Treatment of Massive Metagenomic Data with Graphs

Cristóbal R. Santa María[1], Romina A. Rebrij[2], Victoria Santa María[3] and Marcelo A. Soria[4]

[1] *Departamento de Ingeniería-Universidad Nacional de La Matanza, San Justo, Buenos Aires, B1754JEC, Argentina*
csantamaria@unlam.edu.ar

[2] *Ph.D. Program, Facultad de Ingeniería, Universidad de Buenos Aires, Buenos Aires, C1063ACV, Argentina*
rominarebrij@gmail.com

[3] *Instituto de Investigaciones Médicas "Alfredo Lanari" - Universidad de Buenos Aires, Buenos Aires, C1427ARO, Argentina*
vctrsntmr@gmail.com

[4] *Microbiología – Facultad de Agronomía - Universidad de Buenos Aires, Buenos Aires, C1417DSE, Argentina*
soria@agro.uba.ar

Abstract

Among the *de novo* strategies to assemble metagenomic DNA fragments the application of de Bruijn graphs stands out. These graphs greatly reduce the computational complexity and overload that arises as a consequence of the huge data volume. An Eulerian cycle can be established on a de Bruijn graph that allows the assembly of sequence reads into longer fragments for genome reconstruction. This paper shows the theoretical principles of the computational schema applied. Also, the difficulties that appear in the practical application of the method and the algorithmic features of some of the available open source programs. Finally, the work of the authors research group is summarized.

Keywords: high-throuput DNA, sequencies assembly, metagenomics, graphs, eulerian cycle

1. Introduction

Metagenomics is a branch of biology that studies deoxyribonucleic acid (DNA) extracted from environmental samples, mainly of microbial origin. Upon extraction, the DNA is fragmented and sequenced using some high-throughput sequencing technology. This approach has several advantages, among them two stand out. First, the steps between sampling and data production are minimized, which reduces the number of perturbations introduced by laboratory manipulations. Second, metagenomic analyses do not require the isolation of pure cultures nor their growth in laboratory culture media. This feature allows the detection and quantification of the huge number of microorganism that cannot be cultivated and thus are undetectable through traditional techniques. One of the drawbacks of the sequencing techniques is that they produce fragments, also called reads, that are relatively small compared to a full bacterial genome. The most popular sequencing technology nowadays is Illumina which produces fragments that vary depending the specific protocol between 36 and 250 base pairs of DNA, while a typical bacterial genome range in size between 3 and 6 gigabases. Some newer technologies, such as PacBio or Oxford Nanopore, can reach fragment sizes of several thousand kilobases, which facilitates the process but they are still considerably shorter than the full genome. There are two types of strategies to reconstruct a genome or at least long fragments of it. The first is to map the reads against a similar reference genome. This option is fast and quite accurate but is also possible when a very similar genome is available. The second, known as the *de novo* strategy is to assemble the genome matching and joining reads in progressively longer fragments, or contigs. Although in theory it would be possible to assemble all reads into contigs through a brute force approach consisting of matching all reads vs. all, as the number of reads increases it becomes computationally unfeasibly. Several strategies and heuristics were proposed over the years and many of the successful attempts were based on applications of graph methods, in special De Bruijn graphs [1]. In this work we analyze how the assembly of DNA sequences is achieved using graphs, we also show how these techniques extract information from the pools of reads while reducing the overall requirements of memory.

2. Complexities of contig assembly

Current DNA sequencing technologies can generate 10^6 - 10^{12} reads (fragments) with sizes that typically vary between 50 and 250 base-pairs (bp). In the case of bacterial genomes this short reads must be ordered and assembled into circular DNA molecules with usual lengths ranging between 1 and 6 Mbp, but can be as long as 14 Mbp. Ideally, every single position of the genome would be covered by multiple reads, and every read should have other reads with which it would overlap partially or completely (Fig. 1).

