Graded assignment Comparative Genomics

Biancamaria Florenzi
r0777934
18-12-2020

Introduction

During the course of this assignment, we have applied some of the techniques and principles of comparative genomics that we have seen in class. In particular, we started by comparing protein sequences of two species, which in my case were Blastopirellula marina and Roginitalea biformata. The former is a bacteria member of the planctomycetes phylum, inhabiting aquatic environments, while the latter belongs to the phylum bacteroidetes and is also a marine bacterium. We have compared the two sequence dataset against each other, and then tried to infer orthology based on the best bidirectional hits; in other words, we have looked for those pairs of genes that are more similar to each other than to any other gene, and assume that those were orthologs. We have selected a pair of orthologs among those and found 23 homologs from different species. We have then created a multiple sequence alignment to create a gene tree for those proteins. At the same time we have created a species tree for those 25 species, using the one-copy 16S RNA sequences, as is commonly done in bacteria. We have looked for discordance in topology between the gene tree and species tree in a process called reconciliation, to try and infer what biological events caused the discordance. Then we have further analysed the protein multiple sequence alignment to identify functional regions, keeping in mind one of the pillar elements of comparative genomics, which is that regions of genes that are highly conserved among species are often essential for survival.

Methods

Part I : Compute orthologs using Best Bidirectional Hits
Available protein sequences have been downloaded from NCBI for two species of interest as fasta files. The command line BLAST+ has been used locally to perform blast comparison of the datasets. A blast database has been created from each of the fasta files containing protein sequences, using the command -makeblastdb. Then each of the sequence files has been blasted as a -query with blastp against the -db given by other species proteins' database. From the two runs I have obtained two output files, one contains the blast of the Blastopirellula proteins against the Robiginitalea and the other viceversa.
To analyse this output I have used python, and in particular I have looked for the Best Bidirectional Hits. The two files are imported as a panda dataframe, where each line corresponds to a pair of proteins belonging to two different species. The bit-score is used as a measure of the blast correspondence, as this score measures sequence similarity independent of query sequence length and database size and is normalized based on the raw pairwise alignment score. The higher the bit-score the better the similarity and is appropriate here, also given the fact that the two organisms have a large difference in number of available sequences. The other option would have been to use the e-score, which

is also given in the blastp output and measure the probability for the sequence match to be coincidental. I exploited the dataframe structure to find first the best hits for each protein in each direction, by grouping-by and taking the maximum bit-score. I notice that in some cases, one protein has the same maximum score with more proteins in the other species, and all of these are maintained. Using the e-score would have given rise to the same situation, as these are also not unique in the output file. Then I merged the dataset of best hits in one, and removed those cases in which the best hit was not bidirectional. I then only selected those hits with bit scores higher than 50, which indicates reliable similarity.

Part II: Reconciliation of species and gene tree

I took one of the protein pairs, which are protein A3ZZ40 of blastopirellula and A4CJJ2 of robiginitalea, and I blasted for homologs of A3ZZ40. This is found with the GenBank identification name EAQ78234.1 and corresponds to a glutamate synthase [NADPH] large chain. From NCBI I performed a blastp against the database of non-redundant protein sequences, including robiginitalea as an organism. I then selected among the first 100 matches 23 other proteins from other species. Notice that the last entries among the first 100 matches have a sequence identity of 56% with the queried protein. On the other hand, the robiginitalea ortholog has a sequence similarity of 50%. The species are:

1. Roseimaritima ulvae
2. Mariniblastus fucicola
3. Calyptogena magnifica
4. Rubinisphaera brasiliensis
5. Spirochaeta lutea
6. Bathymodiolus azoricus thioautotrophic gill symbiont
7. Carboxydocella sporoproducens DSM 16521
8. Blastopirellula marina
9. Planctomycetes bacterium
10. Litorilinea aerophila
11. Nitrosomonas halophila
12. Gimesia maris DSM 8797
13. Pedosphaera parvula Ellin514
14. Thermoflexus hugenholtzii
15. Aspergillus carbonarius
16. Malus domestica
17. Candidatus Thioglobus autotrophicus
18. Spirochaeta thermophila
19. Pseudohongiella spirulinae
20. Robiginitalea biformata
21. Fuerstia marisgermanicae
22. Rhodopirellula solitaria
23. Rubripirellula amarantea
24. Alienimonas californiensis
25. Cucumis melo var. makuwa

The fasta file has been manipulated with python to show the name of the species as a header, instead of the name of the protein.

I have then performed multiple sequence alignment of these 25 proteins using clustal Omega, and visualised it in seaview, as it provides a practical and visually pleasing way to inspect the alignment. I then decided to use seaview for the construction of a tree as well, using PhyML based on Blosum 62 matrix and including 100 bootstraps.
In parallel, I have retrieved the 16S rRNA sequences for the chosen 25 species. I have not found a way to automate this, as the variability in nomenclature and formatting requested manual intervention. Most of the sequences have been retrieved from the Silva database, while those that were not available from the rdp 16S rRNA browser. Sequences obtained here have been manipulated to contain U instead of T and merged with those previously found. The final fasta file has been again manipulated in its headers.
Multiple sequence alignment was again performed as before, the maximum likelihood PhyML tree was constructed with HKJ model and bootstrapping.
The two resulting trees have been inspected in FigTree and ITol manually for the identification of duplication events. A reconciliation has been attempted with SeaView ("Reconcile with Treerects"), but without success.

Part III: Identification of functional regions
The clustalw multiple sequence alignment has been parsed with python to look for highly conserved regions. Shannon entropy has been used as a measure of conservation, and it is calculated with the formula:

$$H = -\sum_{i=1}^{M} P_i \log_2 P_i$$

Where Pi is the proportion of residues of amino acid i, and the summatory is done over all amino acids. I first have translated the clustal file to a fasta file using AlignIO from the Bio python package, as this would make it more straightforward to extract sequences. I then have selected each column of the alignment separately, as most conservation measurements rely on a value that is calculated for each column. A problem I encountered soon was the following: from the clustal alignment the extremity of the alignment contains many gaps, as the sequences I started from have different lengths. Initially I inserted the gap symbol, "-" as the 21st possible symbol together with the amino acids, but that obviously lead to a problem, since the score for those positions in which most sequences had a gap was very good, and it looked like the region was highly conserved, where in reality that was not the case. I then tried the approach of inserting a correction for the presence of gaps; eventually I decided to discard the columns that contained more than a certain thresholds of gaps, and to assign to those the arbitrary value of -9 of entropy. The minimum entropy value is obtained when there is one residue that is present in all sequences at that position and the maximum when all the residues are present with the same frequency (and depends on the number of aligned sequences). I have obtained a dataframe that contains the Shannon entropy for each of the positions and the corresponding index.

