Genetic Data Analysis using R on BRCA1 gene

i) Algorithm to find all possible numbers of ORF in the gene for all the 6 Possibilities for BRCA1 gene.

Attached is python code **CountORF.py**.

Explanation: The sequence is read and for all the start codons 'ATG' or equivalently, 'AUG', an algorithm is designed to find sequences AUG……..(until stop codon). No of ORFs are calculated for 6 possible orientations and the output is:

```
visheshrangwani@Visheshs-MacBook-Air Question 2 % python3 CountORF.py
No of ORF while reading from sequence in forward direction from 1st index: 330
No of ORF while reading from sequence in forward direction from 2nd index: 318
No of ORF while reading from sequence in forward direction from 3rd index: 303
No of ORF while reading from reverse complement from 1st index: 303
No of ORF while reading from reverse complement from 2nd index: 301
No of ORF while reading from reverse complement from 3rd index: 324
Total no of ORFs = 1879
```

CountORF_different_algo.py considers that if inside an ORF, if we find another start codon we consider that also as an ORF. So the o/p is:

```
visheshrangwani@Visheshs-MacBook-Air Question 2 % python3 CountORF2.py
No of ORF while reading from sequence in forward direction from 1st index: 452
No of ORF while reading from sequence in forward direction from 2nd index: 405
No of ORF while reading from sequence in forward direction from 3rd index: 395
No of ORF while reading from reverse complement from 1st index: 386
No of ORF while reading from reverse complement from 2nd index: 372
No of ORF while reading from reverse complement from 3rd index: 422
Total no of ORFs = 2432
```

ii) Calculation of the possible number of exons

Python Code in the attached file **CountExons.py**. In this, the ORFs containing more than 30 nucleotide sequences(including the 3 nucleotides of the stop codon, if not to be considered, the code can easily be modified to l>=33) are counted. The ORFs are counted as per CountORF.py code. The no of exons are counted for all 6 possibilities of reading the sequence. The output on running the code is:

```
visheshrangwani@Visheshs-MacBook-Air Question 2 % python3 CountExons.py
No of ORF while reading from sequence in forward direction from 1st index: 215
No of ORF while reading from sequence in forward direction from 2nd index: 204
No of ORF while reading from sequence in forward direction from 3rd index: 190
No of ORF while reading from reverse complement from 1st index: 200
No of ORF while reading from reverse complement from 2nd index: 198
No of ORF while reading from reverse complement from 3rd index: 213
Total no of ORFs = 1220
```

iii) Finding number of homologous structural proteins possible

Suppose k exons form a protein and the total number of exons are n. So the no of different possibilities, by Permutation and combination are k! * C(n, k). We take all the 6 possibilities of different ways of reading the sequence and add them.
Python Code in CountProteinStructures.py file. Output:

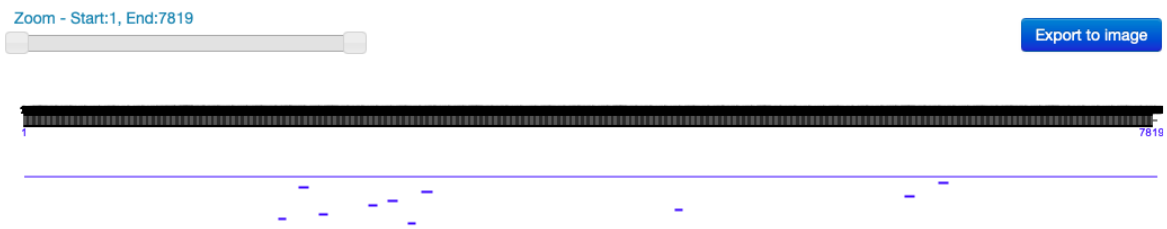
```
visheshrangwani@Visheshs-MacBook-Air Question 2 % python3 CountProteinStructures
.py
a) When 5 exons combine to form protein: 1731935722560 possible structures
b) When 3 exons combine to form protein: 42324750 possible structures
c) When 10 exons combine to form protein: 5551882144701323086656 00 possible stru
ctures
d) When 6 exons combine to form protein: 348736467532320 possible structures
```

iv) Obtaining structure of BRCA1 gene

BRCA1 gene's fasta in 'BRCA1 sequence.fasta'.
Converted to mRNA and primary structure of protein using this website.

```
DNA:
GCT GAG ACT_ TCC TGG ACG GGG GAC AGG CTG TGG GGT TTC TCA GAT AAC TGG GCC CCT GCG CTC AGG AGG CCT TCA CCC TCT GCT CTG GGT AAA GGT AG
mRNA:
CGA CUC UGA_ AGG ACC UGC CCC CUG UCC GAC ACC CCA AAG AGU CUA UUG ACC CGG GGA CGC GAG UCC UCC GGA AGU GGG AGA CGA GAC CCA UUU CCA UC
Protein:
ARG LEU STOP ARG THR CYS PRO LEU SER ASP THR PRO LYS SER LEU LEU THR ARG GLY ARG GLU SER SER GLY SER GLY ARG ARG ASP PRO PHE PRO SE
```

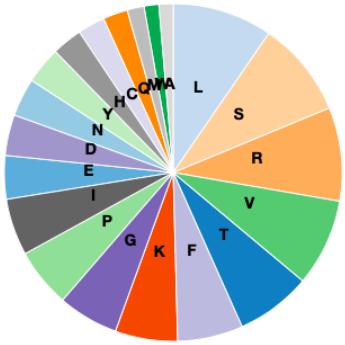
Results attached in BRCA1mRNA.txt for mRNA and prProtein.txt for primary structure of protein.
In order to translate mRNA for 6 frames this website is used. The results are stored as: 5'->3'(1).txt, 5'->3'(2).txt, 5'->3'(3).txt, 3'->5'(1).txt, 3'->5'(2).txt, 3'->5'(3).txt for each corresponding frame.
These 6 are the different possible translation to amino acid sequence from mRNA. We convert 5'->3'(2).txt to secondary and tertiary protein structures.
This is done using this site. (Since it didnt allow sequences of length > 8000, some last amino acids were trimmed).



Summary

Sequence Length	7819
Number of Aligned Proteins	10

Amino Acid composition



Primary Structure of Protein: We know that proteins are formed by the peptide bonds in the sequence of amino acids. This sequence of amino acids is the primary structure of the protein.

