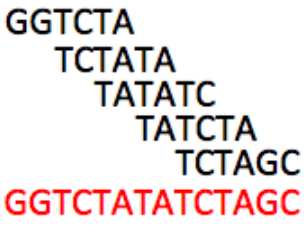
Series 3 - solution

Genomics and bioinformatics - Week 3

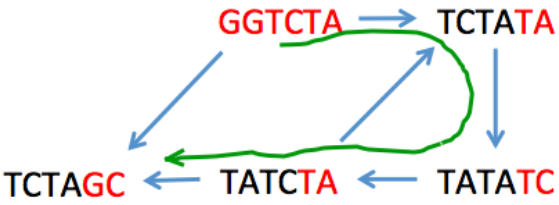
1 Overlap graphs

1. Just by looking manually at the overlaps (of size 4+):



2. The overlap (Hamiltonian) graph is build by taking the reads as vertices and adding an edge each time two reads have an overlap (of a chosen minimal size, here 4). Then find a path going through every vertex once and only once: this is called an Hamiltonian path. The contig is made of the first read plus, from each read in the path, its sub-sequence (in red below) that does not belong to the overlap. There may be several such paths; the number of them depends on the minimal overlap one chooses.

   As in part 1, one finds GGTCTATATCTAGC.



3. To build the "De Bruijn" graph, one proceeds as follows:

   - Choose an integer l (here $l = 4$ was given).
   - Build S_{l-1}, the set of all *unique* $l-1$-mers that the reads contain:
     $S_{l-1} = \{$GGT, GTC, TCT, CTA, TAT, ATA, ATC, TAG, AGC$\}$
     These are the vertices of the graph.
   - Build S_l, the set of all l-mers that the reads contain:
     $S_l =$ {GGTC, GTCT, TCTA,
     TCTA, CTAT, TATA,
     TATA, ATAT, TATC,
     TATC, ATCT, TCTA,
     TCTA, CTAG, TAGC}
     Those are the edges: GGTC binds GGT and GTC, etc.

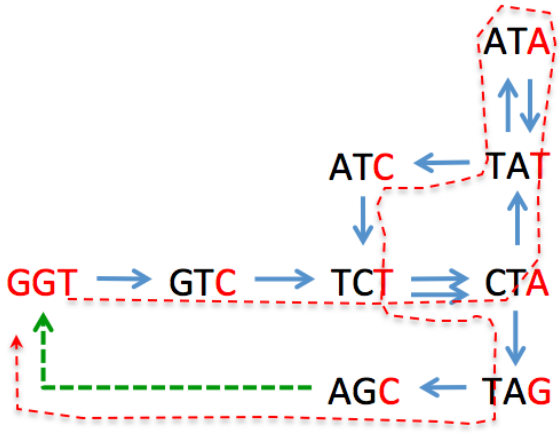
- Make the graph in which the vertices are the elements of S_{l-1} and the edges are those of S_l. Note that some edges are duplicated due to (1) the overlaps between reads and (2) some repeats in the sequence of the contig itself. In general, duplicates are a real problem that is not easily solved in practice. Real algorithms for example make estimates of the frequency of a repeat in the contig, and allow some errors. However, if you have an idea of which ones come from overlaps between reads (as we have here), discard them. In this case,

  $S_l =$ {GGTC, GTCT, TCTA,
  ~~TCTA~~, CTAT, TATA,
  ~~TATA~~, ATAT, TATC,
  ~~TATC~~, ATCT, TCTA,
  ~~TCTA~~, CTAG, TAGC}

  We see that only TCTA is a real repeat. If you do not know, simply reduce the number of edges of non-balanced vertices until you get a balanced graph when including the edge connecting the start and end vertices.

- Close the graph by joining the starting vertex - the only one that has more outgoing than incoming edges - to the end vertex - with more incoming than out coming edges. This should make the graph balanced, i.e. every vertex having the same number of incoming and outgoing edges.

- Find a cycle (which exists by Theorem 1) that goes through each *edge* once and only once: this is called an Eulerian cycle. Note that in general there may be more than one Eulerian cycle. The contig is made of the first vertex, plus the last nucleotide of each vertex in the path. In the figure below, a green arrow joins the start and end, and the cycle is drawn in red.



As in part 1, one finds GGTCTATATCTAGC. Note the TCTA repeat in the sequence.