Assignment 4

1. a. Write a program to construct a dot plot for the alignment of human and chicken hemoglobin β chain. Identify the segments, which are same in both sequences.

import matplotlib.pyplot as plt

def matchstr(s1, s2):

seq=[]

seqrev=[]

tempstr=''

for i in range(len(s1)):

if s1[i]==s2[i]:

tempstr+=s1[i]

else:

seq.append(tempstr)

tempstr=''

return seq

def plot(s1, s2):

x=[]

y=[]

for i in range(len(s1)):

for j in range(len(s2)):

if s1[i]==s2[j]:

x.append(i)

y.append(j)

plt.scatter(x, y)

plt.xlabel("string1")

plt.ylabel("string2")

plt.title("Dot Plot")

plt.show()

if \_\_name\_\_=="\_\_main\_\_":

string1='''MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK

VKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFG

KEFTPPVQAAYQKVVAGVANALAHKYH'''

string2='''MVHWTAEEKQLITGLWGKVNVAECGAEALARLLIVYPWTQRFFASFGNLSSPTAILGNPM

VRAHGKKVLTSFGDAVKNLDNIKNTFSQLSELHCDKLHVDPENFRLLGDILIIVLAAHFS

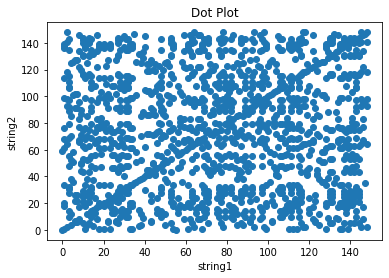
KDFTPECQAAWQKLVRVVAHALARKYH'''

plot(string1, string2)

match=matchstr(string1, string2)

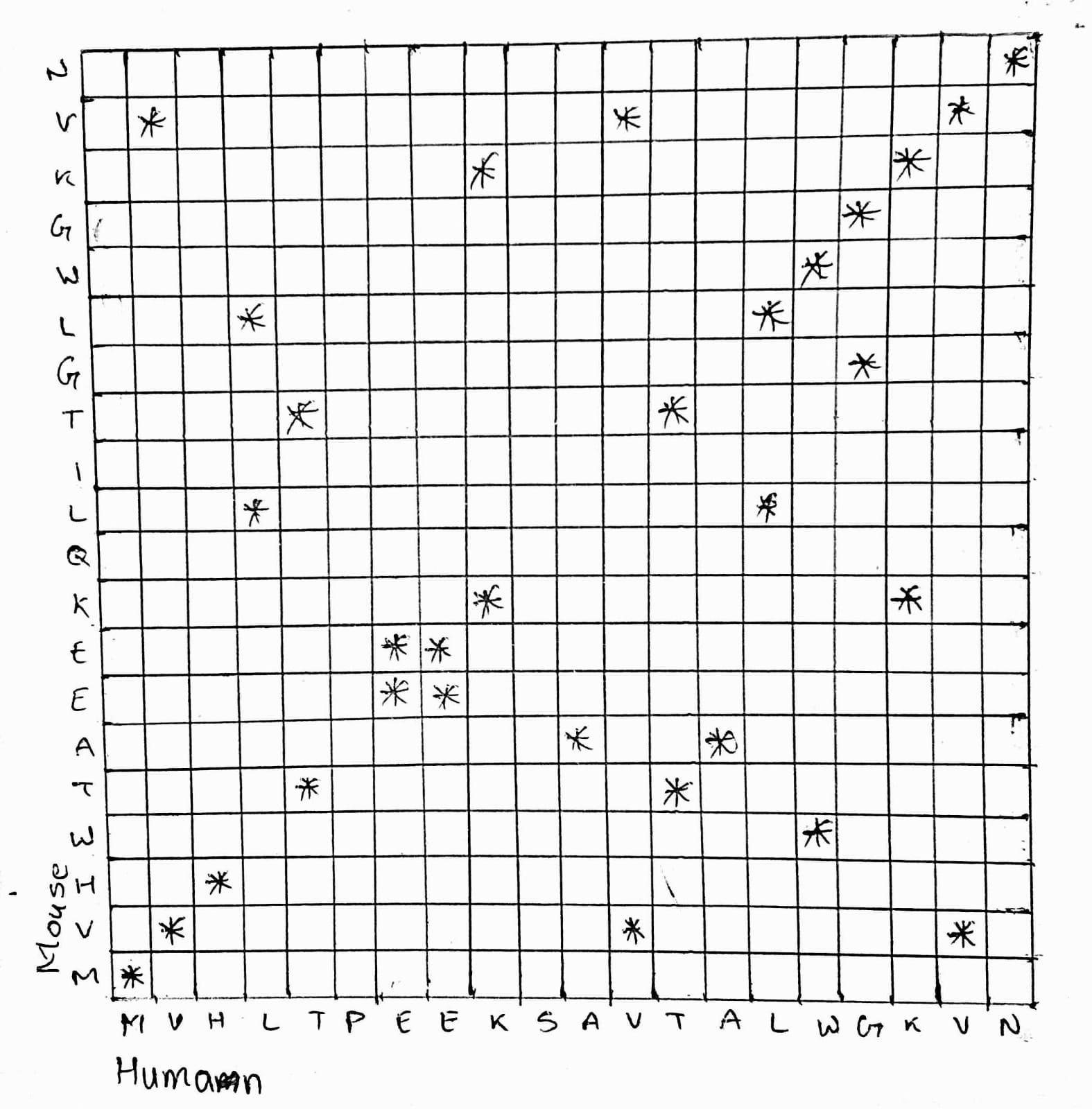
print(match)

