

Übung 2

Aufgabe 1 Sequenz-Assembler

Programmieren Sie einen kleinen Sequenz-Assembler, bei dem Sie die einzelnen Fragmente wie in der Vorlesung besprochen als Knoten eines gerichteten Graphen umsetzen, bei dem jede Kante mit der jeweiligen Überlappung gewichtet ist. Der Assembler soll die einzelnen Fragmente einlesen, den Overlap-Graphen aufbauen und dann auf diesen einen Greedy-Algorithmus wie in der Vorlesung besprochen anwenden. Sie können dabei von folgenden in der Vorlesung besprochenen Idealisierungen ausgehen:

- Die Fragmente enthalten keine Sequenzierungsfehler (d.h. es gibt keine Insertionen, Deletionen oder Substitutionen).
- Alle Fragmente stammen aus einem DNA-Einzelstrang (kein Doppelstrang).
- Repeats können vernachlässigt werden.

Hinweise:

- Achten Sie bei Ihrer Implementierung auf die Laufzeit. Aktualisieren Sie z.B. nur die passenden Knoten und Kanten und bauen Sie den Graphen nicht in jedem Schritt komplett neu auf.

Aufgabe 2 Sequenz-Assembler mit Sequenzierungsfehlern

Erweitern Sie nun Ihren in der vorhergehenden Aufgabe implementierten Sequenz-Assembler so, dass auch Sequenzen mit Substitutionen berücksichtigt werden. Sie können dabei weiterhin von folgenden in der Vorlesung besprochenen Idealisierungen ausgehen:

- Die Fragmente enthalten nur Substitutionen als Sequenzierungsfehler, keine Insertionen oder Deletionen!
- Alle Fragmente stammen aus einem DNA-Einzelstrang (kein Doppelstrang).
- Repeats können vernachlässigt werden.

Hinweise:

- **Diese Aufgabe ist ausschließlich von den Zweitergruppen zu implementieren!**
- Bewerten Sie eine Substitution mit 1 in der Edit-Funktion.
- Lassen Sie eine Fehlerquote von 1 zu 10 Basen zu.

Aufgabe 3 *DNA-Doppelstrang*

Erweitern Sie nun Ihren in der vorhergehenden Aufgabe implementierten Sequenz-Assembler so, dass auch Sequenzen eines DNA-Doppelstrangs eingegeben werden können. D.h. implementieren Sie das in der Vorlesung besprochen Vorgehen zur Zuordnung der Fragmente zu einer Orientierung, die dann als Input für Ihren vorherigen Sequenzer dient.

Weitere Infos:

1. Die Programmiersprache ist - in kurzer Absprache mit mir (per Mail oder während den Webvorlesungen) - von Ihnen wählbar.
2. Zum Testen stelle ich eine Datei frag.dat mit Fragmenten zur Verfügung.
3. **Abgabe als Zip-Archiv mit README und Programm bis Montag, 07. Juni 2021** über Moodle. Die README sollte alle Informationen enthalten die ich benötige, um Ihr Programm auszuführen.