OPEN ACCESS Freely available online

PLOS ONE

Quickly Finding Orthologs as Reciprocal Best Hits with BLAT, LAST, and UBLAST: How Much Do We Miss?

Natalie Ward, Gabriel Moreno-Hagelsieb*

Department of Biology, Wilfrid Laurier University, Waterloo, Ontario, Canada

Abstract

Reciprocal Best Hits (RBH) are a common proxy for orthology in comparative genomics. Essentially, a RBH is found when the proteins encoded by two genes, each in a different genome, find each other as the best scoring match in the other genome. NCBI's BLAST is the software most usually used for the sequence comparisons necessary to finding RBHs. Since sequence comparison can be time consuming, we decided to compare the number and quality of RBHs detected using algorithms that run in a fraction of the time as BLAST. We tested BLAT, LAST and UBLAST. All three programs ran in a hundredth to a 25th of the time required to run BLAST. A reduction in the number of homologs and RBHs found by the faster algorithms compared to BLAST becomes apparent as the genomes compared become more dissimilar, with BLAT, a program optimized for quickly finding very similar sequences, missing both the most homologs and the most RBHs. Though LAST produced the closest number of homologs and RBH to those produced with BLAST, UBLAST was very close, with either program producing between 0.6 and 0.8 of the RBHs as BLAST between dissimilar genomes, while in more similar genomes the differences were barely apparent. UBLAST ran faster than LAST, making it the best option among the programs tested.

Citation: Ward N, Moreno-Hagelsieb G (2014) Quickly Finding Orthologs as Reciprocal Best Hits with BLAT, LAST, and UBLAST: How Much Do We Miss? PLoS ONE 9(7): e101850. doi:10.1371/journal.pone.0101850

Editor: Valerie de Crécy-Lagard, University of Florida, United States of America

Received April 12, 2014; **Accepted** June 11, 2014; **Published** July 11, 2014

Data Availability: The authors confirm that all data underlying the findings are fully available without restriction. All files are available at http://microbiome.wlu.ca/Orthologs/.

Funding: Research supported by Wilfrid Laurier University and by a Discovery grant to GMH by The Natural Sciences and Engineering Research Council of Canada (NSERC). Wilfrid Laurier University provided funds for equipment. The Discovery grant from Natural Sciences and Engineering Research Council of Canada (NSERC) provided funds for equipment and publication fees. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: Gabriel Moreno-Hagelsieb is an academic editor for PLOS ONE. This does not alter the authors' adherence to PLOS ONE Editorial policies and criteria.

* Email: gmoreno@wlu.ca

Introduction

The purpose of this work is to evaluate the speed, number and quality of orthologs mapped as reciprocal best hits (RBHs) as detected and scored using NCBI's BLAST [1,2], the Blast-Like Alignment Tool (BLAT) [3], LAST [4], and UBLAST [5]. The need for this work stems from three main problems in comparative genomics: (i) The exponential increment in the number of genomes available in public databases; (ii) The concomitant need for methods to quickly find homologous sequences in general, and orthologs in particular, across available genomes (for definitions see below); (iii) The appearance of faster software for sequence comparison whose adequacy for particular tasks compared to commonly used software should be assessed.

Several research groups have made orthologs available through web services to a wider community (see for example: [6–12]). However, particular researchers might still prefer to make their own calculations due to reasons such as those that we have listed before [13]: (a) researchers' own newly sequenced genomes under analyses; (b) a need for updated ortholog mappings not available in published ortholog databases; (c) lack of agreement about the genome annotations to use, for instance, those provided by the authors of a genome, corrections such as those within the RefSeq database [14,15], the HAMAP project [16,17], or even those re-annotations produced by other research groups (*e.g.* [18–21]).

Orthologs, which could be referred to as the "same genes" in different species, are defined as homologous genes diverging after a speciation event [22]. Because of this evolutionary relationship, orthologous genes are expected to keep their original functions. Paralogs, defined as homologous genes diverging after a duplication event [22], have been proposed as a source of functional innovation [23,24], and are therefore less expected to have similar functions. Since it seems safer to infer similar functions between orthologs than between paralogs [25–28], it is important to be able to differentiate between orthologs and extra-paralogs, paralogous genes residing in different organisms [29].

Evidently, the definitions provided above are based on the event separating the histories of the genes in question. In practice, researchers rely on sequence similarity and suitable statistics for detecting homologs. After detecting putative homologs, producing evolutionary models such as phylogenetic trees, though performed by some groups (*e.g.* [30–32]), would be too computationally intensive to run in order to differentiate between orthologs and paralogs across available genomes. The growth of the sequence databases does not make a phylogenetic approach practical. Thus, most research in comparative genomics relies on shortcuts, or working definitions, for orthology. Probably the most common working definition of orthology is that of Reciprocal Best Hits (RBH) [33,34], whereby two genes residing in two different

genomes are deemed orthologs if their protein products find each other as the best hit in the opposite genome.

The task of finding homologs to a sequence of interest (the *query*) in a database containing many other sequences (the *subjects*) can be conceptualized as getting the best possible alignment of the query against all the subjects, scoring each of these alignments, and choosing those whose scores surpass a given threshold, or that comply with some alignment statistic. An exhaustive process using the dynamic programming algorithm by Smith and Waterman [35] could be so time consuming that researchers have developed heuristic algorithms. One of these heuristic algorithms, BLAST [36], has been the program of choice to compare proteins and therefore to produce RBHs, because of its speed compared to the exhaustive algorithm mentioned above, and to another heuristic algorithm, namely FASTA [37].

However, the constant increase in genomic sequences make it increasingly harder to rely on BLAST. With the pressure for faster results, other authors have produced faster heuristic algorithms. Among them, the most commonly used ones seem to be the BLAST-Like alignment Tool (BLAT) [3] and UBLAST [5], with the most recent addition of LAST [4]. These programs implement an indexed subject database, which allows to quickly find the most promising proteins to align; they use different methods to seed a pairwise alignment, such as stretches of identical amino-acid residues in BLAT, or variable size seeds implemented into a suffix tree in LAST; and they quickly drop the search for further protein comparisons to avoid wasting time on less likely matches. Further details can be found in the respective references and manuals [3–5]. While these programs run in a fraction of the time required to run BLAST, the speed comes at the cost of missing some matches otherwise found by BLAST.

In this work we used the genomes of four organisms: *Escherichia coli* K12 [38], *Bacillus subtilis* [39], *Methanosarcina mazei* Go1 [40], and *Saccharomyces cerevisiae* [41], as query genomes and a database of around 2750 genomes, to compare the speed, the number and the quality of orthologs found as RBH using four programs to finding similar protein sequences; namely, NCBI's BLAST, LAST, UBLAST, and BLAT.

Results and Discussion

The number of RBHs found decreases from BLAST to LAST to UBLAST to BLAT

Both BLAT and UBLAST ran in close to a hundredth of the time taken by NCBI's BLAST, while LAST ran in about a 25th of the time required for BLAST (Fig. 1). LAST was the program showing the most variation in time to run when compared to BLAST, as well as the most variation in numbers of homologs and reciprocal best hits found. This is probably due to LAST's use of adaptive alignment seeds, similar 'words' shared by sequences, in its strategy for quickly finding sequences that might produce a significant alignment. Adaptive seeds will be different in length and effectiveness across different databases. Thus, LAST's results with different sequence databases should vary the most when compared to results with BLAST, than results obtained using tools whose difference to the way BLAST works is more constant. For example, BLAT normally searches for identical 'tiles' of length 5 when comparing proteins before attempting an alignment.

The programs tested can be ordered from the one producing the highest number of RBHs to the one producing the lowest number of RBHs as BLAST>LAST>UBLAST>BLAT (paired t-tests $p < 1 \times 10^{-9}$; Table set S1 and Table set S2). As it might be expected, the decrease in the number of RBHs found by the faster programs (LAST, UBLAST and BLAT) becomes more pronounced with the overall dissimilarity between the genomes compared (Fig. 2a, 2b; Table sets S1 and S2). The lowest proportion of RBHs found by LAST was close to 0.8 of those found by BLAST, while for UBLAST it was between 0.6 and 0.7. However, these proportions remained very close to 1 in other, more similar, genomes. Given that BLAT is optimized for quickly finding very similar nucleotide sequences [3], it was the program producing the lowest number of RBHs. BLAT showed a quick loss of sensitivity with genome dissimilarity. The program did not find RBHs for a few genome comparisons, and found just a few RBHs between distantly related genomes (Tables with RBHs and homologs are available at: http://microbiome.wlu.ca/Orthologs/).

