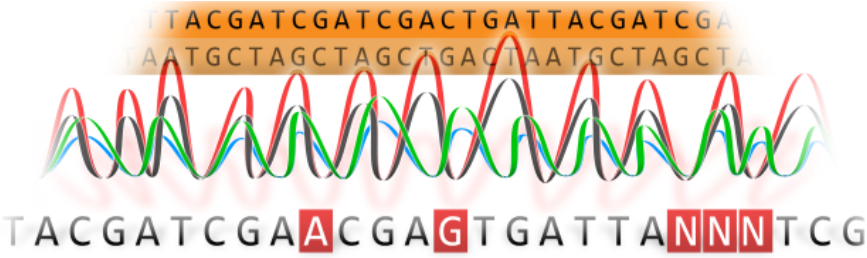
U.PORTO

FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Bioinformatics

Sequence Alignment



Vanda Azevedo 201305778

Introduction

In Bioinformatics, a sequence alignment consists of rearranging sequences of primary structures, such as DNA, RNA or proteins, in order to maximize the similarity score. These similarities can be caused by functional, structural or even evolutionary relationships between them.
Sequence alignments can be divided in two categories:
- Global Alignment: the alignment is "forced" over the full extension of the sequences;
- Local Alignment: the alignment tries to find similarity regions within long sequences that might be overall really different.

The purpose of doing this is to homology: the similarity due to descent from a common ancestor. Often, can infer homology from similarity. If the two sequences originate from individuals with a common ancestor, mismatches can be interpreted as mutations, and gaps can be interpreted as insertions/deletions in one or another sequence that occurred ever since both species diverged in time.

Test Genes

The chosen gene was HERC2 variations for Homo Sapiens (human) and Mus Musculus (domestic rat) species. HERC2 gene belongs to the HERC gene family that encodes unusually large proteins. Genetic variations are associated with skin/hair/eye pigmentation variability.

Test Proteins

The chosen protein family was insulin-like growth factor-binding proteins (IGFBP) for Homo Sapiens (human), IGFBP-4, and Rattus Norvegicus (rat), IGFBP-5, species. "The IGFBPs help to lengthen the half-life of circulating insulin-like growth factors (IGFs) in all tissues, including the prostate. Individual IGFBPs may act to enhance or attenuate IGF signaling depending on their physiological context (i.e. cell type). Even with these similarities, some characteristics are different: chromosomal location, heparin binding domains, RGD recognition site, preference for binding IGF-I or IGF-II, and glycosylation and phosphorylation differences. These structural differences can have a tremendous impact on how the IGFBPs interact with cellular basement membranes."

Sequence Extraction

To extract gene sequences using gene IDs, I use the website https://www.ebi.ac.uk/ena as base.
To extract protein sequences using protein accession codes, I use the website www.ebi.ac.uk/proteins/api as base.
The returned info comes in fasta format. However, I format it in order to obtain the sequence alone. This way it works both with the server and with my algorithms.

Scoring Matrices

PAM and BLOSUM are used as scoring matrices for proteins.

PAM matrices are highly criticized due to its assumption that each aminoacid in a sequence is equally mutable, which isn't true. Another problem is that in the extrapolation of the PAM-1 matrix into higher order PAM-n matrices erros inherent in the PAM-1 matrix data are highly magnified. But the biggest issue is the dataset used to create PAM which basically consisted of globin proteins alone.

BLOSUM was created in order to address the issue of variable amino acid mutation rates within sequences. They are designed to improve the accuracy of alignments between distantly related protein sequences. Multiple alignments of related sequences were made, evolutionary distances were taken into account, etc. This is why BLOSUM is a better alternative to PAM.

For genes, I use the following matrix:

-	A	G	C	T
A	10	-1	-3	-4
G	-1	7	-5	-3
C	-3	-5	9	0
T	-4	-3	0	8

Server Connection

Using tools from the website https://www.ebi.ac.uk/ we are able to redeem a sequence and perform sequence alignment on both gene and protein sequences, while having the option to adjust the algorithms' parameters.
The algorithms used using this resource are:

Needleman -Wunsch

Global alignment algorithm. Given a sequence of n characters A, and a sequence of m characters B, we build a matrix F with dimensions (n+1)*(m+1) that stores the edit distances of the sequences. It uses a scoring matrix S to attribute a value to each pair.

Knowing g is the gap cost, the principle of optimality is then applied as follows:

Initialization: F(0,j) = g*j, F(i,0) = g*i

Recursion:

$$F(i,j) = \max \begin{cases} F(i-1,j-1) + s(x_i, y_j) & \text{match } x_i \text{ with } y_j \\ F(i-1,j) + g & \text{insertion in } x \\ F(i,j-1) + g & \text{insertion in } y \end{cases}$$

For each $F_{i,j}$ we save the pointer(s) to the cell(s) that resulted in the best score.

To then obtain the optimal result, we take it from F(n,m) and follow the pointers until we reach F(0,0).

Smith-Waterman

A local alignment algorithm. Given a sequence of n characters A, and a sequence of m characters B, we build a matrix F with dimensions (n+1)*(m+1) that stores the edit distances of the sequences. It uses a scoring matrix S to attribute a value to each pair.
The principle of optimality is then applied as follows:

Initialization: F(0,j) = 0, F(i,0) = 0

Recursion:

$$F(i,j) = \max \begin{cases} F(i-1,j-1) + s(x_i, y_j) \\ F(i-1,j) + g \\ F(i,j-1) + g \\ 0 \end{cases}$$

For each F(i,j) we save the pointer(s) to the cell(s) that resulted in the best score.

To then obtain the optimal result, we have to find F(i,j) with the maximum value (that can be anywhere in the matrix) and follow the pointers until we reach some cell with value 0.

Implementation and Alignment Cost

The algorithm takes into account affine cost. A gap of length k is more probable than k gaps of length 1; a gap may be due to a single mutational event while separated gaps are probably due to distinct mutational events. A linear gap penalty function treats these cases the same, so, in order to implement affine function, other than the 3 matrices strategy, I've h as a penalty associated with opening a gap, and g as a smaller penalty for extending the gap.

Global Algorithm

Initialization: $I_x(i,0) = h + g*i$
$M(i,0) = -inf$
$I_y(i,0) = -inf$
$I_y(0,j) = h + g*j$
$M(0,j) = -inf$
$I_x(0,j) = -inf$
$M(0,0) = 0$
$I_x(0,0) = h$
$I_y(0,0) = h$

Recursion:

$$M(i,j) = \max \begin{cases} M(i-1,j-1) + s(x_i, y_j) & \text{match } x_i \text{ with } y_j \\ I_x(i-1,j-1) + s(x_i, y_j) & \text{insertion in } x \\ I_y(i-1,j-1) + s(x_i, y_j) & \text{insertion in } y \end{cases}$$

$$I_x(i,j) = \max \begin{cases} M(i-1,j) + h + g & \text{open gap in } x \\ I_x(i-1,j) + g & \text{extend gap in } x \end{cases}$$

$$I_y(i,j) = \max \begin{cases} M(i,j-1) + h + g & \text{open gap in } y \\ I_y(i,j-1) + g & \text{extend gap in } y \end{cases}$$

To then obtain the optimal result, we start at max(M(n,m), I_x(n,m), I_y(n,m)) and stop at any M(0,0), I_x(0,0) or I_y(0,0). To traceback, if the max value in the cell is M(i,j) we came from the diagonal cell, if the max value in the cell is I_x(i,j) we came from the top cell if the max value in the cell is I_y(i,j) we came from the left cell. It is possible, that we have ties, in this case, we have different paths.

Local Algorithm

Initialization: M(i,0) = 0
M(0,j) = 0
I_x(i,0) = -inf
I_y(i,0) = -inf
I_x(0,j) = -inf
I_y(0,j) = -inf

Recursion:

$$M(i,j) = \max \begin{cases} M(i-1,j-1) + s(x_i, y_j) \\ I_x(i-1,j-1) + s(x_i, y_j) \\ I_y(i-1,j-1) + s(x_i, y_j) \\ 0 \end{cases}$$

$$I_x(i,j) = \max \begin{cases} M(i-1,j) + h + g \\ I_x(i-1,j) + g \end{cases}$$

$$I_y(i,j) = \max \begin{cases} M(i,j-1) + h + g \\ I_y(i,j-1) + g \end{cases}$$

To then obtain the optimal result, we have to find the largest M(i,j) (that can be anywhere in the matrix) and stop when we reach some cell with M(i,j)=0. To traceback, if the max value in the cell is M(i,j) we came from the diagonal cell, if the max value in the cell is I_x(i,j) we came from the top cell if the max value in the cell is I_y(i,j) we came from the left cell. It is possible, that we have ties, in this case, we have different paths.