Thus, a brute force approach of finding overlapping reads should accomplish the task or sorting and assembling the genome. However, the huge size of the sequencing data prevents the implementation of such an approach. Besides, there are certain genomic features (repetitive sequences, duplicated genes, etc.) that cannot be sequenced properly or that introduce uncertainties in the ordering of reads. Besides, the sequencing methodology introduces errors. In consequence the result produced by an assembler is not a single strand of DNA per chromosome, but a set of contigs built from overlapping reads (Fig. 2).

```
---ACCGT--
-----CGTGC
CCTACC----
-CTACCGT-
__________
CCTACCGTGC
```

Fig. 1. Optimal alignment for sequence reconstruction

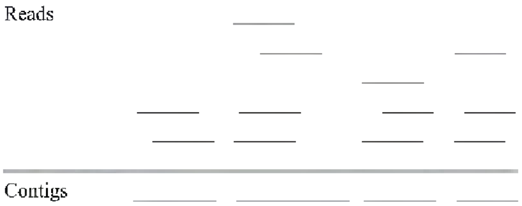
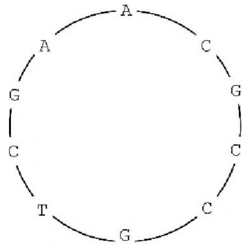


Fig. 2. Optimal alignment for sequence reconstruction

To convert the assembly of genomes into a more tractable problem, several strategies based on the use of graph theory were proposed [2].

One variant is to consider that each sequence read is fragmented into overlapping strings of length *k*, known as *k-mers*. So that two-adjacent *k-mers* overlap by *k-1* nucleotides. Then a graph is built considering that each *k-mer* is a node and the edges are the overlaps of length k-1 between adjacent *k-mers*. For example, consider the toy circular genome in Fig. 3 (left), if the sequence reads generated by the sequencer (Fig. 3, right) are divided into k-mers of length three, the list of all 3-mers is ACG, CGC, GCC, CCG, CGT, GTC, TCG, CGA, GAA, AAC. After that, all two-base overlaps between pairs of list elements could be searched, counted and organized in a Hamiltonian cyclic graph (Fig. 4) in which nodes represent the k-mers, and the directed edges represent the k-1 mers of the overlap.

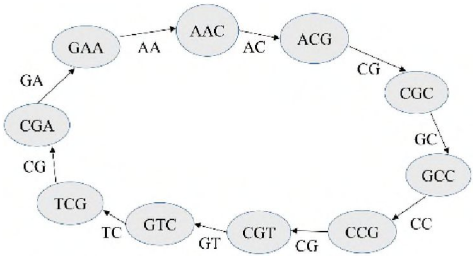


```
ACGCCGT
  GCCGTCG
    CGTCGAA
      TCGAACG
        GAACGCC
          ACGCCGT
ACGCCGTCGA
```

Fig. 3. Circular genome (left) and reads obtained from it and the resulting contig (right)

In other words, the k-mer is formed by removing the prefix from the source node and suffix of the destination node. In this way, by moving from node ACG to node CGC the sequence ACGCC is obtained. By completing this walk along the cyclic graph, the whole sequence of the toy genome (ACGCCGTCGA) is reconstructed.

As a more real example, we could consider the 10^6 – 10^9 million 150-bp single-end reads that a typical Illumina run generate assume that the assembly would be carried out using k-mers of length length 55. Finding the Hamiltonian cycle under these conditions would require between 10^{12} and 10^{18} comparisons of 55-base pairs of k-mers. This is a currently unsolvable NP-complete problem.



ACGCCGTCGA

Fig. 4. Hamiltonian cycle and reconstructed genome sequence

In contrast, the application of the De Bruijn graphs renders the problem solvable. Nicolaas De Bruijn extended the results that Leonhard Euler obtained in 1741 while analyzing the famous seven bridges problems [3]. In short, Euler found the conditions for the existence of what was later named the Eulerian cycle. In this cycle Euler considered that the bridges were edges of a graph that connected different nodes representing the different city neighborhoods. His goal was to determine whether there was a path that started and finished in the same node and visited every other node traversing each edge at most once. Euler determined that:

a) If there are more than two nodes with an odd number of incident edges, the cycle does not exist.
b) If there are only two nodes and they are connected by an odd number of edges, the cycle does exist and can be started from any node.
c) If there is no node with an odd number of incident edges, the cycle exists and can be started from any node.

The graph also must be connected, that is, every node can be visited from any other node in the graph. If the graph is directed, it is required that the number of output edges is equal to the number of input edges for every node. This is known as a balanced directed graph.

