MedBioInfo Algorithms in Bioinformatics project

Alexandros Karagiannopoulos

10/17/2019

Dungeom: a de Bruijn graph-based genome assembler

Motivation

Genome assembly has been a big challenge in the genomic research due to the large amount of sequenced data that needs to be processed. De Bruijn graph-based algorithms, which extract the assembled sequence through the construction of a de Bruijn graph of the initial reads, are widely considered as the most efficient way of approaching the genome assembly process. Dungeom (**D**e Br**u**ij**n** **g**raph-based g**e**n**o**me asse**m**bler) is based on the same principal and can reassemble DNA sequences after splitting them into smaller fragments of predefined size (kmers).

Introduction

Even though Next-Generation Sequencing (NGS) technologies revolutionized genomic research, the limitations of the sequencing instruments allowed for the creation of several downstream problems, such as sequence mapping, assembly, analysis and even data storage. The first programs used in genome assembly used to merge the generated sequence reads with an overlap-consensus-layout method, which was based on identifying and merging the reads with overlapping ends. This approach, though, could not handle effectively the long repetitive genomic regions found, especially, in eukaryotic genomes [1]. This problem was merely solved with the introduction of de Bruijn graph-based algorithms and, as a result, many software packages that are typically used nowadays to assemble the underlying genomes and transcriptomes (Velvet, Spades, Trinity, etc.,) use de Bruijn graphs to reconstruct genomes from NGS libraries.

A de Bruijn graph can be used for the representation of a DNA sequence as the kmer components it is constituted from. If we define a kmer size, we can split a DNA sequence in its kmer components. After that, a directed graph can be constructed by connecting pairs of kmers with overlaps between the first k-1 nucleotides and the last k-1 nucleotides. The direction of arrow goes from the kmer, whose last k-1 nucleotides are overlapping, to the kmer, whose first k-1 nucleotides are overlapping. This is known as de Brujin graph and provides an efficient way to assemble a DNA sequence.

Although running those packages do not require any knowledge of de Bruijn graphs, such understanding is necessary to improve the quality of work in terms of minimizing errors in assembly and memory optimization. Thus, the design of a de Bruijn graph-based DNA sequence assembler DNA seemed like a reasonable step to that direction.

For this purpose Dungeom has been developed. This tool can be used for the reassembly of a single DNA sequence using the same two step workflow as other de Bruijn graph-based assemblers. In the first step, the DNA sequence is broken into small pieces (kmers) and a de Bruijn graph is constructed from those short pieces. In the next step, the sequence is derived from the de Bruijn graph. Dungeom's code is available as a python script. In order for the user to get a better insight of how Dungeom works, a Jupyter Notebook with a step-to-step walkthrough of the code is also provided.

Mehtods

Dungeom is written in Python (3.6.7) with Atom text editor (1.40.1). Random DNA sequences for testing the program's efficacy were generated with the Random Sequence Generator online tool of the molbiotools suite (http://molbiotools.com/randomsequencegenerator.html).

Features

Dungeon is a de Bruijn graph-based assembler that gets a DNA sequence and a kmer size as inputs and outputs the de Bruijn graph of the sequence and the reassembled sequence, reporting if it is identical or no to the input one. The sequence is first broken into small pieces (kmers) and the de Bruijn graph is constructed from the generated kmers. Then all kmers are shuffled in order to assure the randomness of the assembly except the first one, which is the starting point of the reassembly. The process of the assembly is based on finding the Eulerian path from the graph, that is the trail in the de Bruijn graph which visits every edge exactly once. After the Eulerian walk is executed, the reassembled sequence is generated.

It is well known that GC content varies significantly between organisms and within different parts of the same organism. For instance in human genomes the average GC-content ranges from 35% to 60% [2], in yeast from 35% to 50% [3] and in bacteria from less than 15% to more than 75% [4]. In order to define the minimum kmer size for the successful reassembly of a DNA sequence, random DNA sequences of different sizes and GC contents were generated and were given as input to Dungeom. For each sequence size of a specific GC content 3 separate sequences were randomly generated and run using a low kmer size. If the kmer size resulted in a reassembled sequence which was not identical with the input sequence, the tool was run again with a kmer size of a larger size until 10 consecutive reassembles were successful. This was defined as the minimum kmer size which leads to successful reassembly. The results are presented in Table 1.

SIZE(BP)	GC CONTENT		
	0.4	**0.6**	**0.8**
100	6	6	7
500	8	8	10
1000	9	9	12
5000	11	11	15
10000	13	13	16

Discussion

For the purpose of this project Dungeom, a de Bruijn graph-based assembler, was developed. It gets a DNA sequence and a kmer size as inputs and outputs the reassembled sequence and "identical" or "not identical" if the reassembled sequence is identical or not with the input sequence. The user should always take into account the size of the DNA sequence and its GC content before assigning the kmer size, as longer DNA sequences with a higher GC content require a larger kmer size.

Nevertheless, Dungeom provides a very basic functionality of de Bruijn graph-based assembly compared to similar widely-used tools. These tools can deal with more complex problems regarding sequencing errors or biological variation (e.g. polymorphisms) that lead to the creation of bubbles (two distinct paths start and end at the same nodes) or tips (node that is is disconnected on one of its ends) in the de Bruijn graph.

Although not a proper genome assembler, Dungeom can provide a useful insight into how the most widely used assemblers work. Consequently, it might be helpful to new bioinformaticians who would like a better conceptual understanding of the de Bruijn graph-based algorithm, as the Dungeom's workflow is quite simple and easy to follow.

References

1. Li Z, Chen Y, Mu D, Yuan J, Shi Y, Zhang H, et al. Comparison of the two major classes of assembly algorithms: overlap-layout-consensus and de-bruijn-graph. Brief Funct Genomics. 2012;11(1):25-37.
2. Lander ES, Linton LM, Birren B, Nusbaum C, Zody MC, Baldwin J, et al. Initial sequencing and analysis of the human genome. Nature. 2001;409(6822):860-921.
3. Lynch DB, Logue ME, Butler G, Wolfe KH. Chromosomal G + C content evolution in yeasts: systematic interspecies differences, and GC-poor troughs at centromeres. Genome biology and evolution. 2010;2:572-583.
4. Bohlin J, Eldholm V, Pettersson JHO, Brynildsrud O, Snipen L. The nucleotide composition of microbial genomes indicates differential patterns of selection on core and accessory genomes. BMC genomics. 2017;18(1):151-151.