

Needleman-Wunsch Algorithm

Overview

	S	E	Q	U	E	N	C	E	1
S	0								
E									
Q									
E									
N								$S_{i-1,j-1}$	$S_{i-1,j}$
2								$S_{i,j-1}$	$S_{i,j}$

First row initialized with gap value

First col. initialized with gap value

Cell score derives from

$$\max \begin{cases} S_{i-1,j} + gap \\ S_{i,j-1} + gap \\ S_{i-1,j-1} + match/mismatch \end{cases}$$

TRACEBACK: computed starting from last cell and tracing back the ones that had max scores after calculations

DNA R Optimal Global Alignment

INPUTS

- ❖ Two strings
- ❖ Scoring system for match/mismatch/gap

OUTPUTS

- ❖ Console: all possible alignments
- ❖ Text file: Summary about alignment score, inputs details and all possible alignments

```
G_align <- function(seq1, seq2, match, mismatch, gap, plot = 'no') {
```

Block 1: matrix computing

```
  res <- Nw_scores(seq1, seq2, match, mismatch, gap) # returns score and movement matrices
```

```
  if (file.exists('tempFile_Aln.txt')) { # delete (permanently) the file if exists
    file.remove('tempFile_Aln.txt')
  }
```

Block 2: find all possible paths

```
  moves <- threading(res$movements)
```

```
  directions <- list(t(read.table('tempFile_Aln.txt')))
```

```
  global_score <- res$score[length(res$score[,1]), length(res$score[1,])] # last cell of score matrix
  pox_aligns <- length(directions[[1]])
```

```
  write.table(matrix(c(length(res$score[,1]), # initialize output file
                        length(res$score[1,]), # → aligned sequences will be appended here
                        global_score,
                        pox_aligns,
                        ''),
                nrow = 5,
                dimnames = list(c('Sequence 1 length:',
                                   'Sequence 2 length:',
                                   'Global Alignment Score:',
                                   'Possible alignments:',
                                   'SUMMARY')),
                file = 'Test_output_final.txt',
                row.names = T,
                col.names = T,
                append = F)
```

```
  mclapply(directions[[1]],
            FUN = 'nw_traceback')
```

Block 3: translate paths in alignments

```
  if(plot == 'yes') {
    par(mar = c(3.5,3.5,3.5,3.5))
```

```
    break1 <- min(res$scores)
    break2 <- max(res$scores)
    break3 <- (min(res$scores)+max(res$scores))/2
    break4 <- max(res$scores)-1
    sort(c(break1, break2, break3, break4, 1, 0)) # compute the plot breaks
                                                    # to point out alignment direction
```

```
    plot(res$scores,
          breaks = sort(c(break1, break2, break3, break4, 0)),
          main = 'Alignment visualization',
          xlab = 'sequence 1',
          ylab = 'sequence 2')
```

Block 4: optional visual output for the score matrix

```
  }
```

Block 1

INPUTS

sequences and scoring system



Creates two matrices, to store scores and the other for directions



Fills them



OUTPUTS

A list containig the computed matrices

Score matrix

		A	C	G	G	C	A	T	A	A
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
A	-1	1	0	-1	-2	-3	-4	-5	-6	-7
C	-2	0	2	1	0	-1	-2	-3	-4	-5
T	-3	-1	1	1	0	-1	-2	-1	-2	-3
G	-4	-2	0	2	2	1	0	-1	-2	-3
A	-5	-3	-1	1	1	1	2	1	0	-1
C	-6	-4	-2	0	0	2	1	1	0	-1
A	-7	-5	-3	-1	-1	1	3	2	2	1

Directions matrix

		A	C	G	G	C	A	T	A	A
	"0"	"H"	"H"	"H"	"H"	"H"	"H"	"H"	"H"	"H"
A	"V"	"D"	"H"	"H"	"H"	"H"	"DH"	"H"	"DH"	"DH"
C	"V"	"V"	"D"	"H"	"H"	"DH"	"H"	"H"	"H"	"H"
T	"V"	"V"	"V"	"D"	"DH"	"DH"	"DH"	"D"	"H"	"H"
G	"V"	"V"	"V"	"D"	"D"	"H"	"H"	"H"	"DH"	"DH"
A	"V"	"DV"	"V"	"V"	"DV"	"D"	"D"	"H"	"DH"	"DH"
C	"V"	"V"	"DV"	"V"	"DV"	"D"	"VH"	"D"	"DH"	"DH"
A	"V"	"DV"	"V"	"V"	"DV"	"V"	"D"	"H"	"D"	"DH"

Block 2

INPUTS

Directions matrix



Traces back all possible paths starting from bottom left cell



OUTPUTS

A temp file containig all possible paths

Ex. of output paths

[1]	"DHHHDDVDDVVD"
[1]	"DHHHDDVDVDVD"
[1]	"DHHHDDVDVVDD"
[1]	"HDHHDDVDDVVD"
[1]	"HDHHDDVDVDVD"
[1]	"HDHHDDVDVVDD"
[1]	"HHDHDDVDDVVD"
[1]	"HHDHDDVDVDVD"
[1]	"HHDHDDVDVVDD"
[1]	"HHHDDD VDDVVD"
[1]	"HHHDDD VD VD VD"

Thanks to @jeff (stackoverflow user) for pointing out using a recursive threading function to solve the task

Block 3

INPUTS

The generated temp file



OUTPUTS

Console return of all alignments

Textfile return with summary+ alignments

```
"SUMMARY"
"Sequence 1 length:" "11"
"Sequence 2 length:" "10"
"Global Alignment Score:" "3"
"Possible alignments:" "15"
" _____ " ""
"Possible alignment"
"seq1" " A A T C G A C - A A A"
"seq2" " A A - C - C C G A A A"
" _____ " ""
"Possible alignment"
"seq1" " A A T C G A C - A A A"
"seq2" " A A - C C - C G A A A"
" _____ " ""
"Possible alignment"
"seq1" " A A T C G A C - A A A"
"seq2" " A A C C - - C G A A A"
" _____ " ""
"Possible alignment"
"seq1" " A A - T C G A C A A A"
"seq2" " A A C C C G - - A A A"
```

Block 4

INPUTS

The sequences to align



OUTPUTS

Alignment visualization through plotting the scoring matrix (sort of a heatmap)

Alignment visualization

