Practical 5

Question 1. Analyze the occurrence of similar proteins in the **"nr"** and **SWISS-PROT** databases for the sequence given below:

FASTA file for the protein sequence

```
>1336093|Genbank|Outer membrane integral membrane protein|HrcC
MVEKRELRCRLLGALLMLCATLPAGAQTPADWKEQSYAYSADRTPLSTVLQDFADGHSVD
LHLGNVEDTEVTAKIRAENASAFLDRLALEHHFQWFVYNNTLYVSPQDEQSSERLEISPD
AAPDIKQALSGIGLLDPRFGWGELPDDGVVLVTGPPQYLELVKRFSEQREKKEDRRKVMT
FPLRYASVADRTIHYRDQTVVIPGVATMLNELMNGKRAAPASASGIDSTPGGPDTNSMMQ
NTQTLLSRLSSRNKTSNRAGGRDNEIEDVSGRISADVRNNALLIRDDDKRHDEYSQLIAK
IDVPQNLVEIDAVILDIDRTALNRLEANWQATLGGVTGGSSLMSGSGTLFVSDFKRFFAD
IQALEGEGTASIVANPSVLTLENQPAVIDFSQTAYITATGERVADIQPVTAGTSLQVTPR
AVGNEGHSSIQLMIDIEDGHVQTNGDGQATGVKRGTVSTQALISENRALVLGGFHVEESA
DRDRRIPLLGDIPWLGQLFSSKRHEISQRQRLFILTPRLIGDQTDPTRYVTADNRQQLSD
AMGRVERRHSSVNQHDVVENALRDLAEGQSPAGFQPQTSGTRLSEVCRSTPALLFESTRG
QWYSSSTNGVQLSVGVVRNTSSKPLRFDEANCASKRTLAVAVWPHSALAPGESAEVYLAM
DPSRVLHASRESLLNR
```

Solution. I am using **BLASTP** for this task. **BLASTP** program searches protein databases using a protein query. It is basically a **protein-protein BLAST**.
First, I have chosen the standard database as **Non-redundant protein sequences (nr)**. Below is the image for the corresponding query search.

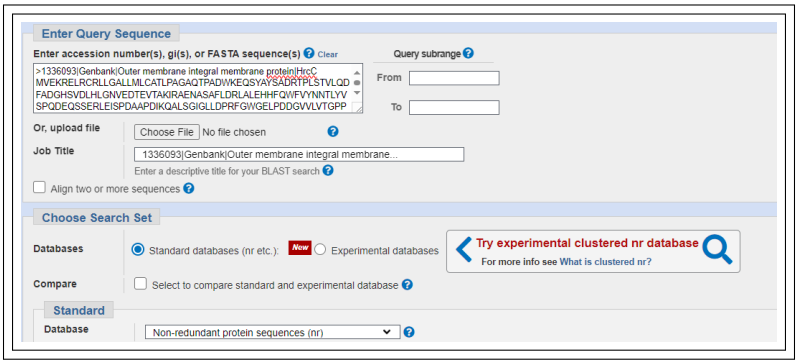


Figure 1. Search parameters for blastp

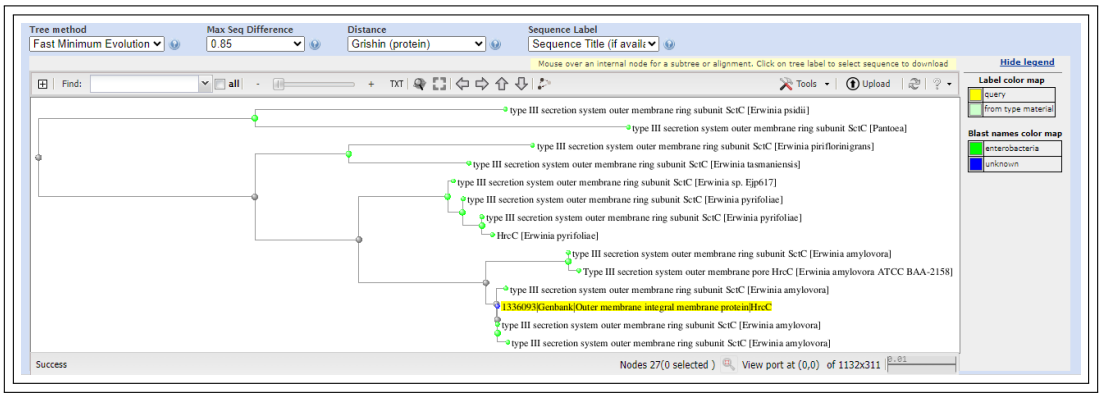
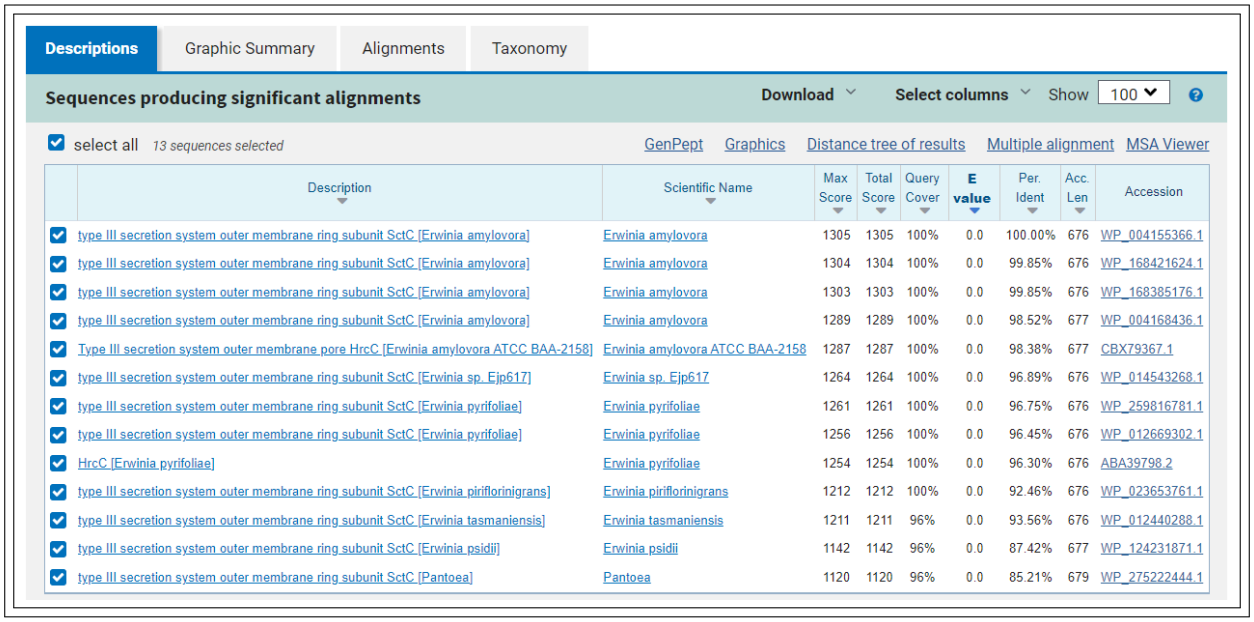


Figure 2. Blast tree view using pairwise alignment



Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession
type III secretion system outer membrane ring subunit SctC [Erwinia amylovora]	Erwinia amylovora	1305	1305	100%	0.0	100.00%	676	WP_004155366.1
type III secretion system outer membrane ring subunit SctC [Erwinia amylovora]	Erwinia amylovora	1304	1304	100%	0.0	99.85%	676	WP_168421624.1
type III secretion system outer membrane ring subunit SctC [Erwinia amylovora]	Erwinia amylovora	1303	1303	100%	0.0	99.85%	676	WP_168385176.1
type III secretion system outer membrane ring subunit SctC [Erwinia amylovora]	Erwinia amylovora	1289	1289	100%	0.0	98.52%	677	WP_004168436.1
Type III secretion system outer membrane pore HrcC [Erwinia amylovora ATCC BAA-2158]	Erwinia amylovora ATCC BAA-2158	1287	1287	100%	0.0	98.38%	677	CBX79367.1
type III secretion system outer membrane ring subunit SctC [Erwinia sp. Ejp617]	Erwinia sp. Ejp617	1264	1264	100%	0.0	96.89%	676	WP_014543268.1
type III secretion system outer membrane ring subunit SctC [Erwinia pyrifoliae]	Erwinia pyrifoliae	1261	1261	100%	0.0	96.75%	676	WP_259816781.1
type III secretion system outer membrane ring subunit SctC [Erwinia pyrifoliae]	Erwinia pyrifoliae	1256	1256	100%	0.0	96.45%	676	WP_012669302.1
HrcC [Erwinia pyrifoliae]	Erwinia pyrifoliae	1254	1254	100%	0.0	96.30%	676	ABA39798.2
type III secretion system outer membrane ring subunit SctC [Erwinia piriflorinigrans]	Erwinia piriflorinigrans	1212	1212	100%	0.0	92.46%	676	WP_023653761.1
type III secretion system outer membrane ring subunit SctC [Erwinia tasmaniensis]	Erwinia tasmaniensis	1211	1211	96%	0.0	93.56%	676	WP_012440288.1
type III secretion system outer membrane ring subunit SctC [Erwinia psidii]	Erwinia psidii	1142	1142	96%	0.0	87.42%	677	WP_124231871.1
type III secretion system outer membrane ring subunit SctC [Pantoea]	Pantoea	1120	1120	96%	0.0	85.21%	679	WP_275222444.1

FIGURE 3. Results from blastp

The above results for the **nr** database include:

- Description statistics showing the protein sequences similar to the given protein sequence. It includes their **Total score**, **Query coverage**, **E value**, **Percentage identity**, and **Accession length**.
- **Blast Tree View**, produced using **BLAST pairwise alignment**. In this representation, an explicit multiple alignment is not computed between different database sequences. Rather, an **implicit alignment is constructed** based on the alignment of database sequences to the query.

Second, I have chosen the standard database as **SWISS-PROT**. Below is the image for the corresponding query search.

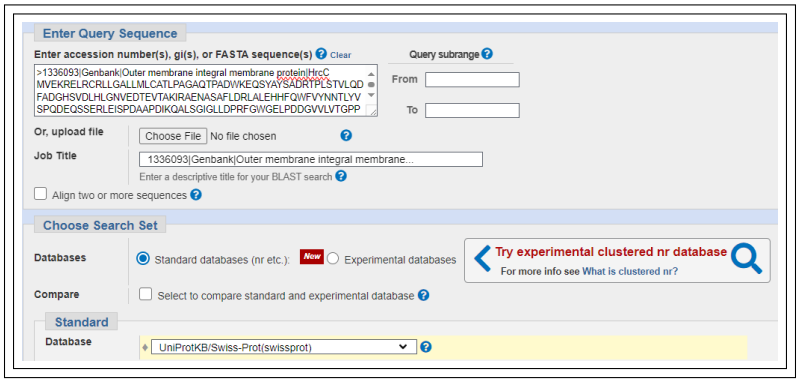


FIGURE 4. Search parameters for blastp

The below results for the **SWISS-PROT** database include:

- Description statistics showing the protein sequences similar to the given protein sequence. It includes their **Total score**, **Query coverage**, **E value**, **Percentage identity**, and **Accession length**.
- **Blast Tree View**, produced using **BLAST pairwise alignment**. In this representation, an explicit multiple alignment is not computed between different database sequences. Rather, an **implicit alignment is constructed** based on the alignment of database sequences to the query.