Secondary structure of protein: These long amino acids sequences, fold over each other and form polypeptide linkages between atoms. This folding of the amino acid sequence is the scondary structure of protein

Tertiary Structure of Protein: There are further linkages due to H-bonds, electrostatic forces, van der Waals forces of attraction and disulphide linkages. These cause further folding of the secondary structure of protein and is called tertiary structure.

v) Finding the number of genes on each chromosomes

Attached is an R file named **BioMart.R**.

i) A loop is run over all 22 chromosomes and the no of genes on each chromosome is stored in a vector 'v'. The no of genes is found using the getBM() function in which we find the number of all possible gene IDs on each of the 22 normal chromosomes and then for the 'X' and 'Y' chromosomes. Then the values of these number of genes is printed.

```
[1] "No of genes on chromosome 1 5557"
[1] "No of genes on chromosome 2 4274"
[1] "No of genes on chromosome 3 3252"
[1] "No of genes on chromosome 4 2704"
[1] "No of genes on chromosome 5 3036"
[1] "No of genes on chromosome 6 3125"
[1] "No of genes on chromosome 7 3061"
[1] "No of genes on chromosome 8 2523"
[1] "No of genes on chromosome 9 2361"
[1] "No of genes on chromosome 10 2380"
[1] "No of genes on chromosome 11 3401"
[1] "No of genes on chromosome 12 3091"
[1] "No of genes on chromosome 13 1422"
[1] "No of genes on chromosome 14 2318"
[1] "No of genes on chromosome 15 2249"
[1] "No of genes on chromosome 16 2588"
[1] "No of genes on chromosome 17 3086"
[1] "No of genes on chromosome 18 1254"
[1] "No of genes on chromosome 19 3018"
[1] "No of genes on chromosome 20 1480"
[1] "No of genes on chromosome 21 892"
[1] "No of genes on chromosome 22 1404"
> print(paste("No of genes on chromosome X", v[23]))
[1] "No of genes on chromosome X 2452"
> print(paste("No of genes on chromosome Y", v[24]))
[1] "No of genes on chromosome Y 522"
>
```

vi) Finding the sum of total number of genes on each chromosomes

```r
## Code for part ii) of the question
print(paste("Sum total of all the genes", sum(v)))

##From the above code, we observe that chromosome 1 has the maximum amount of genes i.e. 5557

#Store all gene ids in 'genes' variable for chromosome 1
genes <- getBM(attributes = "external_gene_name", filters = "chromosome_name", values = "1", mart = mart)

#Finding number of transcripts by iterating over all genes and finding their 'transcript_count'
#also finding gene with max no of transcripts and transcript having maximum length

max_transcripts <- getBM(attributes="transcript_count", values = genes[1,], mart = mart,filters = "external_gene_name")[[1]]
no_of_transcripts <- max_transcripts
max_transcripts_index <- 1
longest_tr <- max(getBM(attributes="transcript_length", values = genes[1,], mart = mart,filters = "external_gene_name"))
longest_tr_idx <- 1
for (i in 2:dim(genes)) {
```

25:51 (Top Level) R Script

Console Terminal Jobs

R 4.1.2 · ~/

```
[1] "No of genes on chromosome 13 1422"
[1] "No of genes on chromosome 14 2318"
[1] "No of genes on chromosome 15 2249"
[1] "No of genes on chromosome 16 2588"
[1] "No of genes on chromosome 17 3086"
[1] "No of genes on chromosome 18 1254"
[1] "No of genes on chromosome 19 3018"
[1] "No of genes on chromosome 20 1480"
[1] "No of genes on chromosome 21 892"
[1] "No of genes on chromosome 22 1404"
> print(paste("No of genes on chromosome X", v[23]))
[1] "No of genes on chromosome X 2452"
> print(paste("No of genes on chromosome Y", v[24]))
[1] "No of genes on chromosome Y 522"
> print(paste("Sum total of all the genes", sum(v)))
[1] "Sum total of all the genes 61450"
>
```

vii) Finding the transcript having maximum length

We see that the chromosome with the max no of genes is chromosome 1. We perform further calculations for the above chromosome.

a) In order to find this, we first store all the genes available in biomaRt in a data structure and then we iterate over all those genes and find the 'transcript_count' attribute of all genes by providing the filter as 'external_gene_name' and value as the ith index of the data structure(data frame used). This done in **getBM()** function. We sum over all these iterations and store and print them in the 'no_of_transcripts' variable.
NOTE: The program runs very slowly and due to this may give a warning. However, after 5-6 tries, it didn't give a warning.

Theoretical rough calculations:
According to the article, there are on average 3.42 transcripts per gene. Since by analysis in part i), there were 5557 genes on chromosome 1. So the total no of transcripts should be roughly 19005 transcripts. However, the program would show a much smaller no as no of genes stored in the 'genes' variable would be much lesser than 5557 as there may not be sufficient data on biomart.

b) Using the above approach, we find genes having the maximum number of transcripts also. By storing it in a variable ‘max_transcript’ and its variable in ‘max_transcript_index’.
Using this we get the maximum number of transcripts for genes[max_transcript_index,]

c) For this, we again do the same. We store the gene index with the currently maximum transcript in the ‘longest_tr_idx’ variable and maximum length in ‘longest_tr’ variable. We find the max length of transcript for each gene, compare with ‘longest_tr’ and update ‘longest_tr’ and ‘longest_tr_idx’. Finally, we print its Ensembl transcript ID.

