BF527: Applications in Bioinformatics

> **Note:** Your code should follow the guidelines laid out in class, including commenting. Partial credit will be given for nonfunctional code that is logical and well commented. This assignment must be completed on your own.

HOMEWORK 5

See Blackboard for assignment and due dates

PROBLEM 5.1 (30%):

In this problem you will be writing a python script to extract information from a **CSV** (**c**omma **s**eparated **v**alues) file. The file is called "**blast_results.csv**" and can be downloaded from Blackboard. The file contains the top 100 hits from a BLAST search. You can open up the file in Excel to view the contents:

- **Row 1**: the headers describing each field (column).
- **Column 2**: this contains the ID for each hit (subject). Note that several hits actually contain multiple IDs corresponding to redundant entries in the NCBI **nr** database. If a hit contains multiple IDs, the IDs are separated by a semi-colon (;).
- **Column 13**: this contains the bit score for each hit.

Create a script called "**parse_blast_hits.py**" or write your code in the box below. This script should do the following:

1. Counts the total number of subject IDs in the file.

   > **Hint:** there is always *at least* 1 ID per line. If there are 2 IDs, there is 1 semi-colon (;) in the subject ID field. If there are 3 IDs, there are 2 semi-colons, etc. You can use the `count` function to count the number of semi-colons in a string.

2. Calculates the average bit score of the top 100 hits (all the scores in the file).

   > **Hint:** You can use the `int(string)` or `float(string)` functions to convert the bit score, which is stored as a string in the file, to a number. You will have to store all the bit scores to be able to later calculate the mean.

Your output should look like:

Total subject IDs: 181 Average bit score: 346.53

In [5]:
```
# Write your code here
```

```python
import csv #import csv module
#initialize variables
count=100 #there is always at least 1 ID per line
score=0
#open file
with open('blast_results.csv','r') as file:
    reader = csv.reader(file,delimiter=',') #read cvs file
    next(reader) #skip the first line
    for row in reader:
        count += row[1].count(';') #count ';' in second column in each row and
        score += float(row[-1]) #total score
    ave_score= score/100
    print('Total subject IDs:',count)
    print('Average bit score:',ave_score)
```

```
Total subject IDs: 181
Average bit score: 346.53
```

PROBLEM 5.2 (40%):

Protein tyrosine kinases are implicated in several forms of cancer. In this problem you will use **ClustalW** to identify the functional tyrosine kinase domain in several proteins.

> **Hint:** the domain is about 250 residues long and is well conserved.

- **(A)** Gather the protein sequences of the following four human tyrosine kinases from the **UniProt database** (http://www.uniprot.org/). A simple search of “Human” plus the gene symbols (given below) will be enough to find these four proteins. Check that the entry names you select make sense - the first search hit may not be the right one!
  1. JAK2
  2. SRC
  3. EGFR
  4. LYN

  > **Hint:** entries with **Star** are manually annotated and reviewed.

```
>sp|O60674|JAK2_HUMAN Tyrosine-protein kinase JAK2 OS=Homo sapiens OX=9606 GN=JAK2 PE=1 SV=2
MGMACLTMTEMEGTSTSSIYQNGDISGNANSMKQIDPVLQVYLYHSLGKSEADYLTFPSG
EYVAEEICIAASKACGITPVYHNMFALMSETERIWYPPNHVFHIDESTRHNVLYRIRFYF
PRWYCSGSNRAYRHGISRGAEAPLLDDFVMSYLFAQWRHDFVHGWIKVPVTHETQEECLG
MAVLDMMRIAKENDQTPLAIYNSISYKTFLPKCIRAKIQDYHILTRKRIRYRFRRFIQQF
SQCKATARNLKLKYLINLETLQSAFYTEKFEVKEPGSGPSGEEIFATIIITGNGGIQWSR
GKHKESETLTEQDLQLYCDFPNIIDVSIKQANQEGSNESRVVTIHKQDGKNLEIELSSLR
EALSFVSLIDGYYRLTADAHHYLCKEVAPPAVLENIQSNCHGPISMDFAISKLKKAGNQT
GLYVLRCSPKDFNKYFLTFAVERENVIEYKHCLITKNENEEYNLSGTKKNFSSLKDLLNC
YQMETVRSDNIIFQFTKCCPPKPKDKSNLLVFRTNGVSDVPTSPTLQRPTHMNQMVFHKI
RNEDLIFNESLGQGTFTKIFKGVRREVGDYGQLHETEVLLKVLDKAHRNYSESFFEAASM
MSKLSHKHLVLNYGVCVCGDENILVQEFVKFGSLDTYLKKNKNCINILWKLEVAKQLAWA
MHFLEENTLIHGNVCAKNILLIREEDRKTGNPPFIKLSDPGISITVLPKDILQERIPWVP
PECIENPKNLNLATDKWSFGTTLWEICSGGDKPLSALDSQRKLQFYEDRHQLPAPKWAEL
ANLINNCMDYEPDFRPSFRAIIRDLNSLFTPDYELLTENDMLPNMRIGALGFSGAFEDRD
PTQFEERHLKFLQQLGKGNFGSVEMCRYDPLQDNTGEVVAVKKLQHSTEEHLRDFEREIE
```