In 1946 de Bruijn discovered how to search the shortest circular superstring that contains all substrings of k elements from an alphabet with n elements [1]. Given the four symbols (bases) that comprise DNA, there are $4^3 = 64$ possible 3-mers. If the size of DNA strings is fixed at k=55, as can easily be the case with next generation sequencing technologies, 4^{55} different 55-mers could exist. The application of de Bruijn's idea to DNA sequencing is to consider each directed edge as a k-mer connecting a source node representing the *k-1* prefix of the k-mer, and a target node representing its k-1 suffix [4]. If all k-mers of a DNA sequencing procedure are considered, the resulting Eulerian cycle will represent the shortest superstring that contains every k-mer once. That is, the superstring that contains all the k-1 overlaps. Fig. 5 shows how this graph representation operates on the given example.

The problem of finding an Eulerian cycle was algorithmically solved by Euler itself in 1741 and has, in consequence, a viable computational solution [3]. In the case of sequence assembly, the DNA reads are fragmented into k-mers and connected in a de Bruijn graph. which is then used to find the Eulerian cycles that will constitute the contigs.

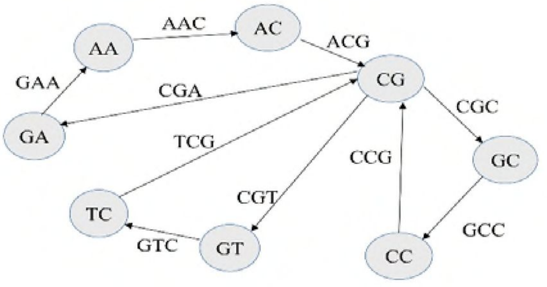


Fig. 5. Eulerian cycle of the de Bruijn graph

3. Application

The practical applications of this assembly method reveal some difficulties. First, the DNA reads may not cover all of the possible k-mers the target genome could produce. Second, sequence duplications occur in most genomes with varying frequencies depending on the organisms under study. This fact determines a multiplicity of the number of inward and outward edges a node can have. Third, the chemical reactions that occur in the sequencing equipment can produce erroneous base calls. These sequencing errors must be addressed before the assembly because they will generate k-mers that do not exist in the genome.

Different software solutions, mainly open-source, have been developed and published in order to solve the difficulties pointed out in the previous paragraph and to include heuristics that accelerate the assembly process. For example, IDBA-UD compare the quality parameters in complementary reads to infer the most likely k-mer [5]. MEGAHIT [6] builds a succinct version of the de Bruijn graph that code m edges using a vector of O(m) bits that marks-up the validity or invalidity of every edge. It starts with suffixes and prefixed of length eight and iteratively increment the graph size by increasing k. This way, in the first iterations edges with errors are removed and later, with larger k's, the sequence repetitiosn are removed.

Within the framework of the project "Applications of Data Mining Techniques for the Analysis of the Human Microbiom through Metabolic Functionalities" of the Universidad Nacional de la Matanza, we carried out the assembly of 143 metagenomes obtained from bacterial DNA extracted from stool, rectal swab, and mucosal samples deposited in the NCBI under the Bioproject accession number PRJNA397450. The data from this study was originally analyzed only at the DNA read level, but we are extending it by assembling the metagenomes to determine whether the use of longer contig sequences allows us to detect genes or bacteria that has potential as biomarkers of health/disease conditions.

4. References

[1] N. de Bruijn. "A combinatorial problem," *Proceeding Nederlands Akademiks Wetensch,* vol. 49, pp. 758-764, 1946

[2] D. Coil, G. Jospin and A. Darling, "A5-miseq: an updated pipeline to assemble microbial genomes from Illumina MiSeq data," *Bioinformatics*, Vol. 31, no. 4, pp. 587-589, 2015

[3] L. Euler, "Solutio problematis ad geometriam situs pertinentis," *Commentarii Academiae*

Scientiarum Petropolitanae, vol. 8, pp. 128-140, 1741

[4] P. Campeau, P. Pevzner, and G. Tesler, "How to apply de Bruijn graphs to genome assembly," *Nature Biotechnology,* vol. 29, no. 11, pp. 987-991, 2011

[5] Y.Peng, H. Leung, S. Yiu, and F. Chin, " IDBA-UD: de novo assembler for single-cell and metagenomic sequencing data with hibhly uneven depth," *Bioinformatics,* vol. 28, no. 11, pp. 1420-1428, 2012

[6] D. Li, C. Liu, R. Luo, K. Sadakane, and T. Lam, "MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph," *Bioinformatics*, vol. 31, no.10, pp. 1674-1676, 2015