EXPLANATION for R CODE2

#In this code, I tried to create a logic which finds out whether a local sequence alignment of two protein sequences also implies structural similarity of the aligned parts.

# Firstly, I tried to demonstrate the steps in detail. Then I added the whole code to the end.

# This code provides an easy way to catch the similarity and it is simple in terms of being deductive.

#Here are the directions for a person who will use my codes:

#1-) Please go to http://www.rcsb.org/pdb/home/home.do

#2-) Write the protein names that you want to analyze respectively to the column and click Go.

#3-) This program specifically use the FASTA format for easy understanding, so choose

#download file at the right side and choose fasta format.

#You can also download the PDB. File format but you should go and specifically choose the
#SEQRES information from that file. Each of two is acceptable but code uses the FASTA format.

FASTA format look like this:

#>1QK9:A|PDBID|CHAIN|SEQUENCE

#ASASPKQRRSIIRDRGPMYDDPTLPEGWTRKLKQRKSGRSAGKYDVYLINPQGKAFR
#SKVELIAYFEKVGDTSLDPNDFD

#FTVTGRGSGSGC

#Also, PDB.text format includes SEQRES information and look like that:

```
’’’SEQRES   1 A   92  ALA SER ALA SER PRO LYS GLN ARG ARG SER ILE
ILE ARG
SEQRES   2 A   92  ASP ARG GLY PRO MET TYR ASP ASP PRO THR LEU
PRO GLU

SEQRES   3 A   92  GLY TRP THR ARG LYS LEU LYS GLN ARG LYS SER
GLY ARG
SEQRES   4 A   92  SER ALA GLY LYS TYR ASP VAL TYR LEU ILE ASN
PRO GLN

SEQRES   5 A   92  GLY LYS ALA PHE ARG SER LYS VAL GLU LEU ILE
ALA TYR
SEQRES   6 A   92  PHE GLU LYS VAL GLY ASP THR SER LEU ASP PRO
ASN ASP

SEQRES   7 A   92  PHE ASP PHE THR VAL THR GLY ARG GLY SER GLY
SER GLY

SEQRES   8 A   92  CYS
```

User has many options for further analysis:

i-) choose the protein sequence from FASTA format and create a vector includes the information and use it.

ii-) use the FASTA format directly and use the appropriate codes for it.
iii-) Eliminate the SEQRES information from text file, copy and paste it to the notepad and

use it for further analysis.

For example the file for i will look like this: >1QK9:A|PDBID|CHAIN|SEQUENCE

ASASPKQRRSIIRDRGPMYDDPTLPEGWTRKLKQRKSGRSAGKYDVYLINPQGKAFR
SKVELIAYFEKVGDTSLDPNDFDFTVTGRGSGSGC

For example the file for iii will look like this:

```
ALA SER ALA SER PRO LYS GLN ARG ARG SER ILE ILE ARG
ASP ARG GLY PRO MET TYR ASP ASP PRO THR LEU PRO GLU
GLY TRP THR ARG LYS LEU LYS GLN ARG LYS SER GLY ARG
SER ALA GLY LYS TYR ASP VAL TYR LEU ILE ASN PRO GLN
GLY LYS ALA PHE ARG SER LYS VAL GLU LEU ILE ALA TYR
PHE GLU LYS VAL GLY ASP THR SER LEU ASP PRO ASN ASP
PHE ASP PHE THR VAL THR GLY ARG GLY SER GLY SER GLY
CYS
```

But the iii seems confusing so I preferred to use FASTA format. ‘’

```
install.packages("seqinr") # this package is installed for being able to use FASTA format
library("seqinr")

1xyx <- read.fasta(file = "1qk9.fasta")

1xu0 <- read.fasta(file = "1bb8.fasta")

1xyxseq <- 1QK9[[1]]
1xu0seq <- 1BB8[[1]]
seq1string <- toupper ( c2s(1qk9seq[[1]])) # toupper function convert the sequence elements to an upper letter.
seq2string <- toupper ( c2s(1bb8seq[[2]]))

library(Biostrings) # this is special package and used for alignment analysis.

data(BLOSUM62) #it was pointed out in the question that BLOSUM62 matrix should be used for scoring.
```

→In bioinformatics, the BLOSUM (BLOcks SUbstitution Matrix) matrix is a substitution matrix used for sequence alignment of proteins. BLOSUM matrices are used to score alignments between evolutionarily divergent protein sequences.

# pairwiseAlignment(seq1string, seq2string, substitutionMatrix ="BLOSUM62", gapOpening = -5, scoreOnly = FALSE, type = "local") # here is the code that computes pairwise alignment between two sequences. As it was said, BLOSUM62 matrix is used and “-5” score is used for gap penalty. It tries to find out the best local sequence alignment.

Also; you can prefer to copy and paste the sequence information from the FASTA format and use directly the information, it also gives the same result.

