Algorithm Engineering for DNA Assembly

Ben Bals[1,2] and Solon P. Pissis[1,2]

[1]CWI, Amsterdam, The Netherlands
[2]Vrije Universiteit, Amsterdam, The Netherlands

April 16, 2025

Background Recently, our research group has developed a theoretically faster algorithm for string assembly based on Eulerian trails in a special class of graphs (so called de Bruijn graphs) [1]. In string assembly, we receive fragments (i.e., strings of length k). Our goal now is to find a larger string such that each fragment maps to a unique substring of that larger string. In the variation of this classical problem that we consider, we also have access to additional information on where each fragment can appear in the assembled string. These kinds of algorithms lay at the heart of genome assembly and also have applications in differential privacy.

Your Contribution The core of this project will consist of you implementing our improved algorithm as well as previous algorithms. The goal is to test the theoretical advantages of the novel algorithm in practice and compare it to previous approaches. Additionally, we see seek to better understand how real-world data (i.e., real genomes) behave (as opposed to a theoretical worst-case analysis).

Requirements You should have solid skills in efficient programming (e.g., good practical grasp of standard data structures and memory management). Prior knowledge of C++ or Rust is advantageous. A good knowledge of common string data structures and algorithms may help, but is not required. Prior knowledge in bioinformatics or graph theory is not required.

Rough Research Plan

Setup

(1) Implement our new algorithm. Implement the Ben-Dor et al. [2] algorithm.[1]
(2) Obtain a set of dataset DNA sequences, for example from the NCBI [4, 5].
(3) Establish two simple schemes for determining the interval for each fragment.
    (a) First and last occurrence in the dataset.
    (b) Randomly distributed around the actual occurrence.
(4) Bonus: Set-up the Conte et al. [3] assessment algorithm.

Research Questions

(1) How fast is our new algorithm compared to the Ben-Dor et al. algorithm?
    - Experiment with the role of the individual parameters, in particular w and k. For which parameter values can the problem still be solved?
    - How does this compare to the simple Eulerian trail algorithm without interval information?
    - Using the scheme 4(b), measure the impact of having few large intervals and many short ones.
(2) What values of w result from the scheme in 5a (as a function of k)?
(3) Possibly: Implement the counting extension of or algorithm. Analyse the number of ETs as a function of the variance of the distribution used in scheme 4(b).
(4) Bonus: What is the relationship between the fragment length and the possible number of assemblies using the Conte et al. algorithm? What about with our novel algorithm under the two proposed schemes?

[1]If the code is designed correctly, this should be possible by making relatively few modifications to our algorithm.

References

[1] Ben Bals, Sebastiaan van Krieken, Solon P. Pissis, Leen Stougie, and Hilde Verbeek. When is string reconstruction using de Bruijn graphs hard? *Private communication*, 2025.

[2] Amir Ben-Dor, Itsik Pe'er, Ron Shamir, and Roded Sharan. On the complexity of positional sequencing by hybridization. *J. Comput. Biol.*, 8(4):361–371, 2002. doi:10.1089/106652701752236188.

[3] Alessio Conte, Roberto Grossi, Grigorios Loukides, Nadia Pisanti, Solon P. Pissis, and Giulia Punzi. Beyond the BEST theorem: Fast assessment of Eulerian trails. In Evripidis Bampis and Aris Pagourtzis, editors, *Fundamentals of Computation Theory - 23rd International Symposium, FCT 2021, Athens, Greece, September 12-15, 2021, Proceedings*, volume 12867 of *Lecture Notes in Computer Science*, pages 162–175. Springer, 2021. doi:10.1007/978-3-030-86593-1_11.

[4] National Center for Biotechnology Information. Escherichia coli. In National Library of Medicine. URL: https://www.ncbi.nlm.nih.gov/datasets/taxonomy/562/.

[5] National Center for Biotechnology Information. Homo sapiens. In National Library of Medicine. URL: https://www.ncbi.nlm.nih.gov/datasets/taxonomy/9606/.