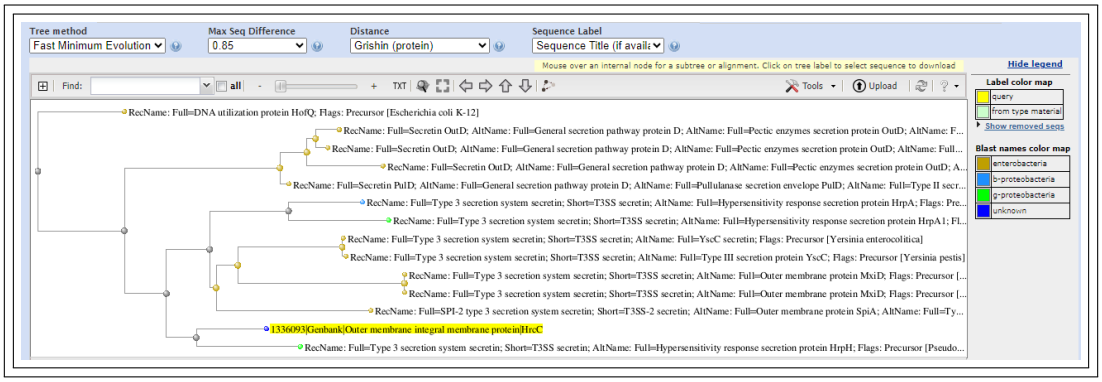
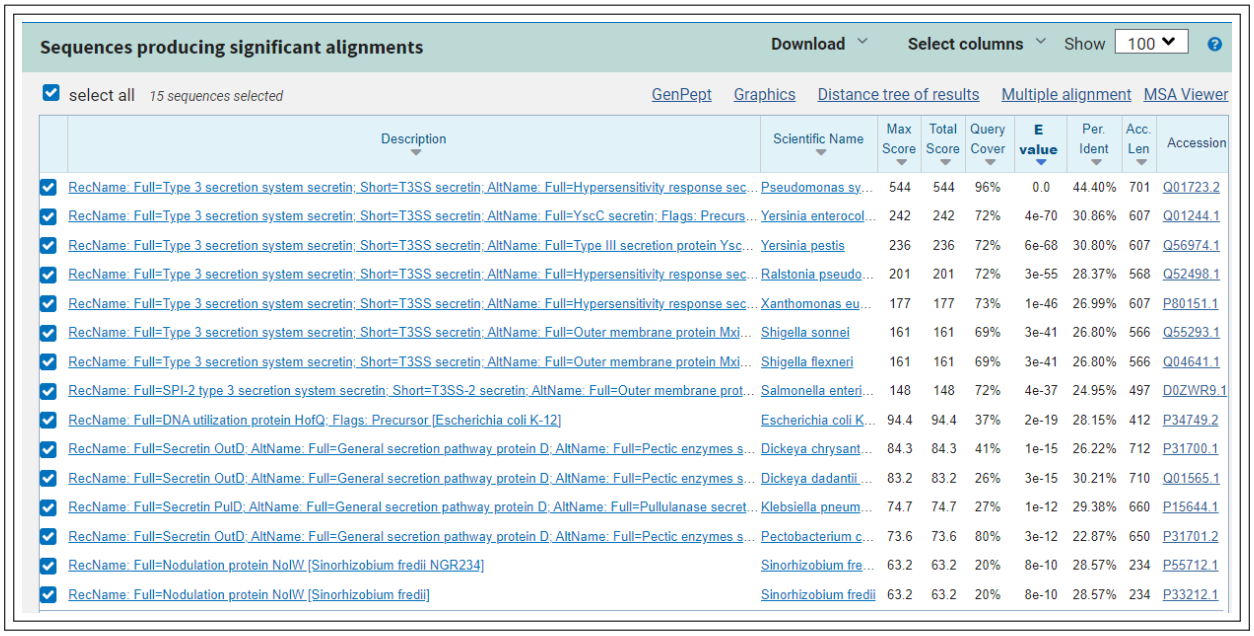


FIGURE 5. Blast tree view using pairwise alignment



Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession
RecName: Full=Type 3 secretion system secretin; Short=T3SS secretin; AltName: Full=Hypersensitivity response sec...	Pseudomonas sy...	544	544	96%	0.0	44.40%	701	Q01723.2
RecName: Full=Type 3 secretion system secretin; Short=T3SS secretin; AltName: Full=YscC secretin; Flags: Precurs...	Yersinia enterocol...	242	242	72%	4e-70	30.86%	607	Q01244.1
RecName: Full=Type 3 secretion system secretin; Short=T3SS secretin; AltName: Full=Type III secretion protein Ysc...	Yersinia pestis	236	236	72%	6e-68	30.80%	607	Q56974.1
RecName: Full=Type 3 secretion system secretin; Short=T3SS secretin; AltName: Full=Hypersensitivity response sec...	Ralstonia pseudo...	201	201	72%	3e-55	28.37%	568	Q52498.1
RecName: Full=Type 3 secretion system secretin; Short=T3SS secretin; AltName: Full=Hypersensitivity response sec...	Xanthomonas eu...	177	177	73%	1e-46	26.99%	607	P80151.1
RecName: Full=Type 3 secretion system secretin; Short=T3SS secretin; AltName: Full=Outer membrane protein Mxi...	Shigella sonnei	161	161	69%	3e-41	26.80%	566	Q55293.1
RecName: Full=Type 3 secretion system secretin; Short=T3SS secretin; AltName: Full=Outer membrane protein Mxi...	Shigella flexneri	161	161	69%	3e-41	26.80%	566	Q04641.1
RecName: Full=SPI-2 type 3 secretion system secretin; Short=T3SS-2 secretin; AltName: Full=Outer membrane prot...	Salmonella enteri...	148	148	72%	4e-37	24.95%	497	D0ZWR9.1
RecName: Full=DNA utilization protein HofQ; Flags: Precursor [Escherichia coli K-12]	Escherichia coli K...	94.4	94.4	37%	2e-19	28.15%	412	P34749.2
RecName: Full=Secretin OutD; AltName: Full=General secretion pathway protein D; AltName: Full=Pectic enzymes s...	Dickeya chrysant...	84.3	84.3	41%	1e-15	26.22%	712	P31700.1
RecName: Full=Secretin OutD; AltName: Full=General secretion pathway protein D; AltName: Full=Pectic enzymes s...	Dickeya dadantii ...	83.2	83.2	26%	3e-15	30.21%	710	Q01565.1
RecName: Full=Secretin PulD; AltName: Full=General secretion pathway protein D; AltName: Full=Pullulanase secret...	Klebsiella pneum...	74.7	74.7	27%	1e-12	29.38%	660	P15644.1
RecName: Full=Secretin OutD; AltName: Full=General secretion pathway protein D; AltName: Full=Pectic enzymes s...	Pectobacterium c...	73.6	73.6	80%	3e-12	22.87%	650	P31701.2
RecName: Full=Nodulation protein NolW [Sinorhizobium fredii NGR234]	Sinorhizobium fre...	63.2	63.2	20%	8e-10	28.57%	234	P55712.1
RecName: Full=Nodulation protein NolW [Sinorhizobium fredii]	Sinorhizobium fredii	63.2	63.2	20%	8e-10	28.57%	234	P33212.1

FIGURE 6. Results from blastp

Question 2. List the algorithm parameters used for the search (Q1).

Solution. Below is a list of the algorithm parameters used for the search in Q1.

Algorithm parameters

Max target sequences: *Select the maximum number of aligned sequences to display*: **Set to 100**
Short queries: *Automatically adjust parameters for short input sequences*: **Set to Yes**
Expect threshold: *Expected number of chance matches in a random model*: **Set to 0.05**
Word Size: *The length of seed that initiates an alignment*: **Set to 5**
Max matches in a query range: *Limit number of matches to a query range*: **Set to 10**
Matrix: *Assigns a score for aligning pairs of residues and determines the overall alignment score*: **Set to BLOSUM62**
Gap costs: *Cost to create and extend a gap in an alignment*: **Set to Existence: 11; Extension: 1**
Compositional adjustments: *Compensation method*: **Set to Conditional score matrix**
Filters: *Mask regions of low compositional complexity that may cause spurious or misleading results*: **Set to Yes**

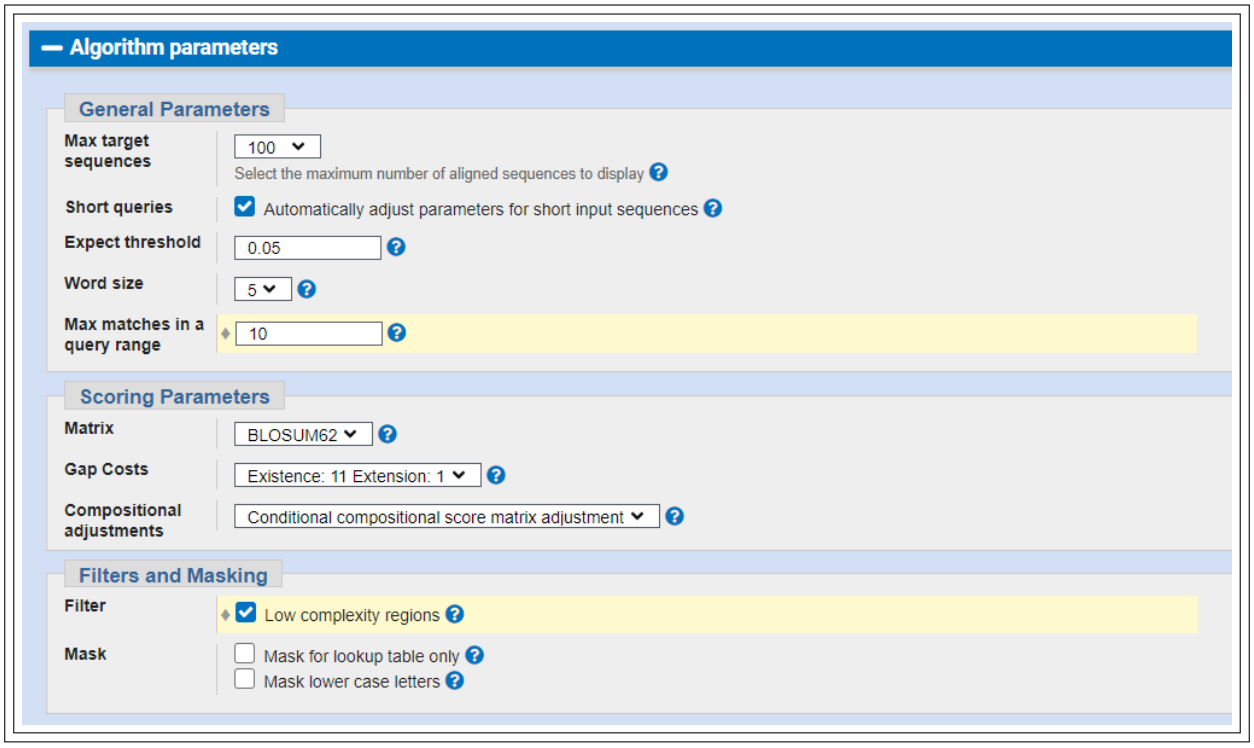


FIGURE 7. Algorithm parameters

Question 3. What is the sequence identity of the query sequence (given in Q1) with AAK81929.1?

