Übungen zum Sequenzanalyse-Praktikum

Universität Bielefeld, SoSe 2014
Prof. Dr. Jens Stoye · M.Sc. Nina Luhmann · M.Sc. Linda Sundermann
http://wiki.techfak.uni-bielefeld.de/gi/Teaching/2014summer/SequaPrak
praktikum-seqan@CeBiTec.Uni-Bielefeld.DE

Übungsblatt 8 vom 02.06.2014
Abgabe des Protokolls bis Donnerstag, 05.06.2014

Aufgabe 1 (q-Gram-Distanz als Filter)

Schreibe eine Klasse, die die q-Gram-Distanz und die Edit-Distanz zweier Strings berechnen kann und die Laufzeiten, die die unterschiedlichen Distanzen benötigen, miteinander vergleicht. Die Methode zur Berechnung der Edit-Distanz und zum Ranken von q-Grammen kannst du dir unter `http://wiki.techfak.uni-bielefeld.de/gi/Teaching/2014summer/SequaPrak` downloaden. Weitere Methoden zur Berechnung der q-Gram-Distanz sollst du in den folgenden Schritten selbst implementieren:

1. Mache dir kurz für dein weiteres Vorgehen klar, weshalb man einem q-Gram einen Rank zuordnen sollte.
2. Schreibe eine Funktion `profile`, die für eine DNA-Sequenz über dem Alphabet $\Sigma = \{A, C, G, T\}$ ein q-Gram-Profil berechnet. Dabei sei die Länge des q-Grams als Parameter einstellbar.
3. Schreibe eine Funktion `qgramDist`, die die Distanz zwischen zwei q-Gram-Profilen berechnet.
4. Schreibe eine Funktion `compDistsTime`, die die q-Gram-Distanz und die Edit-Distanz von zwei Strings berechnen lässt und jeweils die Laufzeiten der beiden unterschiedlichen Berechnungen ausgibt.

Nachdem du nun den Großteil der Programmierarbeit hinter dir hast, jetzt noch einige kleine Ergänzungen:

5. Berechne die 7-Gram- und die Edit-Distanz zweier beliebiger Strings von DNA-Sequenzen mit den Längen 100, 500, 1000 und 2000. Gib die Laufzeiten an und stelle sie grafisch in einem Diagramm gegenüber.
6. Erkläre, warum sich die Zeiten zwischen der Berechnung der 7-Gram-Distanz und der Edit-Distanz unterscheiden.
7. Wie kann die q-Gram-Distanz als Filter für die Edit-Distanz genutzt werden?

Aufgabe 2 (De Buijn-Graphen für Assembly)

In dieser Aufgabe werden wir ein Assembly mit *Velvet* durchführen. Velvet läuft unter Linux oder OS X, lässt sich aber auch mit Cygwin durchführen.

1. Gehe auf die Seite `http://www.ebi.ac.uk/~zerbino/velvet/` und lade dir die aktuelle Version von Velvet herunter. Kompiliere anschließend das Programm. Informationen dazu findest du in der Datei `README.txt` und `For_MAC_or_SPARC_users.txt`. Nun solltest du in dem Verzeichnis von Velvet `velvetg` und `velveth` vorfinden.
2. Informiere dich im Manual darüber, was die Aufgaben von `velveth` und `velvetg` sind. Mache dich außerdem mit der einfachen Syntax der beiden Programme bekannt.
3. Wie beeinflusst die Wahl der k-mer-Länge das Ergebnis des Assemblys? Was muss beachtet werden?
4. Teste Velvet mit drei unterschiedlichen k-mer-Längen. Benutze dazu die Daten *data/test_reads.fa*. Rufe zuerst `velveth` auf:

```
./velveth <myDirectory> <k-mer size> -shortPaired data/test_reads.fa
```

Und danach `velvetg`:

```
./velvetg <myDirectory>
```

Vergleiche anschließend die Ergebnisse. Wie viele Knoten haben deine Graphen und wie unterscheiden sich die n50-Werte? Erkläre, wie die Ergebnisse zustande gekommen sind.