Results

Starting Menu

In here you can choose the default genes and proteins, insert your own, or submit a sequence. All of these are implemented using both the server and the implemented algorithms.

```
Select one of the options:

1) Default GENES: AAO27473 and OWJ99766
2) Default PROTEINS: P17936 and P16611
3) Insert GENE accession codes
4) Insert PROTEIN accession codes
5) Insert GENE sequence
6) Insert PROTEIN sequence
7) EXIT
```

To demonstrate, I've picked the default proteins. Now it asks to insert the scoring matrix we want to use, as well as the penalty associated with opening a gap (gapopen), and the smaller penalty for extending the gap (gapextend).

```
Insert the following parameters INLINE:
 matrix [EBLOSUM$, EPAM$]
 gapopen [1, 5, 10, 15, 20, 25, 50, 100]
 gapextend [0.0005, 0.001, 0.05, 0.1, 0.2, 0.5, 0.6, 0.8, 1.0, 5.0, 10.0]
```

For this example, I've introduced "EBLOSUM62 10 5". Notice that if the values aren't inserted inline or if they don't exist in the server, only the content of the server error will be returned. Meanwhile, my program takes these values and makes them its own in order to later run my implementation using them. For proteins, if the server uses any BLOSUM, my algorithm will use BLOSUM62, if the server uses any PAM, my algorithm will use PAM120.

Needleman-Wunsch

```
#######################################
# Program: needle
# Rundate: Tue 10 Apr 2018 22:15:00
# Commandline: needle
#    -auto
#    -stdout
#    -asequence emboss_needle-R20180410-221458-0240-27073313-p1m.asequence
#    -bsequence emboss_needle-R20180410-221458-0240-27073313-p1m.bsequence
#    -datafile EBLOSUM62
#    -gapopen 10.0
#    -gapextend 5.0
#    -aformat3 pair
#    -sprotein1
#    -sprotein2
# Align_format: pair
# Report_file: stdout
#######################################

#=======================================
#
# Aligned_sequences: 2
# 1: EMBOSS_001
# 2: EMBOSS_001
# Matrix: EBLOSUM62
# Gap_penalty: 10.0
# Extend_penalty: 5.0
#
# Length: 275
# Identity:     106/275 (38.5%)
# Similarity:   138/275 (50.2%)
# Gaps:          21/275 ( 7.6%)
# Score: 398.0
#
#
#=======================================

EMBOSS_001         1 MLPLCLVAALLLAAGPGPSLG-DEAIHCPPCSEEKLARCRP-PVGCEELV     48
                       .:..|..|||||...|:.| ...:||.||.|:.|:.|.| |:|| |||
EMBOSS_001         1 --MVISVVLLLLAACAVPAQGLGSFVHCEPCDEKALSMCPPSPLGC-ELV     47

EMBOSS_001        49 REPGCGCCATCALGLGMPCGVYTPRCGSGLRCYPPRGVEKPLHTLMHGQG     98
                     :|||||||.||||..|..|||||.||..||||.|.:..|||||.|:||:|
EMBOSS_001        48 KEPGCGCCMTCALAEGQSCGVYTERCAQGLRCLPRQDEEKPLHALLHGRG     97

EMBOSS_001        99 VCM---ELAEIEAIQ-ESLQPSDKDEGDHPNNSFSP---CSAHDRRCLQK    141
                     ||:   ...|...|: :|.:..:....:....::||   ...|.|....|
EMBOSS_001        98 VCLNEKSYGEQTKIERDSREHEEPTTSEMAEETYSPKVFRPKHTRISELK    147

EMBOSS_001       142 HFAKIRDRSTSGGKMKVNGAPREDARP--VPQGSCQSELHRA-LER-LAA    187
                     ..|..:||.....:.|..|.....|.|  :|....:.|..:. ..| :.|
EMBOSS_001       148 AEAVKKDRRKKLTQSKFVGGAENTAHPRVIPAPEMRQESDQGPCRRHMEA    197

EMBOSS_001       188 SQSRTHEDLYIIP----IPNCDRNGNFHPKQCHPALDGQRGKCWCVDRKT    233
                     |.........::|    :|||||.|.:..|||.|:...:||.||||| |.
EMBOSS_001       198 SLQEFKASPRMVPRAVYLPNCDRKGFYKRKQCKPSRGRKRGICWCVD-KY    246

EMBOSS_001       234 GVKLPGGLEPKGELDCHQLADSFRE    258
                     |:||||.....|:..||....|..|
EMBOSS_001       247 GMKLPGMEYVDGDFQCHAFDSSNVE    271


#---------------------------------------
#---------------------------------------
```