Results

Orthologous

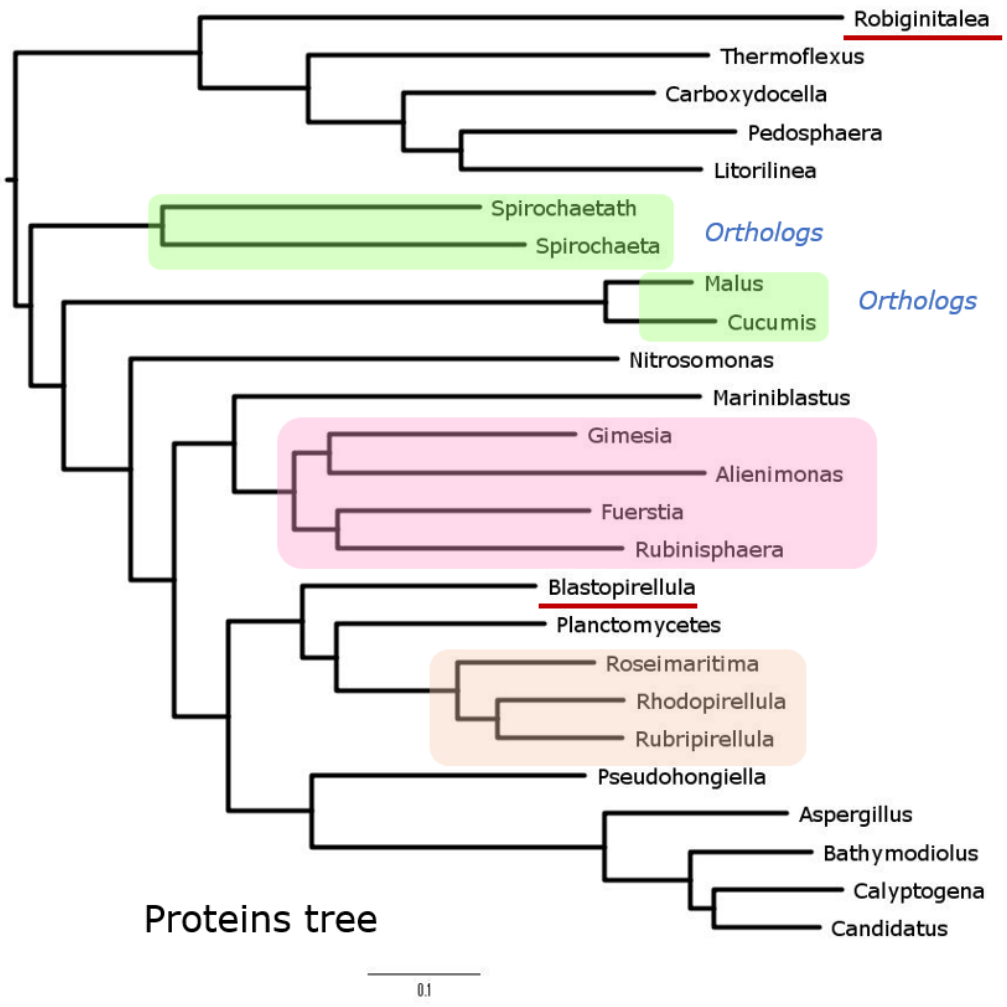
There are approximately 28,000 protein sequences for bastopirurella marina and 3,000 for the robiginitalea biformata. Best bidirectional hits correspond to orthologs, which means that they rose from the same ancestor gene in the past, and I found 125 between the two species.