Solution. In the same **BLASTP**, by selecting **Align 2 or more sequences**, I have added the accession number given in the question. This is now compared against the query sequence given in Q1. The query search is given below:

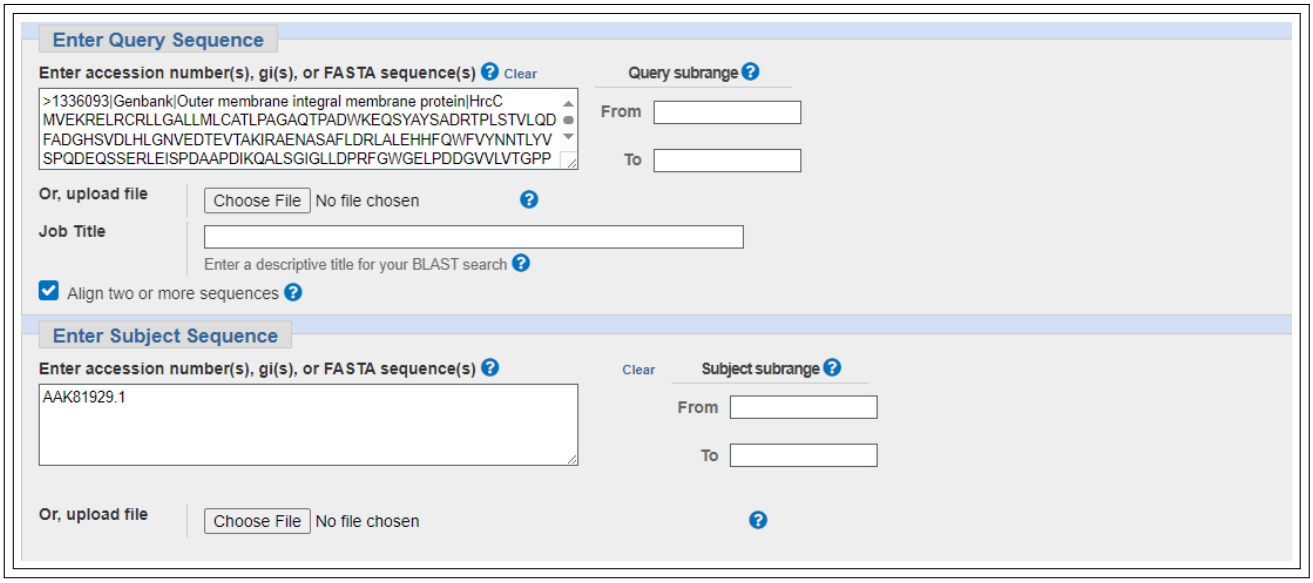


FIGURE 8. blastp to align 2 sequences

The above is the query search on **BLASTP**. The results consist of two components.

- One is the descriptive statistics, which contain the total score, E value, percentage identity, and accession length.
- The other is pairwise sequence alignment. It compares the two sequences.

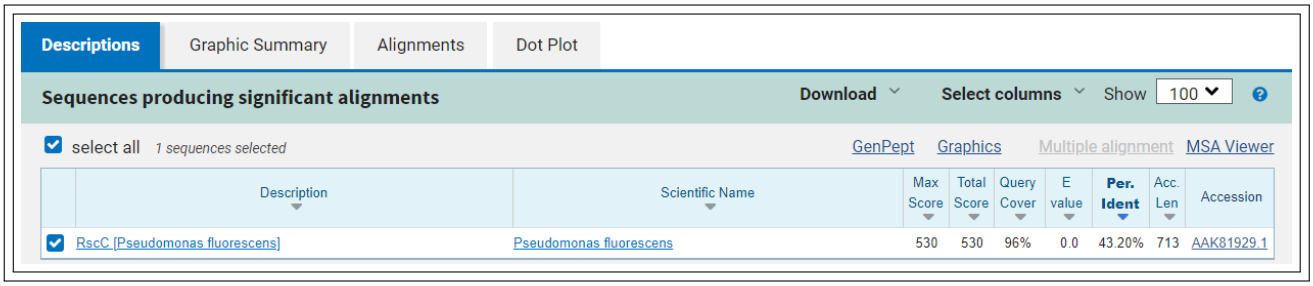


FIGURE 9. Descriptive statistics

FIGURE 10. Pairwise Alignment

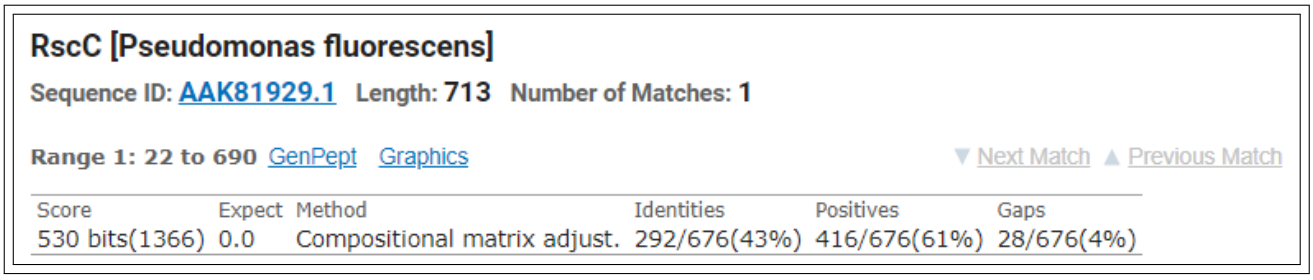


FIGURE 11. Alignment Window in blastp

From the above descriptive statistics and pairwise alignment, it can be seen that the **sequence identity** is about **43.20%**.
From the above alignment window for both sequences, it can be seen that the percentage of sequence identities is **43.20%**, and the percentage of sequence positives is **61.54%**.

Question 4. How far are hemoglobin (beta) sequences in humans and chicken similar?

Solution. In the same **BLASTP**, by selecting **Align 2 or more sequences**, I have added the **hemoglobin (beta) sequences** of both **human** and **chicken**. The query search is given below:

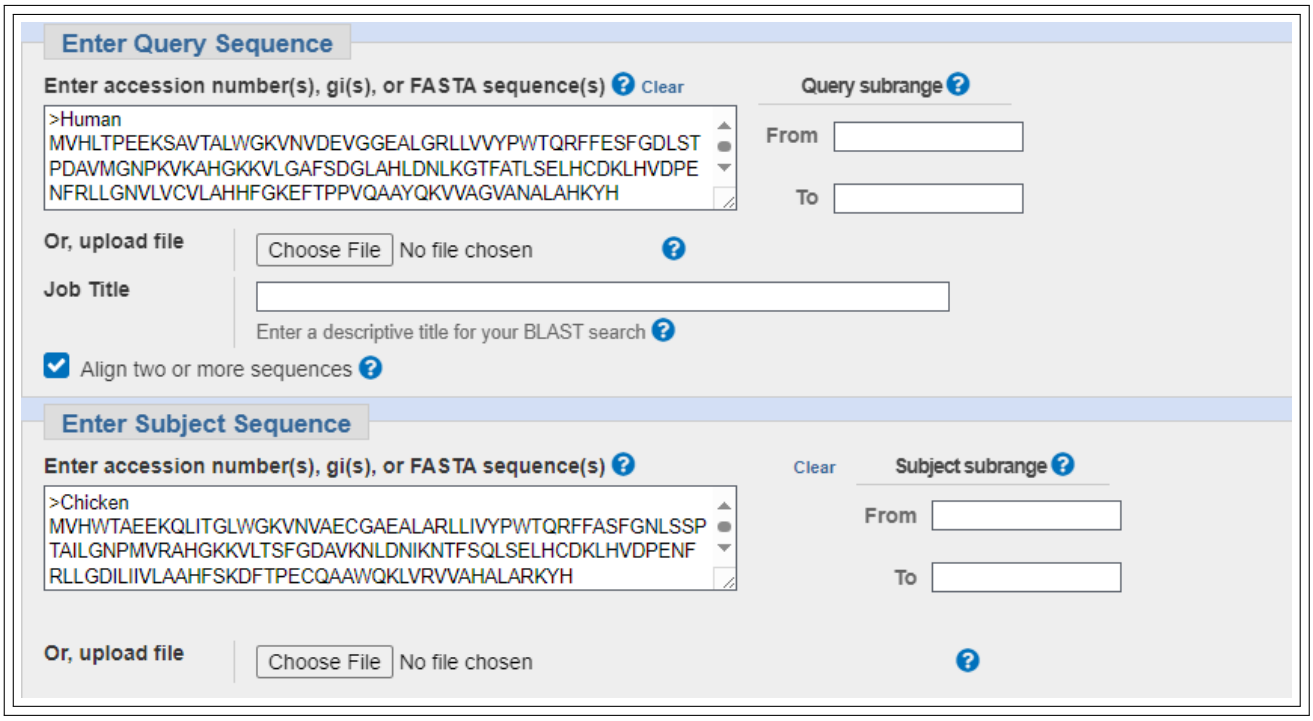


FIGURE 12. blastp to align 2 sequences

The above is the query search on **BLASTP**. The results are given below:

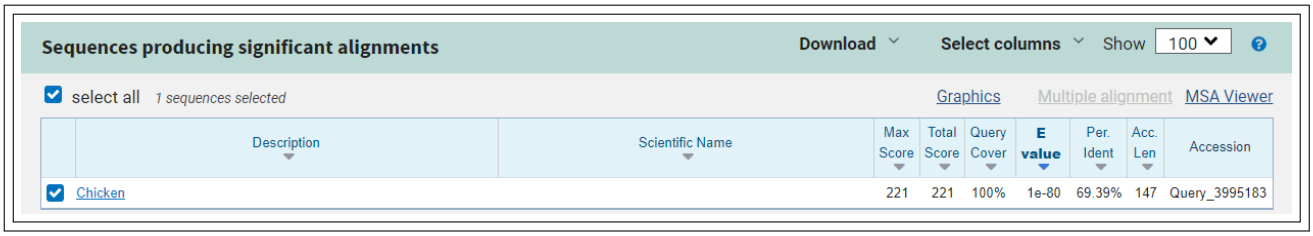


FIGURE 13. Descriptive statistics

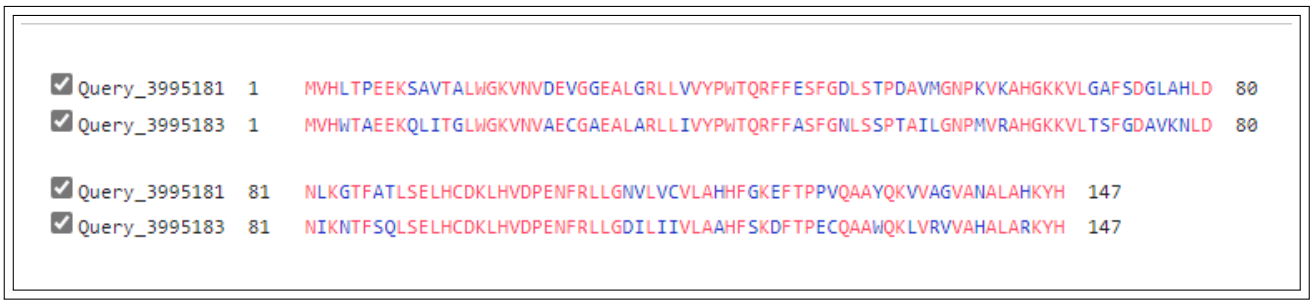
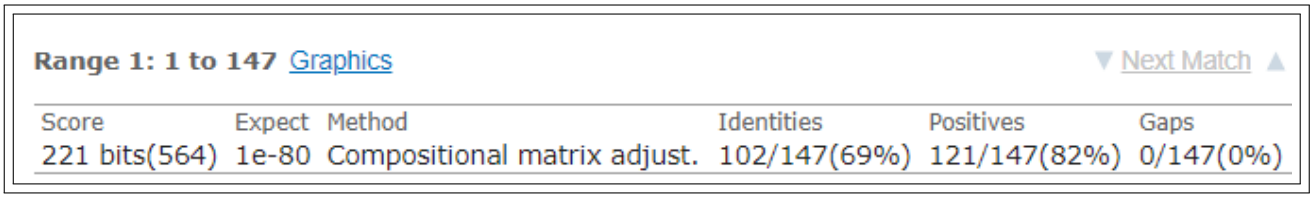


FIGURE 14. Pairwise Alignment

From the above descriptive statistics and pairwise alignment, it can be seen that the **sequence identity** is about **69.39%**.
From the below alignment window for both sequences, it can be seen that the percentage of sequence identities is **69.39%**, and the percentage of sequence positives is **82.31%**.