Smith-Waterman

```
#######################################
# Program: water
# Rundate: Tue 10 Apr 2018 22:15:06
# Commandline: water
#    -auto
#    -stdout
#    -asequence emboss_water-R20180410-221504-0869-75867616-p2m.asequence
#    -bsequence emboss_water-R20180410-221504-0869-75867616-p2m.bsequence
#    -datafile EBLOSUM62
#    -gapopen 10.0
#    -gapextend 5.0
#    -aformat3 pair
#    -sprotein1
#    -sprotein2
# Align_format: pair
# Report_file: stdout
#######################################

#=======================================
#
# Aligned_sequences: 2
# 1: EMBOSS_001
# 2: EMBOSS_001
# Matrix: EBLOSUM62
# Gap_penalty: 10.0
# Extend_penalty: 5.0
#
# Length: 266
# Identity:     105/266 (39.5%)
# Similarity:   136/266 (51.1%)
# Gaps:          19/266 ( 7.1%)
# Score: 403.0
#
#
#=======================================

EMBOSS_001         7 VAALLLAAGPGPSLG-DEAIHCPPCSEEKLARCRP-PVGCEELVREPGCG     54
                     |..|||||...|:.| ...:||.||.|:.|:.|.| |:|| |||:|||||
EMBOSS_001         5 VVLLLLAACAVPAQGLGSFVHCEPCDEKALSMCPPSPLGC-ELVKEPGCG     53

EMBOSS_001        55 CCATCALGLGMPCGVYTPRCGSGLRCYPPRGVEKPLHTLMHGQGVCM---    101
                     ||.||||..|..|||||.||..||||.|.:..|||||.|:||:|||:
EMBOSS_001        54 CCMTCALAEGQSCGVYTERCAQGLRCLPRQDEEKPLHALLHGRGVCLNEK    103

EMBOSS_001       102 ELAEIEAIQ-ESLQPSDKDEGDHPNNSFSP---CSAHDRRCLQKHFAKIR    147
                     ...|...|: :|.:..:....:....::||   ...|.|....|..|..:
EMBOSS_001       104 SYGEQTKIERDSREHEEPTTSEMAEETYSPKVFRPKHTRISELKAEAVKK    153

EMBOSS_001       148 DRSTSGGKMKVNGAPREDARP--VPQGSCQSELHRA-LER-LAASQSRTH    193
                     ||.....:.|..|.....|.|  :|....:.|..:. ..| :.||.....
EMBOSS_001       154 DRRKKLTQSKFVGGAENTAHPRVIPAPEMRQESDQGPCRRHMEASLQEFK    203

EMBOSS_001       194 EDLYIIP----IPNCDRNGNFHPKQCHPALDGQRGKCWCVDRKTGVKLPG    239
                     ....::|    :|||||.|.:..|||.|:...:||.||||| |.|:||||
EMBOSS_001       204 ASPRMVPRAVYLPNCDRKGFYKRKQCKPSRGRKRGICWCVD-KYGMKLPG    252

EMBOSS_001       240 GLEPKGELDCHQLADS    255
                     .....|:..||....|
EMBOSS_001       253 MEYVDGDFQCHAFDSS    268


#---------------------------------------
#---------------------------------------
```

Implementations

My implementation takes different paths. However, for larger sequences (1500+), I've had time-related trouble. So, in order to solve this, I've narrowed down 1000 different optimal paths. This might leave out the possibility with the most matches. My approach will also only print the top 5 paths regarding identity.

