

Inan Khan  
Dynamic programming assignment

	A		G	C	T	C	A	G
G	-1	0	0	1	0	1	0	0
C	-1	0	1	0	0	0	0	1
A	-1	0	0	0	1	0	0	0
G	-1	0	0	1	0	1	0	0
G	-1	0	0	1	0	1	0	0

The code that I created for this dynamic programming assignment is from biopython

```
1  # Import pairwise2 module
2  from Bio import pairwise2
3
4  # Import format_alignment method
5  from Bio.pairwise2 import format_alignment
6
7  # Define two sequences to be aligned
8  X = "ACGGGT"
9  Y = "ACG"
10
11 # Get a list of the global alignments between the two sequences ACGGT and ACG
12 # No parameters. Identical characters have score of 1, else 0.
13 # No gap penalties.
14 alignments = pairwise2.align.globalxx(X, Y)
15
16 # Use format_alignment method to format the alignments in the list
17 for a in alignments:
18     print(format_alignment(*a))
19
```

Essentially what this program does is what is shown above where you define two strings and it applies -1 for a blank space, a 0 for a mismatch and a 1 for a match.

```
ACGGGT
|||||
AC--G-
Score=3

ACGGGT
|||||
AC-G--
Score=3

ACGGGT
|||||
ACG---
Score=3
```

I went a bit further where I made it so at the end it would add all these values up and give you a score. This score can be useful if you want to attach a value to describe the similarity between two strings. This value can be valuable if you want to compare similarity between separate sets of strings. What is interesting about my code is that you can use two strings of any length and any amount of gaps, so there are many possibilities