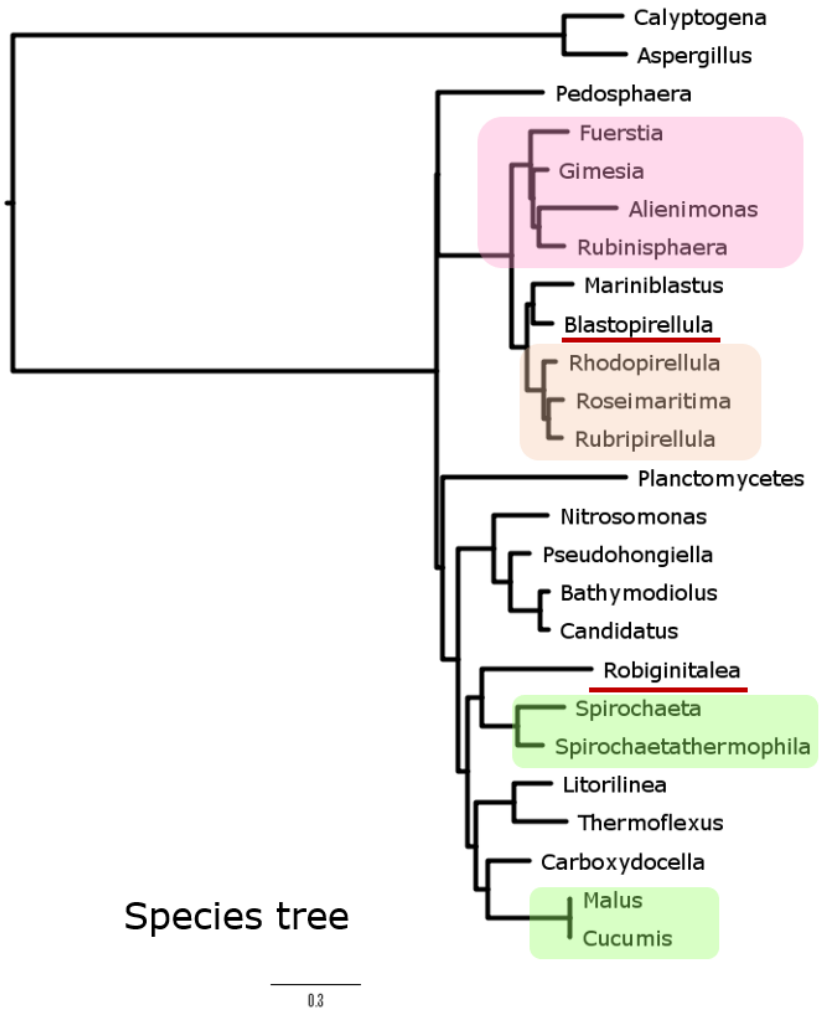
Index	qseqid_x	sseqid_x	bitscore_x	qseqid_y	sseqid_y	bitscore_y
36531	WP_105356685.1	WP_049764876.1	664	WP_049764876.1	WP_105356685.1	669
13172	WP_105357976.1	WP_148214315.1	647	WP_148214315.1	WP_105357976.1	655
15769	WP_158265593.1	WP_148214371.1	638	WP_148214371.1	WP_158265593.1	138
31010	WP_105359175.1	WP_015753775.1	629	WP_015753775.1	WP_105359175.1	625
41672	WP_105355915.1	WP_015755045.1	624	WP_015755045.1	WP_105355915.1	628
28176	WP_105357385.1	WP_041327530.1	485	WP_041327530.1	WP_105357385.1	275
11015	WP_105359282.1	WP_041327231.1	431	WP_041327231.1	WP_105359282.1	449
49120	WP_105359650.1	WP_041327170.1	354	WP_041327170.1	WP_105359650.1	357
9596	WP_158265302.1	WP_148214460.1	335	WP_148214460.1	WP_158265302.1	386
42158	WP_105356980.1	WP_015755829.1	325	WP_015755829.1	WP_105356980.1	356
9476	WP_116344554.1	WP_041326912.1	321	WP_041326912.1	WP_116344554.1	342
28473	WP_158261274.1	WP_015754285.1	310	WP_015754285.1	WP_158261274.1	181
4253	WP_105350097.1	WP_148214487.1	305	WP_148214487.1	WP_105350097.1	327
52070	WP_105357622.1	WP_148214257.1	304	WP_148214257.1	WP_105357622.1	309

Trees reconciliation

I expected my resulting trees to be more similar to each other, and I hoped to be able to over impose them and therefore distinguish between duplication events and speciation events; however, the discordance between the two trees was significant and probably has multiple causes. My process of thinking was: when I see that there is correspondence between the species tree and the proteins tree, for instance two leaf exactly correspond, I can say that the two proteins originated from the same ancestor protein with a speciation event (the same speciation that I see happening in the species tree!). On the other hand, when there is no correspondence, it is much more complicated to say exactly what happened, and we can most make a hypothesis.