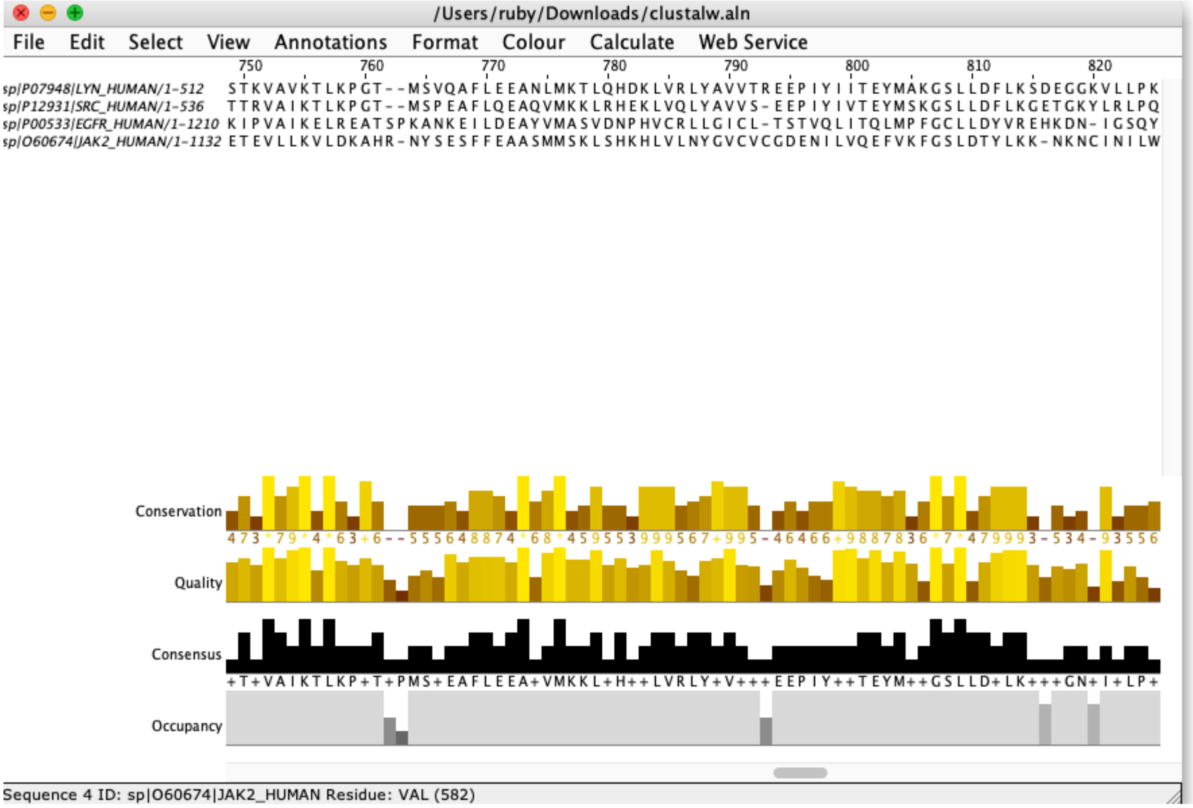
```
ILKSLQHDNIVKYKGVCYSAGRRNLKLIMEYLPYGSLRDYLQKHKERIDHIKLLQYTSQI
CKGMEYLGTKRYIHRDLATRNILVENENRVKIGDFGLTKVLPQDKEYYKVKEPGESPIFW
YAPESLTESKFSVASDVWSFGVVLYELFTYIEKSKSPPAEFMRMIGNDKQGQMIVFHLIE
LLKNNGRLPRPDGCPDEIYMIMTECWNNNVNQRPSFRDLALRVDQIRDNMAG >sp|P00533|EGFR_HUMAN
Epidermal growth factor receptor OS=Homo sapiens OX=9606 GN=EGFR PE=1 SV=2
MRPSGTAGAALLALLAALCPASRALEEKKVCQGTSNKLTQLGTFEDHFLSLQRMFNNCEV
VLGNLEITYVQRNYDLSFLKTIQEVAGYVLIALNTVERIPLENLQIIRGNMYYENSYALA
VLSNYDANKTGLKELPMRNLQEILHGAVRFSNNPALCNVESIQWRDIVSSDFLSNMSMDF
QNHLGSCQKCDPSCPNGSCWGAGEENCQKLTKIICAQQCSGRCRGKSPSDCCHNQCAAGC
TGPRESDCLVCRKFRDEATCKDTCPPLMLYNPTTYQMDVNPEGKYSFGATCVKKCPRNYV
VTDHGSCVRACGADSYEMEEDGVRKCKKCEGPCRKVCNGIGIGEFKDSLSINATNIKHFK
NCTSISGDLHILPVAFRGDSFTHTPPLDPQELDILKTVKEITGFLLIQAWPENRTDLHAF
ENLEIIRGRTKQHGQFSLAVVSLNITSLGLRSLKEISDGDVIISGNKNLCYANTINWKKL
FGTSGQKTKIISNRGENSCKATGQVCHALCSPEGCWGPEPRDCVSCRNVSRGRECVDKCN
LLEGEPREFVENSECIQCHPECLPQAMNITCTGRGPDNCIQCAHYIDGPHCVKTCPAGVM
GENNTLVWKYADAGHVCHLCHPNCTYGCTGPGLEGCPTNGPKIPSIATGMVGALLLLLVV
ALGIGLFMRRRHIVRKRTLRRLLQERELVEPLTPSGEAPNQALLRILKETEFKKIKVLGS
GAFGTVYKGLWIPEGEKVKIPVAIKELREATSPKANKEILDEAYVMASVDNPHVCRLLGI
CLTSTVQLITQLMPFGCLLDYVREHKDNIGSQYLLNWCVQIAKGMNYLEDRRLVHRDLAA
RNVLVKTPQHVKITDFGLAKLLGAEEKEYHAEGGKVPIKWMALESILHRIYTHQSDVWSY
GVTVWELMTFGSKPYDGIPASEISSILEKGERLPQPPICTIDVYMIMVKCWMIDADSRPK
FRELIIEFSKMARDPQRYLVIQGDERMHLPSPTDSNFYRALMDEEDMDDVVDADEYLIPQ
QGFFSSPSTSRTPLLSSLSATSNNSTVACIDRNGLQSCPIKEDSFLQRYSSDPTGALTED
SIDDTFLPVPEYINQSVPKRPAGSVQNPVYHNQPLNPAPSRDPHYQDPHSTAVGNPEYLN
TVQPTCVNSTFDSPAHWAQKGSHQISLDNPDYQQDFFPKEAKPNGIFKGSTAENAEYLRV APQSSEFIGA
>sp|P07948|LYN_HUMAN Tyrosine-protein kinase Lyn OS=Homo sapiens OX=9606 GN=LYN PE=1 SV=3
MGCIKSKGKDSLSDDGVDLKTQPVRNTERTIYVRDPTSNKQQRPVPESQLLPGQRFQTKD
PEEQGDIVVALYPYDGIHPDDLSFKKGEKMKVLEEHGEWWKAKSLLTKKEGFIPSNYVAK
LNTLETEEWFFKDITRKDAERQLLAPGNSAGAFLIRESETLKGSFSLSVRDFDPVHGDVI
KHYKIRSLDNGGYYISPRITFPCISDMIKHYQKQADGLCRRLEKACISPKPQKPWDKDAW
EIPRESIKLVKRLGAGQFGEVWMGYYNNSTKVAVKTLKPGTMSVQAFLEEANLMKTLQHD
KLVRLYAVVTREEPIYIITEYMAKGSLLDFLKSDEGGKVLLPKLIDFSAQIAEGMAYIER
KNYIHRDLRAANVLVSESLMCKIADFGLARVIEDNEYTAREGAKFPIKWTAPEAINFGCF
TIKSDVWSFGILLYEIVTYGKIPYPGRTNADVMTALSQGYRMPRVENCPDELYDIMKMCW
KEKAEERPTFDYLQSVLDDFYTATEGQYQQQP >sp|P12931|SRC_HUMAN Proto-oncogene tyrosine-protein
kinase Src OS=Homo sapiens OX=9606 GN=SRC PE=1 SV=3
MGSNKSKPKDASQRRRSLEPAENVHGAGGGAFPASQTPSKPASADGHRGPSAAFAPAAAE
PKLFGGFNSSDTVTSPQRAGPLAGGVTTFVALYDYESRTETDLSFKKGERLQIVNNTEGD
WWLAHSLSTGQTGYIPSNYVAPSDSIQAEEWYFGKITRRESERLLLNAENPRGTFLVRES
ETTKGAYCLSVSDFDNAKGLNVKHYKIRKLDSGGFYITSRTQFNSLQQLVAYYSKHADGL
CHRLTTVCPTSKPQTQGLAKDAWEIPRESLRLEVKLGQGCFGEVWMGTWNGTTRVAIKTL
KPGTMSPEAFLQEAQVMKKLRHEKLVQLYAVVSEEPIYIVTEYMSKGSLLDFLKGETGKY
LRLPQLVDMAAQIASGMAYVERMNYVHRDLRAANILVGENLVCKVADFGLARLIEDNEYT
ARQGAKFPIKWTAPEAALYGRFTIKSDVWSFGILLTELTTKGRVPYPGMVNREVLDQVER
GYRMPCPPECPESLHDLMCQCWRKEPEERPTFEYLQAFLEDYFTSTEPQYQPGENL
```

- **(B)** Use ClustalW to align the four protein sequences. Qualitatively and quantitatively evaluate the alignment, i.e., does this look like a good alignment? Does the alignment score support your opinion?

Yes, this looks like a good alignment as there're a lot conserved residues and regions. And the highest pairwise score is 52.3438, multiple sequene alignment score is 2472.

- **(C)** Identify the tyrosine kinase domain. Specifically, report its start and stop positions in the alignment. Provide a screenshot of the JalView output for part of the domain.

The tyrosine kinase domain starts near 750 and stops near 1000 in the alignment.



```
                             750       760       770       780       790       800       810       820
sp|P07948|LYN_HUMAN/1-512    STKVAVKTLKPGT--MSVQAFLEEANLMKTLQHDKLVRLYAVVTREEPIYIITEYMAKGSLLDFLKSDEGGKVLLPK
sp|P12931|SRC_HUMAN/1-536    TTRVAIKTLKPGT--MSPEAFLQEAQVMKKLRHEKLVQLYAVVS-EEPIYIVTEYMSKGSLLDFLKGETGKYLRLPQ
sp|P00533|EGFR_HUMAN/1-1210  KIPVAIKELREATSPKANKEILDEAYVMASVDNPHVCRLLGICL-TSTVQLITQLMPFGCLLDYVREHKDN-IGSQY
sp|O60674|JAK2_HUMAN/1-1132  ETEVLLKVLDKAHR-NYSESFFEAASMMSKLSHKHLVLNYGVCVCGDENILVQEFVKFGSLDTYLKK-NKNCINILW

Conservation                 473*79*4*63+6--555648874*68*459553999567+995-46466+9887836*7*479993-534-93556
Consensus                    +T+VAIKTLKP+T+PMS+EAFLEEA+VMKKL+H++LVRLY+V+++EEPIY++TEYM++GSLLD+LK+++GN+I+LP+
```

Sequence 4 ID: sp|O60674|JAK2_HUMAN Residue: VAL (582)

- **(D)** Replace one of the tyrosine kinase sequences with an unrelated protein sequence of your choice. Report the sequence that you used. Rebuild the alignment and compare it to the one obtained in **(B)**.

I replaced JAK2 with IL-7(Interleukin-7). >sp|P13232|IL7_HUMAN Interleukin-7 OS=Homo sapiens OX=9606 GN=IL7 PE=1 SV=1 MFHVSFRYIFGLPPLILVLLPVASSDCDIEGKDGKQYESVLMVSIDQLLDSMKEIGSNCL NNEFNFFKRHICDANKEGMFLFRAARKLRQFLKMNSTGDFDLHLLKVSEGTTILLNCTGQ VKGRKPAALGEAQPTKSLEENKSLKEQKKLNDLCFLKRLLQEIKTCWNKILMGTKEH

- **(E)** Can you still identify the tyrosine kinase domain even though you have thrown an unrelated sequence into the mix? Why or why not?

Yes, but with lower certainty. Because we have multiple sequences, the other three tyrosine kinase proteins can still provide a good alignment in that tyrosine kinase domain, although the unrelated sequences introduce some disturbances.

PROBLEM 5.3 (30%):

Thought question: describe in **English words** or **pseudocode** how you would write a program that works like a virtual ribosome, *i.e.* a script that takes an mRNA sequence and translates it to its corresponding protein sequence. Assume that the input mRNA sequence is stored in a FASTA file and the output protein sequence must be written out to a FASTA file.

> **Hint:** your program will need some additional information to be able to translate from mRNA to protein, which you must describe how to store and use.

First, establish a library of genetic code (mRNA codons as keys, amino acids as values, e.g. {"UUU":"F","UUC":"F"}, there're 3 stop codons with a value "STOP"); Input, read and save mRNA sequence FASTA file as mRNA; Create a new FASTA file to store the output protein sequence; For i in the length of mRNA: If mRNA[i:i+3] is "AUG"(“start” signal to kick off translation): translation starts,add 'M' to protein; read every three nucleotides from here and find corresponding amino acids from genetic code libriray defined at the beginning; add amino acid to the protein sequence one by one; If met a stop codon: stop translation. End End End Close mRNA file. Close protein file.