picture(0,0)(-35,0)(1,0)30 (0,35)(0,-1)30 picture picture(0,0)(35,0)(-1,0)30 (0,35)(0,-1)30 picture

TBD
doi.TBD
Advance Access Publication Date: Day Month Year
Sequence analysis

Sequence analysis

Progress in quickly finding orthologs as reciprocal best hits

Julie Hernández-Salmerón and Gabriel Moreno-Hagelsieb*

Department of Biology, Wilfrid Laurier University, 75 University Ave W, Waterloo, Ontario, Canada N2L 3C5.

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: While the authors of software for the quick comparison of protein sequences evaluate the speed of their software and compare their results against the most usual software for the task, it is not common for them to evaluate their software for more particular uses, such as finding orthologs as reciprocal best hits (RBH). Here we focused on comparing RBH results using software that runs faster than blastp. Namely, lastal, diamond, and MMseqs2.

Results: Of the programs tested, lastal required the least time to produce results. However, it produced fewer results than any other program when comparing evolutionarily distant genomes. The program producing the most similar number of RBH as blastp was MMseqs2. This program was also resulted in the lowest error estimates compared to any of the programs used. The results with diamond were very close to those obtained with MMseqs2, with diamond running much faster. Our results suggest that the best of the programs tested was diamond, ran with the "sensitive" option, which took 7% of the time as blastp to run, and produced results with lower error rates than blastp.

Availability: A program to obtain reciprocal best hits using each sequence comparison program is maintained at https://github.com/Computational-conSequences/SequenceTools

Contact: gmorenohagelsieb@wlu.ca

Supplementary information: There is no Supplementary data associated with this manuscript.

1 Introduction

Finding orthologs is a determining step in comparative genomics and represents a central concept in evolution. Orthologs are defined as characters that diverge after a speciation event (Fitch, 2000). In other words, if the characters are genes, then they are thought of as the same genes in different species. It is expected that they typically conserve their original function, an inference that has been supported by several lines of evidence (Chen and Zhang, 2012; Altenhoff *et al.*, 2012; Gabaldón and Koonin, 2013; Esposito and Moreno-Hagelsieb, 2018).

Efforts in standardizing orthology inference methods remain in constant evaluation, with over forty web services already exist available to the community (Dessimoz *et al.*, 2012; Boeckmann *et al.*, 2016). Few of these methods are based on phylogenetic analyses (tree-based approach), which, despite expected to be the most accurate, tend to be computationally intensive and impractical for big databases (Kuzniar *et al.*, 2008; Kristensen *et al.*, 2011). Some methods employ pairwise sequence similarity comparisons (graph-based methods) that have been successful to implement, such as the clusters of orthologous groups (COGs) database (Tatusov *et al.*, 2000; Galperin *et al.*, 2017). However, researchers require having their own sets of orthologs, since genome sequencing has become a commonly available technology.

Perhaps the most common approach for inferring orthologs, or working definition, is Reciprocal Best Hits (RBH), which is a simple method that has shown low false-positive rates and ease of implementation (Moreno-Hagelsieb and Latimer, 2008; Kristensen *et al.*, 2011; Wolf and Koonin, 2012). Essentially, if gene a in genome A finds gene b as its best, highest-scoring, match in genome B; and gene b finds gene a as its best match in genome A, they are RBH and thus inferred to be orthologs. The most commonly used program for finding sequence matches for finding RBH is blastp (Camacho *et al.*, 2009). This program was chosen for being the fastest program available at the time when comparative genomics started. However, the amount of sequences to analyze continues to grow exponentially, making the speed of blastp comparisons too slow for the demand of sequence analysis.

