Wintersemester 2007/2008 Abgabe am 24. Oktober 2007
Prof. Dr. Stefan Posch, Dipl.-Bioinf. Jan Grau

2. Übung „Angewandte Bioinformatik mit Perl und R“

1. Schreiben Sie ein Perl Skript mit zwei Argumenten `file` und `pattern`, das

   a) die Datei im FastA-Format `file` (Beispieldatei auf der Seite zur Übung) öffnet und zeilenweise auf ein Array einliest (1 Punkt)

   b) alle Vorkommen eines Musters `pattern` in Form eines regulären Audrucks in dieser Datei findet, wobei Beschreibungs- und Kommentarzeilen ignoriert werden und Matches innerhalb einer Sequenz auch über Zeilengrenzen hinweg gefunden werden. (3 Punkte)

   Geben Sie alle Fundstellen (Startpositionen innerhalb der Sequenz, nicht der Zeile) und die zugehörigen gematchten Teilsequenzen aus.

2. Schreiben Sie ein Perl-Skript `transcribe.pl` mit einem Argument `file`, das DNA-Sequenzen aus der Datei `file` einliest und zu jeder Sequenz die entsprechende Sequenz im RNA-Alphabet und das reverse Komplement (ebenfalls im RNA-Alphabet) ausgibt. (3 Punkte)
   Beispiel:

   ```
   dna.txt:
   AGGCTA
   ATTC
   ...
   % perl -w transcribe.pl dna.txt
   AGGCUA
   UAGCCU

   AUUC
   GAAU

   ...
   ```

3. Auf der Seite zur Vorlesung finden Sie zwei Dateien, `RNAtoAS.txt` und `rna.txt`. Die Datei `RNAtoAS.txt` hat folgendes Format:

   ```
   UU(U|C)=Phe
   UU(A|G)=Leu
   UC(A|C|G|U)=Ser
   ...
   ```

   Vor dem = steht jeweils eine Menge von Basentriplets (als regulärer Ausdruck) und hinter dem = steht die zugehörige Aminosäure im Drei-Buchstaben-Code.
   Die Datei `rna.txt` enthält eine Menge von Sequenzen über dem Alphabet A,C,G,U. Die einzelnen Sequenzen sind durch Zeilenumbrüche getrennt.

Schreiben Sie ein Perl-Skript, das

a) die Datei `RNAtoAS.txt` einliest und jede Zeile in den regulären Audruck für das Basentriplet und die zugehörige Aminosäure zerlegt. Beide sollen ausgegeben und zudem so gespeichert werden, dass die Zuordnung nicht verloren geht. (3 Punkte)

b) die Datei `rna.txt` einliest und jede DNA-Sequenz von der ersten Position an in Aminosäuren (getrennt durch Komma) „translatiert“. Beispiel:

```
Eingabe: AUGCAGGCC...
Ausgabe: Met,Gln,Ala,...
```

(4 Punkte)

4. In „Wirklichkeit“ startet die Translation nicht zu Beginn irgendeiner Sequenz, sondern mit dem Startcodon `AUG`. Außerdem endet die Translation, sobald ein Stopcodon (in `RNAtoAS.txt` mit `Stop` bezeichnet) gefunden wurde. Modifizieren Sie Ihr Programm aus Aufgabe 3 so, dass der *reading frame* beginnt, sobald das erste Startcodon in einer Sequenz gefunden wurde, und dass die Translation bei Auftreten des ersten Stopcodons (im reading frame) nach dem Startcodon gestoppt wird. (2 Zusatzpunkte)

5. Modifizieren Sie Ihr Programm auf Aufgabe 3 so, dass es auch Eingaben von `STDIN` anstelle einer Datei akzeptiert. (2 Zusatzpunkte)
   Testen Sie die Modifikation, indem Sie die Ausgabe des Programms aus Aufgabe 2 zu Ihrem modifizierten Programm *pipen*:

```
% perl -w transcribe.pl dna.txt | perl -w translate.pl RNAtoAS.txt
```