

# BCBBIO 444 Lab: Dynamic Programming

In this lab, you will construct a program to perform and illustrate pairwise sequence alignment by dynamic programming.

1. Your first task is to build the aligner.

Input:

- A. Two DNA sequences, up to 10 bp each.
- B. Mismatch score (-s), match score (-m) , and gap penalty (-g).

Output:

- A. The global alignment of the two sequences (optional, if user requests it).
- B. The scoring matrix of the two sequences (optional, if user requests it).
- C. The global alignment score.

Example call:

```
$ myaligner -s -1 -m 1 -g -1 TTCGGGAA TTCGGCTAC
```

The score matrix, when requested by the user, can be output in text format, with traceback cells highlighted, such as with asterisks. For extra credit, you can produce a graphical representation of the matrix with the traceback cells in a different color (see the matplotlib library). The alignment, when requested, should be presented as two lines involving the characters A, C, G, T and - only, for example:

```
TTCGG-GAA
TTCGGCTAC
```

Provide the score matrix and alignment for two different sequence pairs, each one with two different combinations of -s, -m and -g. If you could use some inspiration, here is an interesting example of the internal promoter site, including a TATA box, from the Foamy Virus that infects orangutan and a Foamy-like Virus recently discovered in the Coelacanth genome. The two sequences, not necessarily aligned, are:

```
TGCCATTAAAGTCAAACAAGT
GAATATAAAAGATCAAATTGA
```

To get two input sequences less than 11bp, you need to remove nucleotides from the end(s) (or you could try aligning the whole sequences).

2. Your second task is to assess whether the match you find indicates homology. Design and implement a permutation model to test whether the global alignments you have found are

statistically significant. Remember the hypothesis testing framework: Clearly define the null model, choose a relevant test statistic, and simulate under the null model to estimate the probability of obtaining data as or more extreme than the observed, as measured by the test statistic.

Hint 1: You may find <https://docs.python.org/2/library/argparse.html> useful for parsing command-line options and arguments.

Hint 2: See this website for an example, and also to check yourself:  
<https://gtuckerkellogg.github.io/pairwise/demo/>