When authors introduce a software suite for sequence comparison, they often compare the speed and overall performance of their software compared to blastp. Since speed tends to come at a cost in sensitivity and accuracy, the author reports might include differences in performance in overall sequence comparison and number of detected sequences. However, more specialized usages, like detecting orthologs as RBH might be affected differently compared to overall performance. Thus, it becomes necessary

© The Author 2020.

picture(0,0)(-35,0)(1,0)30 (0,-35)(0,1)30 picture picture(0,0)(35,0)(-1,0)30 (0,-35)(0,1)30 picture

picture(0,0)(-35,0)(1,0)30 (0,35)(0,-1)30 picture picture(0,0)(35,0)(-1,0)30 (0,35)(0,-1)30 picture

to test the adequacy of the software in particular tasks. Accordingly, prior work compared the performance of three fast programs as compared to blastp (Ward and Moreno-Hagelsieb, 2014). The programs tested were blat (Kent, 2002), ublast (Edgar, 2010) and lastal (Kiełbasa *et al.*, 2011). Since then, two somewhat recently developed programs for sequence comparison include diamond (Buchfink *et al.*, 2015) and MMseqs2 (Steinegger and Söding, 2017) (from now on mmseqs). Here we use lastal as a reference to the previous report (Ward and Moreno-Hagelsieb, 2014), where lastal was the program producing the most-similar-to-blastp results, and test these two new programs in terms of the proportion of RBH found and their quality compared to blastp.

2 Methods

The annotated protein sequences of 3,312 non-redundant genomes were used as subjects for comparisons against the genome of *Escherichia coli* K-12 MG1655 (GCF_000005845) as the query. The non-redundant representatives were selected from Ĩ6,000 complete genomes available at NCBI's refSeq genome database (Haft *et al.*, 2018) by January 2020. To select these genomes, we clustered them using a trinucleotide DNA signature (Campbell *et al.*, 1999), with a δ-distance cutoff of 0.04 as described previously (Moreno-Hagelsieb *et al.*, 2013), resulting in 3312 clusters. A distance that roughly corresponds to a "species" level. We took one genome per cluster.

Four programs were used to perform protein sequence comparisons: (1) BLAST version 2.10.0+ (Camacho *et al.*, 2009), (2) Diamond version 0.9.30 (Buchfink *et al.*, 2015), (3) LASTAL version 1045 (Kiełbasa *et al.*, 2011), and (4) MMseqs2 version 11-e1a1c (Steinegger and Söding, 2017).

To compare times, the query *versus* targets comparisons were run one by one, in the same computer, with no other programs running at the same time. Times were obtained by using the unix "time" command. This command reports user, cpu and real times. The plotted times were the real times.

To find Reciprocal Best Hits (RBH), we wrote a program, *getRBH.pl*, to standardize the options and outputs from the different sequence comparison programs, following the options previously described by Ward and Moreno-Hagelsieb (2014), the e-value threshold was 1×10^{-6} (1e-6), coverage of 60% of the shortest protein in the alignment, as well as soft-masking and Smith-Waterman alignment, when available (diamond does not have soft masking, thus it was run with no masking). The latter two previously found to improve the finding of RBH (Moreno-Hagelsieb and Latimer, 2008).

Genomic Similarity Scores (GSS) were calculated from blastp results as the sum of the bit scores of all reciprocal best hits *(compScore)* divided by the bit scores of each corresponding ortholog against itself *(selfScore)*, as described in (Moreno-Hagelsieb *et al.*, 2013).

Error estimates in detecting orthologs were calculated following conservation of gene order as means for testing quality. For this, we analyzed adjacent homologs genes, assuming that a pair of conserved genes in the target genome will be both orthologs to the corresponding pair of genes in the query genome (Moreno-Hagelsieb and Latimer, 2008).

3 Results and Discussion

3.1 Runtimes

The computing speeds for finding homologs were timed for each program relative to blastp. Of all the programs tested, lastal was the fastest, finding the corresponding homologous proteins in 1.5% of the time spent by blastp (Fig. 1). The second fastest program was diamond, which took around 7% as long as blastp to run. The program mmseqs took around 28% as long to run as blastp.

