

# Gorilla Report

*abax, bemn, diem, emba, mkk, sebn*

October 5, 2020

## Results

Our implementation produces the expected results on all pairs of species. We show an example of generated matchings.

Human--Sea-Cucumber: 80  
---M-V--H--LTPEEKSAVTALWGK-V-NVDEVGGEALGRLLVVY-PWTQRFESFGDLSTPDAVMG  
NPKVKAHGKKVLGAFSDGLAHLN-LKGTFAATLSELHCDKLH-VDPENFRLLGNVLCVLAHHFGKEF  
TPPVQAAYQKVAVGANA-LAHKYH

XGGTLAIQAQGDLTAAQKKIVRKTWHQLMRNKTSFVTDVFIRIF-AYDPSAQNKFPMAGMSA-SQLRS  
SRQMQAHAIRVSSIMSEYVEELSDILPELLATLARTH-D-LNKVGADHYNLFKVLMEALQELGSDF  
NEKTRDAWAKAFS-VVQAVLLVKHG

Gorilla--Sea-Cucumber: 80  
---M-V--H--LTPEEKSAVTALWGK-V-NVDEVGGEALGRLLVVY-PWTQRFESFGDLSTPDAVMG  
NPKVKAHGKKVLGAFSDGLAHLN-LKGTFAATLSELHCDKLH-VDPENFRLLGNVLCVLAHHFGKEF  
TPPVQAAYQKVAVGANA-LAHKYH

XGGTLAIQAQGDLTAAQKKIVRKTWHQLMRNKTSFVTDVFIRIF-AYDPSAQNKFPMAGMSA-SQLRS  
SRQMQAHAIRVSSIMSEYVEELSDILPELLATLARTH-D-LNKVGADHYNLFKVLMEALQELGSDF  
NEKTRDAWAKAFS-VVQAVLLVKHG

When using the following nucleotide sequence for the rat:

MVHLTDAEKAANALWGKVNPDVGGGEALGRLLVVYPWTQRYFDSFGDLS  
SASAIMGNPKVKAHGKKVINAENDGLKHLNLTGTFATLSELHCDKLHVD  
PENFRLLGNMIVIVLGHHLGKEFTPCAQAQKVVAGVASALAHKYH

We find the closest species to be the Human, with the following score and sequence alignment:

Rat--Human: 653  
MVHLTDAEKAANALWGKVNPDVGGGEALGRLLVVYPWTQRYFDSFGDLSASAIMGNPKVKAHGKKV  
INAENDGLKHLNLTGTFATLSELHCDKLHVDPENFRLLGNMIVIVLGHHLGKEFTPCAQAQKVVAG  
VASALAHKYH

MVHLTPEEKSAVTALWGKVNDEVGGEALGRLLVVYPWTQRFESFGDLSTPDAVMGNPKVKAHGKKV  
LGAFSDGLAHLNLTGTFATLSELHCDKLHVDPENFRLLGNVLCVLAHHFGKEFTPPVQAAYQKVVA  
GVANALAHKYH

## Implementation details

We chose an iterative implementation. It follows the implementation given on p. 356 of KT. However, since the penalties of mismatch and

inserting a space are inverted (compared to what the book expects) we change our decision of what to do (insert both characters or only one) based on what maximizes the penalty rather than what minimizes it. The decision therefore looks like the following:

```
A[i][j] = max(  
    a[x[i - 1]][y[j - 1]] + A[i-1][j-1], // adding character from both sequences  
    d + A[i-1][j],    // adding a dash to the y sequence  
    d + A[i][j-1])    // adding a dash to the x sequence
```

It is apparent from the implementation of `opt` that we only use  $O(mn)$  space and time based on our construction of the memory (A) and the iteration over this.

The *trace* function is then used to find the resulting optimal alignment and then the result is printed (this iterates over the memory A). The traceback runs in  $O(n + m)$ , since in each step it decreases the character index that is inspected by one on either of the strings.