Range 1: 1 to 147 Graphics ▼ Next Match ▲

Score	Expect	Method	Identities	Positives	Gaps
221 bits(564)	1e-80	Compositional matrix adjust.	102/147(69%)	121/147(82%)	0/147(0%)

FIGURE 15. Alignment Window in blastp

Question 5. Write a program to list all the **matching pentapeptides** (which occur in both the sequences) and their **frequency of occurrence** in given sequences.

Solution. Below is the code to find the list of all matching pentapeptides and their frequencies in both sequences.

```
# Finding the matching pentapeptides
def pentapeptides(seq_1, seq_2):
    peptides_list = []
    # Iterating through seq_1 to find matching peptides between seq_1 and
   seq_2
    for i in range(len(seq_1)-5+1):
        peptide = seq_1[i:i+5]
        if peptide in seq_2:
            peptides_list.append(peptide)

    # Calculating the frequency of each pentapeptide in each sequence
    seq_1_freq = []
    seq_2_freq = []
    for x in set(peptides_list):
        seq_1_freq.append([x, seq_1.count(x)])
        seq_2_freq.append([x, seq_2.count(x)])

    return peptides_list, seq_1_freq, seq_2_freq

seq_human =
"""MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPKVKAHGKKVLGAFS
DGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVANALAHKYH"""

seq_chick =
"""MVHWTAEEKQLITGLWGKVNVAECGAEALARLLIVYPWTQRFFASFGNLSSPTAILGNPMVRAHGKKVLTSFG
DAVKNLDNIKNTFSQLSELHCDKLHVDPENFRLLGDILIIVLAAHFSKDFTPECQAAWQKLVRVVAHALARKYH"""


peptide_list, seq1_freq, seq2_freq = pentapeptides(seq_human, seq_chick)

# Converting all outputs to dataframe for easier visualization
seq1_freq = pd.DataFrame(seq1_freq, columns=["Pentapeptides", "Frequency"])
seq2_freq = pd.DataFrame(seq2_freq, columns=["Pentapeptides", "Frequency"])
peptide_list = pd.DataFrame(peptide_list, columns=["Pentapeptides"])

# Printing the output dataframes
print("Sequence_Human")
seq1_freq
print("Sequence_Chicken")
seq2_freq
print("Pentapeptide_List")
peptide_list
```

	Seq_Human			Seq_Chicken			Pentapeptide_List
	Pentapeptides	Frequency		Pentapeptides	Frequency		Pentapeptides
0	NFRLL	1	0	NFRLL	1	0	LWGKV
1	WTQRF	1	1	WTQRF	1	1	WGKVN
2	SELHC	1	2	SELHC	1	2	GKVNV
3	YPWTQ	1	3	YPWTQ	1	3	VYPWT
4	LHCDK	1	4	LHCDK	1	4	YPWTQ
5	KLHVD	1	5	KLHVD	1	5	PWTQR
6	ENFRL	1	6	ENFRL	1	6	WTQRF
7	LWGKV	1	7	LWGKV	1	7	TQRFF
8	AHGKK	1	8	AHGKK	1	8	AHGKK
9	WGKVN	1	9	WGKVN	1	9	HGKKV
10	TQRFF	1	10	TQRFF	1	10	GKKVL
11	DPENF	1	11	DPENF	1	11	LSELH
12	PWTQR	1	12	PWTQR	1	12	SELHC
13	HVDPE	1	13	HVDPE	1	13	ELHCD
14	FRLLG	1	14	FRLLG	1	14	LHCDK
15	GKKVL	1	15	GKKVL	1	15	HCDKL
16	LHVDP	1	16	LHVDP	1	16	CDKLH
17	DKLHV	1	17	DKLHV	1	17	DKLHV
18	GKVNV	1	18	GKVNV	1	18	KLHVD
19	LSELH	1	19	LSELH	1	19	LHVDP
20	VDPEN	1	20	VDPEN	1	20	HVDPE
21	CDKLH	1	21	CDKLH	1	21	VDPEN
22	HCDKL	1	22	HCDKL	1	22	DPENF
23	HGKKV	1	23	HGKKV	1	23	PENFR
24	PENFR	1	24	PENFR	1	24	ENFRL
25	VYPWT	1	25	VYPWT	1	25	NFRLL
26	ELHCD	1	26	ELHCD	1	26	FRLLG

FIGURE 16. Matching pentapeptides between human and chicken sequences

Question 6. Write a program to compute **sequence identity**, **similarity**, **query coverage**, and **gap percentage** from the alignment of human and chicken hemoglobin sequences (refer Q4).

Solution. The code to generate the blosum62 matrix for the calculations ahead is given below, along with an image of the blosum62 matrix:

```
# Blosum dictionary and blosum matrix
blosum_dict = {"A":0,"R":1,"N":2,"D":3,"C":4,"Q":5,"E":6,"G":7,"H":8,"I":9,
        "L":10,"K":11,"M":12,"F":13,"P":14,"S":15,"T":16,"W":17,"Y":18,"V"
    :19}

# Creating blosum matrix by reading from a text file saved from internet
blosum_matrix = []
with open("blosum62.txt", 'r') as f:
    line = f.read()
    line_list = line.split("\n")

    for i in range(len(line_list)):
        line_i = line_list[i].split(" ")
        line_i = [int(value) for value in line_i if value != ""]
        blosum_matrix.append(line_i)
```

LISTING 1. Generate blosum matrix

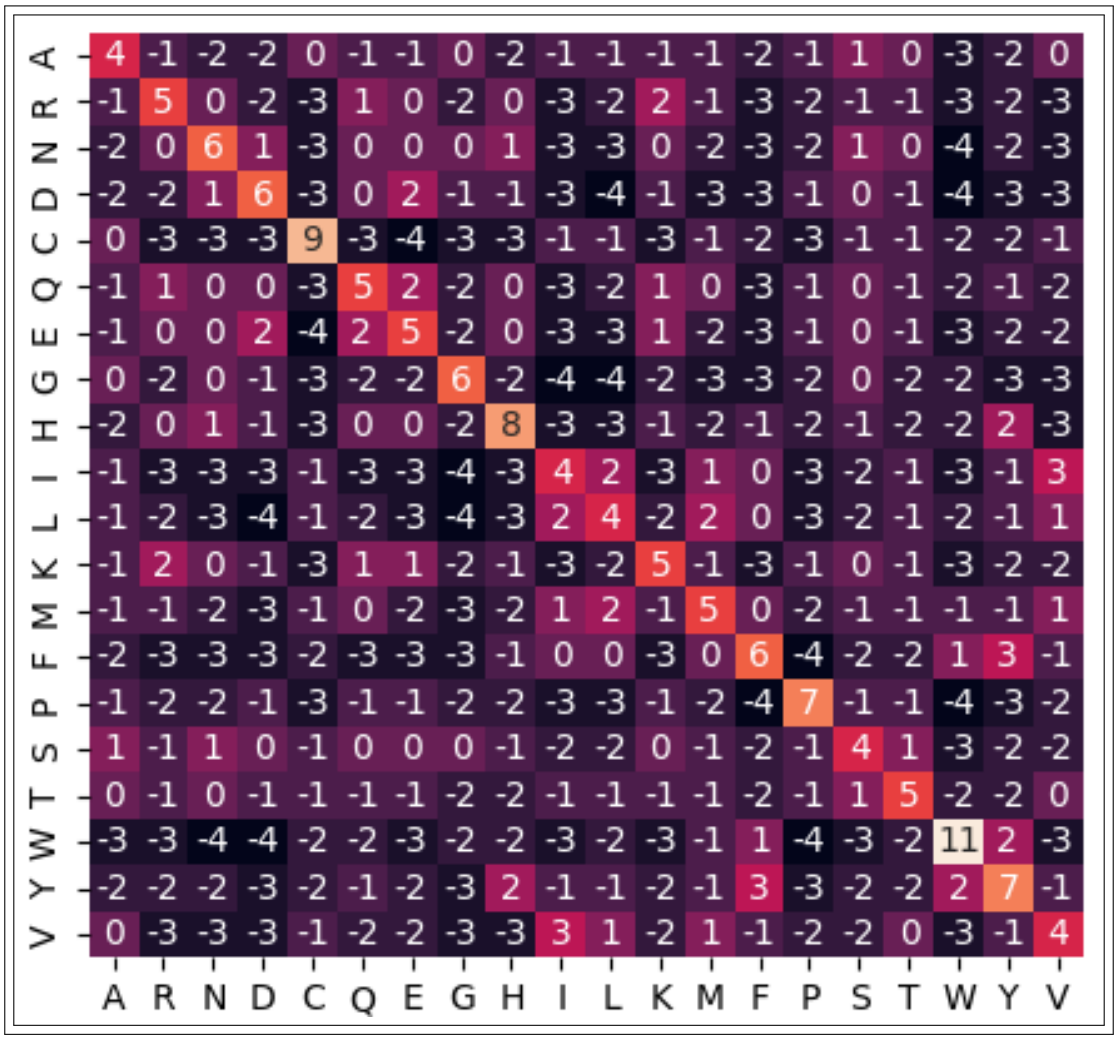


FIGURE 17. BLOSUM62 matrix

The code for the alignment of protein sequences, like **blastp**, is given below:

```
# Calculating the matrix for global alignment in this new scenario
def variation_smith_waterman_matrix(seq_1, seq_2, blosum_matrix, blosum_dict,
                                    create_gap_score, extend_gap_score):
    origin_matrix = [[0 for i in range(len(seq_2)+1)]
                     for j in range(len(seq_1)+1)]
    matrix = [[0 for i in range(len(seq_2)+1)] for j in range(len(seq_1)+1)]
    for i in range(1,len(seq_2)+1):
        origin_matrix[0][i] = 1
        if origin_matrix[0][i-1] == 0:
            matrix[0][i] = matrix[0][i-1] + create_gap_score
        else:
            matrix[0][i] = matrix[0][i-1] + extend_gap_score
    for i in range(1,len(seq_1)+1):
        origin_matrix[i][0] = 1
        if origin_matrix[i-1][0] == 0:
            matrix[i][0] = matrix[i-1][0] + create_gap_score
        else:
            matrix[i][0] = matrix[i-1][0] + extend_gap_score
    for i in range(1,len(seq_1)+1):
        for j in range(1, len(seq_2)+1):
            val_1= matrix[i-1][j] + create_gap_score if origin_matrix[i-1][j]
                   == 0 else matrix[i-1][j] + extend_gap_score
            val_2= matrix[i][j-1] + create_gap_score if origin_matrix[i][j-1]
                   == 0 else matrix[i][j-1] + extend_gap_score
            val_3= matrix[i-1][j-1] + blosum_matrix[blosum_dict[seq_1[i-1]]]
                                                   [blosum_dict[seq_2[j-1]]]
            matrix[i][j] = max(val_1, val_2, val_3, 0)

            if matrix[i][j] == val_3:
                origin_matrix[i][j] = 0

            elif(matrix[i][j] == val_1 or matrix[i][j] == val_2):
                origin_matrix[i][j] = 1

            else:
                origin_matrix[i][j] = 0

    return matrix, origin_matrix

# Backtracking to get the sequence alignment with gaps (if any)
def variation_backtrack(seq_1, seq_2, matrix, origin_matrix, blosum_matrix,
    blosum_dict, create_gap_score, extend_gap_score):
    seq_align_1 = ""
    seq_align_2 = ""
    i = 0
    j = 0
    maximum = 0

    for r in range(len(seq_1)+1):
        for c in range(len(seq_2)+1):
            if matrix[r][c] > maximum:
                maximum = matrix[r][c]
                i = r
                j = c
```

```
54
55      while(matrix[i][j] != 0):
56
57          if(matrix[i-1][j] == matrix[i][j] - (1 - origin_matrix[i-1][j])*
                create_gap_score - origin_matrix[i-1][j]*extend_gap_score):
58              seq_align_1 += seq_1[i-1]
59              seq_align_2 += "-"
60              i -= 1
61
62          elif(matrix[i][j-1] == matrix[i][j] - (1 - origin_matrix[i][j-1])*
                create_gap_score - origin_matrix[i][j-1]*extend_gap_score):
63              seq_align_2 += seq_2[j-1]
64              seq_align_1 += "-"
65              j -= 1
66
67          else:
68              seq_align_1 += seq_1[i-1]
69              seq_align_2 += seq_2[j-1]
70              i -= 1
71              j -= 1
72
73      seq_align_1 = seq_align_1[::-1]
74      seq_align_2 = seq_align_2[::-1]
75
76      return seq_align_1, seq_align_2
77
78  seq_human = """
        MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPKVKAHGKKVLGAFS
79  DGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVANALAHKYH"""
80  seq_chick = """
        MVHWTAEEKQLITGLWGKVNVAECGAEALARLLIVYPWTQRFFASFGNLSSPTAILGNPMVRAHGKKVLTSFG
81  DAVKNLDNIKNTFSQLSELHCDKLHVDPENFRLLGDILIIVLAAHFSKDFTPECQAAWQKLVRVVAHALARKYH"""
82
83  create_gap_score = -11
84  extend_gap_score = -1
85
86  matrix, origin_matrix = variation_smith_waterman_matrix(seq_human, seq_chick,
87            blosum_matrix, blosum_dict, create_gap_score, extend_gap_score)
88  seq1_align, seq2_align = variation_backtrack(seq_human, seq_chick, matrix,
89            origin_matrix, blosum_matrix, blosum_dict, create_gap_score,
              extend_gap_score)
90
91  print(seq1_align)
92  print(seq2_align)
```

LISTING 2. Code for sequence alignment

The algorithm used by blastp is **Smith-Waterman** local alignment algorithm, with an existence gap penalty, an extension gap penalty, and a **blosum62** matrix for sequence matching. After computing the matrix by dynamic programming, I have used a backtracking algorithm to obtain the aligned sequences with gaps (if any).

Now, once we have the alignment of two sequences, we need to compute the sequence identity, sequence similarity, query coverage, and gap percentage. In order to do so, first I have a trim function that trims the additional gaps towards both ends of the aligned sequences (if any). After this, I compute each of the above metrics by iterating both sequences through a single **for loop**. The code to achieve the same is given below:

```python
# To trim ends containing -'s because they are not considered in calculations
def trim(seq_1, seq_2):
    count_start_1 = len(seq_1) - len(seq_1.lstrip('-'))
    count_start_2 = len(seq_2) - len(seq_2.lstrip('-'))
    count_end_1 = len(seq_1) - len(seq_1.rstrip('-'))
    count_end_2 = len(seq_2) - len(seq_2.rstrip('-'))
    count_start = max(count_start_1, count_start_2)
    count_end = max(count_end_1, count_end_2)
    seq_1 = seq_1[count_start:len(seq_1) - count_end]
    seq_2 = seq_2[count_start:len(seq_2) - count_end]
    return seq_1, seq_2

# Calculate the desired metrics from the question
def calculate(seq_1, seq_2):
    seq_1, seq_2 = trim(seq_1, seq_2)
    seq_identity = 0
    seq_similarity = 0
    gap_percent = 0
    query_coverage = 0 # seq_human is the query
    k_human = 0
    for i in range(len(seq_1)):
        # Sequene Identity condition
        if seq_1[i] == seq_2[i]:
            seq_identity += 1
        # Sequence similarity condition
        if (seq_1[i] != "-" and seq_2[i] != "-") and
        blosum_matrix[blosum_dict[seq_1[i]]][blosum_dict[seq_2[i]]] > 0:
            seq_similarity += 1
        # Query coverage condition
        if seq_1[i] != "-" and seq_human[k_human] == seq_1[i]:
            query_coverage += 1
            k_human += 1
        # Gap percent conditions
        if seq_1[i] == "-":
            gap_percent += 1
        if seq_2[i] == "-":
            gap_percent += 1

    seq_identity /= len(seq_1)
    seq_identity *= 100
    seq_similarity /= len(seq_1)
    seq_similarity *= 100
    query_coverage /= len(seq_human)
    query_coverage *= 100
    gap_percent /= len(seq_1)
    gap_percent *= 100

    return seq_identity, seq_similarity, query_coverage, gap_percent

seq_id, seq_sim, query_cover, gap_per = calculate(seq1_align, seq2_align)
print("Sequence Identity:", seq_id)
print("Sequence Similarity:", seq_sim)
print("Query Coverage:", query_cover)
print("Gap percentage:", gap_per)
```

LISTING 3. Code to compute the desired metrics

The output of the above code gives the desired metrics. The two sequences chosen are **human and chicken hemoglobin sequences**. The output is given below:

Metrics after sequence alignment

```
# Aligned_Sequence_Human
MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPKVKAHGKKVLGA
FSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVANAL
AHKYH
# Aligned_Sequence_Chicken
MVHWTAEEKQLITGLWGKVNVAECGAEALARLLIVYPWTQRFFASFGNLSSPTAILGNPMVRAHGKKVLTS
FGDAVKNLDNIKNTFSQLSELHCDKLHVDPENFRLLGDILIIVLAAHFSKDFTPECQAAWQKLVRVVAHAL
ARKYH

Sequence Identity: 69.38775510204081%
Sequence Similarity: 82.31292517006803%
Query Coverage: 100.0%
Gap percentage: 0.0%
```

Question 7. Obtain the multiple sequence alignment for TIM barrel proteins from different organisms (select 20 proteins, for example). Compare the results obtained with **Clustal Omega**, **MAFFT**, and **MUSCLE**. List 5 residue positions which are aligned differently in these three methods.

Solution. On UniProtKB, I searched for **TIM barrel proteins**. Then proteins from different organisms are chosen and their **FASTA files** are downloaded. These FASTA file formats for the different proteins (**20 in number**) are fed as input sequences to **Clustal Omega**. The output format for the multiple sequence alignment is chosen as **ClustalW with character counts**. The input page on the same is given below:

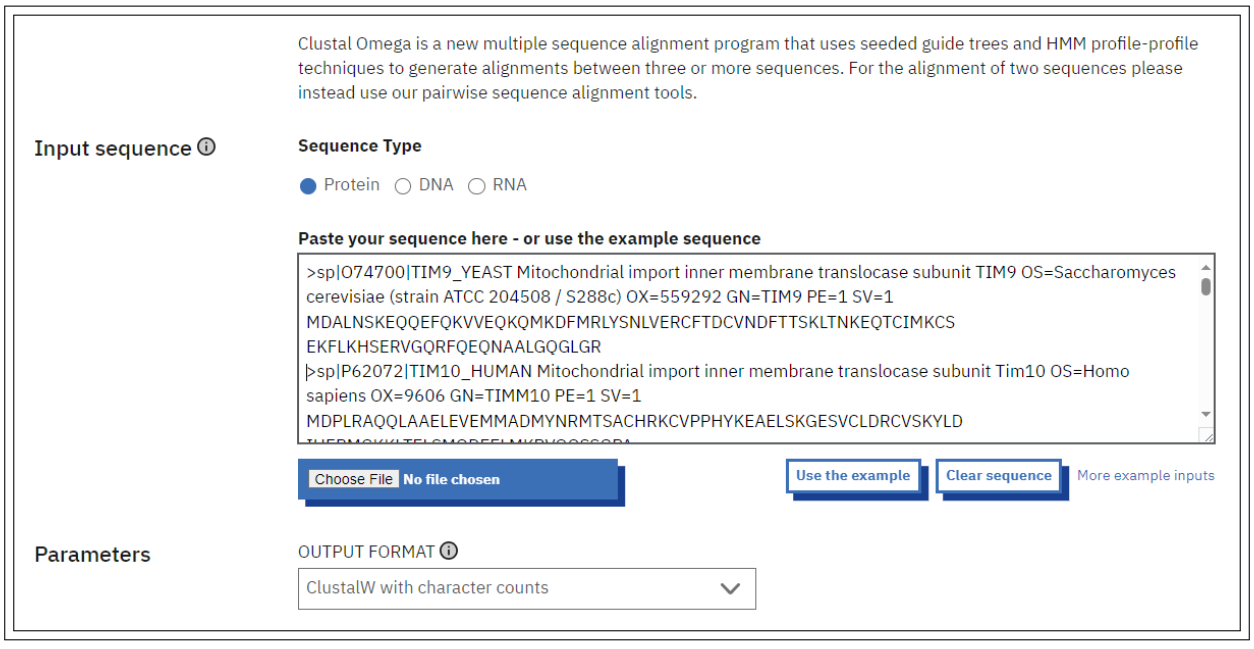


FIGURE 18. 20 input sequences for multiple sequence alignment

The output contains the multiple sequence alignment for all 20 sequences. The color scheme chosen is **clustal2**. The alignment has been zoomed in to a specific segment of the actual complete alignment for easy view. The complete multiple sequence alignment is also given below.

