7.36/7.91/BE.490
Homework 1
Due February 24 at 1:00 PM

Note: Please see the class website for a handout describing how to submit your programming problems electronically.

1. Use of Entrez

Hexokinase is the enzyme that converts glucose to glucose-6-phosphate in the first step of glycolysis. You heard a relative (non-biologist) saying he had a mutation in the hexokinase 4 gene that caused his disease. You are curious to know more about it and go to your favorite site.

a. Go to the protein section of Entrez and do a search that limits your results to matches for 'hexokinase' that are specific to humans and that in the text contain the word 'disease'. What disease was your relative talking about?

b. Click on the 'Blink' link to see related proteins. How many Metazoan species show a match with the default parameters?

c. Look at the "Best hits". How similar is the frog (Xenopus laevis) hexokinase to the human hexokinase? What is the difference between "Identities" and "Positives"?

To learn more about hexokinase, go back to the protein record page, click on "Links" and "Map Viewer".

d. What chromosome and chromosomal region is the gene on? Click on the OMIM link. What is OMIM? Read the information provided. Given what you now know about hexokinase 4, does it make sense that a mutation in this gene causes that disease? Explain briefly.

As indicated in the OMIM record, Danial et al. found an unexpected link between the pro-apoptotic protein BAD and hexokinase. The accession number for BAD is Q61337.

e. Go to Swissprot (expasy.org) to find more information about this protein.
   i) What is its subcellular localization?
   ii) What domain/s does it contain?
   iii) Find the molecular weight for this protein.

2. BLAST

You have recently isolated and sequenced your favorite gene (*yfg*). Yeast mutants in this gene are unable to grow in fructose. You saved the protein sequence onto your hard drive but gave the file an undescriptive name. You know that you only saved two sequences that day. Unfortunately, one of them was a random sequence you generated for assessing the significance of alignments in an unrelated project. You would like to decide which sequence is random and which one is that of *yfg*. The two sequences are provided below in FASTA format:

```
> sequence 1
MPDHDFIDFWIMCAETVEYRVLLGCGEWDAIQVNEHFAIPCSYRSFEGRYPMTTQQTYLTPHQIWLCQMFRCYFEPAHG
ACKTVARTRYQRHVHCRYEKCALESPAVSWSIHMNSSLTLFNQQWSRVYMPSKMEDFDDLSGFWANMQHFKGQWHNDEG
NLYFLMSEWWASWTWEQWGFDIPNVEGHDVVPLLQNEISKRELPLCTEKAHVTHVLNPQPQMRMTDPETKHNPAYVQKR
PGVDGCIHWTGAANRTPGDQWTWHGMEFFQCFQHHRYDCDEWDPGFRMWHRWNVRIREYESPEAGYYFYQCNIFECASA
VIRYEEHAIASYLKDQDLSKLKQPYIMDTSYPARIEDDPFVFLEDTDDIFQKDFGVKTTLPERKLIRRLCEYSETEAAR
LAVCGIAAICQKRGYKTGHIAADGSVYNKYPGFKEAPQSHEVHRKIMEMPATTQPITIVPAEDGSGAGAAVIAALSEKR
IAEGKSLGIIGA
```

```
> sequence 2
PHYRKRGKWQFTPDFPPINLAAHAIQCAPPAAENCIPRQCLKIEQQRLNDLRVGGVFTWFFACPETEEYKHHIINDALV
WGEVFPYQVADTKVRQHEEEKVLTLLLKWAGAQQYNKEPRIAKSSWTIPREWNPFMWHQIPQIKQTIKNNRMSLERYTR
LDQIDNTQYYCIMGNANRYSRKPTCWWPGVMRKYCNGVHQCILKNPDVSSTQFGPMCCGKLWNHLNETYNATPRCKIET
TLYDVSKPYPFIELKLPCHPEPFNMLMWHKHKGIMRHDKLAQRGGRSYLWLTTEIMRNLKCKIHVSWNANTYFRMWRFK
EYIASVGGWDWRTFLCVNHIVICEANDMDSITANWGVDCFWCGYFLGQYSQDCAGTYATPNFTGSQFPPQEPEMPQQVA
HSHWQCCAFMLRNMCEIGSHPYMWTWDTWEDSRQSQVGKFACVHLWFVQVLYIMEMKQQYEDNYAVAMERGWDMVWHKL
DDMRIIGVPFYA
```

a. Go to http://www.ncbi.nlm.nih.gov/BLAST/ and click on "Protein-protein BLAST (blastp)". Run each of the sequences against the *nr* database with default parameters. Answer the following questions for each of the sequences:

   i) What is the E-value, the bit score, and the raw score of the best scoring match?

