needleman_wunsh

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 1-a) Implemente o algoritmo de Needleman-Wunsch para alinhamento global

[3]:
```
def needleman_wunsh(sample1, sample2, match=2, mismatch=-2, gap=-3) ->␣
 ↪tuple[int, np.ndarray]:

    matrix = np.zeros([len(sample2)+1, len(sample1)+1], int)
    matrix[0, 1:] = [gap * (idx+1) for idx, _ in enumerate(sample1)]
    matrix[1:, 0] = [gap * (idx+1) for idx, _ in enumerate(sample2)]
    for i in range(1, matrix.shape[0]):
        for j in range(1, matrix.shape[1]):
            matrix[i, j] = max(
                                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[-1, -1], 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 = matrix.shape
    i -= 1
    j -= 1
    direcao = []
    while True:
        #print(f"amostra1: {sample1[j-1]} amostra2: {sample2[i-1]}", end="    ")
        if 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")

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

                j -= 1
                direcao.append("L")

                #print(f"caso 4 esquerda:{left}  direcao: esquerda")

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

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

            #print(f"caso 5 cima:{upper} direcao: cima")

        else:
            break


    return sample1_aligned, match_mismatch, sample2_aligned
```

0.3 Questão 1-b e 1-c

[5]:
```
seqs = [("korea", "porto_rico"),
        ("korea", "guangdong"),
        ("porto_rico", "guangdong")]
for sample1_name, sample2_name in seqs:
```

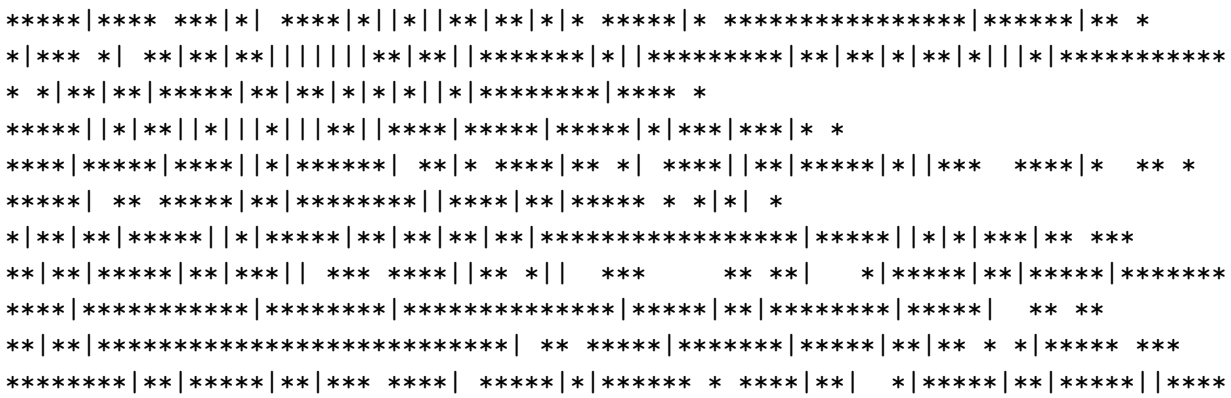
```
    sample1 = parse_sample(sample1_name)
    sample2 = parse_sample(sample2_name)
    score, matrix = needleman_wunsh(sample1, sample2)

    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")

    print("-"*80, end="\n\n")
```

```