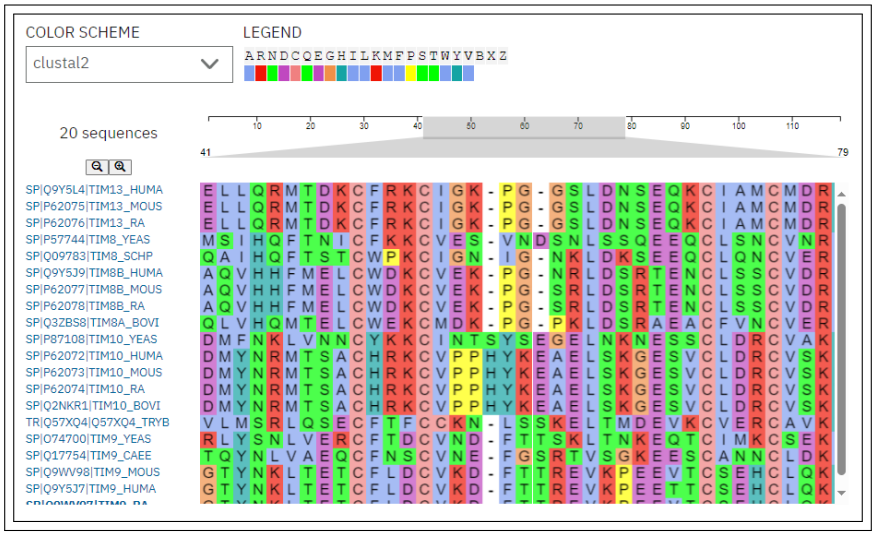


FIGURE 19. Multiple sequence alignment in CLUSTAL OMEGA

The outputs of multiple sequence alignment in **CLUSTAL OMEGA**, **MAFFT** and **MUSCLE** are given below:

```
sp|Q9Y5L4|TIM13_HUMAN       MEGGFGSDFGGSGSGKLD-PGLIMEQVKVQIAVA---NAQELLQRMTDKCFRKCIGK-PG        55
sp|P62075|TIM13_MOUSE       MDSGFGSDFGGTGGGKLD-PGAIMEQVKVQIAVA---NAQELLQRMTDKCFRKCIGK-PG        55
sp|P62076|TIM13_RAT         MDSGFGSDFGGTGGGKLD-PGAIMEQVKVQIAVA---NAQELLQRMTDKCFRKCIGK-PG        55
sp|P57744|TIM8_YEAST        MSSLSTSDLASLDD--TS-KKEIATFLEGENSKQ---KVQMSIHQFTNICFKKCVES-VN        53
sp|Q09783|TIM8_SCHPO        MADATKNPIADLSE--SE-QLELSKFIESEQQKV---KLQQAIHQFTSTCWPKCIGN-IG        53
sp|Q9Y5J9|TIM8B_HUMAN       --------MAELGE--AD-EAELQRLVAAEQQKA---QFTAQVHHFMELCWDKCVEK-PG        45
sp|P62077|TIM8B_MOUSE       --------MAELGE--AD-EAELQRLVAAEQQKA---QFTAQVHHFMELCWDKCVEK-PG        45
sp|P62078|TIM8B_RAT         --------MAELGE--AD-EAELQRLVAAEQQKA---QFTAQVHHFMELCWDKCVEK-PG        45
sp|Q3ZBS8|TIM8A_BOVIN       MDSSSSSSAAGLGS--V--DPQLQHFIEVETQKQ---RFQQLVHQMTELCWEKCMDK-PG        52
sp|P87108|TIM10_YEAST       ------MSFLGFGGGQPQLSSQQ-KIQAAEAELD---LVTDMFNKLVNNCYKKCINTSYS        50
sp|P62072|TIM10_HUMAN       ---------------MD--PLRA-QQLAAELEVE---MMADMYNRMTSACHRKCVPPHYK        39
sp|P62073|TIM10_MOUSE       ---------------MD--PLRA-QQLAAELEVE---MMADMYNRMTSACHRKCVPPHYK        39
sp|P62074|TIM10_RAT         ---------------MD--PLRA-QQLAAELEVE---MMADMYNRMTSACHRKCVPPHYK        39
sp|Q2NKR1|TIM10_BOVIN       ---------------MD--PLRA-QQLAAELEVE---MMADMYNRMTSACHRKCVPPHYK        39
tr|Q57XQ4|Q57XQ4_TRYB2      ----------------------------MRLAVKQESFR-LEVLMSRLQSECFTFCCKN-LS        32
sp|O74700|TIM9_YEAST        ---------------MDALNSKEQQEFQKVVEQKQMKDFMRLYSNLVERCFTDCVND-FT        44
sp|Q17754|TIM9_CAEEL        --------------------------MTSEQNIQTFRDFLTQYNLVAEQCFNSCVNE-FG        33
sp|Q9WV98|TIM9_MOUSE        ----------------------MAAQIPESDQIKQFKEFLGTYNKLTETCFLDCVKD-FT        37
sp|Q9Y5J7|TIM9_HUMAN        ----------------------MAAQIPESDQIKQFKEFLGTYNKLTETCFLDCVKD-FT        37
sp|Q9WV97|TIM9_RAT          ----------------------MAAQIPESDQIKQFKEFLGTYNKLTETCFLDCVKD-FT        37
                                                                  . . *   *

sp|Q9Y5L4|TIM13_HUMAN       -GSLDNSEQKCIAMCMDRYMDAWNTVSRAYNSRLQRERANM------------------ 95
sp|P62075|TIM13_MOUSE       -GSLDNSEQKCIAMCMDRYMDAWNTVSRAYNSRLQRERANM------------------ 95
sp|P62076|TIM13_RAT         -GSLDNSEQKCIAMCMDRYMDAWNTVSRAYNSRLQRERANM------------------ 95
sp|P57744|TIM8_YEAST        DSNLSSQEEQCLSNCVNRFLDTNIRIVNGLQNTR------------------------- 87
sp|Q09783|TIM8_SCHPO        -NKLDKSEEQCLQNCVERFLDCNFHIIKRYALEKFG-FLFCWL--GFSC---------- 98
sp|Q9Y5J9|TIM8B_HUMAN       -NRLDSRTENCLSSCVDRFIDTTLAITSRFAQIVQKGGQ-------------------- 83
sp|P62077|TIM8B_MOUSE       -SRLDSRTENCLSSCVDRFIDTTLAITGRFAQIVQKGGQ-------------------- 83
sp|P62078|TIM8B_RAT         -SRLDSRTENCLSSCVDRFIDTTLAITGRFAQIVQKGGQ-------------------- 83
sp|Q3ZBS8|TIM8A_BOVIN       -PKLDSRAEACFVNCVERFIDTSQFILNRLEQTQKSKPVFS-E--SLSD---------- 97
sp|P87108|TIM10_YEAST       EGELNKNESSCLDRCVAKYFETNVQVGENMQKMGQSFNAAGKF---------------- 93
sp|P62072|TIM10_HUMAN       EAELSKGESVCLDRCVSKYLDIHERMGKKLTELSMQDEELMKR-VQQSSGPA------- 90
sp|P62073|TIM10_MOUSE       EAELSKGESVCLDRCVSKYLDIHERMGKKLTELSMQDEELMKR-VQQSSGPA------- 90
sp|P62074|TIM10_RAT         EAELSKGESVCLDRCVSKYLDIHERMGKKLTELSMQDEELMKR-VQQSSGPA------- 90
sp|Q2NKR1|TIM10_BOVIN       EAELSKGESVCLDRCVSKYLDIHERMGKKLTELSMQDEELMKR-AQQSSGPV------- 90
tr|Q57XQ4|Q57XQ4_TRYB2      SKELTMDEVKCVERCAVKYLQASDIINRALDKGE-------------SGGGAVKQMLKL 78
sp|O74700|TIM9_YEAST        TSKLTNKEQTCIMKCSEKFLKHSERVGQRFQEQNAALGQGLGR---------------- 87
sp|Q17754|TIM9_CAEEL        SRTVSGKEESCANNCLDKFLKMTQRVSQRFQEHQLLNAQANGAAIKVENGGKINKIQ-- 90
sp|Q9WV98|TIM9_MOUSE        TREVKPEEVTCSEHCLQKYLKMTQRISVRFQEYHIQQNEALAAKAGLLGQPR------- 89
sp|Q9Y5J7|TIM9_HUMAN        TREVKPEETTCSEHCLQKYLKMTQRISMRFQEYHIQQNEALAAKAGLLGQPR------- 89
sp|Q9WV97|TIM9_RAT          TREVKPEEVTCSEHCLQKYLKMTQRISMRFQEYHIQQNEALAAKAGLLGQPR------- 89
```

FIGURE 20. Complete Multiple sequence alignment in CLUSTAL OMEGA

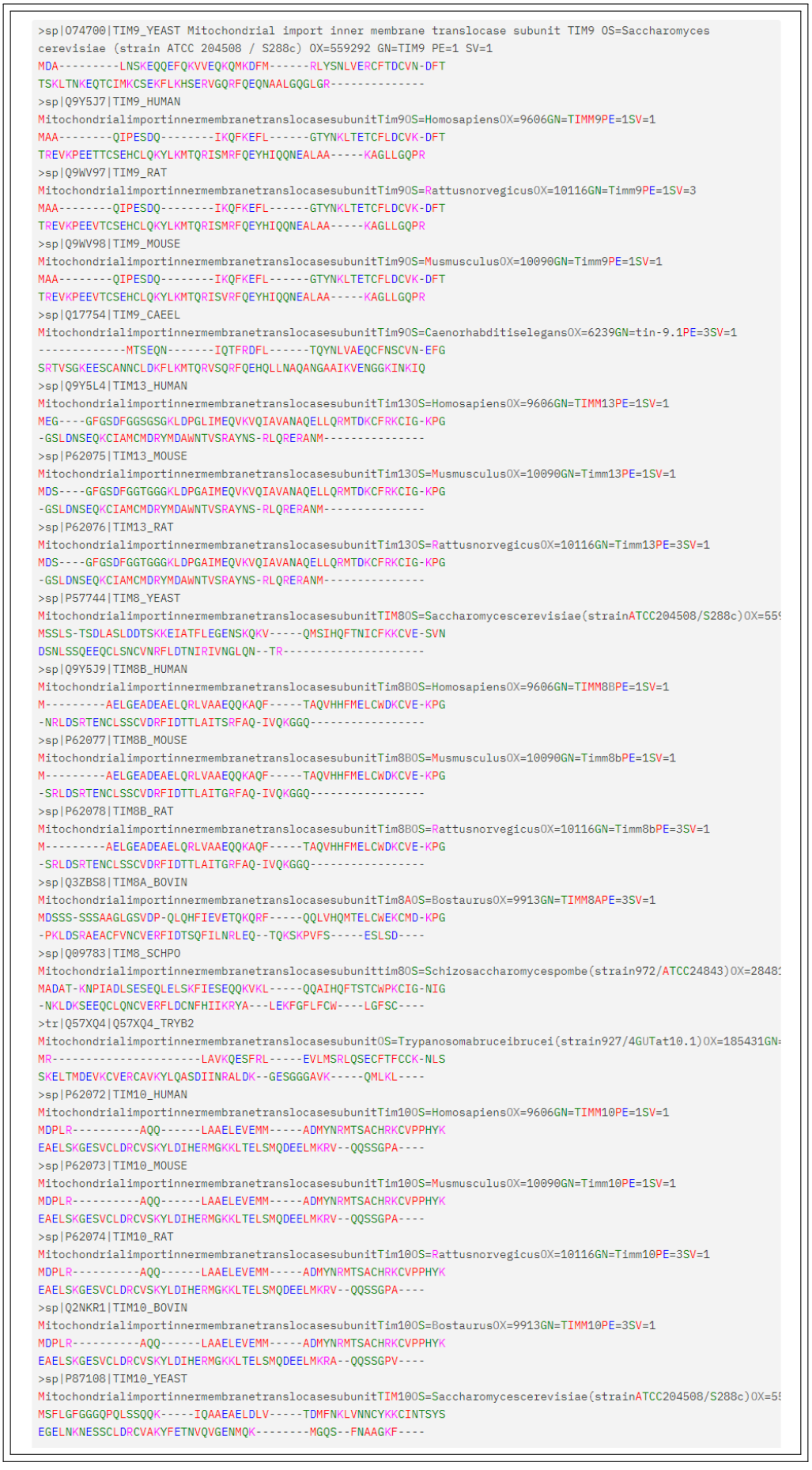


FIGURE 21. Complete Multiple sequence alignment in MAFFT

FIGURE 22. Complete Multiple sequence alignment in MUSCLE

Below, I have shown 5 proteins sequence alignments, with clustal omega, mafft, and muscle sequence alignments for same protein side-by-side.
In the above few images, it can be seen that the relative position of different residues within the same proteins have changed when multiple sequence alignment is done using different methods.