Below is the Screenshot of Code and its o/p for parts a), b) and c):

```r
max_transcripts <- getBM(attributes="transcript_count", values = genes[1,], mart = mart,filters = "external_gene_name")[[1]]
no_of_transcripts <- max_transcripts
max_transcripts_index <- 1
longest_tr <- max(getBM(attributes="transcript_length", values = genes[1,], mart = mart,filters = "external_gene_name"))
longest_tr_idx <- 1
for (i in 2:dim(genes)) {
  n <- max(getBM(attributes="transcript_count", values = genes[i,], mart = mart,filters = "external_gene_name"))
  tr_length <- max(getBM(attributes="transcript_length", values = genes[i,], mart = mart,filters = "external_gene_name"))
  no_of_transcripts = no_of_transcripts + n[[1]]

  #Checking for max no of transcripts
  if (n > max(max_transcripts)){
    max_transcripts=n
    max_transcripts_index=i
  }

  #Checking for max length of transcripts
  if(tr_length > longest_tr){
    longest_tr = tr_length
    longest_tr_idx = i
  }

}
print(paste("Total no of transcripts on chromosome 1:", no_of_transcripts))
print(paste("Gene having max no of transcripts: ", genes[max_transcripts_index, ]))
print(paste("Transcript having maximum length: ", getBM(attributes="ensembl_transcript_id", values = genes[longest_tr_idx,], mart =
```

```
> print(paste("Total no of transcripts on chromosome 1:", no_of_transcripts))
[1] "Total no of transcripts on chromosome 1: 7418"
> print(paste("Gene having max no of transcripts: ", genes[max_transcripts_index, ]))
[1] "Gene having max no of transcripts:  ST3GAL3"
> print(paste("Transcript having maximum length: ", getBM(attributes="ensembl_transcript_id", values = genes[longest_tr_idx,], mart = mar
t,filters = "external_gene_name")))
[1] "Transcript having maximum length:  c(\"ENST00000371191\", \"ENST00000664495\", \"ENST00000476646\", \"ENST00000407417\", \"ENST000004
79364\", \"ENST00000371189\", \"ENST00000664149\", \"ENST00000670151\", \"ENST00000662015\", \"ENST00000403491\", \"ENST00000371187\", \"E
NST00000485903\", \"ENST00000371185\", \"ENST00000371184\", \"ENST00000603233\", \"ENST00000496712\", \"ENST00000663597\", \"ENST000006572
34\", \"ENST00000655770\", \"ENST00000482020\", \"ENST00000357977\", \"ENST00000493627\")"
```