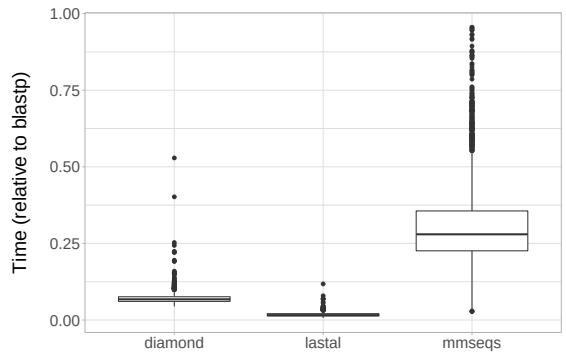

Fig. 1. Difference in speed relative to blastp. Lastal is the fastest program of the three we tested.

The software diamond has three sensitivity modes, "fast", "sensitive" and "more-sensitive". The article presenting the software showed that diamond produced much better results with the sensitive mode than under the default, fast, mode (Buchfink *et al.*, 2015). Thus, our tests here were run with the sensitive option.

All four programs can work with their own target (also called subject) databases. Therefore, we built databases for all programs. We ran mmseqs with the "easy-search" option. This option accepts the target either in plain fasta format, or as a formatted database, but the query has to remain in plain fasta format. However, mmseqs common searches, using the "search" option instead of easy-search, work with databases built for both, query and target. The result is produced as a database too. The results database can then be used to produce other desired output formats. The program can also produce indexes for its databases. We decided not to built indexes because they take very long to be built and use too much space. For example, the database for the longest of the subject genomes (16,040,666 bp) used 5.7M of space, which increased to 898M when building the index.

The mmseqs software also has a rbh function. However, we decided not to use it because we preferred to keep control of the parameters used to produce RBH.

3.2 Reciprocal best hits

The proportion of reciprocal best hits found using the faster programs was evaluated with respect to blastp (Fig. 2). Our results showed a very similar proportion of RBH when genomes are more similar to *E. coli* K12 MG1655. As the similarity decreases, so does the proportion of RBH. However, diamond and mmseqs detected more RBH even at the lowest GSS values compared to lastal (Fig. 2A), which has a significant impact being mmseqs the slowest program, it shows a better efficiency in finding orthologs even amongst distant genomes.

When we compared the numbers of RBH found, about 73% were detected by all programs (Fig. 2B). The second important group shows diamond and mmseqs sharing the majority of RBH with blastp (making up a total of 81.4%). Of the total RBH, 3.2% were only found by blastp, while 7.4% was detected by the fast programs but not by blastp. Unlike our previous analysis which showed higher percentages of RBH only detected by blastp (Ward and Moreno-Hagelsieb, 2014), yet similar numbers of unique RBH were produced by both, mmseqs and lastal (3.1% and 3.5% respectively). These percentages might represent whether those RBH are gained when choosing one or another program, or that number of orthologs might be identified as false positives. Although, no significant difference between lastal, mmseqs and blast was observed, choosing diamond would result in missing less than 1% RBH not obtained by the other programs, or probably obtaining less false positives results.

picture(0,0)(-35,0)(1,0)30 (0,-35)(0,1)30 picture picture(0,0)(35,0)(-1,0)30 (0,-35)(0,1)30 picture

picture(0,0)(-35,0)(1,0)30 (0,35)(0,-1)30 picture picture(0,0)(35,0)(-1,0)30 (0,35)(0,-1)30 picture

