Programmieren in der Physik – PHY.A70 – SS 2021

Prüfung 18. Juni 2021

Programmieraufgabe

Der Bauplan von Proteinen wird durch die Abfolge der vier Nukleobasen Adenin (A), Guanin (G), Cytosin (C) und Thymin (T) festgelegt. In dieser Programmieraufgabe soll diese genetische Information für das sogenannte Spike-Protein des SARS-CoV-2 Virus eingelesen und wie unten genau beschrieben ausgewertet werden. Bitte dokumentieren Sie die geforderten Funktionen, damit uns die Korrektur leichter fällt.

1) Programmname und Struktur des Hauptprogramms:
Schreiben Sie ein Python-Programm mit dem Filenamen `spike.py` mit folgenden Programmzeilen im Hauptteil des Programms:

```
gene = read_sequence('gene.fna')                        # ad 2)
base_frequencies = get_frequencies(gene)                # ad 3)
plot_frequencies(base_frequencies)                      # ad 4)
genetic_code = read_genetic_code('genetic_code.tsv')    # ad 5)
protein = decode_gene(gene, genetic_code)               # ad 6)
```

2) Einlesen der Gensequenz (6 Punkte):
Die Funktion `read_sequence(filename)` soll als input-Argument den Dateinamen des Text-inputfiles `gene.fna` übernehmen und als return-Argument *einen* String ohne Zeilenumbrüche mit der Gensequenz, also `ATGTTT...` zurückgeben. Die Datei `gene.fna` hat 3 Kopfzeilen, die übersprungen werden können, die anschließenden Zeilen 4–58 beinhalten die gewünschte Information:

```
# https://www.ncbi.nlm.nih.gov/gene/43740568
# NC_045512.2:21563-25384 S [organism=Severe acute respiratory syndrome coronavirus
# Also known as: spike glycoprotein
ATGTTTGTTTTTCTTGTTTTATTGCCACTAGTCTCTAGTCAGTGTGTTAATCTTACAACCAGAACTCAAT
TACCCCCTGCATACACTAATTCTTTCACACGTGGTGTTTATTACCCTGACAAAGTTTTCAGATCCTCAGT
...
```

3) Berechnung der Häufigkeiten der Basen (6 Punkte):
Die Funktion `get_frequencies(gene)` soll die Häufigkeiten der Buchstaben `A`, `C`, `G` und `T` im string `gene` berechnen und als Python dictionary in folgender Form zurückgeben (anhand der Zahlen können Sie auch Ihr Resultat überprüfen):

```
{'A': 1125, 'T': 1271, 'G': 703, 'C': 723}
```

Hinweis: Sollten Sie 2) nicht erfolgreich gelöst haben, bearbeiten Sie die Punkte 3)–6) mit folgendem string: `gene='ATGTTTGTTTTTCTTGTTTTATTGCCACTA'`

4) Erzeugen eines Plots (6 Punkte):

Die Funktion `plot_frequencies(base_frequencies)` übernimmt das in 3) erzeugte Python dictionary, wandelt es in zwei Listen `bases` und `frequencies` um, und erzeugt einen matplotlib `bar` plot (`plt.bar(bases, frequencies)`). Versehen Sie Ihren plot auch mit einem geeigneten Titel und Achsenbeschriftungen und speichern Sie die Grafik unter dem Namen `base_frequencies.png`.

5) Einlesen des genetischen Codes (6 Punkte):

Die Funktion `read_genetic_code(filename)` soll das Inputfile `genetic_code.tsv`

```
UUU F Phe Phenylalanine
UUC F Phe Phenylalanine
UUA L Leu Leucine
...
```

einlesen und das Python dictionary `genetic_code` in folgender Form an das Hauptprogramm zurückgeben:

```
{'UUU': 'F', 'UUC': 'F', 'UUA': 'L', ... }
```

6) Dekodieren der Basensequenz (11 Punkte):

Die Funktion `decode_gene(gene, genetic_code)` übernimmt den String `gene` (siehe Punkt 2) sowie das dictionary `genetic_code` (siehe Punkt 5) und erzeugt den string `protein`, der an das Hauptprogramm zurückgegeben wird. Dazu müssen zunächst alle Buchstaben `T` in `gene` durch `U` ersetzt werden und anschließend Basen-Tripel durch das dazugehörige Buchstabenkürzel der Aminosäure ersetzt werden.

Ein Beispiel: Nach dem Ersetzen von `T` durch `U` lauten die ersten Buchstaben in `gene` wie folgt:

```
AUGUUUGUUUUU ...
```

Diese Sequenz soll durch den `genetic_code` in folgende Buchstabenfolge übersetzt werden:

```
MFVF ...
```

Das heißt, `AUG` wird zu `M`, `UUU` wird zu `F` usw. Speichern Sie den so entstehenden string `MFVF...` in der Variable `protein`, und schreiben Sie den string in die Datei `protein.txt`.

Hinweis: Jedes Basen-Tripel kommt auch im `genetic_code` tatsächlich vor.

Achtung: Laden Sie Ihre Lösung bestehend aus den **3 Dateien** `spike.py`, `base_frequencies.png` und `protein.txt` auf `moodle.uni-graz.at` hoch!