- **Clustal Omega** uses **seeded guide trees** and **HMM profile-profile** techniques to generate alignments between three or more sequences.
- **MAFFT** uses an algorithm based on **progressive alignment**, in which the sequences were clustered with the help of the **fast Fourier transform**.
- **MUSCLE** uses an algorithm that includes **fast distance estimation using k-mer counting**, **progressive alignment** using a new profile function called **log-expectation score**, and refinement using **tree-dependent restricted partitioning**.

FIGURE 23. MSA comparison between the three types

Residue differences in Multiple Sequence Alignment

Below, I have tabulated **5 TIM barrel proteins** along with their residue mismatches in **MUSCLE**, **MAFFT**, and **CLUSTAL** after the **Multiple Sequence Alignment**.

TIM barrel protein sequences	Residue position mismatch in the 3 methods			
	Position	MUSCLE	MAFFT	CLUSTAL
TIM10_HUMAN	17	M	Q	D
	26	L	A	Q
	29	E	L	A
	31	E	V	L
	33	E	M	V
TIM9_MOUSE	28	Q	K	P
	29	I	Q	E
	30	K	F	S
	31	Q	K	D
	32	F	E	Q
TIM13_RAT	20	G	K	P
	23	M	P	I
	28	V	E	K
	34	N	I	A
	96	N	Q	R
TIM8A_BOVIN	12	A	G	L
	23	Q	H	L
	27	E	V	I
	34	R	F	Q
	99	S	K	V
TIM8_YEAST	8	T	S	D
	11	L	A	S
	21	E	I	K
	29	E	N	G
	34	K	V	Q

Question 8. Blast the below sequence **EPDMRTPIAHTMAW** against the **PDB database**. Analyze the results and discuss the significance of the results.

Solution. The image for the search query on **blastp** is given below. I have entered the given sequence in **FASTA** format and database is chosen to be **PDB (Protein Data Bank)**.

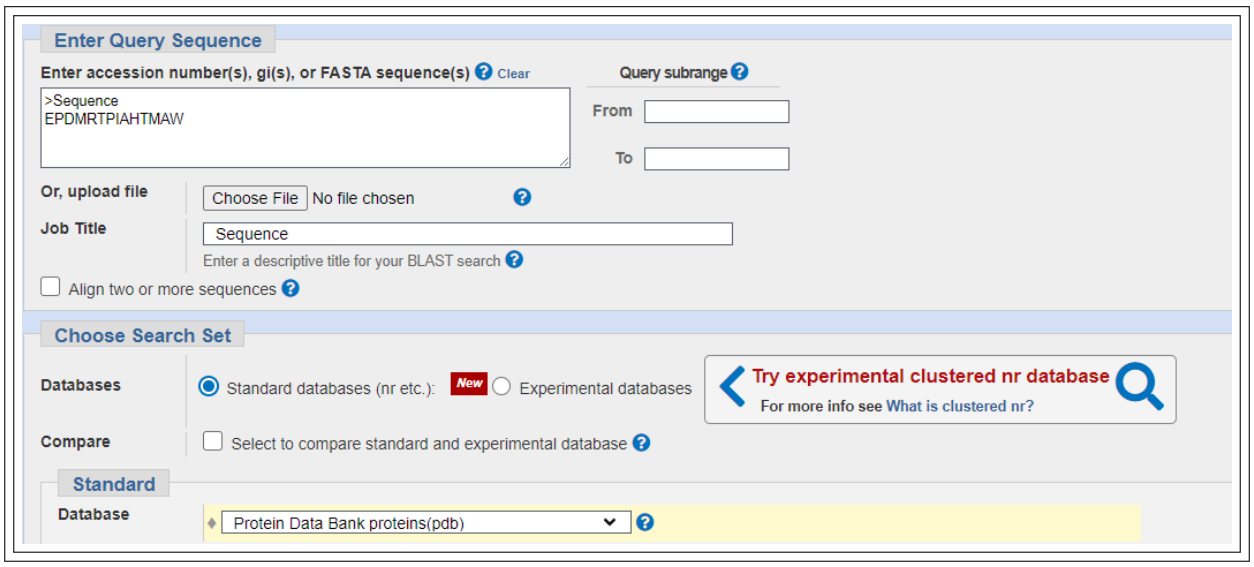
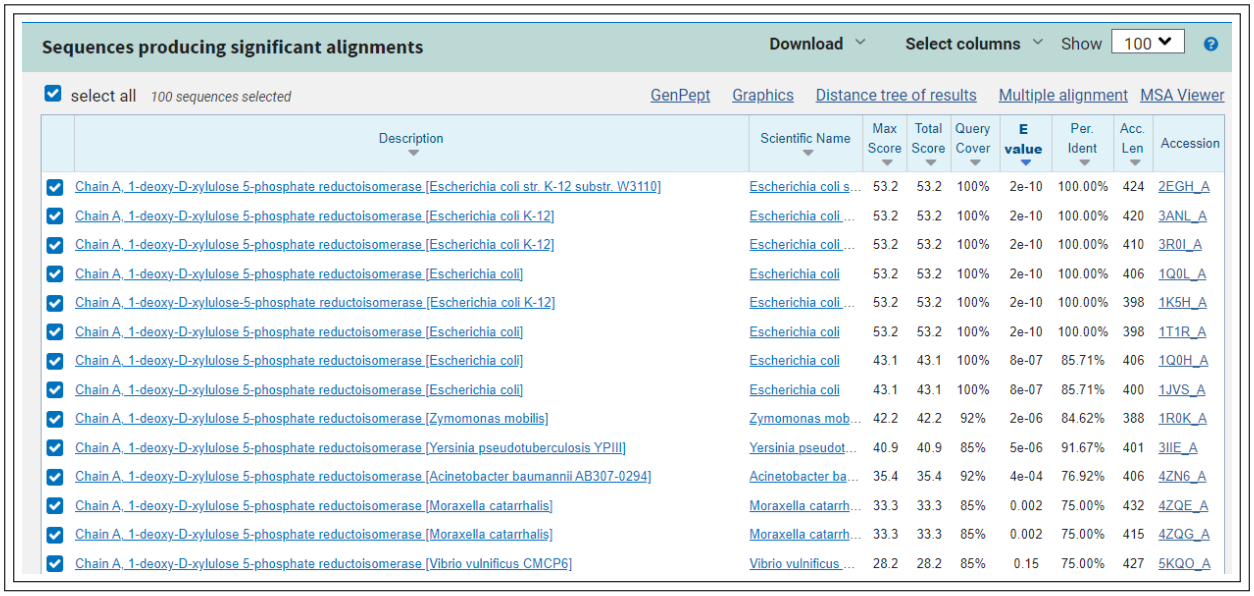


FIGURE 24. blastp against PDB database

Due to **blastp** algorithm parameter limitation of **maximum 100 aligned sequences** to display, only 100 sequences have been displayed. Below is the descriptive statistics for the same.



Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession
Chain A, 1-deoxy-D-xylulose 5-phosphate reductoisomerase [Escherichia coli str. K-12 substr. W3110]	Escherichia coli s...	53.2	53.2	100%	2e-10	100.00%	424	2EGH_A
Chain A, 1-deoxy-D-xylulose 5-phosphate reductoisomerase [Escherichia coli K-12]	Escherichia coli ...	53.2	53.2	100%	2e-10	100.00%	420	3ANL_A
Chain A, 1-deoxy-D-xylulose 5-phosphate reductoisomerase [Escherichia coli K-12]	Escherichia coli ...	53.2	53.2	100%	2e-10	100.00%	410	3R0I_A
Chain A, 1-deoxy-D-xylulose 5-phosphate reductoisomerase [Escherichia coli]	Escherichia coli	53.2	53.2	100%	2e-10	100.00%	406	1Q0L_A
Chain A, 1-deoxy-D-xylulose-5-phosphate reductoisomerase [Escherichia coli K-12]	Escherichia coli ...	53.2	53.2	100%	2e-10	100.00%	398	1K5H_A
Chain A, 1-deoxy-D-xylulose 5-phosphate reductoisomerase [Escherichia coli]	Escherichia coli	53.2	53.2	100%	2e-10	100.00%	398	1T1R_A
Chain A, 1-deoxy-D-xylulose 5-phosphate reductoisomerase [Escherichia coli]	Escherichia coli	43.1	43.1	100%	8e-07	85.71%	406	1Q0H_A
Chain A, 1-deoxy-D-xylulose 5-phosphate reductoisomerase [Escherichia coli]	Escherichia coli	43.1	43.1	100%	8e-07	85.71%	400	1JVS_A
Chain A, 1-deoxy-D-xylulose 5-phosphate reductoisomerase [Zymomonas mobilis]	Zymomonas mob...	42.2	42.2	92%	2e-06	84.62%	388	1R0K_A
Chain A, 1-deoxy-D-xylulose 5-phosphate reductoisomerase [Yersinia pseudotuberculosis YPIII]	Yersinia pseudot...	40.9	40.9	85%	5e-06	91.67%	401	3IIE_A
Chain A, 1-deoxy-D-xylulose 5-phosphate reductoisomerase [Acinetobacter baumannii AB307-0294]	Acinetobacter ba...	35.4	35.4	92%	4e-04	76.92%	406	4ZN6_A
Chain A, 1-deoxy-D-xylulose 5-phosphate reductoisomerase [Moraxella catarrhalis]	Moraxella catarrh...	33.3	33.3	85%	0.002	75.00%	432	4ZQE_A
Chain A, 1-deoxy-D-xylulose 5-phosphate reductoisomerase [Moraxella catarrhalis]	Moraxella catarrh...	33.3	33.3	85%	0.002	75.00%	415	4ZQG_A
Chain A, 1-deoxy-D-xylulose 5-phosphate reductoisomerase [Vibrio vulnificus CMCP6]	Vibrio vulnificus ...	28.2	28.2	85%	0.15	75.00%	427	5KQO_A

FIGURE 25. Descriptive statistics for blastp against PDB

Following are the observations from the descriptive statistics:

- **8** of the aligned sequences have **100% sequence identity** (Obtained by sorting according to percentage identity).
- **8** of the aligned sequences have **100% query coverage**.
- The sequences with the highest sequence identity percentage of 100% belong to the **Escherichia coli**.
- The blastp algorithm automatically **adjusts parameters** for **short input sequences**.
- Scoring matrix used is **BLOSUM62**, with **existence penalty** and **extension penalty**, **11** and **1** respectively.
- The maximum total score attained through alignment is **53.2**.

