The Journal of Open Source Software

VPsearch: fast exact sequence similarity search for genomic sequences

Joris Vankerschaver[1, 2], **Steven J. Kern**[3], **and Robert Kern**[3]

1 Center for Biosystems and Biotech Data Analysis, Ghent University Global Campus, Republic of Korea **2** Department of Applied Mathematics, Computer Science and Statistics, Ghent University, Belgium **3** Enthought Inc., 200 W Cesar Chavez, Austin, TX 78701, United States

DOI: 10.21105/joss.04048

Software

- Review
- Repository
- Archive

Editor: Pending Editor

Submitted: 11 January 2022
Published: 11 January 2022

Summary

Similarity search is a central task in computational biology, and in genomics in particular. In genomics, similarity search usually takes the following form: given an unknown nucleotide or protein sequence (the query), what are the most similar sequences in a given database of known sequences? In this context, similarity search is important for taxonomic determination, to establish phylogenetic relationships, or to annotate sequences and genes with functional information. With the advent of easily accessible high-throughput sequencing technologies, the amount of available genomic data continues to grow rapidly, and the demands for computationally efficient and accurate similarity search implementations have increased accordingly.

Over the years, a number of tools for similarity search have improved upon the venerable BLAST (Altschul et al., 1990) in terms of lookup speed and accuracy. Some of these, such as the FASTA tool suite (Pearson, 2016), provide rapid protein or nucleotide similarity search based on sequence content alone. Others, such as the RDP classifier (Wang et al., 2007) for microbiome analysis, take taxonomic information or other domain-specific information into account to improve classification sensitivity or to provide additional confidence measures. For whole-genome sequences, data structures for approximate similarity search have been adopted to improve sequence lookup speed (Marçais et al., 2019).

Statement of need

VPsearch is a light-weight Python package and command-line tool to perform similarity search. Unlike some of the approximate tools mentioned in the introduction, VPsearch provides an exact similarity search implementation, taking the full sequence content into account (rather than a k-mer spectrum or other approximation).

Given a database of known sequences, VPsearch builds a so-called *vantage point tree* (Uhlmann, 1991; Yianilos, 1993), a data structure that allows for similarity lookups in time proportional to the logarithm of the size of the database. For a set of unknown sequences, VPsearch is then able to query this tree and return the best matching results from the database. In the case study below we show that for short sequences (such as the 16S rRNA gene used in bacterial classification) VPsearch outperforms both BLAST (7x speedup) and ggsearch36 from the FASTA suite (27x speedup) without any loss in accuracy.

The VPsearch tool is implemented in Python, using Cython (Behnel et al., 2011) for performance-critical sections, and to interface with external libraries. To compare sequences during indexing and querying, VPsearch calls out to Parasail (Daily, 2016), a library of SIMD-optimized implementations for global and local sequence alignment.

VPsearch outputs similarity search results in the "BLAST-6" tabular format also used by BLAST, Diamond, the FASTA tool suite and others, so that it can be used as a drop-in replacement for any of these tools. VPsearch is able to return the k most similar sequences for a given query, not just the most similar match, and supports querying the database in multithreaded mode.

Case study

We compare the performance of VPsearch with two standard tools for sequence lookup: Blast+, as a standard tool that is optimized for inexact but fast sequence lookup via the matching sequence pair heuristic, and ggsearch36, part of the FASTA suite, which relies on exact alignment to achieve higher accuracy at the cost of greatly increased lookup times. We show that VPsearch manages to combine the good aspects of both tools, while avoiding the drawbacks.

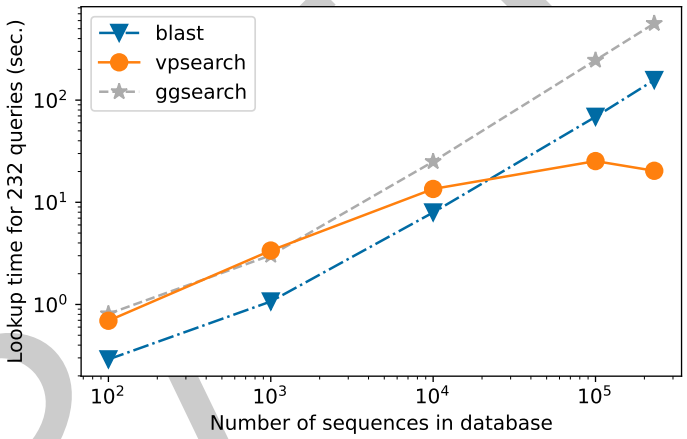

Figure 1: Sequence lookup time for 232 sequences as a function of the size of the database. For small databases (less than 10,000 sequences), VPsearch performs comparably to Blast+ and ggsearch36. For realistic databases (consisting of more than 50,000 sequences), the VPsearch lookup times scales logarithmically as the size of the database increases.

