Smith-Waterman local alignment,and Needleman-Wunsch global alignment using

April 5, 2019

Online demo(1) for local and global alignnment

```
In [7]: # Biopython modules for pairwise alignmnet
        from Bio import pairwise2
        from Bio import SeqIO
        from Bio.SubsMat.MatrixInfo import blosum62
```

1 Smith-Waterman local alignment

Online Demo(2) for Smith-Waterman local alignment

```
In [2]: # alignment with no match score and gap penalty
        local_alignment = pairwise2.align.localxx('AGGTTGCA', 'AGGTCA')
        # formatted output
        for i in range(len(local_alignment)):
            print(pairwise2.format_alignment(*local_alignment[i]))
```

```
AGGTTGCA
||| | ||
AGG-T-CA
  Score=6

AGGTTGCA
||||  ||
AGGT--CA
  Score=6
```

```
In [3]: # local alignment with opengap= -10, extendgap= -0.5
        local_alignmentds = pairwise2.align.localds('AGGTTGCA', 'AGGTCA', blosum62, -10, -0.5)
        # formatted output
        for i in range(len(local_alignmentds)):
            print(pairwise2.format_alignment(*local_alignmentds[i]))
```

```
AGGTTGCA
||||  ||
AGGT--CA
  Score=23.5
```

2 Needleman-Wunsch global alignment

Online Demo(3) for Needleman-Wunsch global alignment

```
In [4]: # alignment with no match score and gap penalty
        global_alignmnet = pairwise2.align.globalxx('ATTAC', 'AATTC')
        # formatted alignment
        for i in range(len(global_alignmnet)):
            print(pairwise2.format_alignment(*global_alignmnet[i]))
```

```
A-TTAC
| || |
AATT-C
  Score=4

-ATTAC
 ||| |
AATT-C
  Score=4
```

```
In [5]: # global alignment using blosum62 substitution matrix, opengap= -1, extendgap= -1
        global_alignmentds = pairwise2.align.globalds('ATTAC', 'AATTC', blosum62, -1, -1)
        # formatted output
        for i in range(len(global_alignmentds)):
            print(pairwise2.format_alignment(*global_alignmentds[i]))
```

```
A-TTAC
| || |
AATT-C
  Score=21

-ATTAC
 ||| |
AATT-C
  Score=21
```

```
In [6]: # global alignmnet with match= 1, mismatch= -1, opengap= -1, endgap= -1
        global_alignmentds = pairwise2.align.globalms('ATTAC', 'AATTC', 1, -1, -1, -1)
```

```
# formatted output
for i in range(len(global_alignmentds)):
    print(pairwise2.format_alignment(*global_alignmentds[i]))
```

```
A-TTAC
| || |
AATT-C
  Score=2

-ATTAC
 ||| |
AATT-C
  Score=2
```

3 xx in localxx and globalxx can have these values

(x)x The match score parameters are:

1. x No parameters. Identical characters have score of 1, otherwise 0.
2. m A match score is the score of identical chars, otherwise mismatch score.
3. d A dictionary returns the score of any pair of characters.
4. c A callback function returns scores.

x(x) The gap penalty parameters are:

1. x No gap penalties.
2. s Same open and extend gap penalties for both sequences.
3. d The sequences have different open and extend gap penalties.
4. c A callback function returns the gap penalties.

4 References:

1. https://en.wikipedia.org/wiki/Smith%E2%80%93Waterman_algorithm

2. https://en.wikipedia.org/wiki/Needleman%E2%80%93Wunsch_algorithm

3. http://biopython.org/DIST/docs/api/Bio.pairwise2-module.html

4. http://biopython.org/DIST/docs/tutorial/Tutorial.html