Stefan Kurtz **Universität Hamburg**
Jeanine Liebold **Zentrum für Bioinformatik**

Programmierung für Naturwissenschaften 1
Wintersemester 2022/2022
Übungen zur Vorlesung: Ausgabe am 26.10.2022

Die Aufgaben auf diesem Übungszettel behandeln das Thema Linux Shell-Kommandos.

Falls Ihre Kommandos für die einzelnen Teilaufgaben mehr als 80 Zeichen umfassen, können Sie ein Kommando auch in mehrere Zeilen aufspalten. Sie müssen dann alle Zeilen, außer die letzte, mit dem Zeichen \ abschliessen. Zu besseren Lesbarkeit sollten Sie die Trennung nach dem Symbol | vornehmen. Hier ist ein Beispiel für drei Teilkommandos `cmd1`, `cmd2`, `cmd3`.

```
cmd1 -a x -b z file.tsv | \
  cmd2 -i y -j 3 | \
  cmd3 -e 0 -f 4
```

Beginnen Sie die Übungen mit den folgenden Schritten (das Zeichen $ steht für das Eingabeprompt im Terminal):

- falls Sie noch kein Repository gecloned haben:

  ```
  $ git clone https://gitlab.rrz.uni-hamburg.de/Bae4410/pfn1_2022.git
  ```

- Beachten Sie den Unterstrich in `pfn1_2022`. Es hat sich gezeigt, dass manchmal der Copy/Paste Mechanismus den Unterstrich nicht wiedergibt. In einem solchen Fall muss man den Unterstrich vor der Ausführung des Kommandos hinzufügen.
- Wechseln in das Repository: `$ cd pfn1_2022`
- Aktualisieren: `$ git pull`
- Wechseln in das HOME-Verzeichnis. Dazu kann man `cd` ohne Argument aufrufen:

  ```
  $ cd
  ```

- Erzeugen eines eigenen Verzeichnisses für die eigenen Lösungen und Materialien aus der Vorlesung, falls es noch nicht existiert. Damit Sie es später nicht vergessen, verwenden Sie gleich den Verzeichnisnamen, der die Nachnamen der Gruppenmitglieder beinhaltet. Unten werden beispielhaft die Platzhalter `Name1` und `Name2` für die Namen der Gruppenmitglieder verwendet. Diese müssen Sie natürlich durch reale Nachnamen ersetzen. Umlaute ü werden durch ue ersetzt und ß durch ss: Falls ein Nachname aus mehreren Worten besteht, werden diese Worte durch einen Unterstrich verbunden.

  ```
  $ mkdir -p my_pfn1_2022/Uebungen/Blatt01.Name1.Name2
  ```

- Wechseln in das neue Verzeichnis:

  ```
  $ cd my_pfn1_2022/Uebungen/Blatt01.Name1.Name2
  ```

- Kopieren der Materialien für Blatt01:

  ```
  $ cp -r ../../../pfn1_2022/Uebungen/Blatt01/* .
  ```

- Erstellen der Lösung in Dateien, entsprechend der Benennungen aus der Übungsaufgabe (falls es Vorgaben gibt).
- Falls Sie noch keinen Texteditor beherrschen, müssen Sie sich mit einem vertraut machen, z.B. mit `vim`. Hierzu gibt es ein interaktives Tutorial: `https://www.openvim.com`

Aufgabe 1.1 (4 Punkte) Das Tabulatorzeichen wird häufig zur Strukturierung von Textdateien verwendet. Z.B. werden in zeilenorientierten Formaten häufig Werte (z.B. Zahlen oder Worte) durch Tabulatorzeichen getrennt. Ein Beispiel ist hier das tsv-Format (tab-separated-values), das z.B. in den Naturwissenschaften häufig verwendet wird.

Eine Datei im tsv-Format enthält tabellarische Daten, in denen einzelne Werte durch einen Tabulator voneinander getrennt werden. Die Werte dürfen dabei das Trennzeichen nicht enthalten. Einzelne Werte dürfen auch fehlen. In diesem Fall kommen mehrere Tabulatoren direkt nacheinander oder ein Tabulator kommt direkt vor dem Zeilenumbruch vor.

In den Materialien zur Übung finden Sie die Datei `islands.tsv`. Diese Tabelle enthält Informationen über vorhergesagte *genomic islands* im Genom eines Bakteriums (*Thiomicrospira* sp. MA2-6). Genomic Islands sind Teile des Genoms, die durch lateralen Gentransfer von anderen Bakterien akquiriert wurden. Diese Eigenschaft ist aber in dieser Aufgabe ohne Relevanz.

Die erste Zeile der Datei ist eine Kopfzeile: Diese enthält die Überschriften der Spalten. Die darauffolgenden Zeilen beschreiben Proteine, die in den Island-Bereichen des Genoms vorkommen.

In jeder Zeile (nach der Kopfzeile) enthalten die erste und zweite Spalte die Start- und Endkoordinaten des jeweiligen Island-Bereiches. Jeder Bereich schließt ggf. mehrere Proteine ein, die dann in verschiedenen Zeilen beschrieben werden (und die gleichen Werte in den ersten zwei Spalten enthalten). Die vorletzte Spalte der Tabelle (*Product*) enthält eine Beschreibung der Funktion des Proteins, wenn diese bekannt ist, ansonsten den Wert *hypothetical protein*. Sie können davon ausgehen, dass dieser Wert in keiner anderen Spalte vorkommt.

Um die Spaltennummern zu den Spaltenköpfen zu ermitteln, führen Sie das folgende Kommando aus: `head -n 1 islands.tsv | tr '\t' '\n' | nl`

Beschreiben Sie in einer Datei `antwort.txt`, was die einzelnen Teile dieses Kommandos bewirken. 1 Pkt

Erzeugen Sie mit Hilfe geeigneter Linux-Kommandos auf der Standardausgabe genau die Informationen entsprechend der Beschreibung in der folgenden Tabelle. Jedes dieser Kommandos muss in einer Datei gespeichert werden, die ebenfalls in der folgenden Tabelle (Spalte Dateiname) angegeben ist:

Ausgabe	Dateiname für Kommando
wie `islands.tsv`, aber ohne Kopfzeile	`islands_nohead.sh`
Proteine, die als *hypothetical protein* annotiert sind	`hypo.sh`
Proteine, die nicht als *hypothetical protein* annotiert sind	`nothypo.sh`
Locus-Bezeichner aller Proteine (Inhalt der Spalte *Locus*)	`locus.sh`
Start- und Endkoordinaten der Islands-Bereiche, ohne Wiederholungen und numerisch aufsteigend nach der Startkoordinate sortiert	`ranges.sh`

Für die Lösung können z.B. die folgenden Linux-Kommandos und Funktionalitäten der Shell verwendet werden:

- Pipes (`|`), um die Standardausgabe eines Kommandos als Standardeingabe eines anderen Kommandos zu verwenden
- `tail` (Wie extrahiert man aus einer Textdatei alle Zeilen außer der ersten Zeile?)
- `sort` (Wie sortiert man die Eingabe in numerischer statt lexikographischer Reihenfolge? Wie zählt man die Häufigkeit jedes Wertes in einer Liste von Werten?).
- `cut` (Wie extrahiert man die Werte einer Spalte?)
- `grep` (Wie sucht man Zeilen, die einen String enthalten und Zeilen, die einen String nicht enthalten?)

Hinweise:

- die notwendige Information über die Optionen der Linux-Kommandos erhalten Sie durch die Manualseiten, die man z.B. durch Aufruf von `man cat` erhält.

Die geforderte Ausgabe der Kommandos steht in den Dateien mit der Endung `_solution.tsv`. Sie dürfen diese Dateien nicht verändern, da sie für Tests notwendig sind. Um zu Testen, ob Ihr Kommando, z.B. das in der Datei `islands_nohead.sh`, korrekt funktioniert, rufen Sie `make islands_nohead.tsv` auf. Entsprechendes gilt für alle anderen Kommandos. Am Ende testen Sie durch `make test`, dass alle Kommandos korrekt funktionieren. In diesem Fall sehen Sie in der Ausgabe am Ende die Meldung `Congratulations: test passed`

Punkteverteilung:

- Beschreibung der einzelnen Teile des Kommandos `head -n 1 ...`: 1 Punkt
- Korrekte Implementierung von `islands_nohead.sh`, `hypo.sh`, `nothypo.sh` und `locus.sh` (inklusive erfolgreicher Tests): jeweils 0.5 Punkt
- Korrekte Implementierung von `ranges.sh` (inklusive erfolgreicher Tests): 1 Punkt

Aufgabe 1.2 (6 Punkte) In den Materialien zur Übung finden Sie die Datei `alignments.sam`. Diese Datei liegt in SAM-Format vor. Das ist ein tsv-Format für die Repräsentation von Alignments.

Hintergrund: Dabei enthält jede Zeile Informationen über das Alignment eines kurzen Sequenz-Abschnitts (Sequencing Read) mit einem Referenzgenom (oder in einigen Fällen die Information, dass ein Read nicht aligniert werden konnte). Das Referenzgenom besteht häufig aus verschiedenen Referenzsequenzen (z.B. Chromosomen).

Die Begriffe Sequencing-Read und Alignment sind für das Verständnis und die Lösung dieser Aufgabe nicht relevant.

Zeilen, die mit einem `@`-Symbol anfangen, sind Kopfzeilen.

Die erste Spalte enthält den Namen eines Reads. In der dritten Spalte des SAM-Formates wird der Name der Referenzsequenz, mit der das Read aligniert wurde, angegeben. Der spezielle Wert $\star$ (der nur in der dritten Spalte vorkommt) bedeutet dabei, dass ein Read nicht aligniert ist.

Erzeugen Sie mit Hilfe geeigneter Linux-Kommandos auf der Standardausgabe genau die Informationen entsprechend der folgenden Tabelle. Jedes dieser Kommandos muss in einer Datei gespeichert werden, die ebenfalls in der folgenden Tabelle (Spalte *Dateiname*) angegeben ist:

Ausgabe	Dateiname für Kommando
lexikographisch sortierte Namen der Referenzsequenzen. Diese Information ist zwar auch im SAM-Header enthalten, soll in der Lösung dieser Teilaufgabe aber aus der dritten Spalte der SAM-Datei extrahiert werden.	`references.sh`
Zeilen, die keine Header-Zeilen sind, aber den Namen einer der Referenzsequenzen enthalten	`aligned_only.sh`
Tabelle mit der Anzahl der Alignments für jede Referenzsequenz sowie die Anzahl der Reads, die nicht aligniert wurden. Ihre Befehle dürfen keine festen Namen für die Referenzsequenzen voraussetzen. Diese Namen müssen sich aus den Ergebnissen vorheriger Befehle ergeben.	`references_count.sh`

Für die Lösung können z.B. die folgenden Linux-Kommandos und Funktionalitäten der Shell verwendet werden:

- Pipes (`|`), um die Standardausgabe eines Kommandos als Standardeingabe eines anderen Kommandos zu verwenden
- `grep`: Wie sucht man Zeilen, die einen der Strings enthalten, die in einer Datei aufgelistet sind?
- `uniq`: Wie zählt man in einer sortierten Eingabe die Anzahl von gleichen Elementen?
- `cut`: Wie gibt man alle Spalten einer Datei außer der ersten aus?

Hinweise:

- die notwendige Information über die Optionen der Linux-Kommandos erhalten Sie durch die Manualseiten, die man z.B. durch Aufruf von `man cut` erhält.
- Die einzelnen Aufgaben bauen aufeinander auf. Sie können daher Ergebnisse aus Zwischenschritten, die in den genannten Dateien abgelegt werden, wiederverwenden.

Die geforderte Ausgabe der Kommandos steht in den Dateien mit der Endung `_solution.tsv`. Sie dürfen diese Dateien nicht verändern, da sie für Tests gebraucht werden. Um zu Testen, ob Ihr Kommando, z.B. das in der Datei `references.sh` korrekt funktioniert, rufen Sie `make references.tsv` auf. Entsprechendes gilt für alle anderen Kommandos. Am Ende testen Sie durch `make test`, dass alle drei Kommandos korrekt funktionieren. In diesem Fall sehen Sie in der Ausgabe am Ende die Meldung `Congratulations: test passed`

Punkteverteilung:

- jeweils 2 Punkte pro Teilaufgabe

Bitte die Lösungen zu diesen Aufgaben bis zum 31.10.2022 um 18:00 Uhr an pfn1@zbh.uni-hamburg.de schicken.