Assembling a genome with the help of de Bruijn graphs

Fiona Walter

Before exploring de Bruijn graphs and their application in genome assembly, we should understand how a genome is assembled in the first place. After the DNA is extracted, it is fragmented into reads of around 150 base pairs with Illumina sequencing (the old Sanger sequencing method produces reads of around 800 bp length). The reads are then assembled into longer "contigs" which are then assembled into the full genome sequence. At the point of assembling reads we come across a few downsides; merging the fragments is particularly tedious with short reads and can often lead to ambiguous sequences. So when the faster next gen sequencing emerged, the programs processing reads and making them into contigs took too much time and often resulted in incorrect assembly[1]. If we think about this logically: there is no point in faster sequencing if the method to assemble the reads is not able to do its job to the degree previously possible.

Solving the superstring problem

Mathematician Nicolaas De Bruijn presented a great solution to this "superstring problem" by making a graph that works quickly and leaves less room for ambiguity when trying. The exact definition of the "superstring problem" is to:

find the shortest *circular superstring* that contains all possible *substrings of length k* over a given *alphabet*.

De Bruijn figured that this problem could be solved by using a "Eulerian circuit". In this method, we show the relationships between substrings like a graph with nodes (= substrings of length k) and arrows (= alignment between each substring). Whilst each arrow can only be used once, each node can be visited several times.

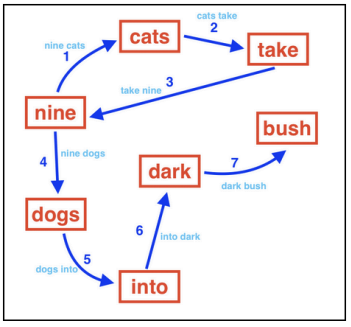

Image 1. Eulerian circuit of a sentence.

As an example for a Eulerian circuit we can chop the sentence "Nine cats take nine dogs into dark bush" into each word-element. We connect each word (= node) using arrows (image 1). This allows us to create an alignment between two words (= light blue text). The sentence can then be re-assembled by following each arrow and overlapping the arrow's substring alignments (image 2).

Image 2. Assembly of eulerian circuit.

How does a de Bruijn graph help with genome sequencing?

Now thinking about our genome and comparing it to the superstring problem, they both are quite similar. The circular superstring represents the circular genome, the substrings represent a sequence read (of length k) and the alphabet represents the nucleotides available (G, C, A, T). Hence, a de Bruijn graph can be used to produce the genome we are sequencing whilst at the same time preventing ambiguity as each arrow in the circuit can only be visited once. In the graph, each node represents a sequence read (of length k), each arrow represents the alignment between each read (or "k-mers").

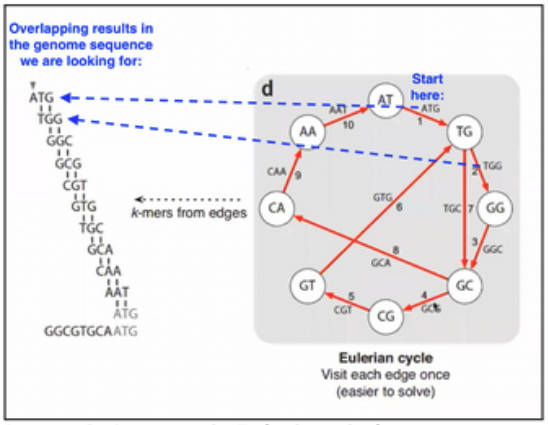

Image 3. A genomic Eulerian circle.

As an example, we can therefore construct a Eulerian circuit for the genome sequence **"GGCGTGCAATG"** (image 3)[2]. We follow any arrow in the circuit until each "read" has been visited (here the "read" is only 2 nucleotides though in a real read it could be any length). We overlap each read to assemble the whole sequence, just like in the example with the sentence.

As illustrated above, de Bruijn graphs make genome sequencing better in two ways:

1. Genome sequencing is now faster by noting double-up reads only once as a node,
2. Genome sequencing is now less ambiguous by using each arrow only once.

Luckily we don't actually need to do this every time we sequence a genome as computer programs are now able to do this work for us. However, it is still important to acknowledge that improved technology has its limitations and a rough understanding of the underlying principles will come in handy when trying to tackle these limitations.

Thank you for your attention!

References:

(1) Henson, J.; Tischler, G.; Ning, Z. Next-Generation Sequencing And Large Genome Assemblies. Pharmacogenomics 2012, 13 (8), 901-915.

(2) Compeau, P.; Pevzner, P.; Tesler, G. How To Apply De Bruijn Graphs To Genome Assembly. Nature Biotechnology 2011, 29 (11), 987-991.

-> please note that this example is part of an image from this source. I have edited the image by adding the writing and arrows in blue.