SEQRES1<-"

ASASPKQRRSIIRDRGPMYDDPTLPEGWTRKLKQRKSGRSAGKYDVYLINPQGKAFR
SKVELIAYFEKVGDTSLDPNDFDFTVTGRGSGSGC"

SEQRES2<”EKRRDNRGRILKTGESQRKDGRYLYKYIDSFGEPQFVYSWKLVATDRVP

AGKRDCISLREKIAELQKDIHD "

pairwiseAlignment(SEQRES1,SEQRES2, substitutionMatrix ="BLOSUM62", gapOpening =

-5, scoreOnly = FALSE, type = "local")

EXTRA INFORMATION:

pairwiseAlignment(SEQRES1,SEQRES2, substitutionMatrix ="BLOSUM62", gapOpening

= -5, scoreOnly = FALSE, type = "local")

If you write , scoreOnly = TRUE , it only gives you the score, do not show the sequences as

detailed.

Also, there exist a code for Amino acid local alignment and it gives the same result with the above one:

gapOpening =-5)

Here is the result:

Local PairwiseAlignmentsSingleSubject (1 of 1) pattern: [33] KQRKSGRSAGKY
subject: [16] SQRKDGRYLYKY
score: 32

pairwiseAlignment(AAString("ASASPKQRRSIIRDRGPMYDDPTLPEGWTRKLKQRKSG

RSAGKYDVYLINPQGKAFRSKVELIAYFEKVGDTSLDPNDFDFTVTGRGSGSGC"),

AAString("EKRRDNRGRILKTGESQRKDGRYLYKYIDSFGEPQFVYSWKLVATDRVPA

GKRDCISLREKIAELQKDIHD"),type = "local", substitutionMatrix = "BLOSUM62",

Now; it is time to use the local sequence alignment results and superimpose the “matching, i.e., positive scoring” part of the alignment structurally

Here is the table that shows positive scorings:

- The root-mean-square deviation (RMSD) is the measure of the average distance between the atoms of superimposed proteins. In the study of globular protein conformations, one customarily measures the similarity in three-dimensional structure by the RMSD of the Cα atomic coordinates after optimal rigid body superposition.
- According to the results from local alignment, I have chosen the aminoacids that have positive score according to the BLOUSUM62 matrix.
- BLOUSUM62 matrix says that, only the aminoacids match with itself and some others has positive scores which are WF,WY,YF,DN,ED,QE,KR,IM,LM,LI,VM,VI and VL (also the reverse versions of course.)
- So, I found out these matches and revealed their ATOM scores by using only Cα coordinates.
- I will use a new code related with RMSD and it needs two matrices in it, which are refX and expX, first one is the coordinates in the reference structure and it can be any protein in this case, second one is the coordinates for structure it needs to be aligned to the reference one.
- All these two are n*3 matrices(the 3 in here shows the x,y,z coordinates of the alpha-Carbon)

QRKGRKY QRKGRKY

→Here are the matches that has positive scores. According to these results, the alpha carbon coordinates should be derived.

Here is the alpha-Carbon xyz coordinate information for the positive matching parts for 1QK9 protein sequence:

```
ATOM    547  CA  GLN A  34
C

ATOM    564  CA  ARG A  35
C

ATOM    588  CA  LYS A  36
C

ATOM    855  CA  GLY A  53
C
```

```
ATOM    914  CA  ARG A  57
C

ATOM    949  CA  LYS A  59
C

ATOM   1050  CA  TYR A  65
C

-14.145  -5.856  -4.797  1.00  0.00

-15.196  -9.456  -5.455  1.00  0.00

-18.592 -11.081  -5.976  1.00  0.00

  4.426   4.161  -8.874  1.00  0.00

 -1.107  -6.492  -1.989  1.00  0.00

  3.338  -7.254   3.655  1.00  0.00

 10.595  -4.761  -2.604  1.00  0.00
```

```
refX <- c(-14.14,-5.856,-4.797,-15.196,-9.456,-5.455,18.592,- 11.081,-5.976,4.426,4.161,-8.874,-
1.107,-6.492,-1.989,3.338,- 7.254,3.655,10.595,-4.761,-2.604)
```

```
matrix(refX,7,3, byrow=TRUE)
```

the matrix looks like this:

```
    [,1]    [,2]   [,3]
[1,] -14.140  -5.856 -4.797
[2,] -15.196  -9.456 -5.455
[3,]  18.592 -11.081 -5.976
[4,]   4.426   4.161 -8.874
[5,]  -1.107  -6.492 -1.989
[6,]   3.338  -7.254  3.655
[7,]  10.595  -4.761 -2.604
```

Here is the alpha-Carbon xyz coordinate information for the positive matching parts for 1BB8 protein sequence:

```
ATOM    275  CA  GLN A  19
C

ATOM    292  CA  ARG A  20
C

ATOM    316  CA  LYS A  21
```

```
C

ATOM    350  CA  GLY A  23
C

-10.146   4.103  -2.373  1.00  0.00

-11.381   4.642  -5.948  1.00  0.00

-14.973   4.215  -7.106  1.00  0.00
-12.838  -0.428  -4.940  1.00  0.00
```

```
ATOM    357  CA  ARG A  24
C

ATOM    442  CA  LYS A  28
C

ATOM    464  CA  TYR A  29
C

-9.079  -0.752  -5.590  1.00  0.00

 1.931   6.633  -1.957  1.00  0.00

 5.495   6.202  -0.615  1.00  0.00
```

```
expX <- c(-10.14,4.103,-2.373,-11.381,4.642,-5.948,-14.973,4.215,-7.106,-12.838,-0.428,- 4.940,-
9.079,-0.752,-5.590,1.931,6.633,-1.957,5.495,6.202,-0.615)

expX<- matrix(expX,7,3, byrow=TRUE)
```

the matrix looks like this:

```
[,1] [,2] [,3]
[1,] -10.140 4.103 -2.373 [2,] -11.381 4.642 -5.948 [3,] -14.973 4.215 -7.106 [4,] -12.838 -0.428 -
4.940 [5,] -9.079 -0.752 -5.590 [6,] 1.931 6.633 -1.957 [7,] 5.495 6.202 -0.615
```

```
rmsd <- function(refX, expX) {
library(MASS)
refo <- apply(refX, 2, mean)
refDX <- refX-matrix(rep(refo,dim(refX)[1]), dim(refX)[1], dim(refX)[2], byrow=T)
```

```
expO <- apply(expX, 2, mean)
expDX <- expX-matrix(rep(expO, dim(expX)[1]), dim(expX)[1], dim(expX)[2], byrow=T)
```

```
A <- matrix(0, 3, 3)
for (i in 1:3) {
for (j in 1:3) {
for (k in 1:dim(refDX)[1]){
A[i,j] <- A[i,j] + refDX[k,i]*expDX[k,j] }
}
}
TR <- sqrt(t(A)*A)*ginv(A) TS <- -TR * expO + refo
DX <- refX - expX*TR - matrix(rep(TS,dim(expX)[1]), dim(expX)[1], dim(expX)[2], byrow=T)
rmsdv <- sqrt(sum(apply(DX^2,1,sum))/dim(DX)[1]) return(rmsdv)
}
```

Here is the result:

rmsd:8.7

This result can be explained by the mathematical formula of RMSD:

→RMSD is used to compare differences between two things that may vary, neither of which is accepted as the "standard".

→So, when the result is too high, it means that the matching parts do not really shows the structural similarity or vice versa.

→My example result is too high and it seems not acceptable that do not confirm sequence alignment showed structural similarity at this time for this example.

→Also, I want to show the PDB 3D structures of these two proteins;

#HERE IS THE CONTINUOUS OF CODE:

```
pairwiseAlignment(AAString("ASASPKQRRSIIRDRGPMYDDPTLPEGWTRKLKQRKSGRSAG
KYDVYLINPQGKAFRSKVELIAYFEKVGDTSLDPNDFDFTVTGRGSGSGC"),
AAString("EKRRDNRGRILKTGESQRKDGRYLYKYIDSFGEPQFVYSWKLVATDRVPAGKR
DCISLREKIAELQKDIHD"),type = "local", substitutionMatrix = "BLOSUM62",
                gapOpening =-5)



refX <- c(-14.14,-5.856,-4.797,-15.196,-9.456,-5.455,18.592,-11.081,-5.976,4.426,4.161,-8.874,-
1.107,-6.492,-1.989,3.338,-7.254,3.655,10.595,-4.761,-2.604)

refX<- matrix(refX,7,3, byrow=TRUE)

expX <- c(-10.14,4.103,-2.373,-11.381,4.642,-5.948,-14.973,4.215,-7.106,-12.838,-0.428,-4.940,-
9.079,-0.752,-5.590,1.931,6.633,-1.957,5.495,6.202,-0.615)
expX<- matrix(expX,7,3, byrow=TRUE
rmsd <- function(refX, expX) {
library(MASS)
refo <- apply(refX, 2, mean)
refDX <- refX-matrix(rep(refo,dim(refX)[1]), dim(refX)[1], dim(refX)[2], byrow=T)

expO <- apply(expX, 2, mean)
expDX <- expX-matrix(rep(expO, dim(expX)[1]), dim(expX)[1], dim(expX)[2], byrow=T)

A <- matrix(0, 3, 3)
for (i in 1:3) {
for (j in 1:3) {
for (k in 1:dim(refDX)[1]){
A[i,j] <- A[i,j] + refDX[k,i]*expDX[k,j]
}
}
}
TR <- sqrt(t(A)*A)*ginv(A)
TS <- -TR * expO + refo

DX <- refX - expX*TR - matrix(rep(TS,dim(expX)[1]), dim(expX)[1], dim(expX)[2], byrow=T)
rmsdv <- sqrt(sum(apply(DX^2,1,sum))/dim(DX)[1])
return(rmsdv)
}
```