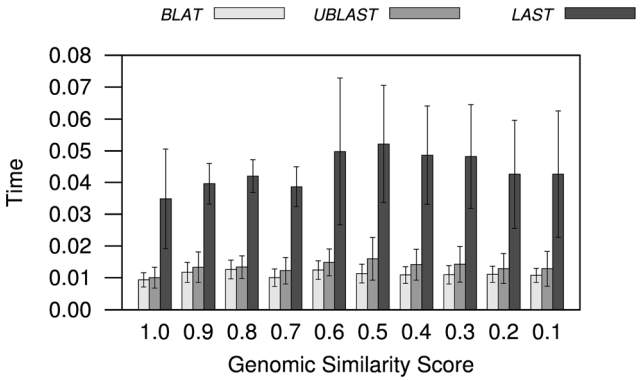

Figure 1. Difference in run time. Both UBLAST and BLAT ran in about a hundredth of the time as NCBI's BLAST, while LAST ran in about a 25th of the time required for BLAST. Note: as in Figures 2 and 4, the bars represent averages for pairwise genome comparisons involving genomes binned at intervals of 0.1 of Genomic Similarity Score (*GSS*); the number of genomes at each *GSS* bin is not the same; and the error bars show standard deviations representing the variability of results among the genomes at each bin.
doi:10.1371/journal.pone.0101850.g001

We also accounted for the number of genes finding homologs (Fig. 2b; Table set S1) and the number of homologous pairs (a gene can have more than one match, and therefore could produce more than one homologous pair) (Fig. 2d; Table set S2). The number of genes finding homologs showed similar tendencies as the number of RBHs above suggesting that the differences in the number of RBHs found is related to a corresponding difference in the number of genes finding homologs. UBLAST had a tendency to find a higher proportion of genes with RBHs per gene finding a homolog than any other program (paired t-tests $p < 1 \times 10^{-9}$), while BLAT had a tendency to find the fewest RBHs per gene finding a homolog (Fig. 2a, 2b). The number of homologous pairs was always smaller for the fastest programs (Fig. 2d). UBLAST and LAST produced the highest proportion of RBHs per homologous pair, while BLAT produced the lowest proportion of RBHs per homologous pair (Fig. 2c, 2d). These results suggest that another source or differences in RBHs is the search depth. UBLAST and LAST would have smaller sources of conflict to decide RBHs than BLAST. However, if the number of homologous pairs is too low, as it is in BLAT, then the reciprocal results might be lacking and RBHs might not be found.

Homologous pairs found by BLAT, UBLAST and LAST are subsets of those found by BLAST

As expected, BLAT, UBLAST and LAST found fewer homologous pairs than BLAST did (Fig. 2b, 3), and most of the matching pairs found by the faster algorithms were subsets of those

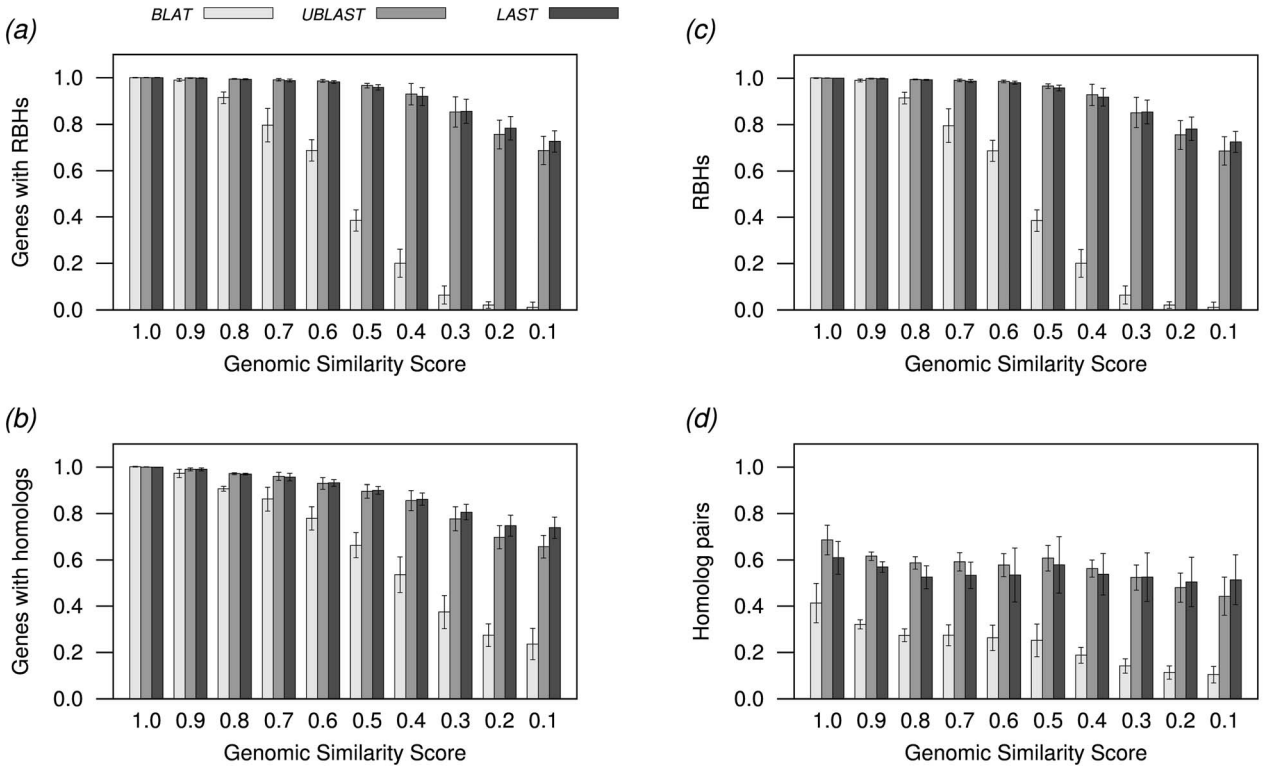

Figure 2. Differences in number of homologs and reciprocal best hits (RBHs). All numbers were normalized against the corresponding numbers obtained with NCBI's BLAST. As expected, the faster programs found fewer RBHs than BLAST. Such differences become more evident as the similarity between genomes decreases (as the genomic similarity score, *GSS*, decreases). This effect was much more pronounced for BLAT. The number of genes finding RBHs (a) and the total RBHs (c) was so small that the two graphs are almost identical. However, the differences between genes finding homologs and the total number of homologous pairs is much more apparent. The number of homolog pairs was always smaller for the fastest programs than for BLAST, suggesting that a good proportion of the differences in RBHs found is due to a lower search depth by the faster programs. See Note in Figure 1.
doi:10.1371/journal.pone.0101850.g002

found by the slowest (Fig. 3), with only 1.2% of the total homologous pairs not being detected by BLAST (0.3% found only by LAST, plus 0.8% found only by UBLAST, and 0.1% found by both UBLAST and LAST—note the intersection between the results of these two programs in the Venn diagram).

The Venn diagram on RBHs, however, was not what would be expected from that of the homologous pairs (Fig. 3). Of the total RBHs found by all programs combined, 3.2% were detected only by LAST; 4% were detected only by UBLAST, and 2% were detected by both UBLAST and LAST, but not by either of BLAST and BLAT. Of the same total RBHs found by all programs, 0.3% were detected only by BLAT. The difference in comparison to the homologous pairs results is most probably due to differences in the scoring systems, since, for example, BLAST modifies its scores by taking into account edge effects, and compositional biases in the sequences being compared [42–45].

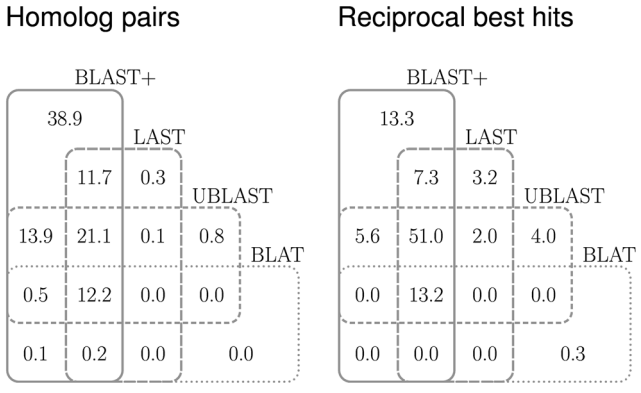

Figure 3. Matching pairs. Most homologous pairs found by LAST, UBLAST and BLAT were also found by BLAST. Differences in scoring resulted in a higher proportion of different RBHs found by either programs than would be expected from the sets of homologous pairs.
doi:10.1371/journal.pone.0101850.g003

The error rates are highest with BLAT

To estimate error rates, we analyzed conservation of adjacency of homologous genes (synteny). Conservation of gene order has been previously suggested to be of limited use for the assignment of orthology in prokaryotes due to the high divergence of gene order prevailing in these organisms [34]. However, synteny can still be used to test for the relative quality of predicted orthologs [13,46,47].