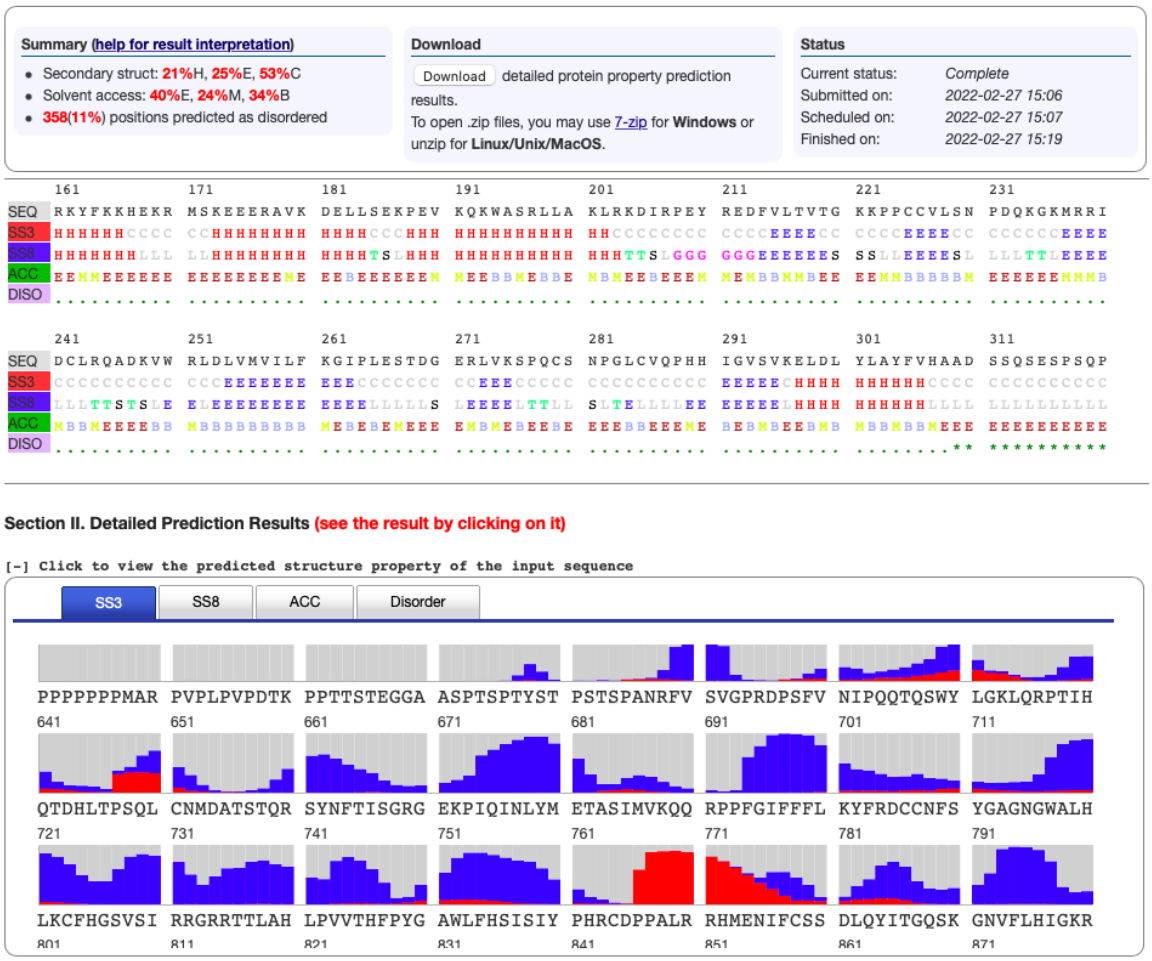
vii) Translation into mRNA

The code for conversion to protein is written in protein.py file.
The output is:

```
visheshrangwani@Visheshs-MacBook-Air Question 3 % python3 protein.py
Ser Pro Leu Ser Leu Thr Ala STOP Leu Thr Leu Lys Thr Phe Gln Ser Met Leu Gly Phe Val Arg Ser Val Leu Val Ser Leu Arg Leu Leu Val Gly Arg Ala Gln Pro Ser Glu Ser Phe Leu Ser His Ser Arg Ser Leu Ala Arg Ser
 Leu Ser Leu Cys His Ser Leu Pro Leu Ser Arg Gly Thr Leu Thr Phe Ser Glu Asp STOP Thr Ser Ile Phe Phe Leu Arg Gly Ser Ser Asp Val Asn Arg Leu Ser Lys Asn Phe Leu Glu Lys Ala Gln Lys Leu Pro Arg Phe Lys ST
OP Ile Tyr Lys Lys Phe Thr Phe Phe Phe Pro Cys Cys Leu Leu Arg STOP Ala Asp Ser STOP Asn Phe Lys Glu Cys Met Arg Lys Met STOP Arg Tyr Leu Leu Lys Val Gly Lys STOP Leu Arg Glu Asp Gly Val Gln Ala Arg Lys A
rg Met Cys Thr Lys Leu Asp Val Arg Ala Phe Ala Phe Met Lys Phe Phe Val Leu Phe Ala Tyr Ser Phe Leu Leu Leu Ser Arg His Phe Leu Leu Asn Asp Ser Leu Phe Gly Leu Gln Phe Val Phe Thr Arg Arg Ala Glu Asp Arg P
he Asn Ala Phe Leu STOP Ala Gly Leu Ile Ala Leu Leu Lys Gln Glu Cys Gln Cys Pro Phe Phe Gly Gly Thr Thr Gln Glu Arg Leu Gly Leu Val Phe Pro Phe Tyr Ala Ser STOP Leu Thr Glu Ala Val Arg Leu Phe Gln Thr Ser
 Asn Leu Glu Gln Tyr His STOP Asn Lys Phe Pro STOP Gly Asp Leu Ser STOP Leu Pro Leu Ala Glu His Phe Arg Gly Val Thr Arg Leu Gly Pro Glu Thr Gln Val Gly Val Val Tyr Pro Gln Arg Gln Phe Leu Glu Leu Asn Ile
Asn Arg Met Lys His Val Arg Arg Leu Ser Ser Val Arg Leu Ser Gly Ser Val Gly Ser Leu Arg Leu STOP Phe Leu Val Gly Leu Leu Pro Val Asn Pro Lys Val Leu Ser Lys Gln Cys Ser Pro Gln Lys Ser Gln STOP Leu Asp Hi
s Ser His Ser Val Cys Gly Tyr Arg Arg Pro STOP Pro Gly Leu Lys Arg Glu Ser Leu Asn Leu Ser Arg Ser Met Met Ser Tyr Ser Gly Pro Arg Tyr Ser Ser Arg Asn Gly Ser Cys Arg Arg Cys Ser Arg Cys Phe Ala Glu Phe A
rg His Leu Leu Leu Tyr Leu Ser Gly Pro Leu Leu Gly Lys Ile Cys Pro Val Pro Ala Arg Gly Pro Ser Pro Ser Val Ser Ser Pro Thr Val Leu His Leu Gly Pro Tyr Gly Arg Gly Trp Cys Asp Phe Phe Ser Leu Phe Arg Pro L
ys Ser Ser Gly Arg Gly Ser Val Trp Arg Arg Asp Pro Cys Arg Lys Cys Val Val Val Ala Gly Gln STOP Cys Pro Gly Ser Arg Ser Gly Val Arg Cys Gly Ser STOP Glu Val Lys Gly Cys Ser Gly STOP STOP Val Val Gly Pro G
ly Met Lys Ser Val Gly Arg STOP Ala Ile Val Gly Val Leu Cys Asp Phe Leu Lys Gln Val Glu Gln Thr Gly Leu Arg Pro Val Val Arg Pro Val His Pro Lys Glu Leu Gly Leu Pro Ser Ser Val Pro Phe His Val Leu Gly Lys
Glu Gly Trp Gly Tyr Asn Gly Gly Gly Gly Gly Gly Gly Tyr Arg Ser Gly His Gly Asp Gly His Gly Leu Cys Phe Gly Gly Trp Cys Ser Cys Leu Pro Pro Arg Arg Arg Gly Cys Ser Gly Trp Met Ser Cys Gly Ser Trp Arg Gly
Arg Leu Ala Lys Gln Ser Gln Pro Gly Ala Leu Gly Ser Lys His Leu STOP Gly Val Val Cys Val Arg Thr Met Asp Pro Ile Phe Asn Val Ala Gly Trp STOP Val Val Cys Leu Val Asp Trp Gly Arg Val Glu Thr Leu Tyr Leu Ar
g Trp Ser Trp Val Ala Ser Met Leu Lys STOP STOP Ser Pro Ser Pro Leu Phe Gly STOP Val STOP Leu Asn Met Tyr Leu Cys Arg Ser STOP Tyr Gln Phe Val Val Ser Gly Ile Gly Lys Pro STOP Lys Lys Lys Asn Phe Met Lys
Ser Leu Thr Thr Leu Lys Ser Ile Pro Arg Pro Leu Pro Thr Arg Asn Ile Val Asn Phe Thr Lys Val Pro Ser His Ser STOP Ser Thr Ala Pro Ile Ala Ser STOP Trp Glu Arg Val Thr Glu Gly Gln His Cys Val Lys Gly Met Pr
o Arg Thr Asp Lys Val Ser STOP Ser Thr STOP Met Gly Val Ser Thr Leu Gly Gly Thr Arg Asn Ser Ser Val Tyr Leu Leu Ile STOP Lys Thr Ser Ser Thr Leu Asp Val Ile Tyr Cys Pro Val Ser Phe Pro Leu Gln Lys Asp Ile
 Val STOP Pro Phe Ser Phe Thr Ser Ile STOP Gly Lys Asn Lys Gln Gly Asn Gln Thr Lys Pro Ser Arg Gln Ser Ser Ile His Ile Leu Tyr Arg Ala Phe Leu Leu Leu Arg STOP Val STOP Arg Leu Val Lys Ala Phe Thr Val STO