   ii) At the bottom of the BLAST results page, you will find two sets of values for □, K, and H – one for gapped and one for ungapped alignments. According to the bit and the raw scores, which set of values for □, K, and H do you think was used? What bit score would the top scoring hit get had the other set of values been used? Show your work.

   iii) Judging by the top scoring hit, which of the two sequences is likely *yfg*? Why?

iv) Does your guess in part (iii) make sense in terms of what you know about the phenotype of yeast deficient in this gene?

b. Assuming that you have correctly identified your gene, go back and BLAST it again, but this time with gap costs of 10 for opening and 1 for extending. How does this change your results? Explain.

c. Go back to the main BLAST page and click on "Align two sequences (bl2seq)". Blast *yfg* against the top scoring hit you identified above using the blastp program and the BLOSUM62 matrix, the PAM 250 matrix and the PAM 30 matrix. Write down the bit scores and the E values for each one. Why are they different? Be sure to explain what the E value means and why it is so much larger for the PAM250 search than for the PAM30 search.

3. Programming in Python

Write a program in python that does all of the following

a. Accepts the name of a file on the command line. This file will contain two DNA sequences in FASTA format.

b. Prints the following statistics concerning each sequence to the screen
   i) length
   ii) % GC content
   iii) % of purines
   iv) % of pyrimidines

c. Sends the amino acid translation of each DNA sequence to the screen in FASTA format.

d. Finds all 8 residue long regions that are identical between the two proteins and prints 8 residue sequence as well as the starting coordinate of this region in both proteins to the screen.

It is always a good idea to perform error checking in your code, but for this assignment it is not required. You can assume that the program is called with exactly one argument, which is the name of an existing file, which indeed contains two DNA sequences in FASTA format (i.e. no errors on the part of the user).

The following is a sample run of the program. Please try to match your program input and output formatting as closely as possible to the example below.

```
[computer] ~/TA/7.91/ps1$ cat input_example.fasta
>Random Coding Sequence 1
ATGCAAAGGCCAGGCAAGAAAGTGGCTGCTGATTCAGAGGAATCAAATGACATCAGCCAACAAGCAGAAA
ACAGAGACCAGCTCCTCCCCCAGGAAGCCAGTCCCAAAGCGTGTGAGGAAGAGGACACAGAGGAACACCG
CAAAGGGGTAACAAGCCGCAGGAAAAGAAGGCCCCCCAGAAGGCAGACAGCCCCTTAA
>Random Coding Sequence 2
ATGGTGGTGAGGAAGAGGACACAGAGGAGCGGTCAAAGGCCAGGCAAGAAAGTGGCTGTGTCTGCCGGGG
TGGGAAGAGGACACAGAGGATCCGCGCGGACCTCGCCCAGCTCAGATAAAGTACAGAAAGACAAGGCTGA
ACTGATCTCAGGGCCCAGGCAGGACAGCCGAATAGGGAAACTCTTGGGTTTTGAGTGGACAGATTTGTCC
AGTTGGCGGAGGCTGGTGACCCTGCTGAATCGACCAACGGACCCTGCAAGCCAAAGGCCAGGCAAGAAAG
TGGCTTGA
[computer] ~/TA/7.91/ps1$ python reshmahw1.py input_example.fasta
Random Coding Sequence 1
Length = 198
percent GC content = 54.04
percent purine content = 64.65
percent pyrimidine content = 35.35

Random Coding Sequence 2
Length = 288
percent GC content = 56.94
percent purine content = 63.19
percent pyrimidine content = 36.81

>Random Coding Sequence 1
MQRPGKKVAADSEESNDISQQAENRDQLLPQEASPKACEEEDTEEHRKGVTSRRKRRPPRRQTAP*
>Random Coding Sequence 2
MVVRKRTQRSGQRPGKKVAVSAGVGRGHRGSARTSPSSDKVQKDKAELISGPRQDSRIGKLLGFEWTDLSS
WRRLVTLLNRPTDPASQRPGKKVA*

The sequence QRPGKKVA is found starting at amino acid position(s) [12,
88] in sequence "Random Coding Sequence 2" and at amino acid
position(s) [2] in sequence "Random Coding Sequence 1".
```

Name this file dnaanalysis.py and submit it online. Your program will be tested on MIT Server. Please make sure that your program runs correctly there.

