

ASE Übung 8 - Sequenzassemblierung

Einführung

Das Ziel dieser Übung ist es sich mit den Verfahren zur Sequenzassemblierung von Genomen vertraut zu machen. Anhand einfachen Beispieldaten werden der *overlap*-Graph und der *de Bruijn*-Graph erstellt, mittels derer die Sequenz rekonstruiert werden kann.

Overlap Graph

Der *overlap*-Graph soll aus folgendem Datensatz von *reads* erstellt werden. Es wird angenommen, dass es sich um überlappende Fragmente einer DNA-Sequenz handelt:

```
1: TTCTTATTCTTAT
2: AAGGCCATTCTT
3: TATTCACGCGG
4: CTTATTCTTATT
5: TTCTTATTCTTA
```

Zunächst soll die Überdeckung zwischen allen Paaren von Fragmenten ermittelt werden. Prüfen Sie für jedes Paar von Fragmenten r_a und r_b , ob der Suffix von r_a gleich dem Präfix von r_b ist (und umgekehrt), und tragen Sie die Längen der maximalen Überdeckungen (beider Richtungen) in eine entsprechende Tabelle oder Matrix ein. Überdeckung eines Fragments mit sich selbst wird nicht benötigt.

Zeichnen Sie nun den Graphen basierend auf den ermittelten Überdeckungen. Für jedes Fragment wird ein Knoten erstellt. Für jede Überdeckung von einer bestimmten Mindestlänge wird zwischen zwei Fragmenten eine der Sequenz entsprechend gerichtete Kante erstellt (falls der Suffix von r_a mit dem Präfix von r_b überlappt, bedeutet dies eine Kante von r_a nach r_b).

In der Praxis wird ein empirischer Parameter für die Mindestlänge der berücksichtigten Überdeckungen verwendet. Fügen Sie hier stattdessen die Kanten absteigend der Länge nach ein, bis alle Knoten (direkt oder indirekt) miteinander verbunden sind. Markieren Sie beim Einfügen der Kanten diese mit der Länge der Überdeckung.

Der fertige Graph erlaubt zwei Pfade, welche alle Knoten genau ein mal besuchen. Ermitteln Sie diese und erstellen Sie die entsprechenden multiplen Sequenzalignments der Fragmente. Erstellen Sie das Alignment der Sequenzen der ersten beiden Knoten entsprechend der ermittelten Überdeckung. Schreiben Sie anschließend die Sequenz des dritten Knotens unter die des zweiten (wieder entsprechend der ermittelten Überdeckung) und fahren Sie so fort, bis alle Knoten berücksichtigt wurden.

Rekonstruieren Sie die vollständigen Sequenzen aus den multiplen Sequenzalignments der beiden Pfade und vergleichen Sie diese. Worin unterscheiden sie sich? Stellt der Unterschied ein Problem dar? Wie lässt dieser sich erklären?

De Bruijn Graph

Erstellen Sie den *de Bruijn*-Graph aus folgendem Datensatz von *reads*:

```
1: GCATATCTGCT
2: CTGCTTGAGTG
```

Zunächst müssen alle in den Fragmenten vorkommenden k -mere (Subsequenzen der Länge k) aufgezählt werden. Verwenden Sie $k = 6$ und tragen Sie die 6-mere in eine Tabelle ein.

Anschließend werden aus jedem k -mer die zwei $k - 1$ -mere gebildet und für jeden dieser jeweils ein Knoten im Graph erstellt, sofern ein entsprechender Knoten nicht bereits vorhanden ist. Zeichnen Sie jeweils eine gerichtete Kante zwischen dem linken und rechten $k - 1$ -mer Knoten. Da die k -mere i.d.R. kurz sind, können mehrere gleich gerichtete Kanten zwischen zwei Knoten entstehen.

Finden Sie einen Pfad durch den fertigen Graphen, welcher bei einem Knoten ohne Eingangskante beginnt und möglichst viele Kanten genau ein mal verwendet. Rekonstruieren Sie die Zielsequenz mittels des Pfads. An die Sequenz des ersten Knotens kann das jeweilige letzte Zeichen der im Pfad folgenden Knoten einfach angehängt werden.

Fragen

1. Bei der Assemblierung der Sequenz mittels des *overlap*-Graphs wurde ein Problem festgestellt. Worum handelt es sich? Wie würde das selbe Problem den *de Bruijn*-Graph Ansatz beeinflussen? Was wären mögliche Lösungsansätze?
2. Welche weiteren Probleme können bei *Shotgun*-Sequenzierung und Graph-Assemblierung auftreten und wie könnte man mit diesen umgehen?
3. Welcher der graphbasierten Ansätze ist komplexer? Welche Komplexität hat die Erstellung der Graphen (grob) in Abhängigkeit von der Anzahl der Fragmente N ?