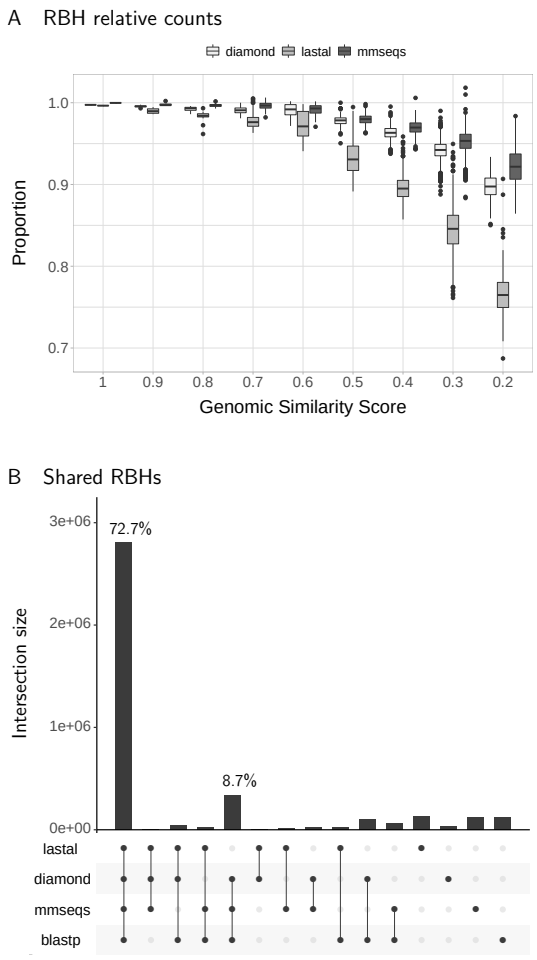

Fig. 2. Reciprocal best hits. A. The proportion of RBH obtained with the fast programs, relative to the number obtained with blastp, is very similar when genomes are very similar to the query genome (here E. coli K12 MG1655). As genomes become less similar, the proportion falls. Both diamond and mmseqs showed improved proportions at low overall similarity compared to lastal. B. Close to 73% of the RBH detected are shared by all programs. Both diamond and mmseqs share the most RBH with blastp.

3.3 Error estimates

Though high divergence and horizontal gene transfer result in divergence of gene order, remnants of highly conserved regions are preserved even between the genomes of evolutionarily distant organisms (Dandekar *et al.*, 1998; Tamames, 2001; Moreno-Hagelsieb *et al.*, 2001). Thus, conservation of adjacency is a very limited source for correction of misidentified orthologs. However, the few pairs of adjacently conserved genes can be used to estimate error rates by extrapolation (Moreno-Hagelsieb and Latimer, 2008). The estimate consists on finding pairs of adjacently conserved gene pairs. If both genes in a target pair are orthologs to the corresponding genes in the query genome, they are inferred to be correct. Errors would consist on conserved pairs where only one of the genes is detected as an ortholog, with the remaining gene being evidence of an ortholog mistaken for a paralog. The formula for error rates is: $E = p/p + o$, where p stands for number of paralogs in conserved pairs, and o for the number of orthologs in conserved pairs.

As expected, error rates increased amongst more distant genomes (Fig. 3). Unexpectedly though, the error rates seem to be similar among all programs, a tendency to be lower for the faster programs than for blastp can be observed, with mmseqs consistently showing the lowest error rate estimates, followed closely by diamond. These results suggest that the quality of orthologs remains as high, if not better, when using software that produces results faster than blastp. Both diamond and mmseqs showed the best quality. One reason for this quality could be that these two programs use a fast implementation of the Smith-Waterman algorithm to produce their final alignments.

