

3. Übung

Allgemeine Hinweise

- Verwenden Sie für die Kompilierung g++ mit den Parametern:
-std=c++11 -Wall -Wextra
(auch andere Standards z. B. c++17 sind erlaubt!)
- Erstellen Sie ein **Makefile** für die Kompilierung (mit o. g. Parametern)
- Achten Sie darauf, dass Sie **keine Compiler-Warnungen** erhalten
- Halten Sie sich an die **Best Practices** bzgl. **Clean Code**
- Es ist nicht erlaubt, externe Libraries (außer Standardlibraries) zu verwenden
- Legen Sie für jede der drei Aufgaben einen **Unterordner** an: u3_1, u3_2 und u3_3
- Entfernen Sie vor dem Hochladen die ausführbaren Kompilate, sowie die Object-Files (*.o). Laden Sie alle benötigten Source-Code-Files (*.cpp, *.h/*.hpp) hoch.
- Erstellen Sie ein **komprimiertes Archiv** (.zip oder .tar.gz), das die drei Unterordner enthält und laden Sie dies rechtzeitig auf Moodle
- Spätester Abgabezeitpunkt: **Mi., 12.06.2024, 23:59**

Aufgabe #1 – GC-Gehalt

Erstellen Sie ein Programm, das den GC-Gehalt beliebiger Gen-Sequenzen berechnet.

Das Programm wird mit zwei Kommandozeilen-Argumenten gestartet:

1. Argument ist der Dateipfad des Input-Files
2. Argument ist der Dateipfad des Output-Files

Das Programm liest den Inhalt des Input-Files zeilenweise, berechnet den prozentuellen Gehalt (gerundet auf ganze Zahlen) der Basen Guanin (G) und Cytosin (C) und schreibt diesen in die Ausgabedatei. Der prozentuelle Gehalt berechnet sich anhand der Anzahl an G- und C-Vorkommen geteilt durch die Gesamtanzahl an Basen in der jeweiligen Zeile multipliziert mit 100%.

Die Basen sind als Kleinbuchstaben im Input-File vorhanden.

Für die Ausgabedatei können Sie Groß-/Kleinschreibung beliebig wählen.

Beispiel-Aufruf: `$./u3_1 /home/xy/input.txt /home/xy/output.txt`

Beispiele

Eingabedatei	Ausgabedatei	Hinweis zur Berechnung
aatgaacacacctttgcacg	45 aatgaacacacctttgcacg	9/20
aattagatgctcatt	27 aattagatgctcatt	4/15
cacagtctcgatgaccgcc	65 cacagtctcgatgaccgcc	13/20

Aufgabe #2 – Übersetzung DNA-Protein

Erstellen Sie ein Programm, das eine beliebige DNA-Kette in ein Protein übersetzt.

Das Programm wird mit zwei Kommandozeilen-Argumenten gestartet:

1. Argument ist der Dateipfad zur DNA-Aminosäuren-Mapping-Datei
2. Argument ist die zu übersetzende DNA-Kette

Das Programm liest die DNA-Kette ein und erzeugt das passende Protein (Aminosäurenkette) aus den einzelnen Codons.

Sollte die Länge der DNA-Kette nicht durch 3 teilbar sein, soll das Programm einen Fehler ausgeben und beenden.

Sequenzen, die zu keinem Protein passen, sollen übersprungen werden (z.B. Stopp-Codons, oder Buchstaben, die kein Nukleotid darstellen)

Die Eingabe-Sequenz liegt in Großbuchstaben vor.

Für die Ausgabedatei können Sie Groß-/Kleinschreibung beliebig wählen.

Verwenden Sie die auf Moodle hinterlegte Mapping-Datei `uebung3_map.txt` – Diese enthält zwei Spalten: Codon und Aminosäure (getrennt durch ein Leerzeichen)

Hinweis: Auf biologische Korrektheit muss nicht geprüft werden (Start-/Stopp-Codons, etc.)

Beispiel-Aufrufe

```
$ ./u3_2 /home/xy/mapping.txt ATGACATCGGGGGCCCACTGCAACCTATGA  
MTSGAHCNL
```

```
$ ./u3_2 /home/xy/mapping.txt ATGA  
Fehler: Länge der DNA-Kette muss durch 3 teilbar sein
```

Aufgabe #3 – Übersetzung Protein-DNA

Erstellen Sie ein Programm, das eine beliebige Aminosäurenkette in eine DNA-Sequenz übersetzt.

Das Programm wird mit zwei Kommandozeilen-Argumenten gestartet:

1. Argument ist der Dateipfad zur DNA-Aminosäuren-Mapping-Datei
2. Argument ist die zu übersetzende Aminosäurenkette

Das Programm liest die Aminosäurenkette ein und erzeugt die passende DNA-Sequenz.

Buchstaben, die zu keinem Protein passen, sollen übersprungen werden.

Die Eingabe-Sequenz liegt in Großschreibung vor.

Für die Ausgabedatei können Sie Groß- oder Kleinschreibung beliebig wählen.

Verwenden Sie die auf Moodle hinterlegte Mapping-Datei `uebung3_map.txt` – Diese enthält zwei Spalten: Codon und Aminosäure (getrennt durch ein Leerzeichen)

Nachdem mehrere Codons für eine Aminosäure stehen können, können Sie aus den passenden Aminosäuren eine beliebige wählen (z.B. die erste in der Liste).

Hinweis: Auf biologische Korrektheit muss nicht geprüft werden.

Beispiel-Aufrufe

```
$ ./u3_3 /home/xy/mapping.txt MTSGAHCNL  
ATGACAAGCGGAGCACACTGCAACCTA
```

Zusätzliche Hinweise

Auf <http://www.bioinformatics.org/sms2/> finden sich diverse Tools mit denen zufällige DNA/Protein-Sequenzen erzeugt werden können, sowie Tools zum Übersetzen zwischen RNA und Proteinen.

Beachten Sie, dass diese Tools z.T. „biologisch korrekt“ arbeiten, d.h., dass sich die Ergebnisse unterscheiden, da diese Überprüfungen bei den Übungen nicht nötig sind.

- http://www.bioinformatics.org/sms2/random_dna.html
- http://www.bioinformatics.org/sms2/random_protein.html
- <http://www.bioinformatics.org/sms2/translate.html>
- http://www.bioinformatics.org/sms2/rev_trans.html