Chapter 3 FAQ

Help Center

Q: Can we construct the overlap graph by considering overlaps of *k*-2 rather than *k*-1 symbols?

A: Yes, but the resulting overlap graph may potentially have more edges. In practice, edges in an overlap graph are formed by pairs of reads that overlap significantly but not perfectly (in order to account for errors in the reads).

Q: Can you give a detailed example of how to find an Eulerian cycle in terms of the adjacency list?

A: Consider the adjacency list of a small graph with ten nodes and twelve edges shown below.

```
0 → 3
1 → 0
2 → 1, 6
3 → 2
4 → 2
5 → 4
6 → 5, 8
7 → 9
8 → 7
9 → 6
```

Start at any node (say, node 0) and walk aimlessly in the graph by taking the first unused edge at each node you encounter until there are no more unused edges. For example:

```
0 → 3 → 2 → 1 → 0
```

Since we haven't traversed all of the edges yet, there exists at least one node in this cycle with still unused edges; in this case, that node is node 2. We therefore rearrange the cycle so that it starts and ends at node 2 instead of node 0:

```
2 → 1 → 0 → 3 → 2
```

After traversing this cycle, start walking again, taking the first untraversed edge at each node until there are no more untraversed edges available.

```
2 → 1 → 0 → 3 → 2 → 6 → 5 → 4 → 2
```

Since we haven't traversed all of the edges yet, there exists a node in the constructed cycle (node 6) with still untraversed edges. We rewrite the cycle so that it starts and ends at node 6 instead of node

2:

```
6 → 5 → 4 → 2 → 1 → 0 → 3 → 2 → 6
```

After traversing this cycle, start walking again, taking the first untraversed edge at each node until there are no more untraversed edges available.

```
6 → 5 → 4 → 2 → 1 → 0 → 3 → 2 → 6 → 8 → 7 → 9 → 6
```

Since all of the edges in our graph have been used, we have constructed an Eulerian cycle.

Q: What is the typical size of a gap in real read-pairs?

A: Biologists have a lot of freedom in selecting the gap size *d* in order to optimize genome assembly quality. For example, in many sequencing projects with Illumina reads, the read length is about 300 nucleotides and the gap length is about 200 nucleotides. However, it is becoming more and more common to generate read-pairs with large gap sizes (e.g., 8,000 nucleotides) because, as explained in the main text, large gap sizes often result in better assemblies.

Q: In the chapter, we assumed that the distance between read-pairs was always equal to *d*. Is this true?

A: In practice, the distance between read-pairs is known only approximately. Although it may seem that the paired de Bruijn graph would become impractical in the case of imprecise distances between reads, recent studies beyond the scope of this book have demonstrated how to adapt de Bruijn graphs in order to analyze inexact read-pair distances.

Q: How does the concept of read breaking work for read-pairs?

A: Assuming that the distance between reads in a read-pair is fixed, say that we want to break them into shorter read-pairs separated by the same fixed distance. The example below illustrates breaking a (5,2)-mer into three (3,5)-mers:

```
ACGTA---GCCTT
ACG-----GCC
 CGT-----CCT
  GTA-----CTT
```

In general, any (*k, d*)-mer can be broken into *t*+1 (*k*-1, *d*+*t*)-mers.

Q: How does real genome assembly software deal with information loss resulting from read breaking?

A: Some Eulerian paths in the de Bruijn graph constructed from *k*-mers after read breaking will be incompatible with the original reads. Real assemblers store information about the read that each *k*-mer comes from after read breaking. This additional information limits analysis to Eulerian paths in de Bruijn graphs that are compatible with reads.

Q: We only saw how to assemble single-stranded DNA in the chapter. How do

bioinformaticians assemble double-stranded DNA?

A: To assemble real genomes, bioinformaticians must handle reads from both DNA strands without knowing in advance which strand each read comes from. To address this challenge, they first add the reverse complement of each read to the collection of reads, effectively doubling the number of reads. In an ideal world, the de Bruijn graph formed from all these reads would consist of two (topologically identical but differently labeled) connected components, one for each DNA strand (see figure below).

