Python for Bioinformatics

Project

The goal of this project is to extract information from a Genbank file file using the format of clustalw and to execute some basic analysis on it. You have until 23:59 of the 11th of April 2015 to pull your data into a github repository and to send an email with the URL of that repository.

The GenBank flat file format is one of the most frequently used formats for genomic data. It contains the sequence but also a set of headers with information of the included sequence, its name, length, description, accession, organism, references and features, among other fields.
A description of the format including an example can be found at
http://www.ncbi.nlm.nih.gov/Sitemap/samplerecord.html

This project consists of a program that can do several operations on files of this type. The file in the URL above is used for all the examples included in this document.

The path of the gbk file should be received as an argument at the moment of running your program:
`python3 gbkreader.py file.gbk`

The program should read the file and present a summary of the information similar to the following:

```
1 GBK Reader - (file.gbk)DNA
2 ================================================================================
3 Sequence: U49845 (SCU49845, 5028bp)
4 Description: Saccharomyces cerevisiae (baker's yeast)
5 Source: Saccharomyces cerevisiae (baker's yeast)
6 Number of References: 3
7 Number of Features: 6
8 ================================================================================
9 R:References S:Sequence M:Motif T:Translate F:Features E:Export Q:Quit
10>
```

As you see in line 9, the user has several options that he can choose from by entering the right letter in line 10. The options of the program are:

- **R:references** (*Articles with information about this protein*) If the user enters `R` the program will display how many papers are reported in this file, followed by a numbered list of the first authors of the publications (lines 12-13). The user can choose to view the details of one of them by the input of the assigned number (lines 16-25). If the user enters M, the main menu is printed again, For example:

```
10>R
11 There are 3 articles reported for the sequence U49845
12 [1] Torpey,L.E. et.al.
13 [2] Roemer,T. et. al.
14 [3] Roemer,T.
15 Input the number of a reference for details (M for the Menu):2
16 Title:
17       Selection of axial growth sites in yeast requires Axl2p, a 18
novel plasma membrane glycoprotein
19 Authors:
20         Roemer,T.
21         Madden,K.
22         Chang,J.
23         Snyder,M.
24 Journal:
25         Genes Dev. 10 (7), 777-793 (1996)
26
27 Input the number of a reference for details (M for the Menu):M
```

- **S:Sequence** (*Getting the sequence*) When the user enters S, they are asked to enter the range to display. Then, the sequence in that region is displayed in upper case with no spaces between nucleotides, with a maximum of 60 characters per line. Validations should be done to ensure that the coordinates are not outside of the range. If no range is entered, the initial summary should be displayed.
  The range is a text string that starts with either `(` or `[` then two values separated by comma: start and stop, and at the end either `)` or `]`. Square brackets imply that the position in the range should be included, while parentheses exclude it.

```
10 >S
11 Range:[5,25)
12 CTCCATATACAACGGTATCTC
13 Range:(5,25)
14 TCCATATACAACGGTATCTC
15 Range:(5,25]
16 TCCATATACAACGGTATCTCC
17 Range:[5,25]
18 CTCCATATACAACGGTATCTCC
19 Range:M
```

- **M:Motif** (*Searching a Motif*) When M is selected, the user will be asked to enter a motif sequence, composed of at least 5bp, written in any case, but characters should be nucleotides(ATGC). The wildcard character '?' represents any nucleotide in that position, and '*' represents none or many nucleotides in that position.
  The program will print all the cases where the pattern matches, including the position between

brackets and 5 nucleotides before and after the motif. The match should be displayed in lowercase and the nucleotides matching a wildcard should be uppercase. After all the matches have been reported, the user can enter another motif or a blank line to go back to the summary.

```
>M
Motif:ATCGTCT?TAG*AAGACA
Searching for the motif atcgtct?tag*aagaca...
Match 1 of 1:
tataa[849]atcgtctGtagACaagaca[867]gctca
Motif:
```

- **T:Translate** (*DNA to Proteins*) The user will input a range similar to the one for the option "sequence" above, and an ORF to be translated. If no ORF is selected (i.e. a blank line entered) the translation should be done for the 3 cases. It should only initiate the translation when it finds the start codon, and finish the translation on a stop codon. The keyword FULL is also allowed as a range to indicate that the whole sequence should be translated. When displaying the sequence it should be in upper case and each line should have a length of 60 characters

```
>T
Range:[687,3158)
ORF:1

Amino acids sequence from nucleotide 687 to 3158 in the 1st ORF:
MTQLQISLLLTATISLLHLVVATPYEAYPIGKQYPPVARVNESFTFQISNDTYKSSVDKTA
QITYNCFDLPSWLSFDSSSRTFSGEPSSDLLSDANTTLYFNVILEGTDSADSTSLNNTYQF
VVTNRPSISLSSDFNLLALLKNYGYTNGKNALKLDPNEVFNVTFDRSMFTNEESIVSYYGR
SQLYNAPLPNWLFFDSGELKFTGTAPVINSAIAPETSYSFVIIATDIEGFSAVEVEFELVI
GAHQLTTSIQNSLIINVTDTGNVSYDLPLNYVYLDDDPISSDKLGSINLLDAPDWVALDNA
TISGSVPDELLGKNSNPANFSVSIYDTYGDVIYFNFEVVSTTDLFAISSLPNINATRGEWF
SYYFLPSQFTDYVNTNVSLEFTNSSQDHDWVKFQSSNLTLAGEVPKNFDKLSLGLKANQGS
QSQELYFNIIGMDSKITHSNHSANATSTRSSHHSTSTSSYTSSTYTAKISSTSAAATSSAP
AALPAANKTSSHNKKAVAIACGVAIPLGVILVALICFLIFWRRRRENPDDENLPHAISGPD
LNNPANKPNQENATPLNNPFDDDASSYDDTSIARRLAALNTLKLDNHSATESDISSVDEKR
DSLSGMNTYNDQFQSQSKEELLAKPPVQPPESPFFDPQNRSSSVYMDSEPAVNKSWRYTGN
LSPVSDIVRDSYGSQKTVDTEKLFDLEAPEKEKRTSRDVTMSSLDPWNSNISPSPVRKSVT
PSPYNVTKHRNRHLQNIQDSQSGKNGITPTTMSTSSSDDFVPVKDGENFCWVHSMEPDRRP
SKKRLVDFSNKSNVNVGQVKDIHGRIPEML
Range:
```

- **F:Features** (*Getting the features*) The user has the option of searching features. They can enter the name of the feature, and all the features with that name will be displayed. The other option is through a range, and all the features covered by the range should be reported. As above, an empty option will display the summary again.

```
10 >F
11 Choose the type of feature query P(Position) or N(Name) :P
12 Range:[600,3200]
13 Searching for features in the range [687-3158]...
14 Feature 1 of 2:
15   gene(687,3158):
16     /gene="AXL2"
17 Feature 2 of 2:
18   CDS(687,3158):
19     /gene="AXL2"/
20     /note="plasma membrane glycoprotein"
21     /codon_start=1
22     /function="required for axial budding pattern of S.cerevisiae"
23     /product="Axl2p"
24     /protein="AAA98666.1"
25     /db_xref=:"GI:1293615"
26     /translation="MTQLQISLLLTATISLLHLVVATPYEAYPIGKQYPPVARVNESF
...
41 Choose the type of feature query P(Position) or N(Name) :N
42 Name:Source
43 Searching for features with name source...
44 Feature 1 of 1:
45   source(1,5028):
46     /organism="Saccharomyces cerevisiae"
47     /db_xref="taxon:4932"
48     /chromosome="IX"
49     /map="9"
50 Choose the type of feature query P(Position) or N(Name) :
```

- **E:Export** (*gbk to fasta*) The user should be asked for the name of a file and a fasta version of this file should be created. You have to do the respective validations to avoid overwriting an important file in your system. The user should press enter to acknowledge the system message, and then back to the summary page.

```
10 >E
11 Filename :/home/user/U49845.fasta
12 File (/home/user/U49845.fasta)Created.
13 >
```

- **Q:Quit** (*Exit or Loading*) The user should choose between exiting the program completely or loading another file by indicating its path.

```
10 >Q
11 Do you want to exit(E) or do you want load another file(F):E
```