4. Dot Matrix

a. Using a dot matrix program (Dotlet at http://www.isrec.isb-sib.ch/java/dotlet/Dotlet.html), compare the following sequence to itself. What can you say about its primary structure? (use the scoring matrix blosum62). You will need to adjust the window, grayscale and zoom.

```
MPCFYLRSCGSLLPELKLEERTEFAHRIWDTLQKLGAVYDVSHYNALLKVYLQNEYKFSP
TDFLAKMEEANIQPNRVTYQRLIASYCNVGDIEGASKILGFMKTKDLPVTEAVFSALVTG
```

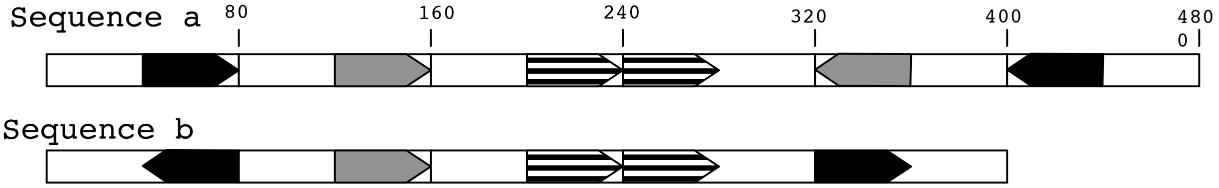
```
HARAGDMENAENILTVMRDAGIEPGPDTYLALLNAYAEKGDIDHVKQTLEKVEKFELHLM
DRDLLQIIFSFSKAGYLSMSQKFWKKFTCERRYIPDAMNLILLLVTEKLEDVALQILLAC
PVSKEDGPSVFGSFFLQHCVTMNTPVEKLTDYCKKLKEVQMHSFPLQFTLHCALLANKTD
LAKALMKAVKEEGFPIRPHYFWPLLVGRRKEKNVQGIIEILKGMQELGVHPDQETYTDYV
IPCFDSVNSARAILQENGCLSDSDMFSQAGLRSEAANGNLDFVLSFLKSNTLPISLQSIR
SSLLLGFRRSMNINVWSEITELLYKDGRYCQEPRGPTEAVGNFLYNLIDSMSDSEVQAKE
EHLRQYFHQLEKMNVKIPENIYRGIRNLLESYHVPELIKDAHLLVERKNLDFQKTVQLTS
SELESTLETLKAENQPIRDVLKQLILVLCSEENMQKALELKAKYESDMVTGGYAALINLC
CRHDKVEDALNLKEEFDRLDSSAVLDTGNYLGLVRVLAKHGKLQDAIKILKEMKEKDVLI
KDTTALSFFHMLNGAALRGEIETVKQLHEAIVTLGLAEPSTNISFPLVTVHLEKGDLSTA
LEVAIDCYEKYKVLPRIHDVLCKLVEKGETDLIQKAMDFVSQEQGEMVMLYDLFFAFLQT
GNYKEAKKIIETPGIRARSARLQWFCDRCVANNQVETLEKLVELTQKLFECDRDQMYYNL
LKLYKINGDWQRADAVWNKIQEENVIPREKTLRLLAEILREGNQEVPFDVPELWYEDEKH
SLNSSSASTTEPDFQKDILIACRLNQKKGAYDIFLNAKEQNIVFNAETYSNLIKLLMSED
YFTQAMEVKAFAETHIKGFTLNDAANSRLIITQVRRDYLKEAVTTLKTVLDQQQTPSRLA
VTRVIQALAMKGDVENIEVVQKMLNGLEDSIGLSKMVFINNIALAQIKNNNIDAAIENIE
NMLTSENKVIEPQYFGLAYLFRKVIEEQLEPAVEKISIMAERLANQFAIYKPVTDFFLQL
VDAGKVDDARALLQRCGAIAEQTPILLLFLLRNSRKQGKASTVKSVLELIPELNEKEEAY
NSLMKSYVSEKDVTSAKALYEHLTAKNTKLDDLFLKRYASLLKYAGEPVPFIEPPESFEF
YAQQLRKLRENSS
```

b. Draw a sketch Dot Matrix plot for:

   i) Sequence a vs. Sequence a

   ii) Sequence a vs. Sequence b

Assume the residues between blocks are unrelated in sequence (and ignore matches involving these sequences). Use a window length of 1.



c. Briefly explain (don't draw) how the plots will change if you instead use a window of size 10 and stringency of 10/10.