3.4 Diamond sensitivity options

The results above show that diamond might be the best option to obtain RBH in terms of the speed, sensitivity and quality. As stated, we ran the program using the program's "sensitive" option. However, the program also has the "fast" and "more-sensitive" options. We thus decided to test how close the results would be in quality running diamond with those two options so that it might run faster (fast mode), or the time costs of perhaps obtaining better quality (more-sensitive mode).

picture(0,0)(-35,0)(1,0)30 (0,-35)(0,1)30 picture picture(0,0)(35,0)(-1,0)30 (0,-35)(0,1)30 picture

picture(0,0)(-35,0)(1,0)30 (0,35)(0,-1)30 picture picture(0,0)(35,0)(-1,0)30 (0,35)(0,-1)30 picture

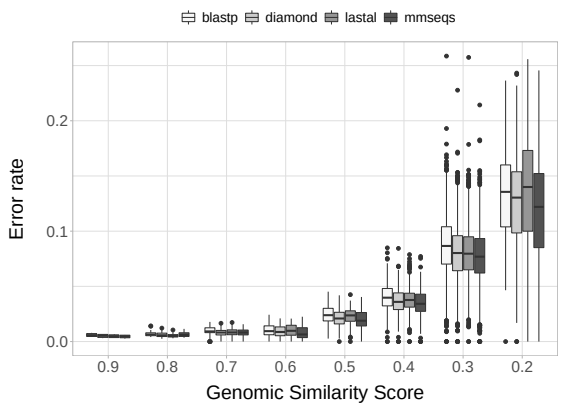

Fig. 3. Error rate estimates. All error rate estimates are very close, indicating that using any of the fast programs would not add errors beyond what would be obtained with blastp.

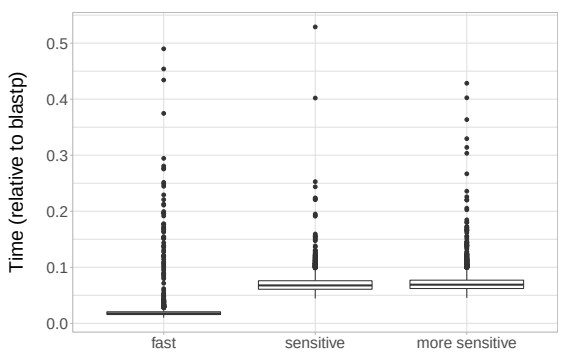

Fig. 4. Times to run diamond with different sensitivity settings. The more-sensitive mode takes barely longer than sensitive mode, while both sensitive and more-sensitive modes take close to three times as long to run as fast mode.

Interestingly, both the sensitive and more-sensitive options showed very similar speeds running in about 7% of the time required by blastp. The fast option took less than 2% of the blastp time to run, which is indeed comparable with lastal times (Fig. 4).

When evaluating the number of RBH detected, the proportions found using the fast setting dropped noticeably below those detected with the other options (Fig. 5A). Furthermore, the sensitive and more sensitive settings had the highest RBH in common with blastp, close to 90% (Fig. 5B). Thus, the fast option did not represent the best option for detecting RBH with diamond. It also appears that the more-sensitive mode has no advantage over the sensitive option.