Global

```
=================================================================================

 [GLOBAL] Algorithm

=================================================================================


# ================================================
#
# [ GLOBAL ]
# Length =  274
# Identity =  107 / 274  (39.1 %)
# Gaps =  19 / 274  (6.9 %)
# Score =  174
#
# ================================================

1 - 75
MLPLCLVAALLLAAGPGPSLGDEAIHCPPCSEEKLARCRP-PVGCEELVREPGCGCCATCALGLGMPCGVYTPRC
|::::::::|:::|:|:::|| :::|:|:|:|::|::|:| |:|| |||:|||||||:||||::|::||||||:||
MVISVVLLLLAACAVPAQGLG-SFVHCEPCDEKALSMCPPSPLGC-ELVKEPGCGCCMTCALAEGQSCGVYTERC

76 - 150
GSGLRCYPPRGVEKPLHTLMHGQGVCM------ELAEIEAIQESLQPSDKDEGDHPNNSFS-PCSAHDRRCLQKH
::||||:|::::|||||:|:||:|||:      |:::||:::::::::::|::::::|:: ::::|:|::::|:
AQGLRCLPRQDEEKPLHALLHGRGVCLNEKSYGEQTKIERDSREHEEPTTSEMAEETYSPKVFRPKHTRISELKA

151 - 225
FAKIRDRSTSGGKMK-VNG----A-PREDARP-VPQGSCQSELHRALER-LAASQSRTHEDLYIIPIPNCDRNGN
:|:::||::::::: |:|    | ||::::| ::|:|:|::::|::|: |::::::::::::::::|||||:|:
EAVKKDRRKKLTQSKFVGGAENTAHPRVIPAPEMRQESDQGPCRRHMEASLQEFKASPRMVPRAVYLPNCDRKGF

226 - 274
FHPKQCHPALDGQRGKCWCVDRKTGVKLPGGLEPKGELDCHQLADSFRE
:::|||:|::::::||:||||| |:|:||||:::::|:::||::::|::|
YKRKQCKPSRGRKRGICWCVD-KYGMKLPGMEYVDGDFQCHAFDSSNVE


# ================================================
#
# [ GLOBAL ]
# Length =  274
# Identity =  107 / 274  (39.1 %)
# Gaps =  19 / 274  (6.9 %)
# Score =  174
#
# ================================================

1 - 75
MLPLCLVAALLLAAGPGPSLGDEAIHCPPCSEEKLARCRP-PVGCEELVREPGCGCCATCALGLGMPCGVYTPRC
|::::::::|:::|:|:::|| :::|:|:|:|::|::|:| |:|| |||:|||||||:||||::|::||||||:||
MVISVVLLLLAACAVPAQGLG-SFVHCEPCDEKALSMCPPSPLGC-ELVKEPGCGCCMTCALAEGQSCGVYTERC

76 - 150
GSGLRCYPPRGVEKPLHTLMHGQGVCM------ELAEIEAIQESLQPSDKDEGDHPNNSFS-PCSAHDRRCLQKH
::||||:|::::|||||:|:||:|||:      |:::||:::::::::::|::::::|:: ::::|:|::::|:
AQGLRCLPRQDEEKPLHALLHGRGVCLNEKSYGEQTKIERDSREHEEPTTSEMAEETYSPKVFRPKHTRISELKA

151 - 225
FAKIRDRSTSGGKMK-VNG----A-PREDARP-VPQGSCQSELHRALER-LAASQSRTHEDLYIIPIPNCDRNGN
:|:::||::::::: |:|    | ||::::| ::|:|:|::::|::|: |::::::::::::::::|||||:|:
EAVKKDRRKKLTQSKFVGGAENTAHPRVIPAPEMRQESDQGPCRRHMEASLQEFKASPRMVPRAVYLPNCDRKGF

226 - 274
FHPKQCHPALDGQRGKCWCVDRKTGVKLPGGLEPKGELDCHQLADSFRE
:::|||:|::::::||:||||| |:|:||||:::::|:::||::::|::|
YKRKQCKPSRGRKRGICWCVD-KYGMKLPGMEYVDGDFQCHAFDSSNVE
```

```
# ==================================================
#
# [ GLOBAL ]
# Length =  274
# Identity =  107 / 274  (39.1 %)
# Gaps =  19 / 274  (6.9 %)
# Score =  174
#
# ==================================================

1 - 75
MLPLCLVAALLLAAGPGPSLGDEAIHCPPCSEEKLARCRP-PVGCEELVREPGCGCCATCALGLGMPCGVYTPRC
|::::::::|:::|:|:::||:: :||:|||:|::|::|:| |:|| |||:|||||||:||||::|::||||||:||
MVISVVLLLLAACAVPAQGLGSF-VHCEPCDEKALSMCPPSPLGC-ELVKEPGCGCCMTCALAEGQSCGVYTERC

76 - 150
GSGLRCYPPRGVEKPLHTLMHGQGVCM------ELAEIEAIQESLQPSDKDEGDHPNNSFS-PCSAHDRRCLQKH
::||||:|::::|||||:|:||:|||:      |:::||:::::::::::|::::::|:: ::::|:|::::|:
AQGLRCLPRQDEEKPLHALLHGRGVCLNEKSYGEQTKIERDSREHEEPTTSEMAEETYSPKVFRPKHTRISELKA

151 - 225
FAKIRDRSTSGGKMK-VNG----A-PREDARP-VPQGSCQSELHRALER-LAASQSRTHEDLYIIPIPNCDRNGN
:|:::||::::::: |:|    | ||::::| ::|:|:|::::|::|: |::::::::::::::::|||||:|:
EAVKKDRRKKLTQSKFVGGAENTAHPRVIPAPEMRQESDQGPCRRHMEASLQEFKASPRMVPRAVYLPNCDRKGF

226 - 274
FHPKQCHPALDGQRGKCWCVDRKTGVKLPGGLEPKGELDCHQLADSFRE
:::|||:|:::::||:||||| |:|:||||:::::|:::||::::|::|
YKRKQCKPSRGRKRGICWCVD-KYGMKLPGMEYVDGDFQCHAFDSSNVE


# ==================================================
#
# [ GLOBAL ]
# Length =  274
# Identity =  107 / 274  (39.1 %)
# Gaps =  19 / 274  (6.9 %)
# Score =  174
#
# ==================================================

1 - 75
MLPLCLVAALLLAAGPGPSLGDEAIHCPPCSEEKLARCRP-PVGCEELVREPGCGCCATCALGLGMPCGVYTPRC
|::::::::|:::|:|:::||::: ||:|||:|::|::|:| |:|| |||:|||||||:||||::|::||||||:||
MVISVVLLLLAACAVPAQGLGSFV-HCEPCDEKALSMCPPSPLGC-ELVKEPGCGCCMTCALAEGQSCGVYTERC

76 - 150
GSGLRCYPPRGVEKPLHTLMHGQGVCM------ELAEIEAIQESLQPSDKDEGDHPNNSFS-PCSAHDRRCLQKH
::||||:|::::|||||:|:||:|||:      |:::||:::::::::::|::::::|:: ::::|:|::::|:
AQGLRCLPRQDEEKPLHALLHGRGVCLNEKSYGEQTKIERDSREHEEPTTSEMAEETYSPKVFRPKHTRISELKA

151 - 225
FAKIRDRSTSGGKMK-VNG----A-PREDARP-VPQGSCQSELHRALER-LAASQSRTHEDLYIIPIPNCDRNGN
:|:::||::::::: |:|    | ||::::| ::|:|:|::::|::|: |::::::::::::::::|||||:|:
EAVKKDRRKKLTQSKFVGGAENTAHPRVIPAPEMRQESDQGPCRRHMEASLQEFKASPRMVPRAVYLPNCDRKGF

226 - 274
FHPKQCHPALDGQRGKCWCVDRKTGVKLPGGLEPKGELDCHQLADSFRE
:::|||:|:::::||:||||| |:|:||||:::::|:::||::::|::|
YKRKQCKPSRGRKRGICWCVD-KYGMKLPGMEYVDGDFQCHAFDSSNVE


# ==================================================
#
# [ GLOBAL ]
# Length =  274
# Identity =  107 / 274  (39.1 %)
# Gaps =  19 / 274  (6.9 %)
# Score =  174
#
# ==================================================

1 - 75
MLPLCLVAALLLAAGPGPSLGDEAIHCPPCSEEKLARCRP-PVGCEELVREPGCGCCATCALGLGMPCGVYTPRC
|::::::::|:::|:|:::|| :::||:|||:|::|::|:| |:||| ||:|||||||:||||::|::||||||:||
MVISVVLLLLAACAVPAQGLG-SFVHCEPCDEKALSMCPPSPLGCE-LVKEPGCGCCMTCALAEGQSCGVYTERC

76 - 150
GSGLRCYPPRGVEKPLHTLMHGQGVCM------ELAEIEAIQESLQPSDKDEGDHPNNSFS-PCSAHDRRCLQKH
::||||:|::::|||||:|:||:|||:      |:::||:::::::::::|::::::|:: ::::|:|::::|:
AQGLRCLPRQDEEKPLHALLHGRGVCLNEKSYGEQTKIERDSREHEEPTTSEMAEETYSPKVFRPKHTRISELKA

151 - 225
FAKIRDRSTSGGKMK-VNG----A-PREDARP-VPQGSCQSELHRALER-LAASQSRTHEDLYIIPIPNCDRNGN
:|:::||::::::: |:|    | ||::::| ::|:|:|::::|::|: |::::::::::::::::|||||:|:
EAVKKDRRKKLTQSKFVGGAENTAHPRVIPAPEMRQESDQGPCRRHMEASLQEFKASPRMVPRAVYLPNCDRKGF

226 - 274
FHPKQCHPALDGQRGKCWCVDRKTGVKLPGGLEPKGELDCHQLADSFRE
:::|||:|:::::||:||||| |:|:||||:::::|:::||::::|::|
YKRKQCKPSRGRKRGICWCVD-KYGMKLPGMEYVDGDFQCHAFDSSNVE
```

Local

```
=================================================================================

 [LOCAL] Algorithm

=================================================================================


# ================================================
#
# [ LOCAL ]
# Length =  77
# Identity =  49 / 77  (63.6 %)
# Gaps =  2 / 77  (2.6 %)
# Score =  266
#
# ================================================

1 - 75
HCPPCSEEKLARCRP-PVGCEELVREPGCGCCATCALGLGMPCGVYTPRCGSGLRCYPPRGVEKPLHTLMHGQGV
||:||:|::|::|:| |:|| |||:|||||||:||||::|::||||||:||::||||:|::::||||||:|:||:||
HCEPCDEKALSMCPPSPLGC-ELVKEPGCGCCMTCALAEGQSCGVYTERCAQGLRCLPRQDEEKPLHALLHGRGV

76 - 77
CM
|:
CL


# ================================================
#
# [ LOCAL ]
# Length =  77
# Identity =  49 / 77  (63.6 %)
# Gaps =  2 / 77  (2.6 %)
# Score =  266
#
# ================================================

1 - 75
HCPPCSEEKLARCRP-PVGCEELVREPGCGCCATCALGLGMPCGVYTPRCGSGLRCYPPRGVEKPLHTLMHGQGV
||:||:|::|::|:| |:||| ||:|||||||:||||::|::||||||:||::||||:|::::||||||:|:||:||
HCEPCDEKALSMCPPSPLGCE-LVKEPGCGCCMTCALAEGQSCGVYTERCAQGLRCLPRQDEEKPLHALLHGRGV

76 - 77
CM
|:
CL


# ================================================
#
# [ LOCAL ]
# Length =  78
# Identity =  49 / 78  (62.8 %)
# Gaps =  2 / 78  (2.6 %)
# Score =  266
#
# ================================================

1 - 75
HCPPCSEEKLARCRP-PVGCEELVREPGCGCCATCALGLGMPCGVYTPRCGSGLRCYPPRGVEKPLHTLMHGQGV
||:||:|::|::|:| |:|| |||:|||||||:||||::|::||||||:||::||||:|::::||||||:|:||:||
HCEPCDEKALSMCPPSPLGC-ELVKEPGCGCCMTCALAEGQSCGVYTERCAQGLRCLPRQDEEKPLHALLHGRGV

76 - 78
CME
|::
CLN


# ================================================
#
# [ LOCAL ]
# Length =  78
# Identity =  49 / 78  (62.8 %)
# Gaps =  2 / 78  (2.6 %)
# Score =  266
#
# ================================================

1 - 75
HCPPCSEEKLARCRP-PVGCEELVREPGCGCCATCALGLGMPCGVYTPRCGSGLRCYPPRGVEKPLHTLMHGQGV
||:||:|::|::|:| |:||| ||:|||||||:||||::|::||||||:||::||||:|::::||||||:|:||:||
HCEPCDEKALSMCPPSPLGCE-LVKEPGCGCCMTCALAEGQSCGVYTERCAQGLRCLPRQDEEKPLHALLHGRGV

76 - 78
CME
|::
CLN
```

Identity Scores

The target identity for BLOSUM62 is around 28.9%. So, we can affirm that the sequences are homologous since they've passed this mark for every algorithm.

Performance

My implementation performed better in this case, both for global and local. However, it isn't always the case. The differences aren't too big either.

References

https://www.ebi.ac.uk/interpro/entry/IPR000867
https://en.wikipedia.org/wiki/Needleman%E2%80%93Wunsch_algorithm
https://en.wikipedia.org/wiki/Smith%E2%80%93Waterman_algorithm
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3848038/
Slides from class