We use VPsearch to look up 232 query sequences from the Mothur SOP dataset (Kozich et al., 2013) in the Silva database of bacterial 16S sequences (Quast et al., 2013). The database was processed by excising the v4 region of the full-length 16S sequences and removing duplicate sequences, resulting in a database of 230,013 sequences (each approximately 250 base pairs in length) with known taxonomies. The Mothur SOP dataset was processed using the dada2 protocol (Callahan et al., 2016), resulting in 232 Amplicon Sequence Variants (ASVs), representing distinct taxonomic units in the dataset.

On the full Silva database, VPsearch is clearly the fastest (20s total lookup time), compared to Blast+ (157s) and ggsearch (561.3s) (Figure 1). For Blast+ and ggsearch36, the total lookup time scales linearly with the size of the database, whereas for VPsearch the scaling is logarithmical (in other words, making the database 10 times larger adds a constant factor to the total lookup time).

For each lookup, we compared the top matches (ranked by alignment score) between VPsearch, ggsearch36, and Blast+. Out of 232 ASVs there is one sequence where the taxonomic assignment differs between ggsearch36 and VPsearch, due to a small difference in how the alignment parameters are chosen for both algorithms. Both algorithms identify the ASV as being in the

family of *Lachnospiraceae*, with the difference on the genus level. Between VPsearch and Blast+, there are three different assignments, due to the fact that several sequences in the database present an equally plausible match for the query sequence.

Acknowledgements

We would like to thank Jun Isayama and Yuko Kiridoshi for stimulating discussions, and Homin Park for help in setting up the computational infrastructure that was used for the case study.

References

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, *215*(3), 403–410. https://doi.org/10.1016/S0022-2836(05)80360-2

Behnel, S., Bradshaw, R., Citro, C., Dalcin, L., Seljebotn, D. S., & Smith, K. (2011). Cython: The best of both worlds. *Computing in Science & Engineering*, *13*(2), 31–39. https://doi.org/10.1109/MCSE.2010.118

Callahan, B. J., McMurdie, P. J., Rosen, M. J., Han, A. W., Johnson, A. J. A., & Holmes, S. P. (2016). DADA2: High-resolution sample inference from Illumina amplicon data. *Nature Methods*, *13*(7), 581–583. https://doi.org/10.1038/nmeth.3869

Daily, J. (2016). Parasail: SIMD C library for global, semi-global, and local pairwise sequence alignments. *BMC Bioinformatics*, *17*(1), 81. https://doi.org/10.1186/s12859-016-0930-z

Kozich, J. J., Westcott, S. L., Baxter, N. T., Highlander, S. K., & Schloss, P. D. (2013). Development of a Dual-Index Sequencing Strategy and Curation Pipeline for Analyzing Amplicon Sequence Data on the MiSeq Illumina Sequencing Platform. *Applied and Environmental Microbiology*, *79*(17), 5112–5120. https://doi.org/10.1128/AEM.01043-13

Marçais, G., Solomon, B., Patro, R., & Kingsford, C. (2019). Sketching and Sublinear Data Structures in Genomics. *Annual Review of Biomedical Data Science*, *2*(1), 93–118. https://doi.org/10.1146/annurev-biodatasci-072018-021156

Pearson, W. R. (2016). Finding Protein and Nucleotide Similarities with FASTA. *Current Protocols in Bioinformatics*, *53*, 3.9.1–3.925. https://doi.org/10.1002/0471250953.bi0309s53

Quast, C., Pruesse, E., Yilmaz, P., Gerken, J., Schweer, T., Yarza, P., Peplies, J., & Glöckner, F. O. (2013). The SILVA ribosomal RNA gene database project: Improved data processing and web-based tools. *Nucleic Acids Research*, *41*(D1), D590–D596. https://doi.org/10.1093/nar/gks1219

Uhlmann, J. K. (1991). Satisfying general proximity / similarity queries with metric trees. *Information Processing Letters*, *40*(4), 175–179. https://doi.org/10.1016/0020-0190(91)90074-R

Wang, Q., Garrity, G. M., Tiedje, J. M., & Cole, J. R. (2007). Naive Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy. *Applied and Environmental Microbiology*, *73*(16), 5261–5267. https://doi.org/10.1128/AEM.00062-07

Yianilos, P. (1993). Data Structures and Algorithms for Nearest Neighbor Search in General Metric Spaces. *Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, *93*. https://doi.org/10.1145/313559.313789