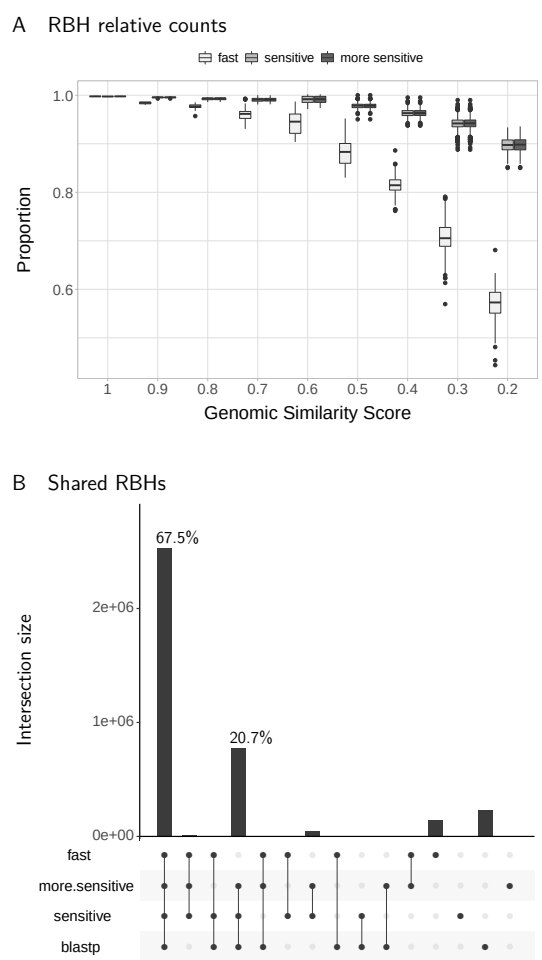

Fig. 5. RBH detected by diamond at different sensitivity settings. Using the fast mode dropped the number of RBH detected the most when compared to blastp, while using the more-sensitive mode did not noticeably produce more results than using the sensitive mode.

4 Conclusion

Our results show that the faster programs produced results with very similar error rate estimates as blastp, with diamond and mmseqs showing slightly lower errors. Since the results using diamond and mmseqs were very similar, yet diamond ran faster than mmseqs, we conclude that diamond, ran with the "sensitive" option, is the best alternative to blastp for obtaining RBH.

Acknowledgements and Funding

This work was supported by a Discovery Grant to GM-H from The Natural Sciences and Engineering Research Council of Canada (NSERC). JH-S

picture(0,0)(-35,0)(1,0)30 (0,-35)(0,1)30 picture picture(0,0)(35,0)(-1,0)30 (0,-35)(0,1)30 picture

picture(0,0)(-35,0)(1,0)30 (0,35)(0,-1)30 picture picture(0,0)(35,0)(-1,0)30 (0,35)(0,-1)30 picture

was supported by a fellowship from Mexico's Consejo Nacional de Ciencia y Tecnología (CONACYT).

References

Altenhoff, A. M., Studer, R. A., Robinson-Rechavi, M., and Dessimoz, C. (2012). Resolving the ortholog conjecture: orthologs tend to be weakly, but significantly, more similar in function than paralogs. *PLoS Comput Biol*, **8**(5), e1002514.

Boeckmann, B., Capella-Gutierrez, S., Dalquen, D. A., DeLuca, T., Huerta-Cepas, J., Linard, B., Pereira, C., da Silva, A. S., Train, C.-M., Bork, P., Lecompte, O., von Mering, C., Sjölander, K., Jensen, L. J., Quest for Orthologs consortium, Altenhoff, A. M., Gabaldón, T., Thomas, P. D., Forslund, K., Sonnhammer, E., Pryszcz, L. P., Schreiber, F., Szklarczyk, D., Xenarios, I., Martin, M. J., Muffato, M., Lewis, S. E., and Dessimoz, C. (2016). Standardized benchmarking in the quest for orthologs. *Nat Methods*, **13**(5), 425–430.

Buchfink, B., Xie, C., and Huson, D. H. (2015). Fast and sensitive protein alignment using DIAMOND. *Nat Methods*, **12**(1), 59–60.

Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., and Madden, T. L. (2009). BLAST+: architecture and applications. *BMC Bioinformatics*, **10**, 421.

Campbell, A., Mrázek, J., and Karlin, S. (1999). Genome signature comparisons among prokaryote, plasmid, and mitochondrial DNA. *Proc Natl Acad Sci USA*, **96**(16), 9184–9189.

Chen, X. and Zhang, J. (2012). The Ortholog Conjecture Is Untestable by the Current Gene Ontology but Is Supported by RNA Sequencing Data. *PLoS Comput Biol*.

Dandekar, T., Snel, B., Huynen, M., and Bork, P. (1998). Conservation of gene order: a fingerprint of proteins that physically interact. *Trends Biochem. Sci.*, **23**(9), 324–328.

Dessimoz, C., Gabaldón, T., Roos, D. S., Sonnhammer, E. L. L., Herrero, J., and the Quest for Orthologs Consortium (2012). Toward community standards in the quest for orthologs. *Bioinformatics*, **28**(6), 900–904.