P STOP STOP STOP STOP Gln Val Lys Lys Phe Ser Lys Leu Tyr Asp Val Asn Val Asn Phe Phe Val Val Val Val Lys Lys Leu Leu Lys Leu Leu Leu Cys Phe STOP Val Tyr Leu Tyr Pro Phe Tyr Gln Thr Lys Leu Ser Leu Thr V
al Leu Ser STOP Phe Cys STOP Leu Lys Thr STOP Leu Ile Ser Arg STOP Ser Asn Leu Ser Ile Tyr Arg Asn Ile Leu Ser Leu Thr Thr Phe Lys Ile Lys Thr Lys Arg Ser Gly Lys Trp Gly Pro Glu Ile STOP STOP Pro His Phe
 Leu Ser Glu Arg Ser Phe Ser Pro Arg STOP Lys Arg Lys Ile Gln Lys Ile Thr Ile STOP Thr STOP Gly Asn Phe STOP Leu Pro Leu Ile Phe Phe Phe Cys Phe Phe Val STOP Ile Thr Thr Lys Met Val Leu Tyr Asn STOP Cys A
sn Leu Arg Leu Ile Asn Lys Arg Gln STOP Phe Arg Val Leu Ile Leu Leu STOP Tyr His Arg Gln Leu Lys Tyr Ser Ile Phe Ser Asp Ile Thr STOP Cys Gly STOP Asp Phe Leu Lys Cys Gly STOP Thr Val Val Ser STOP Ile Val
 Asn Ser Leu Trp Ser Asn Met Val Ser Asn Arg Ser Ser Lys Phe Leu Glu Thr STOP Gly Leu Asp Pro Ile Pro Phe Ile Ile Ile Ile Ile Ile Ile Ile Thr Thr Leu Val Met Thr Val Phe Thr Thr Thr Gly Lys Ser Se
r Ser His Glu Gln Ser Leu Phe Phe Phe Ser Asn Val Gly Val Cys Arg Thr STOP Asn Thr Phe Phe Phe Val Phe Cys Phe Cys Phe Phe Phe Val Phe Val Phe Val Phe Phe His Val His STOP Lys Asp Gly Leu
 Ser Leu Arg Gly Glu Thr Lys Gly Thr Leu Lys Phe Phe Gln Arg Ile STOP Tyr Val Tyr Ile Cys Lys Thr Asn Arg Ser Gly Arg Leu Val Ser Phe Arg Asp Val Gln Asp His Asn His STOP Leu Lys Ile Tyr Gln Leu His Pro Il
e Ser Ser His Val Ile Leu Cys Tyr Asn Ile Arg Asp Glu Ile Arg Glu Arg Lys Phe Asp Thr Lys Ser Arg Glu Val Phe Pro Arg STOP STOP Gln Asp Met Leu Val His STOP Leu STOP Glu Pro Phe Cys Glu Thr Arg Ser Arg Ly
s Arg Gly Phe Val Thr Thr Lys Gln Asn Lys Lys Lys Glu Leu Asn Val Leu Leu Phe Leu Arg Tyr Ser Tyr Thr Glu Arg Asp Leu Ser Leu Ile Asn Thr Leu Thr Thr Cys Thr Arg Arg STOP Leu Ser Pro Lys Arg Asp Gln Phe V
al Asn Leu Cys Glu Lys His Leu His STOP Lys Tyr Ser Arg STOP Gln Ser Lys Leu STOP Ile Leu Ile Tyr Leu Ser Ile Asn Lys Ala Ser Ile Leu Ile Arg Asn Asn Arg Thr Cys Gln Pro Pro Ser Arg STOP Asp Ser Arg His L
eu Arg Ile Leu STOP Ser STOP Asn Leu Gln Arg Arg Lys Met Leu Asn Lys Pro Gln Lys Glu Ile Lys Gly Gly Leu Tyr Thr Phe Ile Arg Tyr Gly Arg Ile STOP Tyr Ile Phe Ser Asn Arg STOP STOP Leu Lys Arg Lys STOP Lys
 Lys Ile Asn Trp Phe Gln Pro Arg Arg Glu Leu Arg His Thr Lys Ser Thr Asn Arg Asn Met Lys Lys His STOP Lys Leu Thr Cys Thr Arg Pro Ser Arg Gly Glu Ser Leu Arg Gln Phe Asp Thr Ser STOP Phe Asp Trp Asp Glu V
al Arg STOP Asn Lys Asn Val Asp Lys Glu Ser Val Arg Ser Glu Phe Phe Phe Phe Leu Leu Lys Glu Lys Glu Val Glu Gly Val Gly STOP Lys Glu Ser Ser Asp Val Ser STOP Val STOP Val Ile Phe His Ile Leu Arg Val Lys I
le Tyr Val Gln Lys Phe Glu Gly Ser Arg Leu Leu Thr Asn Asp Gln STOP Arg STOP Ile Pro Phe Phe Leu Phe Tyr Phe Val Lys Pro Lys Ser Ile His Arg Leu Phe Thr Asn Leu Ser Ala Asn Arg Asp Asn Ile Pro His STOP Gi
n Ser His Thr Thr Val Lys Asp Ile Gly Arg Ser Tyr Glu Thr Glu Gly Asp Ile Val Asn Arg Pro Val Met Lys STOP Tyr Cys Ile His Val Ile Phe Tyr Phe Phe Ser Phe Phe Ser Phe Phe Phe Ser Phe Leu Ser Ser Lys Val T
hr Lys Cys His Phe Pro Phe Phe Gly Ser Val Cys Lys Thr Val Leu Glu Arg Tyr Ser Val Pro STOP Trp Ser Phe Asn Gln His Ile Cys Tyr Phe Asn Val Gly Lys Asn Leu Phe Cys Leu Ile Arg Tyr Gln Thr Arg Gln Asn Gln
 Gln Lys Ser Leu Val Gln His Gln Tyr STOP Arg Phe Ser Leu Leu Phe Asp Gly Arg Val Phe Tyr Thr Leu Asn Ile Thr Thr Tyr Gln Asn Asn STOP Cys Asp STOP Ile Arg Trp Asp Lys Ile Leu Arg Pro Glu Asn Ile Thr Thr A