Amostra 1: korea | Amostra 2: porto_rico | Score: 1198 | Identidade: 69%

ATG--GCCATCATTTATCTCATACTCCTGT-T-CACA-GCAG-TG-AGGGGG-GAC-CAGATATGCATTGGATACCATGC
CAATAATTCCACAGAAAAGGTCGACACAATTCTAGAGCGGAATGTCACTGTGACTCA-
TGCCAAGGACATCCTTGAGAAGAC--CCATAACGGAAAGCTATGCAAACTAAACGGAATC-
CCTCCACTTGAACTAGGGGAC-TGTAGCATTGCCGGATGGCTCCTT-GGAAATCCAGAATGTGAT--AG-
GCTTCTAAGTGTGCCAGAATGGTCCTATATAATGGAGAAAGA--AAACCC-GAGATACAGTTTGTGTTACCCAGGCAGC-
TTCAAT-GACTATGAAGAATTGAAACATCTCCTCAGCAGC-GTGAAA-CATTTTGAGAAAG-TT--AAGAT-
TTTGCCCAAAGATAG---ATGGA-C-A-CAGCAT-ACAA-CAACTGGAGGTTCATGG--GCCTGCGCGG-
TGTCAGGTAAACCATCA-TTCTT-CAGGAACATGGTCTGGCTGACACGTA-AAGGAT--CAAATTATCCG--
GTTGCCAAAGGA-TCGTAC---AACAATACAAGCGGAGAACAAATGCTAATAATTTGGGG-AGTGCACCATCC-
TAATGATGAGGCAGAA-CAA-AGAGCATTGTACCAGAATGTGGGAAC-C-TATGTTTCCGTAGCCACATCAACATTGT-
ACAAAAGGTCAATCCCAGAAATAGCAGCAAGGCCTAAAGTGAATG-GA-CTAGGACGTAGAATGGAATTCTCT--
TGGACCCT-CTTGGATATGTGG-GACACCATAAAT-TTTGAGAGCAC-TGGTAATCTAGTTGCACCAGA-
GTATGGGTTCAAAATATCGAAAAGAGG-TA-GT-TCAGGGATCATGAAGACAGAAGGAA-CACTTG-A-GAACTGTGAA-
ACCAAATGCCAAACTCCTTTGGGAGCAATAAATACAACA--
CTACCTTTTCACAATGTCCACCCACTGACAATAGGTGAATGCCCCAAATATGTAAA-A-TCGGAGAAATTG-
GTCTTAGCAACAGGACTAAGGAATGTTCC--CCAGATTGAATCAAGAGGAT-
TGTTTGGGGCAATAGCTGGTTTTATAGAAGGAGGATGGCAA-GGAATGGTTGATGGTTGGTATGGATACCATCA-
CAGCAATGACCAGGGATCAGGGTATGCAGCAGA-CAAAGAATCCACTCAAAAGG-
CATTTAATGGAATCACCAACAAGGTAAATTCTGTGATTGA-AAAGATGAACACCCAATTTGA-AGCTGTTGGG-
AAAGAATTCAGTAACTTAGAGAAAA-GACTGGAGAACTTGAAC-
AAAAAGATGGAAGACGGGTTTCTAGATGTGTGGACATACAATGCAGAGCTTC-TAGTTCTGA-
TGGAAAATGAGAGGACACTTGACTTT-CATGATTCTAATGTCAAGAATCTGTATGATAAAGTCAGAATG-
CAGCTGAGAGAC-
AACGTCAAAGAACTAGGAAATGGATGTTTTGAATTTTATCACAAATGTGACAATGAATGCATGGATAGTGTGAA-
AAACGGGACATATGATTATCCCAAGTATGAAGAAGAATCTAAACTA-AATAGAAATGAAATCAA-AGGG-
GTAAAATTGAGCAGC-ATGGGGGTTTATCAAATCCTTGCCATTTA-TGCTACAGTAG-CAGGTTCTCT-GT-CAC-
```

TGGCAATCATGA-TGGCTGGG--ATCTCTTTCTGGATGTGCTCCAACGGGTCTCTGCAGTGCAGAATCTGCATATGA

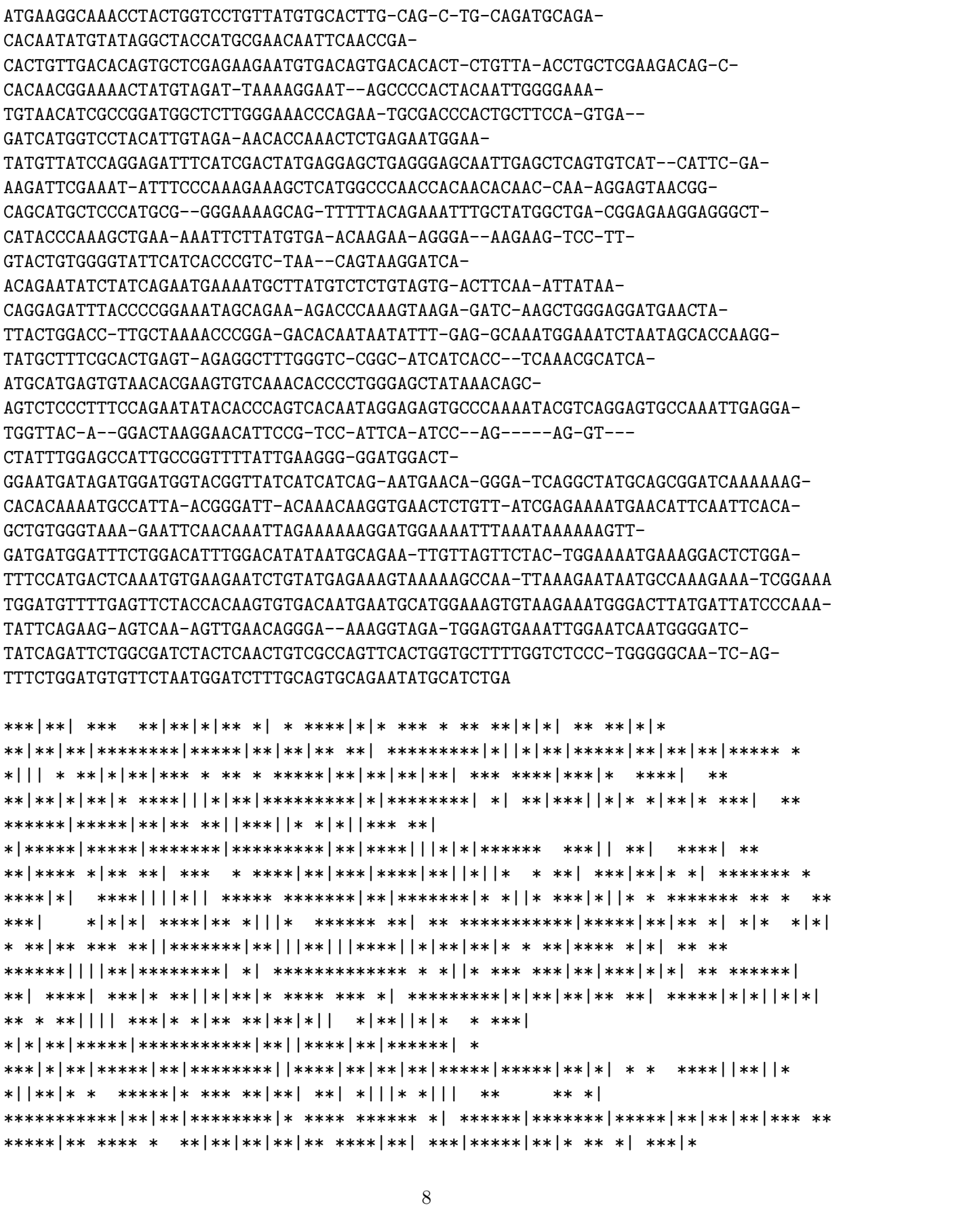
ATGAAGGCAA-ACCTA-CTGGTCCTGTTATGTGCACTTGCAGCTGCAGATGCAGACACA-
ATATGTATAGGCTACCATGCGAACAATTCAACCGACACTGTTGACACAGTGCTCGAGAAGAATGTGACAGTGACACACT-
CTGTTAACCTGCTC--GAAGACAGCCACAACGGAAAACTATGTAGATTAAAAGGAATAGCC-CCACTACAATT-
GGGGAAATGTAACATCGCCGGATGGCT-CTTGGGAAACCCAGAATGCGACCCACTGCTTCC-AGTGAG--
ATCATGGTCCTACAT--TGTAGAAACACCAAACTCTGAGA-ATGGAATATGTTATCCAGG-AGATTTC-
ATCGACTATGAGGAGCTGAGGGAGCAATTGAGCT-CAGTGTCATCATTC---GAAAGATTCGAA-ATATTT-
CCCAAAGAAAGCTCATGGCCCAACCA-CAACACAACCAAA-GGA-GTA-ACGGCAGCATGCTCCCATG-CGGGGAAA--
AGCAGTTTTTACAGAAATTTGCTATGGCTGACG-G-AGAAGGAGGGCTCA-TACCCAAAGCTGA-AAA--
ATTCTTATGTGAACAAGA-AAG-GGA-AAGAAGTCCTTGTACTGTGGGGTATT-CATCACCCGTC-TAAC-
AGTAAGGATCAACAGAATATC-TATCAGAATGAA--AATGCTTATGTCTCTGTAGTGACTTCAA-
ATTATAACAGGAGATTTACCCCGGAAATAGCAGAAAGACCCAAAGT-AA-GAGATCAAGCTGGGAGGAT-GAACTAT-
TACTGGACCTTGCTAA-A-ACCCGGAGACACAAT-AATATTTGAG-GCAAATGGAAATCTAATAGCACCA-
AGGTATGCTTTCGCAC--T-GAGTAGAGGCTTTGGGTCCGGCATCATCACCTCA-AACGCATCAA-TGCATGAG-TGT-
AACACGAAGTGTCAAACACCCCTGGGAGCTAT-AA-
ACAGCAGTCTCCCTTTCCAGAATATACACCCAGTCACAATAGGAGAGTGCCCAAAATACGTCAGGAGT-GCC-
AAATTGAGGATG-GTTACAGGACTAAGGAACATTCCGTCC--ATTCAATCCAGAGG-
TCTATTTGGAGCCATTGCCGGTTTTATTGAAGGGGGATGG-
ACTGGAATGATAGATGGATGGTACGGTTATCATCATCAG-AATGAACAGGGATCAGGCTATGCAGCGGATCAAAAAAG-
CACACAAAATGCCA-TTAACGGGATTACAAACAAGGTGAACTCTGTTATCGAGAAA-ATGAACATTCAATTC-

```
ACAGCTG-TGGGTAAAGAATTCAACAAATTAGAAAAAAGGA-TGGAAAATTTAAATAAAAAAGTT-
GATGATGGATTTCTGGACATTTGGACATATAATGCAGAA-TTGTTAGTTCT-ACTGGAAAATGAAAGGACTCTGGA-
TTTCCATGACTCAAATGTGAAGAATCTGTATGAGAAAGTAAAAA-GCCAATTAA-
AGAATAATGCCAAAGAAATCGGAAATGGATGTTTTGAGTTCTACCACAAGTGTGACAATGAATGCATGGAAAGTGT-
AAGAAATGGGACTTATGATTATCCCAAATATTCAGAAGAGTC-AAAGTTGAACAGGGA--
AAAGGTAGATGGAGTGAAATTG-GAATCAATGGGGATCTATCAGATTCTGGCGATCTACT-CAACTGTCGCCA-
GTTCACTGGTGCTTTTGG---TC-TCCCTGG--
GGGCAATCAGTTTCTGGATGTGTTCTAATGGATCTTTGCAGTGCAGAATATGCATCTGA

--------------------------------------------------------------------------------

Amostra 1: korea | Amostra 2: guangdong | Score: 1380 | Identidade: 71%

ATGGCCATCATTTA-T-CTCATACTCCTGTTCAC-AG-CAG---TG--AGGGGGGACCAGATATGCATTGGATACCATGC
CAATAATTCCACAGAAAAGGTCGACACAATTCTAGAGCGGAATGTCACTGTGACTCATG-
CCAAGGACATCCTTGAGAAGACCCATAACGGAAAGCTATGCAAACTAAACGGAATCCCTCCACTTGAACTAGGG-
GACTGTAGCATTGCCGGATGGCTCCTTGGAAATCCAGAATGTGAT-AGGC-TTC-T-
AAGTGTGCCAGAATGGTCCTATATAATGGAGAAAGAAAACCCGAGATACA-GTT-TGTGTTACCCAGGCAGC-
TTCAATGACTATGAAGAATTGAAACATCTCCTCAGCAGCGTGAAA-CATTTTGAGAAAGTTAAGATTTTGCCC-
AAAGATAGATGGACACAG-CAT-A--CAACAACTGGAGGTTCA--TGGGCCTG-CGCGGTGTCA-GGTAAA-
CCATCATTCTTCAGGAACATGGTCTGGCTGA-CACGTAAAGGATCAAATT-AT-CCGGTTGCC-AA-
AGGATCGTACAACAATA-CAAGCGGAGAACAAATGCTAATAATTT-
GGGGAGTGCACCATCCTAATGATGAGGCAGAACA-A-AGAGCATTGTACCAGAATGTGGGAACCTATGTTTCCGTAGCCA
CATCAACATTGTACAAAAGGTCAATCCCAGAAATAGC-
AGCAAGGCCTAAAGTGAATGGACTAGGACGTAGAATGGAATTCTCT-
TGGACCCTCTTGGATATGTGGGACACCATAAATTTTGAGAGCACTGGTAATCT-
AGTTGCACCAGAGTATGGGTTCAAAATA-TCGA-AAAGAGGTAGTTCAGGGATCATGAAGACAGAA--
GGAACACTTGAG-AACTGTGAA-ACCAAATGCCAAACTCCTTTGGGAGCAATAAA-TACAAC-
ACTACCTTTTCACAATGTCCACCCACTGACAATAGGTGAATGCCCCAAATATGTAAAATCGGAGAAATTGGT-
CTTAGCAACAGGACTAAGGAATGTTCCC-CAGATTGA-ATC--AAG-----AG-GAT---TGTTTGGGGCAATAGCTGGT
TTTATAGAAGGAGGATGGCAAGGAATGGTTGATGGTTGGTATGGATACCATCACAGCAATGACCAGGGATCAG-GG-
TATGCAGCAGACAAAGAATCCACTCAAAAGGCATTTA-ATGGAATCACCAACAAGGTAAATTCTG-TGATTGA-
AAAGATGAACACCCAATTTGAAGCTGTTGGGAAA-GAATTCAGTAACTTAGAGAAAAGAC--TGGAGAACTTGAACAAAA
AGATGGAAGACGGGTTTCTAGATGTGTGGACATACAATGCAGAGCTTCTAGTTCTGATGGAAAATGAGAGGACACTTGAC
TTTCATGATTCTAATGTCAAGAATCTGTATGATAAAGTCAGAATGCAGCTGAGAGACAACGTCAAAGAA-CTAGGAAATG
GATGTTTTGAATTTTATCACAAATGTGACAATGAATGCATGGATAGTGTGAAAAACGGGACATATGATTATCCCA-
AGTATGAAGAAGAATCTAA-ACTAAATAGAAATGAAATCAAAG-GG-GTAAAATTGAGCAGC-