5. Dynamic Programming

Suppose that your professor suddenly wants an alignment of two proteins from different species for use in a grant proposal that is due by the end of the day. The entire campus is experiencing a network outage meaning you can't use any web servers to do the alignment.

a. Use the BLOSUM62 matrix and the Needleman-Wunsch algorithm to provide an alignment of two short regions of the proteins shown below. Create and fill in a dynamic programming matrix for the two sequences as shown in class. Assume a linear gap penalty of –8 for each gap. Show how you moved from each square to the next. Circle the traceback as done in class and show the optimal alignment. *Please use sequence 1 on the top of the matrix and sequence 2 on the left-hand side.*

Sequence 1:   INMWGAF
Sequence 2:   VSTEWGD

b. Suppose that your professor isn't satisfied with the alignment and wants to see the resulting alignment from the Smith-Waterman algorithm. Does using this algorithm change the alignment as compared to your answer in part (a)? Why or why not? *Note: you do not have to repeat the creation of the matrix. Just describe in words how the alignment changes and what about the algorithm causes the change.* Also, how does the score of the resulting alignment change in this situation?

c. Once you email your professor the alignment, he/she soon returns and demands to know why you chose to use the BLOSUM62 matrix rather one of the PAM matrices. Explain the differences between the BLOSUM62 scoring matrix and the PAM matrices that should be taken into account when generating alignments.

6. Amino acid substitution matrices

According to the entries in the PAM1 matrix, the probability that an amino acid will mutate is ~0.98% (thus, the probability that it will not mutate is ~99.02%).

