

smith_waterman

March 22, 2022

```
[1]: !pip install numpy
import numpy as np
```

Requirement already satisfied: numpy in /opt/conda/lib/python3.9/site-packages (1.21.5)

```
[2]: def parse_sample(sample_name: str) -> str:
      sample = open(f"samples/{sample_name}.txt")
      parsed_sample = sample.read().replace("\n", "")
      return parsed_sample
```

0.1 Questão 3-a) Implemente o algoritmo de Smith-Waterman para alinhamento Local

```
[3]: def smith_waterman(sample1, sample2, match=2, mismatch=-2, gap=-3) ->
      tuple[int, np.ndarray]:
      matrix = np.zeros([len(sample2)+1, len(sample1)+1], int)
      for i in range(1, matrix.shape[0]):
          for j in range(1, matrix.shape[1]):
              matrix[i, j] = max( 0,
                                  matrix[i-1, j-1] +
                                  (match if sample2[i-1] == sample1[j-1]
                                   else mismatch),
                                  matrix[i-1, j] + gap,
                                  matrix[i, j-1] + gap
                                  )

      return matrix.max(), matrix
```

0.2 Backtracking

```
[4]: def get_backtracking(sample1: list[str], sample: list[str], matrix: np.ndarray)
      -> list[list[str]]:
      sample1_aligned = []
      sample2_aligned = []
      match_mismatch = []
```

```

i, j = i, j = np.unravel_index(matrix.argmax(), matrix.shape)
direcao = []
while True:
    if matrix[i, j] == 0:
        break
    elif i > 0 and j > 0:
        upper = matrix[i-1, j]
        diagonal = matrix[i-1, j-1]
        left = matrix[i, j-1]

        if sample1[j-1] == sample2[i-1]:
            sample1_aligned.insert(0, sample1[j-1])
            sample2_aligned.insert(0, sample2[i-1])
            match_mismatch.insert(0, "*")

            i -= 1
            j -= 1

            #print(f"caso 0 esquerda:{left} cima:{upper} diagonal:␣
↪{diagonal} direcao: diagonal")

        elif (diagonal >= left and diagonal >= upper):
            sample1_aligned.insert(0, sample1[j-1])
            sample2_aligned.insert(0, sample2[i-1])
            match_mismatch.insert(0, "|")

            i -= 1
            j -= 1

            #print(f"caso 1 esquerda:{left} cima:{upper} diagonal:␣
↪{diagonal} direcao: diagonal")

        elif (left > diagonal and left > upper):
            sample1_aligned.insert(0, sample1[j-1])
            sample2_aligned.insert(0, "-")
            match_mismatch.insert(0, " ")

            j -= 1
            #print(f"caso 2 esquerda:{left} cima:{upper} diagonal:␣
↪{diagonal} direcao: esquerda")

        else:
            sample1_aligned.insert(0, "-")
            sample2_aligned.insert(0, sample2[i-1])
            match_mismatch.insert(0, " ")

```

```

        i -= 1
        direcao.append("U")

        #print(f"caso 3 esquerda:{left} cima:{upper} diagonal:␣
↪{diagonal} direcao: cima")

    else:
        break

return sample1_aligned, match_mismatch, sample2_aligned

```

0.3 Questão 3-b e 3-c

```

[5]: sample1_name = "bacillus_lentus"
sample2_name = "bacillus_halodurans"

sample1 = parse_sample(sample1_name)
sample2 = parse_sample(sample2_name)

score, matrix = smith_waterman(sample1, sample2, gap=-2, match=1, mismatch=-1)

sample1_aligned, match_mismatch, sample2_aligned = get_backtracking(sample1,␣
↪sample2, matrix)

identity = int(match_mismatch.count("*") / len(match_mismatch) * 100)
print(f"Amostra 1: {sample1_name} | Amostra 2: {sample2_name} | Score: {score}␣
↪| Identidade: {identity}%\n")

print(''.join(sample1_aligned), end="\n\n")
print(''.join(match_mismatch), end="\n\n")
print(''.join(sample2_aligned), end="\n\n")

```

Amostra 1: bacillus_lentus | Amostra 2: bacillus_halodurans | Score: 85 |
 Identidade: 67%

AHNRGLTGSQVAVLDTGISTHPDLNIRGGASFVPGEPTQDGNHGHGTHVAGTIAALNNSIGVLGVAPSAELYAVKVLG
 ASGSGSVSSIAQGLEWAGNNGMHVANLSLGSPPSATLEQAVNSATSRGVLVVAASGNSGAGSISYPARYANAMAVGATD
 QNNNRASFSQYGAGLDIVAPGVNVQSTYPGSTYASLNGTSMATPHVAGAAALVKQKNPSWSNVQIRNHLKNTATSLGSTN
 LYGSGLV

```

*****|*|*|*****|****|*|*****||***|*|*****|*****|
|*****|*|***|***|**|**||*|*****|*|***|***|*||*|*|*|*|*|*|*|*****|*****|*|*
**|*****|**||*|*****|***|*||*|**|*****|*****|**||*|*|***|*|*|*|***|***|
***|***

```

AHNRGIFGNGARVAVLDTGIASHPDLRIAGGASFISSEPSYHDNNGHGVAGTIAALNNSIGVLGVAPSADLYAVKVLD
RNGSGSLASVAQGIEWAINNNMHIINMSLGSTSGSSTLELAVNRANNAGILLVGAAGNTGRQGVNYPARYSGVMVAAVD
QNGQRASFSTYGPEIEISAPGVNVNSTYTGNRYVSLSGTSMATPHVAGVAALVKSRYPSYTNQIRQRINQTATYLGSPS
LYGNGLV

[]: