	PhyloBuddy
Clustal	R & W†	R & W	None
EMBL‡	R & W	R† / W	None
FASTA	R & W	R† / W	None
GenBank‡	R & W	R† / W	None
Nexus	R & W†	R & W	R & W
Newick	None	None	R & W
NeXML	None	None	R & W
PHYLIP (interleaved)	R & W†	R & W	None
PHYLIP (sequential)	R & W†	R & W	None
SeqXML	R & W	None	None
Stockholm	R & W†	R & W	None
Swissprot‡	R only	None	None

†All sequences must be the same length
‡Supports rich sequence annotation