Edgar, R. C. (2010). Search and clustering orders of magnitude faster than BLAST. *Bioinformatics*, **26**(19), 2460–2461.

Esposito, M. and Moreno-Hagelsieb, G. (2018). Non-synonymous to synonymous substitutions suggest that orthologs tend to keep their functions, while paralogs are a source of functional novelty. *bioRxiv*.

Fitch, W. M. (2000). Homology a personal view on some of the problems. *Trends Genet*, **16**(5), 227–231.

Gabaldón, T. and Koonin, E. V. (2013). Functional and evolutionary implications of gene orthology. *Nat Rev Genet*, **14**(5), 360–366.

Galperin, M. Y., Kristensen, D. M., Makarova, K. S., Wolf, Y. I., and Koonin, E. V. (2017). Microbial genome analysis: the COG approach. *Brief Bioinformatics*.

Haft, D. H., DiCuccio, M., Badretdin, A., Brover, V., Chetvernin, V., O'Neill, K., Li, W., Chitsaz, F., Derbyshire, M. K., Gonzales, N. R., Gwadz, M., Lu, F., Marchler, G. H., Song, J. S., Thanki, N., Yamashita, R. A., Zheng, C., Thibaud-Nissen, F., Geer, L. Y., Marchler-Bauer, A., and Pruitt, K. D. (2018). RefSeq: an update on prokaryotic genome annotation and curation. *Nucleic Acids Res*, **46**(D1), D851–D860.

Kent, W. J. (2002). BLAT–the BLAST-like alignment tool. *Genome Res*, **12**(4), 656–664.

Kiełbasa, S. M., Wan, R., Sato, K., Horton, P., and Frith, M. C. (2011). Adaptive seeds tame genomic sequence comparison. *Genome Res*, **21**(3), 487–493.

Kristensen, D. M., Wolf, Y. I., Mushegian, A. R., and Koonin, E. V. (2011). Computational methods for Gene Orthology inference. *Brief Bioinformatics*, **12**(5), 379–391.

Kuzniar, A., van Ham, R. C. H. J., Pongor, S., and Leunissen, J. A. M. (2008). The quest for orthologs: finding the corresponding gene across genomes. *Trends Genet*, **24**(11), 539–551.

Moreno-Hagelsieb, G. and Latimer, K. (2008). Choosing BLAST options for better detection of orthologs as reciprocal best hits. *Bioinformatics*, **24**(3), 319–324.

Moreno-Hagelsieb, G., Treviño, V., Pérez-Rueda, E., Smith, T. F., and Collado-Vides, J. (2001). Transcription unit conservation in the three domains of life: a perspective from Escherichia coli. *Trends Genet*, **17**(4), 175–177.

Moreno-Hagelsieb, G., Wang, Z., Walsh, S., and ElSherbiny, A. (2013). Phylogenomic clustering for selecting non-redundant genomes for comparative genomics. *Bioinformatics*, **29**(7), 947–949.

Steinegger, M. and Söding, J. (2017). MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat Biotechnol*, **35**(11), 1026–1028.

Tamames, J. (2001). Evolution of gene order conservation in prokaryotes. *Genome Biol*, **2**(6), RESEARCH0020.

Tatusov, R. L., Galperin, M. Y., Natale, D. A., and Koonin, E. V. (2000). The COG database: a tool for genome-scale analysis of protein functions and evolution. *Nucleic Acids Res*, **28**(1), 33–36.

Ward, N. and 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.

Wolf, Y. I. and Koonin, E. V. (2012). A tight link between orthologs and bidirectional best hits in bacterial and archaeal genomes. *Genome Biol Evol*, **4**(12), 1286–1294.

picture(0,0)(-35,0)(1,0)30 (0,-35)(0,1)30 picture picture(0,0)(35,0)(-1,0)30 (0,-35)(0,1)30 picture