Alignment-Free Sequence Analysis Area: 2019–20 CBB Qualifiers

Lenwood S. Heath

November 23, 2019

1 Introduction

Traditional analysis of biological sequences has depended on various types of sequence alignment, as exemplified by BLAST [1, 2]. Over the years, a number of analysis methods have been developed that are called alignment-free because they do not depend on traditional sequence alignment. Many such methods depend on examining the k-mers in one or more biological sequences; a k-mer is a substring of a biological sequence — DNA, RNA, or amino acid — of length exactly k. For reasonable values of k, it is possible to catalog all the k-mers that exist in a biological sequence and then to use the catalog for comparison or other analysis purposes.

A sample application is the comparison of two genomes for similarity. Yi and Jin [56] develop a method called co-phylog for comparing many genomes based on k-mers and constructing a phylogenetic tree for the genomes; co-phylog software is available. Nordström et al. [34] develop the k-mer based algorithm NIKS (needle in the k-stack) for genome comparison; NIKS is available from SourceForge. Haubold [15] provides a useful review of methods for alignment-free genome comparison, including Table 1, a compendium of existing methods.

One method that was originally developed for comparing documents is MinHash [8, 9]. More recently, MinHash has been employed by many tools for biological sequence comparison, starting with Mash [37]. The idea is to use hashing and the k-mers of a sequence to generate a signature or sketch for the sequence that can be used to compare sequences. A recent tool that employs MinHash to compare biological sequences is sourmash [41].

2 Qualifier Instructions

For the written part of the CBB qualifying exam, you are to do the following steps.

1. Read the following papers for background purposes: [8, 9, 15, 37, 41].
2. Make sure you understand the concepts and issues discussed in Section 1.

3. Choose at least five alignment-free tools for biological sequence analysis that address a particular problem, such as metagenomics, genome comparison, or virus identification. Each paper should be explained in one or more references that you identify. You may find some useful references in the bibliography or through a search of Web of Science.

4. Study your five or more references in depth and examine related Web sites and code.

5. Write a six to eight page document, including bibliography, as a record of what you have done in terms of papers read and approaches investigated. As one section of your document, discuss MinHash [8, 9] and its implementation [37]. Explain the mathematical and computational ideas behind the application of MinHash to both text documents and biological sequences. Next, give a precise definition of your biological sequence analysis task. Make sure to compare your five or more approaches carefully and logically. Be critical about the suitability of each approach to your task. Finally, think creatively about any additional approaches that might be used for this task, and explain how you would go about implementing those approaches.

6. Generate a PDF of your document and submit it as the written part of your exam.

References

[1] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, *Basic local alignment search tool*, Journal of Molecular Biology, 215 (1990), pp. 403–410.

[2] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. H. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, *Gapped BLAST and psi-BLAST: A new generation of protein database search programs*, Nucleic Acids Research, 25 (1997), pp. 3389–3402.

[3] A. Andoni and P. Indyk, *Efficient algorithms for substring near neighbor problem*, in Proceedings of the Seventheenth Annual ACM-SIAM Symposium on Discrete Algorithms, 2006, pp. 1203–1212.

[4] ——, *Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions*, 47th Annual IEEE Symposium on Foundations of Computer Science, Proceedings, (2006), pp. 459–+.

[5] ——, *Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions*, Communications of the ACM, 51 (2008), pp. 117–122.

[6] K. Berlin, S. Koren, C. S. Chin, J. P. Drake, J. M. Landolin, and A. M. Phillippy, *Assembling large genomes with single-molecule sequencing and locality-sensitive hashing*, Nature Biotechnology, 33 (2015), pp. 623–+.

[7] M. Bhushan, M. Singh, and S. K. Yadav, *Big data query optimization by using locality sensitive bloom filter*, in 2015 2nd International Conference on Computing for Sustainable Global Development (Indiacom), 2015, pp. 1424–1428.

[8] A. Z. Broder, *On the resemblance and containment of documents*, in Compression and Complexity of Sequences 1997 — Proceedings, 1998, pp. 21–29.

[9] ——, *Identifying and filtering near-duplicate documents*, Combinatorial Pattern Matching, 1848 (2000), pp. 1–10.

[10] L. H. Chi and X. Q. Zhu, *Hashing techniques: A survey and taxonomy*, ACM Computing Surveys, 50 (2017), p. 36 pages.

[11] M. Cochez, *Locality-sensitive hashing for massive string-based ontology matching*, in 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (Iat), Vol 1, 2014, pp. 134–140.

[12] S. Deorowicz and S. Grabowski, *Data compression for sequencing data*, Algorithms for Molecular Biology, 8 (2013), p. 13 pages.

[13] M. Deraspe, F. Raymond, S. Boisvert, A. Culley, P. H. Roy, F. Laviolette, and J. Corbeil, *Phenetic comparison of prokaryotic genomes using k-mers*, Molecular Biology and Evolution, 34 (2017), pp. 2716–2729.

[14] S. Girotto, M. Comin, and C. Pizzi, *Metagenomic reads binning with spaced seeds*, Theoretical Computer Science, 698 (2017), pp. 88–99.

[15] B. Haubold, *Alignment-free phylogenetics and population genetics*, Briefings in Bioinformatics, 15 (2014), pp. 407–418.

[16] B. Haubold, F. Klotzl, and P. Pfaffelhuber, *Andi: Fast and accurate estimation of evolutionary distances between closely related genomes*, Bioinformatics, 31 (2015), pp. 1169–1175.

[17] B. Haubold, L. Krause, T. Horn, and P. Pfaffelhuber, *An alignment-free test for recombination*, Bioinformatics, 29 (2013), pp. 3121–3127.

[18] B. Haubold and P. Pfaffelhuber, *Alignment-free population genomics: An efficient estimator of sequence diversity*, G3-Genes Genomes Genetics, 2 (2012), pp. 883–889.

[19] B. Haubold, P. Pfaffelhuber, M. Domazet-Loso, and T. Wiehe, *Estimating mutation distances from unaligned genomes*, Journal of Computational Biology, 16 (2009), pp. 1487–1500.

[20] B. Haubold, N. Pierstorff, F. Moller, and T. Wiehe, *Genome comparison without alignment using shortest unique substrings*, BMC Bioinformatics, 6 (2005).

[21] B. Haubold, F. A. Reed, and P. Pfaffelhuber, *Alignment-free estimation of nucleotide diversity*, Bioinformatics, 27 (2011), pp. 449–455.

[22] C. Jain, A. Dilthey, S. Koren, S. Aluru, and A. M. Phillippy, *A fast approximate algorithm for mapping long reads to large reference databases*, Journal of Computational Biology, 25 (2018), pp. 766–779.

[23] Z. M. Jin, C. Li, Y. Lin, and D. Cai, *Density sensitive hashing*, IEEE Transactions on Cybernetics, 44 (2014), pp. 1362–1371.

[24] J. Kawulok and S. Deorowicz, *Cometa: Classification of metagenomes using k-mers*, Plos One, 10 (2015), p. 23 pages.

[25] D. Koslicki and D. Falush, *Metapalette: A k-mer painting approach for metagenomic taxonomic profiling and quantification of novel strain variation*, Msystems, 1 (2016), p. 18 pages.

[26] J. A. Lees, M. Vehkala, N. Valimaki, S. R. Harris, C. Chewapreecha, N. J. Croucher, P. Marttinen, M. R. Davies, A. C. Steer, S. Y. C. Tong, A. Honkela, J. Parkhill, S. D. Bentley, and J. Corander, *Sequence element enrichment analysis to determine the genetic basis of bacterial phenotypes*, Nature Communications, 7 (2016), p. 8 pages.

[27] X. N. Liu, Y. Yu, J. P. Liu, C. F. Elliott, C. Qian, and J. Z. Liu, *A novel data structure to support ultra-fast taxonomic classification of metagenomic sequences with k-mer signatures*, Bioinformatics, 34 (2018), pp. 171–178.

[28] Y. H. Lv, T. H. Ma, M. L. Tang, J. Cao, Y. Tian, A. Al-Dhelaan, and M. Al-Rodhaan, *An efficient and scalable density-based clustering algorithm for datasets with complex structures*, Neurocomputing, 171 (2016), pp. 9–22.

[29] E. Marinier, R. Zaheer, C. Berry, K. A. Weedmark, M. Domaratzki, P. Mabon, N. C. Knox, A. R. Reimer, M. R. Graham, L. Chui, L. Patterson-Fortin, J. Zhang, F. Pagotto, J. Farber, J. Mahony, K. Seyer, S. Bekal, C. Tremblay, J. Isaac-Renton, N. Prystajecky, J. Chen, P. Slade, and G. V. Domselaar, *Neptune: A bioinformatics tool for rapid discovery of genomic variation in bacterial populations*, Nucleic Acids Research, 45 (2017), p. 13 pages.

[30] P. Melsted and J. K. Pritchard, *Efficient counting of k-mers in DNA sequences using a bloom filter*, BMC Bioinformatics, 12 (2011), p. 7 pages.

[31] A. Muller, C. Hundt, A. Hildebrandt, T. Hankeln, and B. Schmidt, *Metacache: Context-aware classification of metagenomic reads using minhashing*, Bioinformatics, 33 (2017), pp. 3740–3748.

[32] K. D. Murray, C. Webers, C. S. Ong, J. Borevitz, and N. Warthmann, *Kwip: The k-mer weighted inner product, a de novo estimator of genetic similarity*, Plos Computational Biology, 13 (2017), p. 17 pages.

[33] S. Nath, P. Singha, and M. S. Islam, *Locality-sensitive hashing scheme for bangla news article clustering using bloom filter*, in 2017 International Conference on Electrical, Computer and Communication Engineering (Ecce), 2017, pp. 17–21.

[34] K. J. V. Nordstrom, M. C. Albani, G. V. James, C. Gutjahr, B. Hartwig, F. Turck, U. Paszkowski, G. Coupland, and K. Schneeberger, *Mutation identification by direct comparison of whole-genome sequencing data from mutant and wild-type individuals using k-mers*, Nature Biotechnology, 31 (2013), pp. 325–+.

[35] S. O'Hara and B. A. Draper, *Are you using the right approximate nearest neighbor algorithm?*, 2013 IEEE Workshop on Applications of Computer Vision (Wacv), (2013), pp. 9–14.

[36] J. Oliver, C. Cheng, and Y. G. Chen, *Tlsh — A locality sensitive hash*, in 2013 Fourth Cybercrime and Trustworthy Computing Workshop (Ctc 2013), 2014, pp. 7–13.

[37] B. D. Ondov, T. J. Treangen, P. Melsted, A. B. Mallonee, N. H. Bergman, S. Koren, and A. M. Phillippy, *Mash: Fast genome and metagenome distance estimation using minhash*, Genome Biology, 17 (2016), p. 14 pages.

[38] P. Pandey, M. A. Bender, R. Johnson, and R. Patro, *Squeakr: An exact and approximate k-mer counting system*, Bioinformatics, 34 (2018), pp. 568–575.

[39] M. Patella and P. Ciaccia, *The many facets of approximate similarity search*, 2008 IEEE 24th International Conference on Data Engineering Workshop, Vols 1 and 2, (2008), pp. 468–479.

[40] D. Pellow, D. Filippova, and C. Kingsford, *Improving bloom filter performance on sequence data using k-mer bloom filters*, Journal of Computational Biology, 24 (2017), pp. 547–557.

[41] N. T. Pierce, L. Irber, T. Reiter, P. Brooks, and C. T. Brown, *Large-scale sequence comparisons with sourmash*, F1000Research, 8 (2019), p. 15 pages.

[42] V. Popic and S. Batzoglou, *A hybrid cloud read aligner based on minhash and kmer voting that preserves privacy*, Nature Communications, 8 (2017), p. 7 pages.

[43] V. Popic, V. Kuleshov, M. Snyder, and S. Batzoglou, *Fast metagenomic binning via hashing and Bayesian clustering*, Journal of Computational Biology, 25 (2018), pp. 677–688.

[44] J. B. Qian, Q. Zhu, and H. H. Chen, *Multi-granularity locality-sensitive bloom filter*, IEEE Transactions on Computers, 64 (2015), pp. 3500–3514.

[45] ——, *Integer-granularity locality-sensitive bloom filter*, IEEE Communications Letters, 20 (2016), pp. 2125–2128.

[46] M. L. Rodríguez-R and K. T. Konstantinidis, *Bypassing cultivation to identify bacterial species*, Microbe, 9 (2014), pp. 111–118.

[47] R. Rozov, R. Shamir, and E. Halperin, *Fast lossless compression via cascading bloom filters*, BMC Bioinformatics, 15 (2014), p. 8 pages.

[48] L. Schaeffer, H. Pimentel, N. Bray, P. Melsted, and L. Pachter, *Pseudoalignment for metagenomic read assignment*, Bioinformatics, 33 (2017), pp. 2082–2088.

[49] C. Y. Shuai, H. C. Yang, O. Y. Xin, S. Q. Li, and Z. Chen, *A novel accuracy and similarity search structure based on parallel bloom filters*, Computational Intelligence and Neuroscience, (2016), p. 12 pages.

[50] M. Slaney, Y. Lifshits, and J. F. He, *Optimal parameters for locality-sensitive hashing*, Proceedings of the IEEE, 100 (2012), pp. 2604–2623.

[51] B. Solomon and C. Kingsford, *Fast search of thousands of short-read sequencing experiments*, Nature Biotechnology, 34 (2016), pp. 300–+.

[52] H. Stranneheim, M. Kaller, T. Allander, B. Andersson, L. Arvestad, and J. Lundeberg, *Classification of DNA sequences using bloom filters*, Bioinformatics, 26 (2010), pp. 1595–1600.

[53] C. Sun, R. S. Harris, R. Chikhi, and P. Medvedev, *Allsome sequence bloom trees*, Journal of Computational Biology, 25 (2018), pp. 467–479.

[54] J. Waldmann, J. Gerken, W. Hankeln, T. Schweer, and F. O. Glöckner, *Fastavalidator: An open-source java library to parse and validate fasta formatted sequences*, BMC Research Notes, 7 (2014), p. 4 pages.

[55] J. Wang, W. Liu, S. Kumar, and S. F. Chang, *Learning to hash for indexing big data-a survey*, Proceedings of the IEEE, 104 (2016), pp. 34–57.

[56] H. G. Yi and L. Jin, *Co-phylog: An assembly-free phylogenomic approach for closely related organisms*, Nucleic Acids Research, 41 (2013), p. 13 pages.

[57] A. Zielezinski, S. Vinga, J. Almeida, and W. M. Karlowski, *Alignment-free sequence comparison: Benefits, applications, and tools*, Genome Biology, 18 (2017), p. 17 pages.