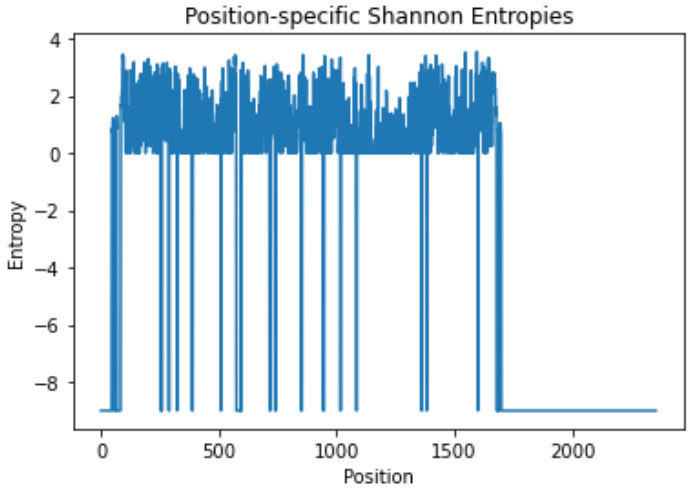
In this particular case, I was not able to determine if the proteins from blastopirellula and robiginitalea are actually orthologous. I was able to determine that for instance the proteins from the two spirochetas (Thermophila and Lutea) and from malus domestica and cucumis melo are probably orthologs, since I can map their speciations to the species tree. There are other events that are quite more complex, for instance the group Gimesia, Aliemonas, Fuerstia and Rubinispheara. It looks like, according to the species tree, that fuerstia speciated earlier, and Gimesia, Alienimonas, Rubinisphaera have a more recent common ancestor. On the other hand, the gene tree shows a different layout, in which the Fuerstia protein and the Rubinisphaera seem to have originated from the same ancestor. These two are therefore probably not orthologs, but rather paralogs originated from duplication.


Robiginitalea
Thermoflexus
Carboxydocella
Pedosphaera
Litorilinea
Spirochaetath
Spirochaeta
Orthologs
Malus
Cucumis
Orthologs
Nitrosomonas
Mariniblastus
Gimesia
Alienimonas
Fuerstia
Rubinisphaera
Blastopirellula
Planctomycetes
Roseimaritima
Rhodopirellula
Rubripirellula
Pseudohongiella
Aspergillus
Bathymodiolus
Calyptogena
Candidatus
Proteins tree
0.1


Calyptogena
Aspergillus
Pedosphaera
Fuerstia
Gimesia
Alienimonas
Rubinisphaera
Mariniblastus
Blastopirellula
Rhodopirellula
Roseimaritima
Rubripirellula
Planctomycetes
Nitrosomonas
Pseudohongiella
Bathymodiolus
Candidatus
Robiginitalea
Spirochaeta
Spirochaetathermophila
Litorilinea
Thermoflexus
Carboxydocella
Malus
Cucumis
Species tree
0.3

Conserved positions

Shannon entropy is often used to identify regions in protein that have low complexity or high conservation. We know that sequence conservation is connected to essentiality, since if a residue or a short fragment of residues is important, it is less likely to change (or, more correctly, changes have been lost). In the sequence alignment I am considering, there are 382 positions with Shannon entropy lower than 0.1.



There are two types of conserved sites: there are small fragments of 3-4 amino acids positions with low scores, and there are large segments of 12+ amino acids with low scores, the latter are mostly found around position 1000 of the alignment. Further inspections, which have not been carried out during this assignment, could be aimed at finding correspondence among the “sites” section on the NCBI page of the blastopirellula protein. For instance in the figure, some of the active or binding sites of blastopirellula are mentioned:

```
Site            order(901..904,931,954,976,982,1002,1044,1074..1076,1115,
                1117,1138..1139,1147,1153,1158)
                /site_type="active"
                /db_xref="CDD:239202"
Site            order(901..904,931,954,976,1044,1074..1075,1115,1117,
                1138..1139)
                /site_type="other"
                /note="FMN binding site [chemical binding]"
                /db_xref="CDD:239202"
Site            order(982,1002,1075..1076)
                /site_type="other"
                /note="substrate binding site [chemical binding]"
                /db_xref="CDD:239202"
Site            order(1147,1153,1158)
                /site_type="other"
                /note="3Fe-4S cluster binding site [ion binding]"
                /db_xref="CDD:239202"
```

Conserved sites indeed are likely involved in some essential activity, often structural, as they might correspond to a binding site when the protein is folded, or be located right at the interface of protein-protein interactions.

It is also important to note: if is the structure of the protein and its binding site that determine which amino acids are to be found at the interface, it is also true that the amino

acids involved will not necessarily be adjacent; what matters is the 3D structure, and not the order of the amino acids in the sequence. This is why, above, one site is scattered on non-adjacent indices. For the same reason, I have decided not to perform a selection of conserved sites based on their index during this assignment, as I thought this would be misleading.

Discussion

During the assignment we have applied concept and infer methods that are commonly used in comparative genomics. Best bidirectional hits are one of the ways in which we can infer orthology between two species; given the fact that verifying the evolutionary definition of orthology is not possible, it is necessary to make approximations and try to deduce the evolutionary events that happened.
By reconstructing the phylogenetic tree of individual proteins that are found across a set of species, we were again trying to reconstruct the evolutionary history of those genes, and trying to infer what events occurred. In these cases, it is common that the reconstructed tree does not agree with the species tree, since evolution is not linear nor simple. Even so, the two trees should not be completely different. We wished to distinguish families of orthologous genes, that have actually originated from the same ancestral gene through speciation, and exclude paralogous, that have been generated from a duplication event, and xenologous.
We have then used Shannon entropy to identify low complexity regions in a protein multiple sequence alignment. Shannon entropy measures the homogeneity of a position in the alignment in terms of amino acid content. Conserved regions across proteins likely have a key role in the function of the proteins themselves, and are often also connected to diseases in which that function is altered.

References

Bastien Boussau, Celine Scornavacca. Reconciling Gene trees with Species Trees. Scornavacca, Celine; Delsuc, Frédéric; Galtier, Nicolas. Phylogenetics in the Genomic Era, No commercial publisher | Authors open access book, pp.3.2:1--3.2:23, 2020. ⟨hal-02535529⟩

Chrysa Ntountoumi, Panayotis Vlastaridis, Dimitris Mossialos, Constantinos Stathopoulos, Ioannis Iliopoulos, Vasilios Promponas, Stephen G Oliver, Grigoris D Amoutzias, Low complexity regions in the proteins of prokaryotes perform important functional roles and are highly conserved, Nucleic Acids Research, Volume 47, Issue 19, 04 November 2019, Pages 9998–10009, https://doi.org/10.1093/nar/gkz730

Kristensen DM, Wolf YI, Mushegian AR, Koonin EV. Computational methods for Gene Orthology inference. Brief Bioinform. 2011 Sep;12(5):379-91. doi: 10.1093/bib/bbr030. Epub 2011 Jun 19. PMID: 21690100; PMCID: PMC3178053.

Guharoy, M., Chakrabarti, P. Conserved residue clusters at protein-protein interfaces and their use in binding site identification. BMC Bioinformatics 11, 286 (2010). https://doi.org/10.1186/1471-2105-11-286

Python code

```
"""
@author: Biancamaria
"""
#Create a python script that generates BBH for your two species.

#Import packages
import os
import numpy as np
import pandas as pd
import statistics as stat

os.chdir("C:/Users/Biancamaria/Documents/Courses/Comparative Regulatory
gen/Comparative/Assignment")

#Take the blastp outputs as input files
#In forward file the query are blastopirellula proteins and the db are the robiginitalea proteins
#In the backward file, viceversa

fwd="Blasto_to_robi"
bk="Robi_to_blasto"
score="bitscore"

#Takes the standard blastp results and returns dataframes of bit-scores
def getResults(fwd,bk, score):
    fwd_results = pd.read_csv(fwd, sep="\t", header=None)
    bk_results = pd.read_csv(bk, sep="\t", header=None)
    #These are standard column headers from blast
    headers = ["qseqid", "sseqid", "pident", "length",
      "mismatch", "gapopen", "qstart",
      "qend", "sstart", "send", "evalue", "bitscore"]
    fwd_results.columns = headers
    bk_results.columns = headers
    #each line is a pair of proteins;
    #notice that the bit-scores are already in decreasing ordered for each query protein
    fwd_scores = fwd_results[["qseqid", "sseqid", score]]
    bk_scores = bk_results[["qseqid", "sseqid", score]]
    return fwd_scores, bk_scores

#from the scores dataset, selects max score for each
def getBBH(fwd_scores, bk_scores):
  #only keeps the best hit of backward forward score
  #notice that some proteins have the same bit-score compared to multiple other proteins
  fwd_bh = fwd_scores.groupby(['qseqid']).max().reset_index()
  bk_bh = bk_scores.groupby(['qseqid']).max().reset_index()
  bbh = pd.merge(fwd_bh, bk_bh, how='outer',
          left_on='sseqid', right_on='qseqid')
```

```
    #these are not bidirectional hits
    bbh = bbh.loc[bbh.qseqid_x == bbh.sseqid_y]
    bbh_sorted = bbh.sort_values(by='bitscore_x', ascending=False)
    #A bit score of 50 indicates good similarity and I select with this cut-off
    BBHs=bbh_sorted.loc[(bbh_sorted.bitscore_x>50) & (bbh_sorted.bitscore_y>50)]
    return BBHs

#main
fwd_scores, bk_scores = getResults(fwd,bk,score)
BBHs=getBBH(fwd_scores, bk_scores)
#print the number f BBHs
print("There are %d pairs of best bidirectional hits" % len(BBHs))
BBHs.head()
```

```
"""
@author: Biancamaria
"""
#import packages
import sys, os, re
import matplotlib.pyplot as plt
import numpy as np
import math
from Bio.Seq import Seq
from Bio import SeqIO
from Bio.SeqRecord import SeqRecord
from Bio import AlignIO
import pandas as pd
from pandas import DataFrame

os.chdir("C:/Users/Biancamaria/Documents/Courses/Comparative Regulatory
gen/Comparative/Assignment")
# The input is a multiple alignment of protein sequences.
#In our case these are 25 proteins of different species, two of which are orthologous
#because were BBH in the first part of the assignment

#Transform the clustalo alignment file to a fasta file
def toFasta(clustal, fasta):
    clustal = open(clustal, "rU")
    fasta = open(fasta, "w")
    alignments = AlignIO.parse(clustal, "clustal")
    AlignIO.write(alignments, fasta, "fasta")
    fasta.close()
    clustal.close()

#exctract sequences from fasta file
def getSequences(fasta):
    sequences=[]
    with open(fasta, 'r') as handle:
        for record in SeqIO.parse(handle, "fasta"):
            sequences.append(str(record.seq))
    return sequences

#I want each string in columns to be a column of the alignment
def getColumns(sequences):
    columns=[]
    for i in range(0,len(sequences[1])): #taking the length of one,they are all same size
        column=''
        for seq in sequences:
            column+=seq[i]
        columns.append(column)
```

```
    return columns

def calculateFreq(column):
    #Amino acids
    AAs=['A','R','N','D','C','E','Q','G','H','I','L','K','M','F','P','S','T','W','Y','V','X']
    freq={}
    for aa in AAs:
        freq[aa]=float(column.count(aa)/len(column))
    return freq

def calculateShannon(columns):
     #for each column, i will calculate the shannon entropy.
    Hs=[]
    for col in columns:
        freq=calculateFreq(col)
        H=0
        for value in freq.values():
            if value!=0:
                H+=float(value)*math.log2(float(value))
        if col.count('-')>20:
            H=+9
        Hs.append(-H)
    return Hs

#main
#take the input file
clustal="25prot_multiple_seq_al.clustal_num"
fasta="25prot_multiple_seq_al.fasta"
toFasta(clustal, fasta)
sequences=getSequences(fasta)
columns=getColumns(sequences)
Hs= calculateShannon(columns)
print(Hs)
sites=DataFrame(range(1,2358))
sites["Hs"]=Hs
print(sites)

#Highly conserved positions are those with H close to 0
flags=[]
for H in Hs:
    if H<0.1:
        flags.append(True)
    else:
        flags.append(False)

flags.count(True)
```