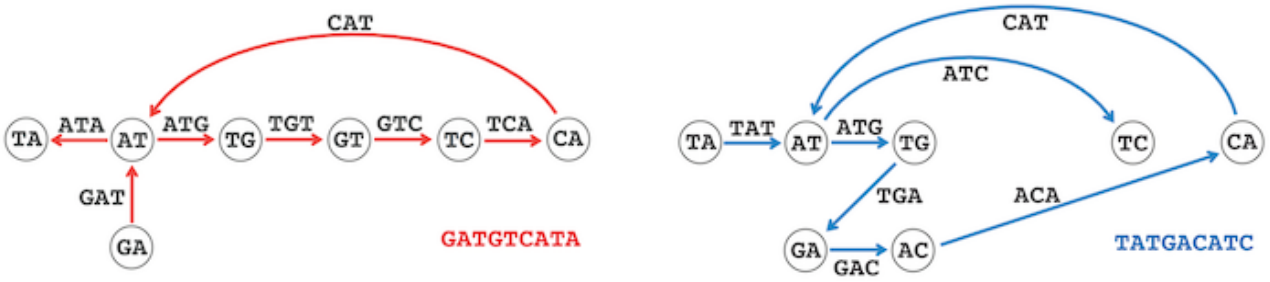

Figure: Reconstructing genomes **GATGTCATA** and **TATGACATC** from their corresponding graphs *DeBruijn*$_3$(**GATGTCATA**)and *DeBruijn*$_3$(**TATGACATC**) can be done easily, as there is only one way to traverse each graph.

In reality, these two components will be glued together, since there are many reverse complementary *k*-mers within a single strand in genomes. Indeed, in addition to direct repeats (like **ATG** in **GATGTATGA**), genomes have many **inverted repeats**, in which one substring is the reverse complement of another (like **ATG**/**CAT** in **GATGTCATA**). As a result, while the single strand **GATGTCATA** has no repeated 3-mers, making its assembly trivial, the reverse complementary strands **GATGTCATA** and **TATGACATC** have repeated 3-mers. The figure below shows the de Bruijn graph for the reverse complementary strings **GATGTCATA** and **TATGACATC**, which requires gluing nodes from two different strings, thus complicating assembly.

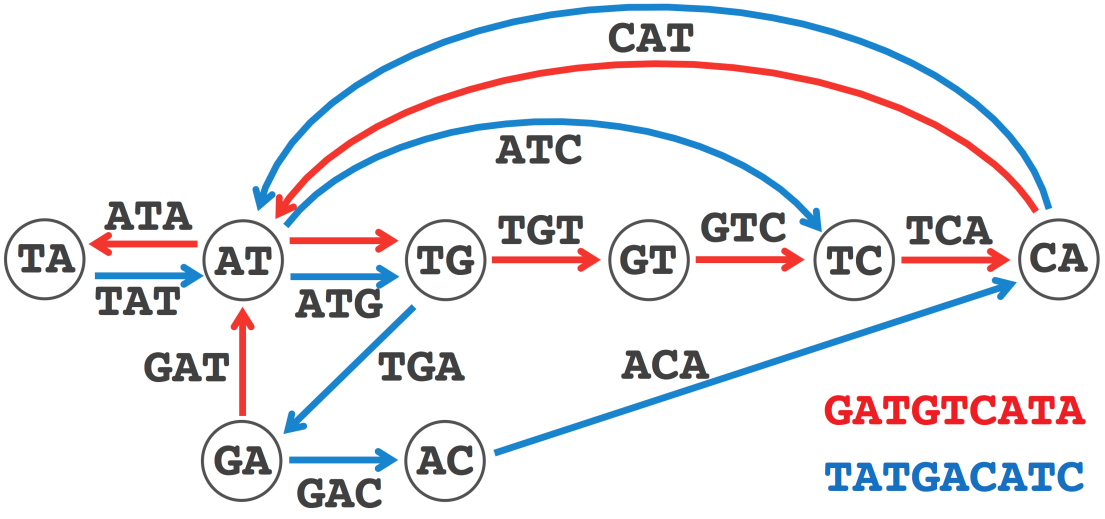

Figure: The de Bruijn graph formed by combining 3-mers from **GATGTCATA** and its reverse complement **TATGACATC**. Reconstructing the original genome is now a nontrivial problem.

Q: Are real de Bruijn graphs balanced and Eulerian?

A: In reality, the situation is more complicated. Some parts of a genome may not be covered by any reads at all, resulting in de Bruijn graphs with unbalanced nodes. Also, while bacterial genomes typically consist of a single circular chromosome, eukaryotic genomes typically have multiple linear chromosomes. Even in the case of perfect coverage by reads, ends of linear chromosomes result in unbalanced nodes. Existing assembly tools successfully address this applications by finding contigs in de Bruijn graphs, which they then combine into scaffolds using paired reads.

Q: In the case when some regions of a genome have no read coverage, is it sufficient to find an Eulerian path in each connected component of the de Bruijn graph?

A: No.

Created Fri 21 Nov 2014 4:03 PM PST

Last Modified Fri 3 Jul 2015 6:40 AM PDT