Krish Patel
CS C121

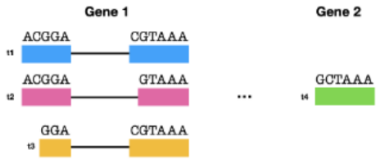
Homework 4

Pseudoalignment

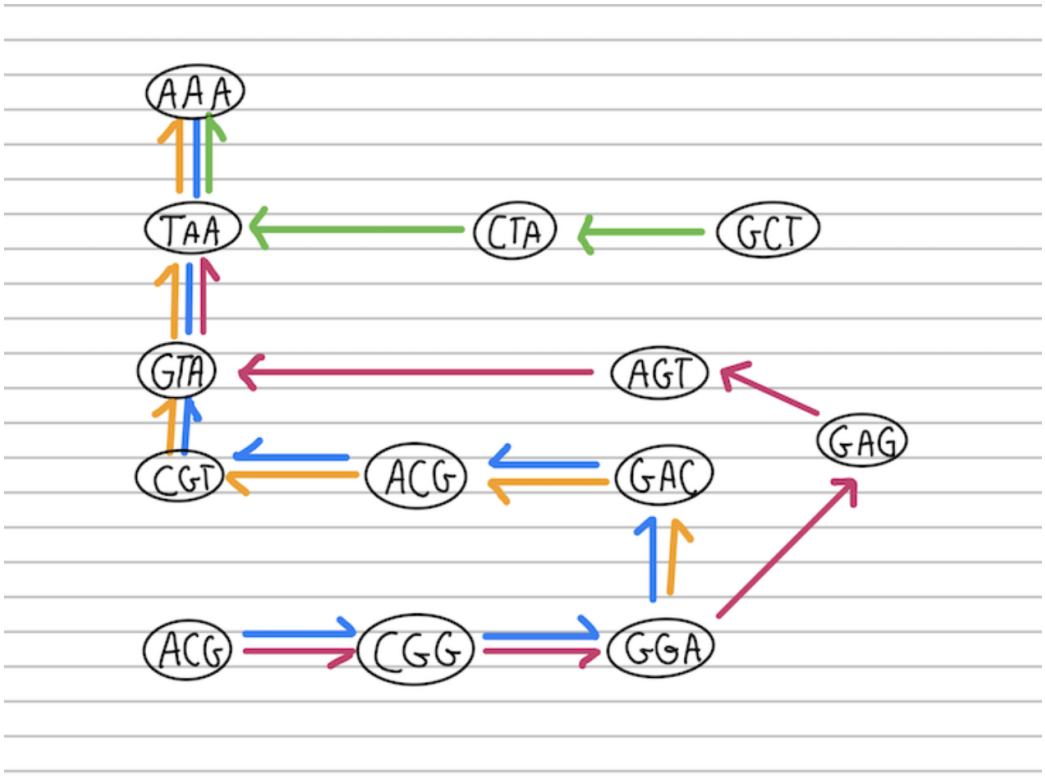
Consider the following transcriptome:

```
>t1_g1
ACGGACGTAAA
>t2_g1
ACGGAGTAA
>t3_g1
GGACGTAAA
>t4_g2
GCTAAA
```

Here is a visual representation of the transcriptome:



(a) Draw the colored de Bruijn graph that one would use for pseudoalignment with $k = 3$. In this particular case, the nodes have 3-mers, not the edges (this is the setup we used in class). Refer to the slides on construction.



b) Pseudoalign GGACGT and show the relevant steps you might take (including skips).Refer to the slides for the pseudoalignment.

Firstly, I broke GGACGT into 3 kmers: GGA, GAC, ACG, and CGT. I then start with the first k-mer: GGA. I would then map it to the graph, which gives me a match.

In this graph, GGA, GAC, ACG, and CGT were all present. I traced a path through the graph corresponding to these 3-mers: GGA → GAC → ACG → CGT. Finally, I identified the transcript(s) associated with this path. This path mapped to two transcripts which can be seen in the figure above, such as t1_g1 and t3_g1, indicating overlaps of these 3-mers in their sequences.

c) Pseudoalign GGATGT and show the relevant steps you might take (including skips). Remember, every k-mer has an equivalence class which is a set. If a k-mer is in your data but not in your transcriptome, its equivalence class is the null set

I pseudoaligned the sequence "GGATGT" using the provided De Bruijn graph by first generating the 3-mers: GGA, GAT, ATG, and TGT. I then mapped these 3-mers to the graph and found that GGA and GAT were present, corresponding to parts of t1_g1. However, ATG and TGT were not present in the graph, indicating their equivalence class was the null set.

Next, I traced a path through the graph starting with GGA, which mapped to nodes in t1_g1, t2_g1, and t3_g1. Following GGA, I mapped GAT to a node in t1_g1. Since ATG and TGT were not found in the graph, the sequence "GGATGT" could not be fully aligned with any transcript. Only the initial segment GGA → GAT was mapped, while the remaining segment ATG → TGT could not be traced, indicating that the sequence did not have a complete alignment in the given transcriptome.

1 d) The previous read might arise if there is an error at position 4 (using 1-based indexing). Describe an algorithm to deal with the error. Describe the benefits and drawbacks of your algorithm. These properties might come in speed, loss of data, or false alignments (or other things). Note: there isn't one correct answer.

Algorithm:

1. **Generate k-mers**:
    - Generate all k-mers from the read, including the ones that potentially contain the error.
2. **Allow for mismatches**:
    - For each k-mer, generate all possible k-mers with a single mismatch. This involves changing each nucleotide position to the other three possible nucleotides.
3. **Map k-mers to the De Bruijn graph**:
    - Map each original and mismatched k-mer to the De Bruijn graph.

- For each k-mer containing the error, consider its variations and map them to the graph.

4. **Trace paths and score them**:
    - Trace all possible paths in the graph using the original and mismatched k-mers.
    - Score each path based on the number of exact matches and mismatches. Assign higher scores to paths with more exact matches.
5. **Select the best path**:
    - Choose the path with the highest score or paths that meet a predefined threshold of acceptable mismatches.

This fuzzy matching algorithm provides a balance between robustness to errors and computational efficiency, making it suitable for applications where sequence errors are common. However, tuning of mismatch allowances and scoring thresholds is essential to minimize false positives and ensure meaningful alignments.

1 e) The following sequence is a valid RNA-seq read T T T ACG. Clearly it won't give you a non-empty equivalence class as-is. How did this data arise and how might you pseudoalign it? Hint: think about how RNA-seq is generated. That is, the orientation of the reads matters and can generate some annoying things we have to keep track of.

The sequence "TTTACG" is a valid RNA-seq read, but it does not directly provide a non-empty equivalence class when pseudoaligned. This situation arises because RNA-seq reads can come from either strand of the DNA, and often the reads need to be considered in their reverse complement form to align correctly. In RNA-seq, the RNA molecules are reverse-transcribed into cDNA, which is then sequenced. The sequenced reads can come from either the forward or reverse strand of the original RNA. Therefore, a read might be the reverse complement of the actual sequence present in the transcriptome.

To pseudoalign the sequence, I first took the reverse complement of the read "TTTACG," which is "CGTAAA." Then, I generated the 3-mers from the reverse complement: CGT, GTA, and TAA and AAA. Mapping these 3-mers to the De Bruijn graph, I found that CGT is present in t1_g1 and t3_g1, GTA is present in t1_g1, t2_g1, and t3_g1, and TAA is present in t1_g1, t2_g1, t3_g1, and t4_g2. Tracing the path CGT → GTA → TAA in the graph revealed that this path maps to multiple transcripts, such as t1_g1, t2_g1, and t3_g1. By considering the reverse complement of the read "TTTACG," the sequence "CGTAAA" can be aligned to the De Bruijn graph, providing a non-empty equivalence class.