a. Write a python program, which takes two command line parameters: the name of a file containing a PAM1 matrix and an integer ***n***. The program should then read in the matrix from the file and calculate the PAM***n*** matrix. It then should output the resulting matrix one row per line separating entries with spaces. Although it is always good to perform error checking in your code, it is not required for this assignment. You can assume that the program is called with exactly two parameters: 1) the name of an existing file, which indeed contains a square matrix and 2) a positive integer. Below is a sample run of the program. Make your output look as close to the output below as possible. In particular when printing out matrices multiply the entries by 10000 and display them as integers (this makes the high probability pairs more apparent).

```
[computer] ~/TA/7.91/ps1$ cat PAM1.txt
0.9867 0.0002 0.0009 0.0010 0.0003 0.0008 0.0017 0.0021 0.0002 0.0006 0.0004 0.0002 0.0006 0.0002 0.0022 0.0035 0.0032 0.0000 0.0002 0.0018
0.0001 0.9913 0.0001 0.0000 0.0001 0.0010 0.0000 0.0000 0.0010 0.0003 0.0001 0.0019 0.0004 0.0001 0.0004 0.0006 0.0001 0.0008 0.0000 0.0001
0.0004 0.0001 0.9822 0.0036 0.0000 0.0004 0.0006 0.0006 0.0021 0.0003 0.0001 0.0013 0.0000 0.0001 0.0002 0.0020 0.0009 0.0001 0.0004 0.0001
0.0006 0.0000 0.0042 0.9859 0.0000 0.0006 0.0053 0.0006 0.0004 0.0001 0.0000 0.0003 0.0000 0.0000 0.0001 0.0005 0.0003 0.0000 0.0000 0.0001
0.0001 0.0001 0.0000 0.0000 0.9973 0.0000 0.0000 0.0000 0.0001 0.0001 0.0000 0.0000 0.0000 0.0000 0.0001 0.0005 0.0001 0.0000 0.0003 0.0002
0.0003 0.0009 0.0004 0.0005 0.0000 0.9876 0.0027 0.0001 0.0023 0.0001 0.0003 0.0006 0.0004 0.0000 0.0006 0.0002 0.0002 0.0000 0.0000 0.0001
0.0010 0.0000 0.0007 0.0056 0.0000 0.0035 0.9865 0.0004 0.0002 0.0003 0.0001 0.0004 0.0001 0.0000 0.0003 0.0004 0.0002 0.0000 0.0001 0.0002
0.0021 0.0001 0.0012 0.0011 0.0001 0.0003 0.0007 0.9935 0.0001 0.0000 0.0001 0.0002 0.0001 0.0001 0.0003 0.0021 0.0003 0.0000 0.0000 0.0005
0.0001 0.0008 0.0018 0.0003 0.0001 0.0020 0.0001 0.0000 0.9912 0.0000 0.0001 0.0001 0.0000 0.0002 0.0003 0.0001 0.0001 0.0001 0.0004 0.0001
0.0002 0.0002 0.0003 0.0001 0.0002 0.0001 0.0002 0.0000 0.0000 0.9872 0.0009 0.0002 0.0012 0.0007 0.0000 0.0001 0.0007 0.0000 0.0001 0.0033
0.0003 0.0001 0.0003 0.0000 0.0000 0.0006 0.0001 0.0001 0.0004 0.0022 0.9947 0.0002 0.0045 0.0013 0.0003 0.0001 0.0003 0.0004 0.0002 0.0015
0.0002 0.0037 0.0025 0.0006 0.0000 0.0012 0.0007 0.0002 0.0002 0.0004 0.0001 0.9926 0.0020 0.0000 0.0003 0.0008 0.0011 0.0000 0.0001 0.0001
0.0001 0.0001 0.0000 0.0000 0.0000 0.0002 0.0000 0.0000 0.0000 0.0005 0.0008 0.0004 0.9874 0.0001 0.0000 0.0001 0.0002 0.0000 0.0000 0.0004
0.0001 0.0001 0.0001 0.0000 0.0000 0.0000 0.0000 0.0001 0.0002 0.0008 0.0006 0.0000 0.0004 0.9946 0.0000 0.0002 0.0001 0.0003 0.0028 0.0000
0.0013 0.0005 0.0002 0.0001 0.0001 0.0008 0.0003 0.0002 0.0005 0.0001 0.0002 0.0002 0.0001 0.0001 0.9926 0.0012 0.0004 0.0000 0.0000 0.0002
0.0028 0.0011 0.0034 0.0007 0.0011 0.0004 0.0006 0.0016 0.0002 0.0002 0.0001 0.0007 0.0004 0.0003 0.0017 0.9840 0.0038 0.0005 0.0002 0.0002
0.0022 0.0002 0.0013 0.0004 0.0001 0.0003 0.0002 0.0002 0.0001 0.0011 0.0002 0.0008 0.0006 0.0001 0.0005 0.0032 0.9871 0.0000 0.0002 0.0009
0.0000 0.0002 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0001 0.0000 0.0001 0.0000 0.9976 0.0001 0.0000
0.0001 0.0000 0.0003 0.0000 0.0003 0.0000 0.0001 0.0000 0.0004 0.0001 0.0001 0.0000 0.0000 0.0021 0.0000 0.0001 0.0001 0.0002 0.9945 0.0001
0.0013 0.0002 0.0001 0.0001 0.0003 0.0002 0.0002 0.0003 0.0003 0.0057 0.0011 0.0001 0.0017 0.0001 0.0003 0.0002 0.0010 0.0000 0.0002 0.9901
[computer] ~/TA/7.91/ps1$ python pamn.py PAM1.txt 120
PAM1 is:
9867     2     9    10     3     8    17    21     2     6     4     2     6     2    22    35    32     0     2    18
   1  9913     1     0     1    10     0     0    10     3     1    19     4     1     4     6     1     8     0     1
   4     1  9822    36     0     4     6     6    21     3     1    13     0     1     2    20     9     1     4     1
   6     0    42  9859     0     6    53     6     4     1     0     3     0     0     1     5     3     0     0     1
   1     1     0     0  9973     0     0     0     1     1     0     0     0     0     1     5     1     0     3     2
   3     9     4     5     0  9876    27     1    23     1     3     6     4     0     6     2     2     0     0     1
  10     0     7    56     0    35  9865     4     2     3     1     4     1     0     3     4     2     0     1     2
  21     1    12    11     1     3     7  9935     1     0     1     2     1     1     3    21     3     0     0     5
   1     8    18     3     1    20     1     0  9912     0     1     1     0     2     3     1     1     1     4     1
   2     2     3     1     2     1     2     0     0  9872     9     2    12     7     0     1     7     0     1    33
   3     1     3     0     0     6     1     1     4    22  9947     2    45    13     3     1     3     4     2    15
   2    37    25     6     0    12     7     2     2     4     1  9926    20     0     3     8    11     0     1     1
   1     1     0     0     0     2     0     0     0     5     8     4  9874     1     0     1     2     0     0     4
   1     1     1     0     0     0     0     1     2     8     6     0     4  9946     0     2     1     3    28     0
  13     5     2     1     1     8     3     2     5     1     2     2     1     1  9926    12     4     0     0     2
  28    11    34     7    11     4     6    16     2     2     1     7     4     3    17  9840    38     5     2     2
  22     2    13     4     1     3     2     2     1    11     2     8     6     1     5    32  9871     0     2     9
   0     2     0     0     0     0     0     0     0     0     0     0     0     1     0     1     0  9976     1     0
   1     0     3     0     3     0     1     0     4     1     1     0     0    21     0     1     1     2  9945     1
  13     2     1     1     3     2     2     3     3    57    11     1    17     1     3     2    10     0     2  9901
PAM120 is:
2658   331   727   735   327   583   830  1113   343   580   365   378   463   223  1124  1279  1302    86   206   859
 151  3796   254   130   101   517   154    78   553   193   119   968   300    96   291   302   197   538    54   125
 343   257  1626   928    83   371   527   372   700   180   104   529   143   112   244   568   438    92   205   155
 417   147  1078  2578    51   598  1547   432   409   153    74   331   105    47   221   416   323    33    78   158
 118    95    65    33  7262    34    32    45    90   106    26    34    36    43   116   266   130    18   238   155
 242   471   353   489    37  2605   892   150   926   135   193   377   232    54   349   216   198    49    59   132
 486   185   640  1632    50  1161  2691   369   386   208   130   337   162    53   304   363   296    28    91   206
1124   227   779   788   197   387   648  4904   255   223   168   308   212   134   475  1076   580    60    88   434
 135   446   597   296    88   804   251    79  3705    77   104   204    83   162   230   173   143   108   263    95
 225   149   166   119   155   128   140    85    92  2611   555   165   603   389    96   158   345    38   151  1186
 313   197   253   129    68   407   181   155   343  1344  5601   265  2181   966   278   212   348   349   327  1078
 352  1881  1030   596    80   774   567   284   465   343   198  4531   923    89   370   601   671   145   131   229
  77    95    55    32    17   100    41    29    41   252   386   184  2331    94    39    71   114    21    29   220
 109    91   116    43    58    55    45    94   178   439   450    47   307  5509    44   137   124   277  1806   141
 658   367   291   229   143   471   305   283   369   175   188   240   174   113  4307   603   411    51    60   231
1024   546   962   602   580   412   511   832   354   309   181   527   314   213   847  2086  1225   291   203   351
 886   303   629   401   197   306   339   384   225   548   234   492   394   156   491  1033  2589    72   175   546
  11   134    13     5     7    10     5     7    12     8     6    18     8    89    10    54    14  7507    90     4
  88    37   157    56   239    52    74    37   256   138   134    31    70  1358    34    98    96   190  5404    96
 612   207   233   192   262   236   234   281   229  2058   794   198   927   264   298   321   600    47   193  3620
[computer] ~/TA/7.91/ps1$
```

