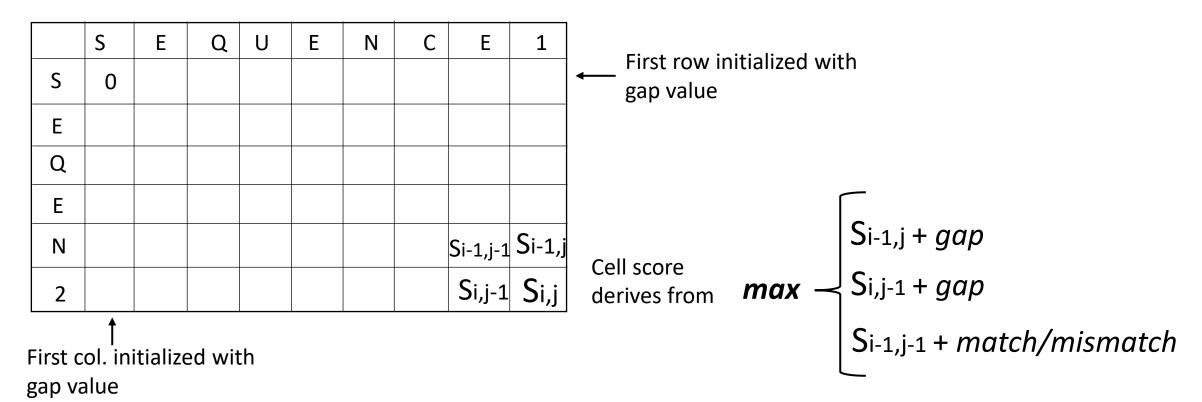
Needleman-Wunsch Algorithm

Overview



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

if (file.exists('tempFile_Aln.txt')) { # delete (permanetly) the file if exists

file.remove('tempFile_Aln.txt')

break4 ← max(res\$scores)-1

xlab = 'sequence 1',

ylab = 'sequence 2')

plot(res\$scores,

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 \leftarrow length(directions[[1]])
write.table(matrix(c(length(res$score[1,]), # initialize output file
                     length(res\$score[,1]), # \longrightarrow aligned sequences will be appended here
                     global_score,
                     pox_aligns,
                   nrow = 5,
                   dimnames = list(c('Sequence 1 length:',
                                      'Sequence 2 length:',
                                      'Global Alignment Score:',
                                      'Possible alignments:',
                                    'SUMMMARY')),
            file = 'Test_output_final.txt',
            row.names = T.
            col.names = T
            append = F)
mclapply(directions[[1]],
         FUN = 'nw_traceback')
if(plot == 'ves') {
  par(mar = c(3.5, 3.5, 3.5, 3.5))
  break1 ← min(res$scores)
  break2 ← max(res$scores)
  break3 ← (min(res$scores)+max(res$scores))/2
```

to point out alignment direction

sort(c(break1, break2, break3, break4,1, 0)) # compute the plot breaks

breaks = sort(c(break1, break2, break3, break4, 0)),

main = 'Alignment visualization',

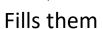
Block 3: translate paths in alignments

Block 4: optional visual output for the score matrix

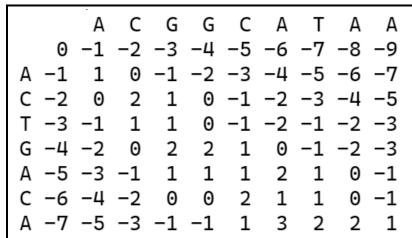
INPUTS

sequences and scoring system

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



Score matrix

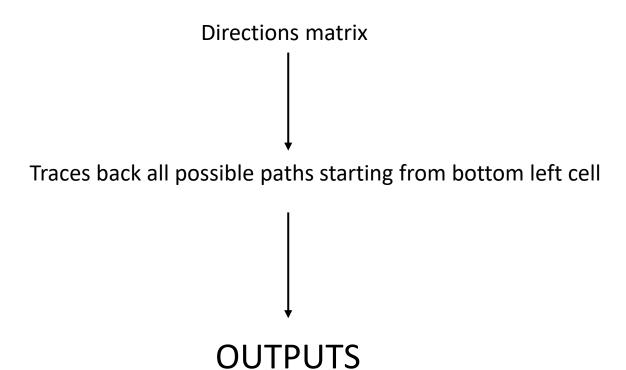


Directions matrix

OUTPUTS

A list containig the computed matrices

INPUTS



Ex. of output paths

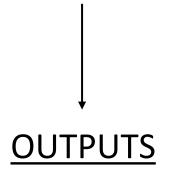
[1]	"DHHHDDVDDVVD"
[1]	"DHHHDDVDVDVD"
[1]	"DHHHDDVDVVDD"
[1]	"HDHHDDVDDVVD"
[1]	"HDHHDDVDVDVD"
[1]	"HDHHDDVDVVDD"
[1]	"HHDHDDVDDVVD"
[1]	"HHDHDDVDVDVD"
[1]	"HHDHDDVDVVDD"
[1]	"HHHDDDVDDVVD"
[1]	"HHHDDDVDVDVD"

A temp file containig all possible paths

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

INPUTS

The generated temp file

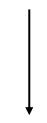


Console return of all alignments
Textfile return with summary+ alignments

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

INPUTS

The sequences to align Scoring matrix



OUTPUTS

Alignment vizualization through plotting the scoring matrix (sort of a heathmap)

Alignment visualization