Output:



['MVH', 'T', 'EEK', '', '', 'T', 'LWGKVNV', 'E', 'G', 'EAL', 'RLL', 'VYPWTQRFF', 'SFG', 'LS', 'P', 'A', '', 'GNP', '\nV', 'AHGKKVL', '', 'F', 'D', '', '', '', 'LDN', 'K', 'TF', '', 'LSELHCDKLHVDPENFRLLG', '', 'L', '', 'VLA', 'HF', '\nK', 'FTP', '', 'QAA', 'QK', 'V', '', 'VA', 'ALA']

b. Construct the dot plot manually for the residues 1-20 and verify with the plot obtained using program



1. Calculate the score for the following alignments using code:

AATCTATA

AAG--ATA

Assume that the match score is 1, mismatch score is 0, origination penalty is -2, and length penalty is -1.

def score(s1, s2):

score=0

for i in range(len(s1)):

if (s1[i]==s2[i]):

score+=1

elif (s1[i]=='-' or s2[i]=='-'):

if (s1[i-1]=='-' or s2[i-1]=='-'):

score-=1

else:

score-=3

return score

string1='AATCTATA'

string2='AAG--ATA'

print(score(string1, string2))

Output:

1

1. Verify the Q2 manually

String1=AATCTATA

String2=AAG--ATA

Starting from first character of two strings:

1. A – A = Match score=+1 (0+1)
2. A – A = Match score=+2 (1+1)
3. T – G = mismatch score=+2 (0+2)
4. C - - = OP + LP score=-1 (-3+2)
5. T - - = LP score=-2 (-1-1)
6. A – A = Match score=-1 (1-2)
7. T – T = Match score= 0 (1-1)
8. A – A = Match score=+1 (1+1)
9. Using the Needleman and Wunsch dynamic programming method, construct the partial alignment score table and align the following two sequences (using code): ACAGTCGAACG and ACCGTCCG

use the scoring parameters: match score: +2; mismatch score: -1 and gap penalty: -2

import numpy as np

import pandas as pd

def NnWa(s1, s2):

arr=np.zeros((len(s1)+1, len(s2)+1))

arr[:,0]=np.arange(0,2\*(-len(s1)-1), -2)

arr[0,:]=np.arange(0,2\*(-len(s2)-1), -2)

temp=[0, 0, 0]

for i in range(len(s1)):

for j in range(len(s2)):

if (s1[i]==s2[j]):

temp[0]=arr[i,j]+2

else:

temp[0]=arr[i,j]-1

temp[1]=arr[i,j+1]-2

temp[2]=arr[i+1,j]-2

arr[i+1,j+1]=max(temp)

str1=" "+s1

str2=" "+s2

panda\_df=pd.DataFrame(data=arr.astype(int), index=[i for i in str1], columns=[i for i in str2])

print(panda\_df)

x=[]

y=[]

i=len(s1)

j=len(s2)

x.append(s1[i-1])

y.append(s2[j-1])

i-=1

j-=1

while i>0 or j>0:

if max(arr[i][j], arr[i][j+1],arr[i+1][j])==arr[i][j]:

x.append(s1[i-1])

y.append(s2[j-1])

i-=1

j-=1

elif max(arr[i][j], arr[i][j+1],arr[i+1][j])==arr[i][j+1]:

x.append(s1[i-1])

y.append('-')

i-=1

elif max(arr[i][j], arr[i][j+1],arr[i+1][j])==arr[i+1][j]:

x.append('-')

y.append(s1[j-1])

j-=1

x=''.join(x[::-1])

y=''.join(y[::-1])

print(x)

print(y)

#-----MAIN PROGRAM-------

string1='ACAGTCGAACG'

string2='ACCGTCCG'

NnWa(string1, string2)

Output:

A C C G T C C G

0 -2 -4 -6 -8 -10 -12 -14 -16

A -2 2 0 -2 -4 -6 -8 -10 -12

C -4 0 4 2 0 -2 -4 -6 -8

A -6 -2 2 3 1 -1 -3 -5 -7

G -8 -4 0 1 5 3 1 -1 -3

T -10 -6 -2 -1 3 7 5 3 1

C -12 -8 -4 0 1 5 9 7 5

G -14 -10 -6 -2 2 3 7 8 9

A -16 -12 -8 -4 0 1 5 6 7

A -18 -14 -10 -6 -2 -1 3 4 5

C -20 -16 -12 -8 -4 -3 1 5 3

G -22 -18 -14 -10 -6 -5 -1 3 7

ACAGTCGAACG

ACCGTC---CG

1. Verify Q4 manually