sn Leu Leu Leu Phe Lys Asp Ser Leu Ile Gly Ser Phe STOP Arg Leu Ala Ser Asn Ile Ser Val Ala Ser Ile Leu Thr Lys Arg Lys Lys Asn Gly Gly Glu Thr Leu Asp Leu Asn Ser Asn Phe Glu Arg Thr Lys Glu Lys Lys Lys
 Lys Lys His Ser STOP Ile Tyr Val Asn Lys Lys Arg Glu Leu Pro Tyr Arg Ile Arg Ile Arg Thr Leu His Ile Gln Met Thr Phe Trp STOP His Leu Pro Ala Met Ser Ile Arg His Lys Asn Phe His Lys Thr Thr STOP Phe Lys L
eu Leu Arg Phe STOP Ser Leu Glu Thr Cys Leu Asn Val Leu Phe Tyr Lys Ile Ser Glu Ile Leu STOP Ser Leu Arg Gln Ile Pro Leu Leu Leu Val Pro Asn Ser STOP Phe Met Cys Phe Asn Val Lys Thr Ser Thr Lys STOP Gln V
al Phe Arg Arg Pro Leu Ile Ser Gln His STOP Phe Val Leu Phe Ser Leu Trp Leu Ser Thr Val Thr Lys Thr Lys Asn Arg Ile Phe Val Ser Leu Leu Arg Thr Pro Ser Glu Thr Met Asp Pro Thr His Leu Ser Val STOP Leu Ser
 Ile Val Val Asn Phe Arg Val Gly Asn STOP Asp Ser Val STOP Thr STOP Lys Arg Lys Gly Ser Leu Pro STOP Leu Phe STOP Pro Val Ser Thr Gln Gly Val STOP Arg His Lys Lys Gln Thr Asn Lys Gln Thr Lys Lys Lys Pro L
eu Thr Ser Val Ser Val Leu Val Ile Glu Asn Arg Lys Lys Thr Ser Glu Lys Glu Asp Arg Glu Gln Gly Arg Ser Phe Cys Gly Glu Lys Thr Ser Lys His Ala Ser Pro Gly Gly Ser Leu Arg Cys Glu Arg Tyr Asp Leu Ser Glu G
lu Lys Gly Asp Ser Lys Leu Asp Tyr Lys Thr Gln Lys Cys Gly Thr Val Leu Ser Met His Thr Ser Arg Ala Arg Gly Pro Gly Arg Asp Gly Gly Gly Pro His Pro Arg Ser Val Arg Leu Phe Gln Cys Gly Gly Ser Arg Thr Leu G
lu Val Lys Asp Cys Gln Phe His Thr Ser Leu Gly Leu Lys Gly Leu Gly Ala Thr Ser Gln Ser Gly Val Leu Ile Arg Ser Gly STOP Arg Phe Ser Asn Arg Asn STOP Lys Arg Phe Gly Asn Arg Gly Ile Arg Asn Leu Gly His Gln
 Cys His Arg Ile Ser STOP Leu Phe Trp Thr Tyr Ser Phe Leu Glu Lys Lys Asp Thr STOP Lys Lys Gln Lys Ile Pro Phe Pro STOP Asp Ser Tyr Ser Lys Leu His Glu Thr STOP Tyr Arg Asp Leu Tyr Arg Asn Arg Tyr His His
 His Leu Arg Glu Arg Glu Arg Glu Lys Asn Lys Arg Arg Gln Lys Lys Glu Gln Lys STOP Glu Arg Lys Glu Lys Ile Lys Thr Ile Val Arg Arg Tyr Leu Val STOP Ile Leu Cys Trp Trp Tyr His Tyr Ile Pro Leu Glu Val Thr V
al Glu Phe Trp Val Gly Ser Ser Arg Gln Val Ser Val Arg Val Phe Arg Arg Asp Gly Trp Pro Ser Arg Lys STOP Val Phe Pro Lys Val Pro Val Lys Thr Glu Leu Gln Leu Arg Thr STOP Met Thr Asn Cys His Leu Gly Glu Thr
 Glu Glu Val Gln Val Pro Thr Arg Lys Gln Ile Asn Arg Ser Thr Tyr Lys Arg Lys Ile Lys Lys Val Ile Phe Ser Ser Pro Tyr Phe STOP Arg Asn Tyr Ile Ser Ser Ile Val Thr Ser Ile Arg Tyr Arg Glu Leu Thr Ser Leu Me
t Thr Gln Trp Leu Lys Leu STOP Leu Asn Tyr Lys Leu Arg Leu Gly Asn Val Pro Arg STOP Tyr Pro Thr Val Leu Leu Asn Val Phe Thr Leu Leu Asn Pro Ser STOP Val Arg Arg Lys Val Val Thr Phe Thr Tyr Arg His Lys STO
P Thr Tyr Ser STOP Tyr Val Ile Arg STOP Thr Ser Phe Phe Asp Ile Val Glu Thr Ile Asp Glu Arg Gln Met Lys Leu Gly Thr Glu Lys Pro Val Thr Ile Leu Arg Ser Lys Ser Cys Leu Ser STOP Tyr Ser Phe Gly Arg Ile Met
 Asp Tyr Ile Ile Leu His His Gly His Phe Arg His Asn Ile Thr Arg Arg Leu His Ser Lys Lys Met Trp Pro Ser Glu Ala Thr Asp Ser Gln Ser Trp Lys Thr Ile Ser Thr Thr Phe Tyr Trp His Val Ile Ile Thr Tyr Ala Lys
 Val Leu Arg Ile Thr STOP His Thr Tyr Asn Glu Phe Gln Tyr Asn Gln Lys Lys Arg Leu Asn Gln Glu Gln Ser Ile Gln Ile Ser Ile Lys STOP Leu Leu Arg Leu Arg Gln Lys Arg Thr Lys Ala Asp Lys Ser His Lys Thr Ser I
le Cys Lys STOP Ile STOP Val Ile Met Ser Thr Lys Gln Lys Gln Lys Thr Lys Glu Lys Gly Ser Pro Val Arg Thr His Lys Leu Lys Ser Leu Val Met Asp Arg Ser Lys Asn Arg Val Leu Leu Ile Asn Leu Lys Phe Leu
 Thr Arg Arg Phe Cys Glu Thr Leu Val Asn Val Gln Ser Pro STOP Lys Glu Leu Gly Thr Asp Tyr Asp Thr Pro Leu Thr Met STOP Lys Asn Tyr Leu Leu Gln Val Arg Ser Met Gly Thr Asn STOP Thr Glu Val Trp Ser Gly Gly
```

vii) Primary structure output

Primary structure is the output of the program protein.py.
To find secondary, tertiary and quaternary structure, this website was used and the results obtained are downloaded in the folder 'Structure of Protein'.
This is the link to the result of the job.
Following is a screenshot of the result's summary:



A more holistic and clear result was obtained from this website.
Results occur in various formats.
Stored in ProteinStruct.png.
Screenshot:

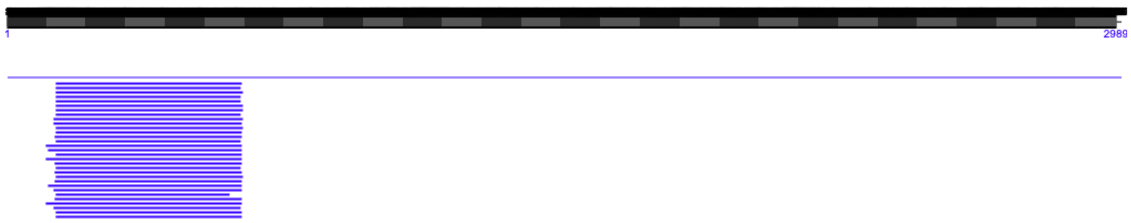
Predicted features

What am I seeing Here? This viewer lays out predicted features that correspond to regions within the queried sequence. Mouse over the different colored boxes to learn more about the annotations

Export to PNG

Zoom - Start:1, End:2989

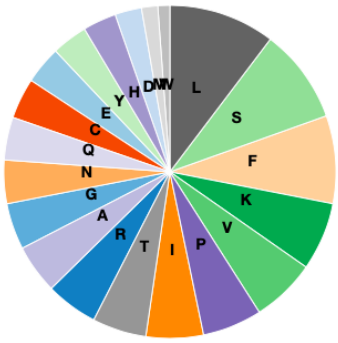
Export to image



Summary

Sequence Length	2989
Number of Aligned Proteins	31

Amino Acid composition

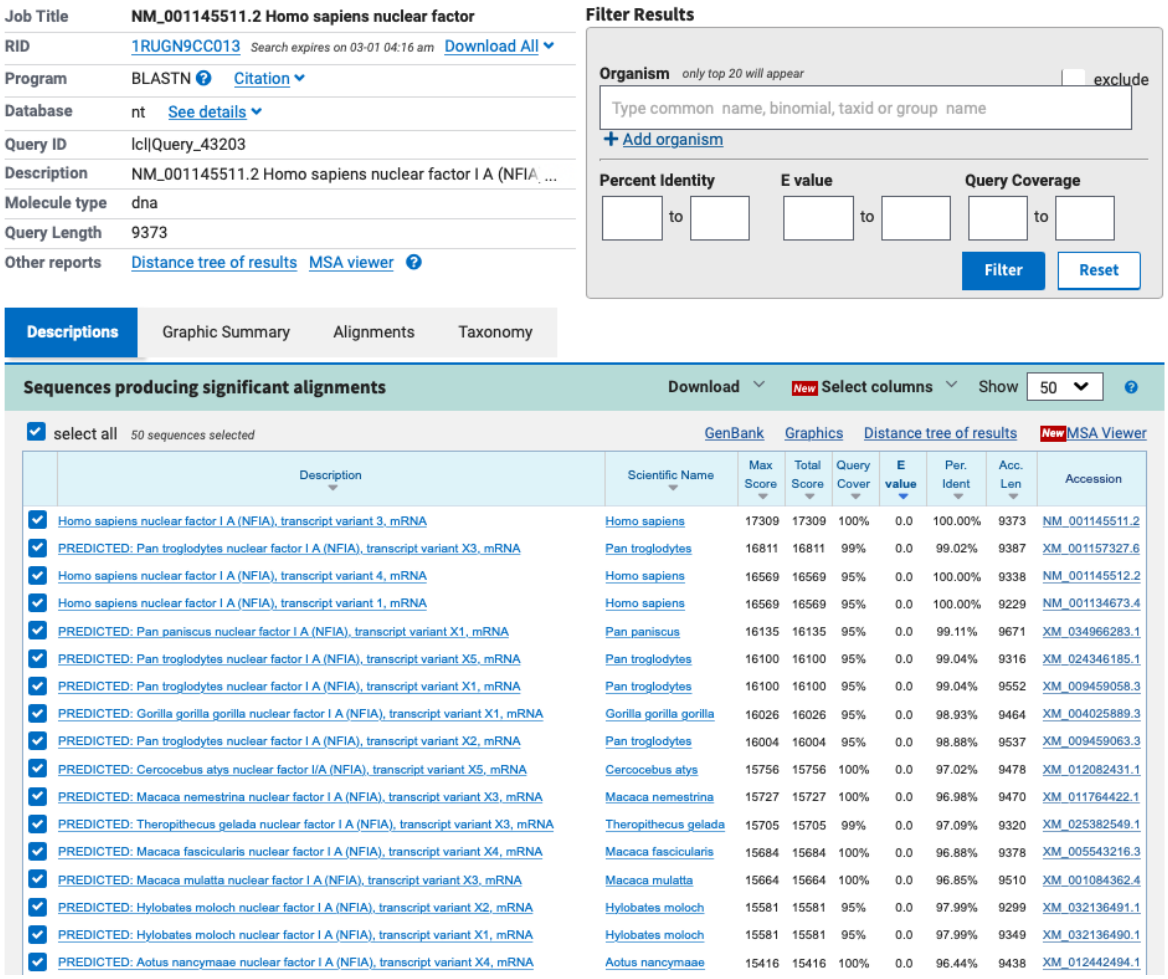


viii) BLAST queries on NCBI

To find this, we blast the transcript sequence on NCBI website and analyze the result.

5 organisms sharing similarity:

- Homo Sapiens (Humans)
- Pan troglodytes (Chimpanzee)
- Pan paniscus (Bonobo - Great Ape)
- Gorilla gorilla gorilla (Gorilla)
- Cercocebus atys (sooty mangabey - a type of monkey)
- Macaca nemestrina (Southern pig-tailed macaque)



Job Title	NM_001145511.2 Homo sapiens nuclear factor
RID	1RUGN9CC013 Search expires on 03-01 04:16 am Download All
Program	BLASTN Citation
Database	nt See details
Query ID	lcl|Query_43203
Description	NM_001145511.2 Homo sapiens nuclear factor I A (NFIA ...
Molecule type	dna
Query Length	9373
Other reports	Distance tree of results MSA viewer

Filter Results

Organism only top 20 will appear exclude

Type common name, binomial, taxid or group name

+ Add organism

Percent Identity to E value to Query Coverage to

Filter Reset

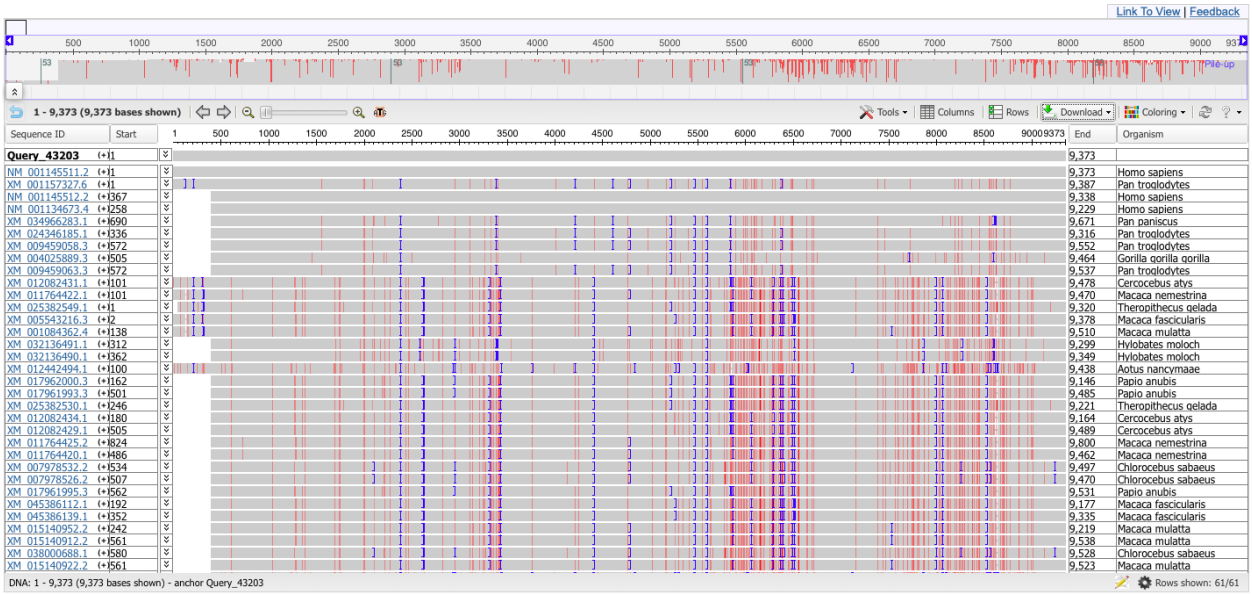
Descriptions Graphic Summary Alignments Taxonomy

Sequences producing significant alignments Download New Select columns Show 50

select all 50 sequences selected GenBank Graphics Distance tree of results New MSA Viewer

Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession
Homo sapiens nuclear factor I A (NFIA), transcript variant 3, mRNA	Homo sapiens	17309	17309	100%	0.0	100.00%	9373	NM_001145511.2
PREDICTED: Pan troglodytes nuclear factor I A (NFIA), transcript variant X3, mRNA	Pan troglodytes	16811	16811	99%	0.0	99.02%	9387	XM_001157327.6
Homo sapiens nuclear factor I A (NFIA), transcript variant 4, mRNA	Homo sapiens	16569	16569	95%	0.0	100.00%	9338	NM_001145512.2
Homo sapiens nuclear factor I A (NFIA), transcript variant 1, mRNA	Homo sapiens	16569	16569	95%	0.0	100.00%	9229	NM_001134673.4
PREDICTED: Pan paniscus nuclear factor I A (NFIA), transcript variant X1, mRNA	Pan paniscus	16135	16135	95%	0.0	99.11%	9671	XM_034966283.1
PREDICTED: Pan troglodytes nuclear factor I A (NFIA), transcript variant X5, mRNA	Pan troglodytes	16100	16100	95%	0.0	99.04%	9316	XM_024346185.1
PREDICTED: Pan troglodytes nuclear factor I A (NFIA), transcript variant X1, mRNA	Pan troglodytes	16100	16100	95%	0.0	99.04%	9552	XM_009459058.3
PREDICTED: Gorilla gorilla gorilla nuclear factor I A (NFIA), transcript variant X1, mRNA	Gorilla gorilla gorilla	16026	16026	95%	0.0	98.93%	9464	XM_004025889.3
PREDICTED: Pan troglodytes nuclear factor I A (NFIA), transcript variant X2, mRNA	Pan troglodytes	16004	16004	95%	0.0	98.88%	9537	XM_009459063.3
PREDICTED: Cercocebus atys nuclear factor I/A (NFIA), transcript variant X5, mRNA	Cercocebus atys	15756	15756	100%	0.0	97.02%	9478	XM_012082431.1
PREDICTED: Macaca nemestrina nuclear factor I A (NFIA), transcript variant X3, mRNA	Macaca nemestrina	15727	15727	100%	0.0	96.98%	9470	XM_011764422.1
PREDICTED: Theropithecus gelada nuclear factor I A (NFIA), transcript variant X3, mRNA	Theropithecus gelada	15705	15705	99%	0.0	97.09%	9320	XM_025382549.1
PREDICTED: Macaca fascicularis nuclear factor I A (NFIA), transcript variant X4, mRNA	Macaca fascicularis	15684	15684	100%	0.0	96.88%	9378	XM_005543216.3
PREDICTED: Macaca mulatta nuclear factor I A (NFIA), transcript variant X3, mRNA	Macaca mulatta	15664	15664	100%	0.0	96.85%	9510	XM_001084362.4
PREDICTED: Hylobates moloch nuclear factor I A (NFIA), transcript variant X2, mRNA	Hylobates moloch	15581	15581	95%	0.0	97.99%	9299	XM_032136491.1
PREDICTED: Hylobates moloch nuclear factor I A (NFIA), transcript variant X1, mRNA	Hylobates moloch	15581	15581	95%	0.0	97.99%	9349	XM_032136490.1
PREDICTED: Aotus nancymaae nuclear factor I A (NFIA), transcript variant X4, mRNA	Aotus nancymaae	15416	15416	100%	0.0	96.44%	9438	XM_012442494.1

The top 50 aligned sequences are downloaded in file named '50AlignedSequences.txt'.
MSA done on NCBI website. Downloaded in file named MSA.pdf. Screenshot:



viii) Phylogenetic Trees (NCBI)

The phylogenetic tree is made on NCBI website itself. Downloaded in file named ‘PhylTree.pdf’. Screenshot: