Sequence Alignment

M. Kurka

June 9, 2021

- Rationale Behind
- Scoring System
- Needleman-Wunsch Algorithm
- Smith-Waterman Algorithm
- Situation Today

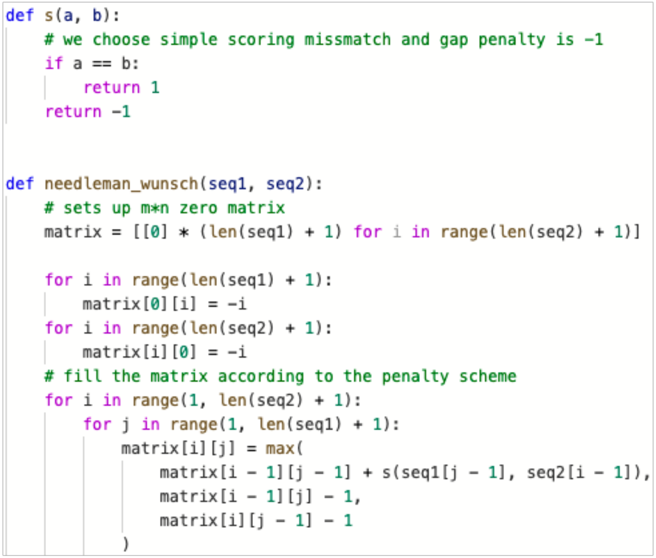
Why is it useful

- Homology search
- Genetic diversity
- Active sites, Reading frames

Keeping Scores

- mathematically describe similarity
- selecting weights
  - BLOSUMXY
  - mathematical (log likelihood)

Needleman-Wunsch Algorithm

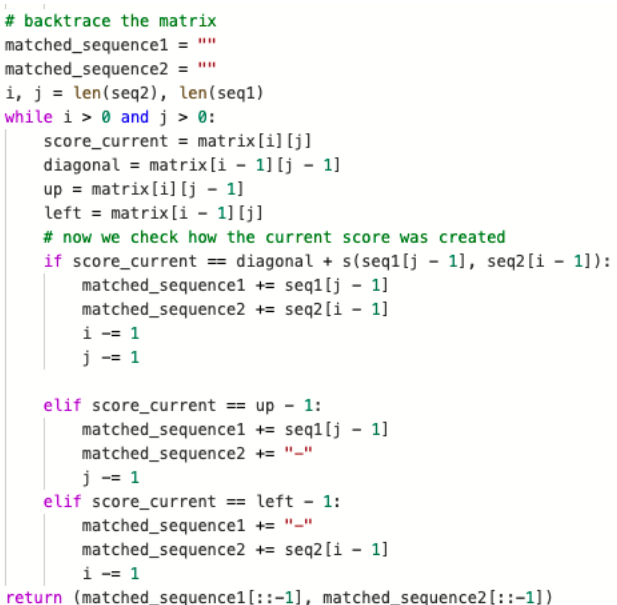


```python
def s(a, b):
    # we choose simple scoring missmatch and gap penalty is -1
    if a == b:
        return 1
    return -1


def needleman_wunsch(seq1, seq2):
    # sets up m*n zero matrix
    matrix = [[0] * (len(seq1) + 1) for i in range(len(seq2) + 1)]

    for i in range(len(seq1) + 1):
        matrix[0][i] = -i
    for i in range(len(seq2) + 1):
        matrix[i][0] = -i
    # fill the matrix according to the penalty scheme
    for i in range(1, len(seq2) + 1):
        for j in range(1, len(seq1) + 1):
            matrix[i][j] = max(
                matrix[i - 1][j - 1] + s(seq1[j - 1], seq2[i - 1]),
                matrix[i - 1][j] - 1,
                matrix[i][j - 1] - 1
            )
```

Needleman-Wunsch Algorithm



```
# backtrace the matrix
matched_sequence1 = ""
matched_sequence2 = ""
i, j = len(seq2), len(seq1)
while i > 0 and j > 0:
    score_current = matrix[i][j]
    diagonal = matrix[i - 1][j - 1]
    up = matrix[i][j - 1]
    left = matrix[i - 1][j]
    # now we check how the current score was created
    if score_current == diagonal + s(seq1[j - 1], seq2[i - 1]):
        matched_sequence1 += seq1[j - 1]
        matched_sequence2 += seq2[i - 1]
        i -= 1
        j -= 1

    elif score_current == up - 1:
        matched_sequence1 += seq1[j - 1]
        matched_sequence2 += "-"
        j -= 1
    elif score_current == left - 1:
        matched_sequence1 += "-"
        matched_sequence2 += seq2[i - 1]
        i -= 1
return (matched_sequence1[::-1], matched_sequence2[::-1])
```

Smith-Waterman Algorithm

- took 10 years to develop
- minor change

$$matrix[\,i\,][\,j\,] = max \begin{cases} matrix[i-1][j-1] + s(x,y) \\ matrix[i-1][\,j\,] - 1 \\ matrix[\,i\,][j-1] - 1 \\ 0 \end{cases}$$

Smith-Waterman Algorithm

How is it done in practice

- Both algorithms are $O(mn)$ and provably correct
- Genome of organsims are really long
- Poses a challenge $\rightarrow$ Heuristics
  - BLAST
  - FASTA