The error rates increased with the evolutionary distance as measured using Genomic Similarity Scores (Fig. 4; Table set S3). These error rates were more similar for RBH produced between closely related genomes (Fig. 4). Error rates using LAST and UBLAST were similar to those produced by BLAST, except between the least similar genomes (Fig. 4; Table set S3), where both programs showed higher error rates than BLAST, and UBLAST having higher error rates than LAST among the most divergent genomes. BLAT consistently produced the highest error

rates. Though error rates across the board are high among more divergent genomes, we must bear in mind that errors might relate to biologically meaningful events whose probability increases with divergence. For example, gene conversions (recombination between homologous genes), or gene divergences so high that their status as orthologs or extra-paralogs are barely discernible. One more biological source of confusion might be the possibility of co-orthology, which occurs when a duplication event happens after a speciation event [22]. It is possible that some apparent errors arise from divergent co-orthologs that therefore produce slightly different results. However, since the background biological events should be the very same, the difference in error rates should still reflect differences in the results obtained with each program.

Concluding remarks

We tested three programs that run considerably faster than BLAST for the task of detecting reciprocal best hits (RBHs). These programs have options that alter their default running methods in ways that might improve their performance in terms of sensitivity and thus increase the proportion of homologs found when compared to BLAST. Such increase might result in a concomitant increase in detection of RBHs. However, changing those options also increases the time required to run these programs. Thus, playing with options to try and attain results somewhat more similar to those obtained with BLAST would defy the purpose of this work.

While evaluating the results presented, we must bear in mind that none of the programs tested, not even NCBI's BLAST, was designed for the task of finding reciprocal best hits. The results show that, as would be expected from programs that miss homologs otherwise found by BLAST, the number of RBHs found by LAST, UBLAST and BLAT are mostly a subset of those found by NCBI's BLAST. When dealing with the most dissimilar genomes, both LAST and UBLAST kept between 0.6 and 0.8 of the number of RBHs found by BLAST. Results with more closely related genomes were more similar to those produced using BLAST. Given that BLAT is optimized for finding very similar sequences quickly, it should not be surprising that it missed most of the RBHs between the least similar genomes, and a high proportion of RBHs in the rest. Overall, UBLAST might be the best compromise between speed and sensitivity of the faster programs tested.

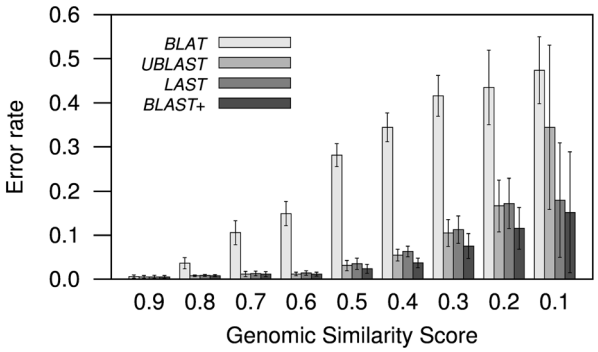

Figure 4. Error rate estimated using conservation of gene order. The estimate consists on false negatives (a paralog conserved next to a RBH) divided by the sum of false negatives + true positives (RBHs showing conservation of gene order). BLAST consistently showed the lowest error rates. Both LAST and UBLAST showed the most similar error rates to those produced by BLAST except when the genomes compared had low *GSS*, where UBLAST had higher error rates than LAST. BLAT showed the highest error rates. See Note in Figure 1.
doi:10.1371/journal.pone.0101850.g004

Methods

In this work we used the genomes of *Escherichia coli* K-12 MG1655 (uid57779) [38], *Bacillus subtilis* 168 (uid57675) [39], *Methanosarcina mazei* Go1 (uid57893) [40], and of *Saccharomyces cerevisiae* (uid128) [41], as testing genomes, and compared their annotated protein sequences against those annotated in the 2754 prokaryotic and fungal genomes available at the RefSeq database [14,15] (ftp://ftp.ncbi.nih.gov/genomes/Bacteria/) by the end of December, 2013. We calculated Genomic Similarity Scores (*GSSa*) as described previously [29,48,49]. Briefly, the *GSS* is the normalized BLAST bit score of all reciprocal best hits between any two genomes. In this work *GSS*s were calculated with RBHs produced from NCBI's BLAST results.

The protein sequence comparisons were performed using four programs: (i) NCBI's BLAST version 2.2.28+ [2], which is the BLAST suite of programs implemented in C++; (ii) LAST version 393 [4]; (iii) UBLAST, as implemented in the sequence analysis multitool USEARCH (version 7.0.1001) [5]; and (iv) the BLAST-Like Alignment Tool (BLAT) [3] version 35. We compiled all these programs at 64 bits, except for USEARCH, which is kindly provided by the author precompiled at 32 bits for academic use. All sequence comparisons were run with testing genomes as queries and database genomes as subjects, as well as database genomes as queries and testing genomes as subjects (reciprocal sequence comparisons).

The specific command lines used to run each program are presented in Table 1. The options for NCBI's BLAST different to the defaults were a maximum *E-value* threshold of 1×10^{-6} (*-evalue 1e-6*), and a final Smith-Waterman alignment (*-use_sw_tback*). For UBLAST we also specified an e-value threshold of 1×10^{-6} (*-evalue 1e-6*). Since LAST and BLAT do not offer an option to control e-value thresholds, they were run with default values only (BLAT's minimal score is 30, and minimal identity is 25%). However, BLAT calculates an e-value when the output sequence is specified as "blast8" (*-out = blast8*). LAST's e-values can be estimated using the command *lastex* from this program suite. We therefore filtered BLAT and LAST results using their calculated e-values during the process of finding reciprocal best hits. We also required coverage of at least 50% of any of the protein sequences in the alignments.

Finding best hits involved sorting the results for a query-genome-to-subject-genome comparison from highest to lowest score. The first hit for each query protein within the sorted results would therefore be the best hit. If the next hit had the very same score there would be more than one best hit (the method can therefore produce co-orthologs). We performed the very same procedure for the results ran in the opposite direction. That is, for the results where the subject genome was used as a query, and the query genome was used as a subject. Finally, to find orthologs as reciprocal best hits, for each best hit found by a query protein in the first direction, we checked if it found this query gene as a best hit in the opposite direction.

To estimate error rates in orthology detection, we used a test based on synteny [13,46,47]. For every pair of adjacent genes in the testing genomes, we found pairs of corresponding adjacently conserved homologs in any other genome. We then checked if those conserved homologs were also RBHs. If both conserved homologs were RBHs, the pair was considered to consist of two true positives (*TP*). If one gene was a RBH, but the other was not, then we counted the former as a *TP*, and the latter as a false

Table 1. Command lines used to run each genome to genome comparison.

Command lines
blastp -num_threads 2 -evalue 1e-6 -use_sw_tback -outfmt 6 -query query_genome -db subject_genome > result
lastal -f 0 subject_genome query_genome.faa > result
usearch7 -ublast query_genome.faa -db subject_genome.udb -evalue 1e-6 -blast6out result
blat -prot subject_genome.faa query_genome.faa -out = blast8 result

doi:10.1371/journal.pone.0101850.t001

negative (*FN*). With these definitions, error rates (*ER*) were calculated as:

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

Note that despite this measure might not test for other measures of quality, like true negatives and false positives, the aim here is not to produce a standard error rate, but to compare the relative quality of results among the programs tested.

Supporting Information

Table Set S1 These tables contain counts for genes finding reciprocal best hits and genes finding homologs organized on a per query genome fashion. The directory contains the R-script used to run the t-tests comparing the results from each program, and a table with the results of these t-tests.
(ZIP)

Table Set S2 These tables contain reciprocal best hit counts and homolog pair counts organized on a per query genome fashion. The directory contains the R-script used to run the t-tests comparing the results from each program, and a table with the results of these t-tests.
(ZIP)

Table Set S3 These tables contain counts for error estimates organized on a per query genome fashion. The directory contains the R-script used to run the t-tests comparing the results from each program, and a table with the results of these t-tests.
(ZIP)

Acknowledgments

GMH acknowledges the Shared Hierarchical Academic Research Computing Network (SHARCNET) for computer cluster usage.

Author Contributions

Conceived and designed the experiments: GMH. Performed the experiments: NW GMH. Analyzed the data: NW GMH. Contributed reagents/materials/analysis tools: NW GMH. Contributed to the writing of the manuscript: GMH NW.

References

1. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, et al. (1997) Gapped BLAST and PSIBLAST: a new generation of protein database search programs. Nucleic Acids Res 25: 3389–3402.
2. Camacho C, Coulouris G, Avagyan V, Ma N, Papado poulos J, et al. (2009) BLAST+: architecture and applications. BMC Bioinformatics 10: 421.
3. Kent WJ (2002) BLAT–the BLAST-like alignment tool. Genome Res 12: 656–64.
4. Kie lbasa SM, Wan R, Sato K, Horton P, Frith MC (2011) Adaptive seeds tame genomic sequence comparison. Genome Res 21: 487–493.
5. Edgar RC (2010) Search and clustering orders of magnitude faster than blast. Bioinformatics 26: 2460–1.
6. Tatusov RL, Fedorova ND, Jackson JD, Jacobs AR, Kiryutin B, et al. (2003) The cog database: an updated version includes eukaryotes. BMC Bioinformatics 4: 41.
7. Alexeyenko A, Tamas I, Liu G, Sonnhammer EL (2006) Automatic clustering of orthologs and inparalogs shared by multiple proteomes. Bioinformatics 22: e9–15.
8. Deluca TF, Wu IH, Pu J, Monaghan T, Peshkin L, et al. (2006) Roundup: a multi-genome repository of orthologs and evolutionary distances. Bioinformatics 22: 2044–6.
9. Powell S, Forslund K, Szklarczyk D, Trachana K, Roth A, et al. (2014) eggNOG v4.0: nested orthology inference across 3686 organisms. Nucleic acids research 42: D231–9.
10. Nakaya A, Katayama T, Itoh M, Hiranuka K, Kawashima S, et al. (2013) KEGG OC: a large-scale automatic construction of taxonomy-based ortholog clusters. Nucleic acids research 41: D353–7.
11. Whiteside MD, Winsor GL, Laird MR, Brinkman FSL (2013) OrthologeDB: a bacterial and archaeal orthology resource for improved comparative genomic analysis. Nucleic acids research 41: D366–76.
12. Waterhouse RM, Tegenfeldt F, Li J, Zdobnov EM, Kriventseva EV (2013) OrthoDB: a hierarchical catalog of animal, fungal and bacterial orthologs. Nucleic acids research 41: D358–65.
13. Moreno-Hagelsieb G, Latimer K (2008) Choosing BLAST options for better detection of orthologs as reciprocal best hits. Bioinformatics 24: 319–324.
14. Maglott DR, Katz KS, Sicotte H, Pruitt KD (2000) NCBI's LocusLink and RefSeq. Nucleic Acids Res 28: 126–128.
15. Tatusova T, Ciufo S, Fedorov B, O'Neill K, Tolstoy I (2014) RefSeq microbial genomes database: new representation and annotation strategy. Nucleic acids research 42: D553–9.
16. Boeckmann B, Bairoch A, Apweiler R, Blatter MC, Estreicher A, et al. (2003) The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. Nucleic Acids Res 31: 365–70.
17. Gattiker A, Michoud K, Rivoire C, Auchincloss AH, Coudert E, et al. (2003) Automated annotation of microbial proteomes in SWISS-PROT. Comput Biol Chem 27: 49–58.
18. Besemer J, Lomsadze A, Borodovsky M (2001) GeneMarkS: a self-training method for prediction of gene starts in microbial genomes. implications for finding sequence motifs in regulatory regions. Nucleic Acids Res 29: 2607–2618.
19. Aziz RK, Bartels D, Best AA, DeJongh M, Disz T, et al. (2008) The RAST Server: rapid annotations using subsystems technology. BMC Genomics 9: 75.
20. Chen IMA, Markowitz VM, Chu K, Anderson I, Mavromatis K, et al. (2013) Improving microbial genome annotations in an integrated database context. PLoS ONE 8: e54859.
21. Wattam AR, Abraham D, Dalay O, Disz TL, Driscoll T, et al. (2014) PATRIC, the bacterial bioinformatics database and analysis resource. Nucleic acids research 42: D581–91.
22. Fitch WM (2000) Homology a personal view on some of the problems. Trends Genet 16: 227–231.
23. Ohno S (1970) Evolution by gene duplication. Berlin: Springer-Verlag.
24. Francino MP (2005) An adaptive radiation model for the origin of new gene functions. Nat Genet 37: 573–7.
25. Altenhoff AM, Studer RA, Robinson-Rechavi M, Dessimoz C (2012) Resolving the ortholog conjecture: orthologs tend to be weakly, but significantly, more similar in function than paralogs. PLoS Computational Biology 8: e1002514.
26. Chen X, Zhang J (2012) The Ortholog Conjecture Is Untestable by the Current Gene Ontology but Is Supported by RNA Sequencing Data. PLoS Computational Biology.
27. Thomas PD, Wood V, Mungall CJ, Lewis SE, Blake JA, et al. (2012) On the Use of Gene Ontology Annotations to Assess Functional Similarity among Orthologs and Paralogs: A Short Report. PLoS Computational Biology 8: e1002386.
28. Gabaldón T, Koonin EV (2013) Functional and evolutionary implications of gene orthology. Nature Reviews Genetics 14: 360–366.

29. Janga SC, Moreno-Hagelsieb G (2004) Conservation of adjacency as evidence of paralogous operons. Nucleic Acids Res 32: 5392–7.
30. Ruan J, Li H, Chen Z, Coghlan A, Coin LJM, et al. (2008) TreeFam: 2008 Update. Nucleic Acids Res 36: D735–40.
31. Vilella AJ, Severin J, Ureta-Vidal A, Heng L, Durbin R, et al. (2009) EnsemblCompara GeneTrees: Complete, duplication-aware phylogenetic trees in vertebrates. Genome Research 19: 327–335.
32. Afrasiabi C, Samad B, Dineen D, Meacham C, Sjölander K (2013) The PhyloFacts FAT-CAT web server: ortholog identification and function prediction using fast approximate tree classification. Nucleic Acids Res 41: W242–8.
33. Tatusov RL, Koonin EV, Lipman DJ (1997) A genomic perspective on protein families. Science 278: 631–637.
34. Bork P, Dandekar T, Diaz-Lazcoz Y, Eisenhaber F, Huynen M, et al. (1998) Predicting function: from genes to genomes and back. J Mol Biol 283: 707–25.
35. Smith TF, Waterman MS (1981) Identification of common molecular subsequences. J Mol Biol 147: 195–197.
36. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ (1990) Basic local alignment search tool. J Mol Biol 215: 403–10.
37. Pearson WR (1990) Rapid and sensitive sequence comparison with fastp and fasta. Methods Enzymol 183: 63–98.
38. Blattner FR, Plunkett G 3rd, Bloch CA, Perna NT, Burland V, et al. (1997) The complete genome sequence of *Escherichia coli* K-12. Science 277: 1453–1474.
39. Kunst F, Ogasawara N, Moszer I, Albertini AM, Alloni G, et al. (1997) The complete genome sequence of the gram-positive bacterium Bacillus subtilis. Nature 390: 249–256.
40. Deppenmeier U, Johann A, Hartsch T, Merkl R, Schmitz RA, et al. (2002) The genome of Methanosarcina mazei: evidence for lateral gene transfer between bacteria and archaea. Journal of molecular microbiology and biotechnology 4: 453–461.
41. Goffeau A, Barrell BG, Bussey H, Davis RW, Dujon B, et al. (1996) Life with 6000 genes. Science 274: 546–567.
42. Schaffer AA, Aravind L, Madden TL, Shavirin S, Spouge JL, et al. (2001) Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements. Nucleic Acids Res 29: 2994–3005.
43. Altschul SF, Wootton JC, Gertz EM, Agarwala R, Morgulis A, et al. (2005) Protein database searches using compositionally adjusted substitution matrices. The FEBS journal 272: 5101–5109.
44. Gertz EM, Yu YK, Agarwala R, Schäffer AA, Altschul SF (2006) Composition-based statistics and translated nucleotide searches: improving the TBLASTN module of BLAST. BMC biology 4: 41.
45. Yu YK, Gertz EM, Agarwala R, Schäffer AA, Altschul SF (2006) Retrieval accuracy, statistical significance and compositional similarity in protein sequence database searches. Nucleic Acids Res 34: 5966–5973.
46. Dutilh BE, van Noort V, van der Heijden RTJM, Boekhout T, Snel B, et al. (2007) Assessment of phylogenomic and orthology approaches for phylogenetic inference. Bioinformatics 23: 815–824.
47. Dessimoz C, Gabaldón T, Roos DS, Sonnhammer ELL, Herrero J, et al. (2012) Toward community standards in the quest for orthologs. Bioinformatics 28: 900–904.
48. Moreno-Hagelsieb G, Collado-Vides J (2002) A powerful non-homology method for the prediction of operons in prokaryotes. Bioinformatics 18 Suppl 1: S329–S336.
49. Moreno-Hagelsieb G, Wang Z, Walsh S, ElSherbiny A (2013) Phylogenomic clustering for selecting non-redundant genomes for comparative genomics. Bioinformatics 29: 947–949.