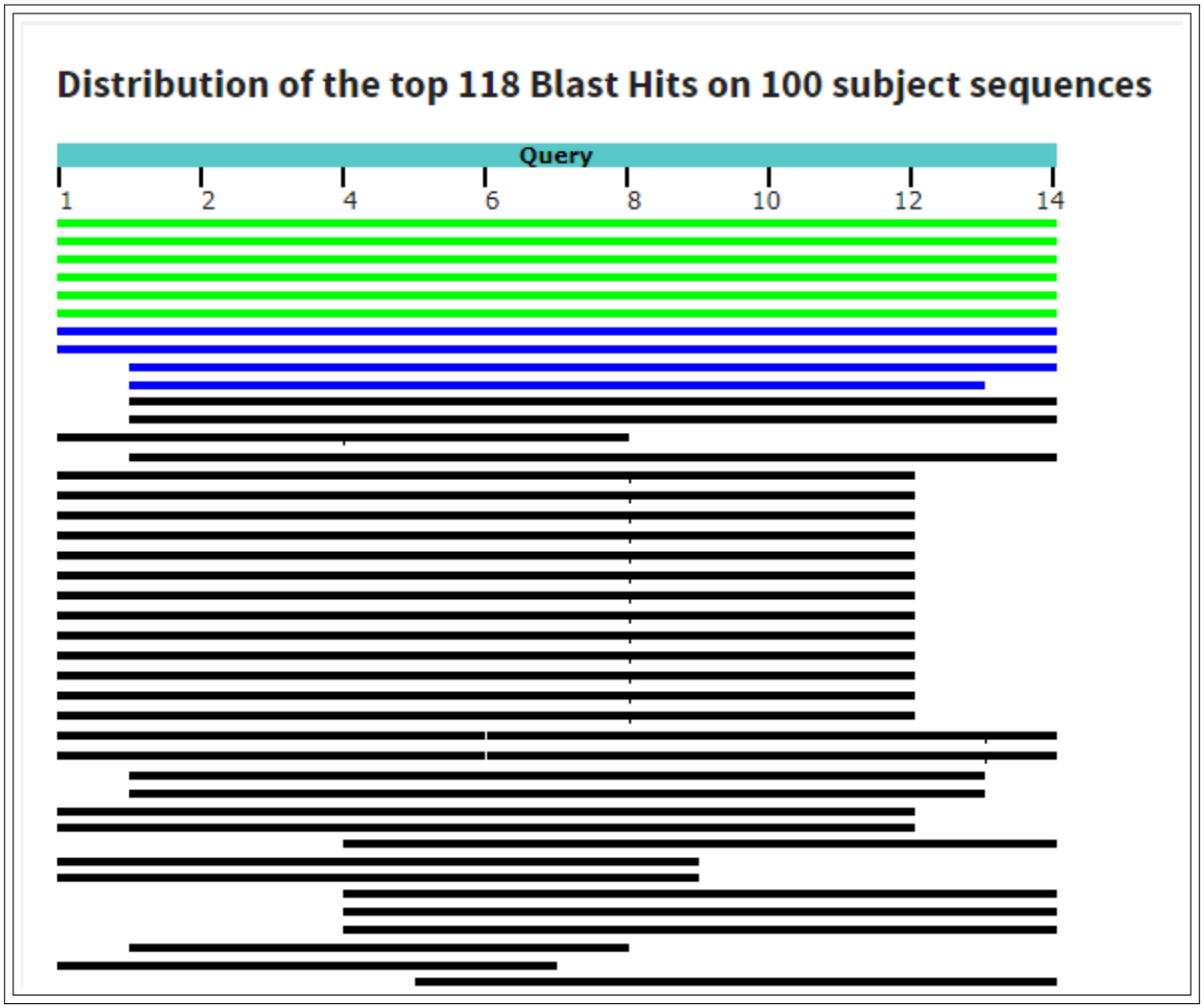


FIGURE 26. Alignment Scoring

The above image shows the distribution of the top **118 blast hits** on some among the **100 subject sequences**. The **green color** indicates that the alignment score is in the range **50-80**. The **blue color** indicates that the alignment score is in the range **40-50**. The **black color** indicates that the alignment score is in the range **below 40**. It is evident from the percentage alignment seen in the image.

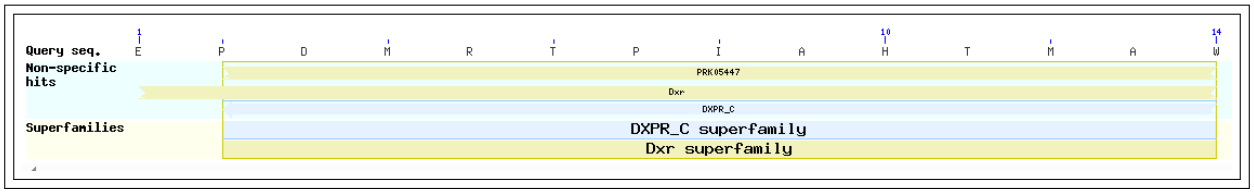


FIGURE 27. Conserved domains in the alignment

In the alignment, putative conserved domains have been detected. Three of the domains are:

- **DXPR_C**: This is the C-terminal domain of the **1-deoxy-D-xylulose-5-phosphate reductoisomerase** enzyme. This domain forms a **left handed super-helix**.
- **Dxr**: This is the domain of the **1-deoxy-D-xylulose 5-phosphate reductoisomerase**. [Lipid transport and metabolism].
- **PRK05447**: This is the member of the superfamily **cl42529**. It catalyzes the formation of an alternative **nonmevalonate pathway for terpenoid biosynthesis**.

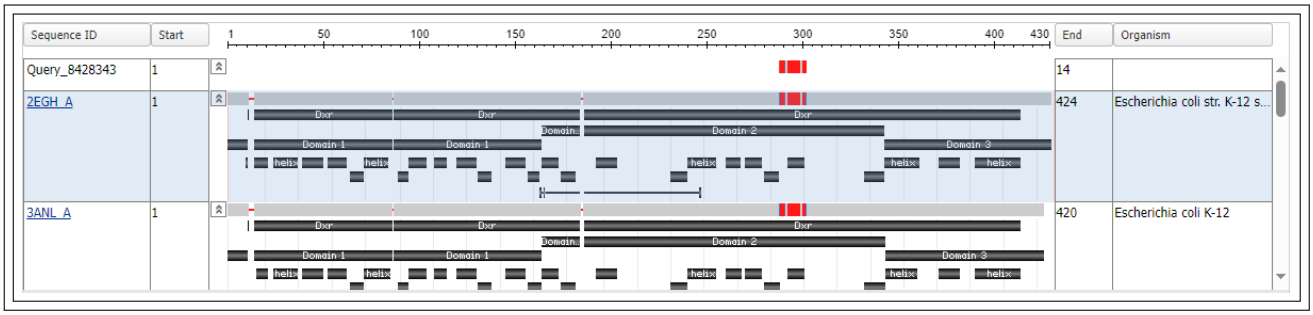


FIGURE 28. Multiple sequence alignment

The above shows the multiple sequence alignment among the multiple matches in the **blastp tool** with the query. The red region in the query line represents the query sequence. As we can see, that the input query is very short. The **blastp** tool is so designed to tackle the case of short input sequences easily.
Even while matching, the regions in the subject sequences that match the query are shown in red. This image also shows the super-family, the domain and helix information about the matches. It also allows us to shift the start and end points for our sequence alignment analysis.

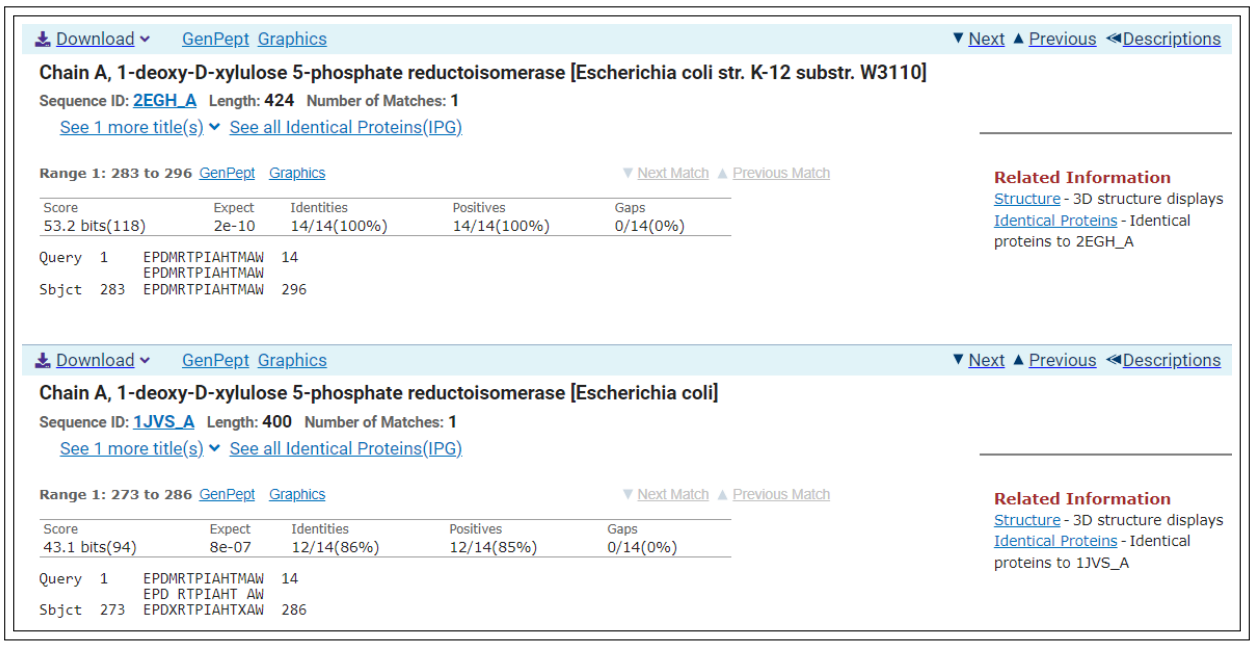


FIGURE 29. Individual pairwise sequence alignment

Above is the alignment view with **pairwise alignment** between the query sequence and one of the subject sequences. This window gives information about the **start and end position** in the subject sequence that aligns best with the query sequence. This page also gives access to the **3D structures** of the matched sequences.
It also gives the **total alignment score**, **expect value**, **percentage sequence identity**, **percentage positives (similarity)**, and **gap percentage**. The number of matches in the sequence is given. There are occurrences where given query has more than one match in the sequence.
Below is a **phylogenetic tree view** of the aligned sequences. This has been generated using the **COBALT (Constraint based multiple alignment tool)**. The tree method used is **Fast Minimum Evolution**. The max sequence distance is set to **0.85**. The distance metric is set to **Grishin (protein)**. Number of aligned sequences in the phylogenetic tree is **12**. The blast names color map is given on the right side as legend.

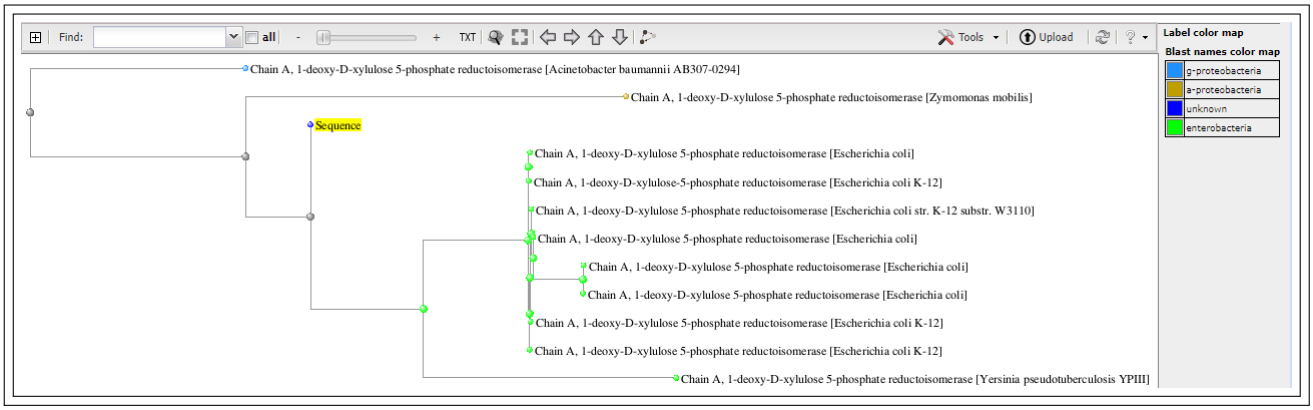


FIGURE 30. Phylogenetic tree view using COBALT

Above, I have discussed the results of the blasting the given sequence against the PDB database and their significance.