Name this program pamn.py and submit it online. Your program will be tested on MIT Server. Please make sure that your program runs correctly there.

b. In the sample run above, the program calculates the PAM120. From the values in this matrix (keeping in mind that the entries are multiplied by 10000) calculate the average probability of amino acid conservation.

7. Phylogenetic analysis

Suppose you are studying a set of proteins, which have a conserved BLOCKS motif. You would like to establish a phylogenetic relationship between these proteins. In order to do this, you decide to look at the degree of divergence among the nucleotide sequences coding for the conserved motif. You obtain these sequences and they look as follows:

```
HXK_PLAFA:  AAA ATT ATA AAT ATC GAA TTT GGT AAT TTT
HXK_SCHMA:  GTC GTC ATA AAC ACA GAG TGG GGT GCA TTC
HXK1_BOVIN: ATG TGC ATT AAC ATG GAG TGG GGT GCT TTT
HXK1_HUMAN: ATG TGC ATC AAC ATG GAG TGG GGG GCC TTT
HXK1_TOBAC: ATG GTT ATC AAC ATG GAA TGG GGT AAT TTT
```

a. Apply the Jukes-Cantor model to find the number of real substitutions for all pairs of the sequences above.

b. Assuming that what you found in a) is a measure of genetic distance, apply the Unweighted Pair-Group Method with Arithmetic mean (UPGMA) to build a phylogenetic tree for the proteins. Draw the resulting tree and label edge length (assuming scaled branch length).

c. The sequences above come from a conserved BLOCKS motif in hexokinases. The species of origin are malaria parasite *P. falciparum*, blood fluke (*Schistosoma mansoni*), cow (*Bos taurus*), human (*Homo sapiens*), and common tobacco (*Nicotiana tabacum*) respectively. Given this information, do you think your tree in part a) is reasonable?