ATGGGGGTTTATCAAATCCTTGCC-ATTTATGCTACAGTAGCAG-GTTCTCT-
GTCACTGGCAATCATGATGGCTGGGATCTCTTTCTGGATGTGCTCCAACGGGTC-TCTGCAGTGCAGAATCTGCATATGA

****||*||*  ** * **  * **|** *|**| ** ***   **  *|||*|**|*****|********|********
|**|**|**|*****||****|********||*|**|||***|**|*****|**|**** ****
*****|**|**|*****|**|**|**|*****|***|*|*****|***|*||||**|**| *||*|*|*
**|*****||*|**|***********|*****|**|  ***** * *|*| *** * **
******|********|**|***|*******|*||*||** **||* * *|| *|**********| *|
*****|************|*******|**||*|******||| *** ************|**|****||*|***
****|*|| ***|* **| *** *  *||**|*|** ***| *  *||**|** * *| *||** **|*|| **
**|**|*****|**||****|*****|* **|  ****|* **| *| ** **|||||| ** ****|*
```



```
ATGGAGAAAA--TAGTGCT--T-CTTCT-TGCAATAGTCAGTCTTGTCAAAAGTGATCAGATTTGCATTGGTTACCATGC
AAACAACTCGACAGAGCAGGTTGACACAATAATGGAAAAGAACGTTACTGTTACACATGCCCAA-
GACATACTGGAAAAGACACACAATGGGAAGCTCTGCGATCTAAATGGAGTGAAGCCTCTC-
ATTTTGAGAGATTGTAGTGTAGCTGGATGGCTCCTCGGAAACCCT--ATGTG-TGACGAATTCATCAA-
TGTGCCGGAATGGTCTTACATAGTGGAGAAGGCCAGTCC-AGCCA-ATGACCTCTGTTACCCAGGG-
GATTTCAACGACTATGAAGAACTGAAACACCTATTGAGCAGAAC-
AAACCATTTTGAGAAAATTCAGATCATCCCCAAAAGTTCT-TGGTC-CAATCATGATGCCTCATCAGG-GGTG-
AGCTCAGCATGTC-CA-TACCATGGGAGGTCC-TCCTTTTTCAGAAATGTGGTATGGCTTATCAA--AAAGAA-CAG-
TGCATACCCAACAATAAAGAGGAGC-TACAATAATACCAACCA-AGAAGATCTTTTAGTACTGTGGGGGATT-
CACCATCCTAATGATGCGGCAGAGCAGACAAAGC-TC-
TATCAAAACCCAACCACTTACATTTCCGTTGGAACATCAACACTGAACCAGAGATTGGTTCCAGAAATAGCTA-
CTAGACCCAAAGTAAACGGGCAAAGTGGAAGAATGGAGTTCT-
TCTGGACAATTTTAAAGCCGAATGATGCCATCAATTTCGAGAGTAATGGAAATTTCA-
TTGCTCCAGAATATGCATACAAAATTGTCAAGAAAGGGG-AC-TCAGCAATTATGAAAAGTGAATTGGAATA--TG-
GTAACTGC-AACACCAAGTGTCAAACTCCAATGGGGGCGATAAACT-CTAGTA-
TGCCATTCCACAACATACACCCCCTCACCATCGGGGAATGCCCCAAATATGTGAAATCAAACAGATTAGTCCTT-
GCGACTGGACTCAGAAATAC-CCCTCAGAGAGAGAGAAGAAGAAAAAAGAGAGGACTATTTGGAGCTATAGCAGGTTTTA
TAGAGGGAGGATGGCAGGGAATGGTAGATGGTTGGTATGGGTACCACCATAGCAATGAGCAGGGG--
AGTGGATACGCTGCAGACAAAGAATCCACTCAAAAGGCAA-TAGATGGAGTCACCAATAAGGTCAACTC-
GATCATTGACAAA-ATGAACACTCAGTTTGAGGCCGTT-GGAAGGGAATTTAATAACTT-G-GAAAGGAGGATAGAGAAT
TTAAACAAGCAGATGGAAGACGGATTCCTAGATGTCTGGACTTATAATGCTGAACTTCTGGTTCTCATGGAAAATGAGAG
AACTCTAGACTTTCATGACTCAAATGTCAAGAACCTTTATGACAAGGTCCGACTACAGCTTAGGGATAATG-CAAAGGAG
CTGGGTAATGGTTGTTTCGAGTTCTATCACAAATGTGATAATGAATGTATGGAAAGTGTAAAAAACGGAACGTATGACTA
-CCCGCAGTATTCAGAAGAAGC-AAGACTAAACAGAGAGGAAAT--AAGTGGAGTAAAATTG-
GAATCAATGGGAACTTACCAAATAC-TGTCAATTTATTCAACAGTGGC-GAGTTCCCTAG-
CACTGGCAATCATGGTAGCTGGTCTATCTTTATGGATGTGCTCCAATGGATCGT-TACAATGCAGAATTTGCATTTAA
```

Amostra 1: porto_rico | Amostra 2: guangdong | Score: 1048 | Identidade: 67%


ATGAAGGCAAACCTACTGGTCCTGTTATGTGCACTTG-CAG-C-TG-CAGATGCAGA-
CACAATATGTATAGGCTACCATGCGAACAATTCAACCGA-
CACTGTTGACACAGTGCTCGAGAAGAATGTGACAGTGACACACT-CTGTTA-ACCTGCTCGAAGACAG-C-
CACAACGGAAAACTATGTAGAT-TAAAAGGAAT--AGCCCCACTACAATTGGGGAAA-
TGTAACATCGCCGGATGGCTCTTGGGAAACCCAGAA-TGCGACCCACTGCTTCCA-GTGA--
GATCATGGTCCTACATTGTAGA-AACACCAAACTCTGAGAATGGAA-
TATGTTATCCAGGAGATTTCATCGACTATGAGGAGCTGAGGGAGCAATTGAGCTCAGTGTCAT--CATTC-GA-
AAGATTCGAAAT-ATTTCCCAAAGAAAGCTCATGGCCCAACCACAACACAAC-CAA-AGGAGTAACGG-
CAGCATGCTCCCATGCG--GGGAAAAGCAG-TTTTTACAGAAATTTGCTATGGCTGA-CGGAGAAGGAGGGCT-
CATACCCAAAGCTGAA-AAATTCTTATGTGA-ACAAGAA-AGGGA--AAGAAG-TCC-TT-
GTACTGTGGGGTATTCATCACCCGTC-TAA--CAGTAAGGATCA-
ACAGAATATCTATCAGAATGAAAATGCTTATGTCTCTGTAGTG-ACTTCAA-ATTATAA-
CAGGAGATTTACCCCGGAAATAGCAGAA-AGACCCAAAGTAAGA-GATC-AAGCTGGGAGGATGAACTA-
TTACTGGACC-TTGCTAAAACCCGGA-GACACAATAATATTT-GAG-GCAAATGGAAATCTAATAGCACCAAGG-
TATGCTTTCGCACTGAGT-AGAGGCTTTGGGTC-CGGC-ATCATCACC--TCAAACGCATCA-
ATGCATGAGTGTAACACGAAGTGTCAAACACCCCTGGGAGCTATAAACAGC-
AGTCTCCCTTTCCAGAATATACACCCAGTCACAATAGGAGAGTGCCCAAAATACGTCAGGAGTGCCAAATTGAGGA-
TGGTTAC-A--GGACTAAGGAACATTCCG-TCC-ATTCA-ATCC--AG-----AG-GT---
CTATTTGGAGCCATTGCCGGTTTTATTGAAGGG-GGATGGACT-
GGAATGATAGATGGATGGTACGGTTATCATCATCAG-AATGAACA-GGGA-TCAGGCTATGCAGCGGATCAAAAAAG-
CACACAAAATGCCATTA-ACGGGATT-ACAAACAAGGTGAACTCTGTT-ATCGAGAAAATGAACATTCAATTCACA-
GCTGTGGGTAAA-GAATTCAACAAATTAGAAAAAAGGATGGAAAATTTAAATAAAAAAGTT-
GATGATGGATTTCTGGACATTTGGACATATAATGCAGAA-TTGTTAGTTCTAC-TGGAAAATGAAAGGACTCTGGA-
TTTCCATGACTCAAATGTGAAGAATCTGTATGAGAAAGTAAAAAGCCAA-TTAAAGAATAATGCCAAAGAAA-TCGGAAA
TGGATGTTTTGAGTTCTACCACAAGTGTGACAATGAATGCATGGAAAGTGTAAGAAATGGGACTTATGATTATCCCAAA-
TATTCAGAAG-AGTCAA-AGTTGAACAGGGA--AAAGGTAGA-TGGAGTGAAATTGGAATCAATGGGGATC-
TATCAGATTCTGGCGATCTACTCAACTGTCGCCAGTTCACTGGTGCTTTTGGTCTCCC-TGGGGGCAA-TC-AG-
TTTCTGGATGTGTTCTAATGGATCTTTGCAGTGCAGAATATGCATCTGA

```
**|**|*****|***** *|* **|**|**********|***|**|| * **|**|** **|
*****|**|**|**|****||*****|**|********|**|| **|*
**|**|*****|**|**||*|*****|********|*** **|*| ***** * **********|**|*****|** ***
**************|*****|**|*****|**|**|||*|||**| ***||* ****** *****|*|
*|**|*****|*****|********|*****|*****|********|*************|***|**|**|*****|**
***||* ********** ** *** *|*| *****|**  ***  ** * ******|*****************|* *
**|**|**|***|*|**|**|*****|**|**|*****|**|*||*   *** *|||* ***||**|| ** *|
***|********|**|********|**|**|********|*****|*|*

ATGGAGA-AAA--TAGTGCTTCT-TC-T-TGCAATAGTCAGTCTTGTCAAAAGT-GATCAGATT-
TGCATTGGTTACCATGCAAACAACTCGACAGAGCAG-GTTGACACAATAATGGAAAAGAACGTTACTGTTACACA-
TGCCCA-AGACATACTGGAA-A-AGACACACAATGGGAAGCTCTGC-GATCTAAATGGAGTGAAGCCT--CT-
CATTTTGAGAGATTGTAGTGTAGCTGGATGGCTCCTCGGAAACCCT-ATGTGTGACGAATT-CATCAATGTGCCGGA--
ATGGTCTTACATAGTGGAGAAGGCCAGTC-CAGCCAAT-
GACCTCTGTTACCCAGGGGATTTCAACGACTATGAAGAACTGAAACACCTATTGAG--CAGAA-
CAAACCATTTTGAGAAAATTC-AGATCATC-CCC--A-AAAGTTCTTGGTCCAATCATGATGC--CTCATCAGGGGTGA-
GCTCAGCATG-T-CCATACCATGGGAGGTCCTCCTTTTT-CAGAAATGTGGTATGGCTTATCAAA-AAGAACAG-
TGCATACCC-AA-C--AATAAAG----
AGGAGCTACAATAATACCAACCAAGAAGATCTTTTAGTACTGTGGGGGATTCACCATCC-TAATGATGCGGC-A-
GAGCAGACA-AAGCTCTATCAAAACCCAACCACTTACATTTCCGTTG-GAACATCAACACTG-AACCA-
GAGATTGGTTCCAGAAATAGCT-ACTAGACCCAAAGTAA-ACGGGCAAAG-TGGAAGAATGGAGTTCTT-
CTGGACAATTT-TAAAG-CCGAATGATGCCATCA-ATTTCGAGAGT-AATGGAAATTTCATTGCTCC-
AGAATATGCATACAAAATT-GTCA-AGAAAG-GGGACTCAGCAATTATGAAAAGTGAATTGGA--ATATGG-
TAACTGCAACACCAAGTGTCAAACTCCAATGGGGGCGATAAACT-
CTAGTATGCCATTCCACAACATACACCCCCTCACCATCGGGGAATGCCCCAAATATGTGAA-A-T--
CAAACAGATTAGTCCTTGCGACTGGACTCA-GAA-
ATACCCCTCAGAGAGAGAGAAGAAGAAAAAAGAGAGGACTATTTGGAGCTATAGCAGGTTTTATAG-AGGGAGGATGG-
CAGGGAATGGTAGATGGTTGGTATGGGTACCACCAT-AGCAATGAGCAGGGGAGT--GGATACGCTGCAGA-
CAAAGAATCCACTCAAAAGGCAA-TAGAT-GGAGTCACCAATAAGGTCAACTC-
GATCATTGACAAAATGAACACTCAGTTTG-AGGCCGTTGG-
AAGGGAATTTAATAACTTGGAAAGGAGGATAGAGAATTTAAACAAGC-
AGATGGAAGACGGATTCCTAGATGTCTGGACTTATAATGCTGAACTTCTG-GTTCT-
CATGGAAAATGAGAGAACTCTAGACTTT-
CATGACTCAAATGTCAAGAACCTTTATGACAAGGTCCGACTACAGCTTAGGG-ATAATG-CAAAGGAGCTGGGTAATGGT
TGTTTCGAGTTCTATCACAAATGTGATAATGAATGTATGGAAAGTGTAAAAAACGGAACGTATGACTA-
CCCGCAGTATTCAGAAGAAG-CAAGACTA-AACAGAGAGGAAA--TA-AGTGGAGTAAAATTGGAATCAATGGGAA-
CTTACCAAATACTGTCAATTTATTCAACAGTGGCGAGTTCCCTAGCAC---TGG-
CAATCATGGTAGCTGGTCTATCTTTATGGATGTGCTCCAATGGATCGTTACAATGCAGAATTTGCATTTAA

--------------------------------------------------------------------------------
```

0.4 Questão 2

[6]:
```
sample1 = "GCCGCCGGC"
sample2 = "CCCC"
score, matrix = needleman_wunsh(sample1, sample2, gap=-4, match=7, mismatch=-3)
```

```
sample1_aligned, match_mismatch, sample2_aligned = get_backtracking(sample1,␣
 ↪sample2, matrix)
identity = int(match_mismatch.count("*") / len(match_mismatch) * 100)
print(f"Score: {score} | Identitidade: {identity}%\n")
print(''.join(sample1_aligned))
print(''.join(match_mismatch))
print(''.join(sample2_aligned))
```

```
Score: 8 | Identitidade: 44%

GCCGCCGGC
  * **  *
--C-CC--C
```

[ ]: