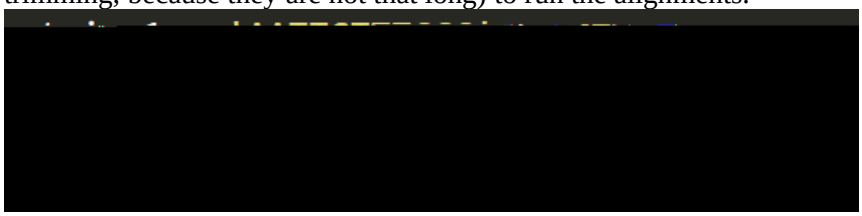
Yu Fu individual project summary. O56922. May 11, 2017.

Affine gap alignment

In homework 7, we did the global alignment of two strings with a fixed indel penalty. However, when there is a gap in either string that was extended over two nucleotides, the biological interpretation is still an indel, but only longer. Therefore, is it more realistic to penalize gap extension less than that of opening a gap.

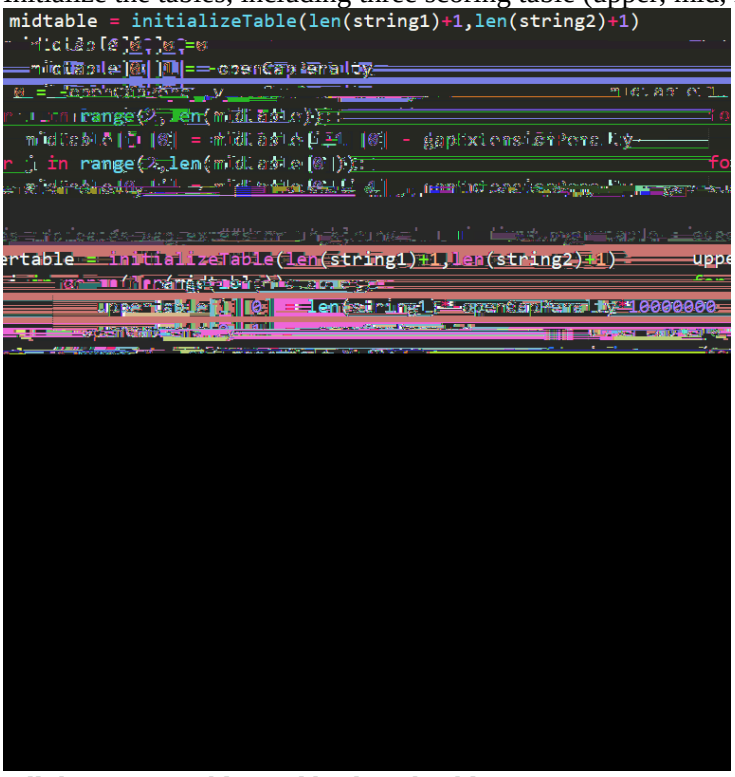
My code is based on the global alignment algorithm from homework 7. So, I simply copied the global alignment function and then stated to code for an affine gap alignment. In my python file, I have two sets of strings. I used a pair of toy strings to test whether my function works or not and used the actual sequence of PDGF and Vsis gene from UCSC genome browser (without any trimming, because they are not that long) to run the alignments.



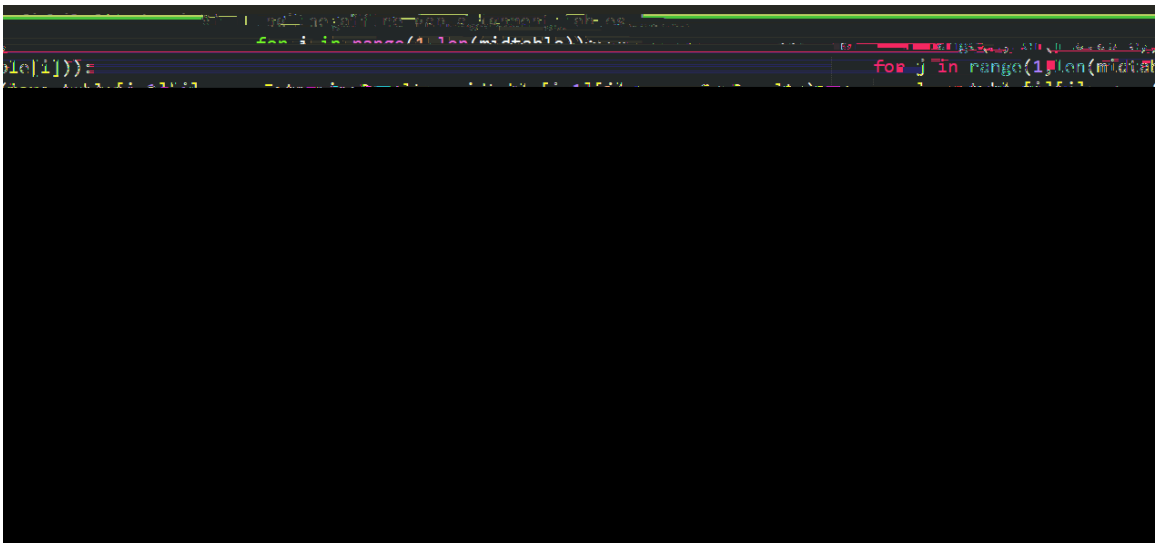
Code structure:

My affine gap alignment function consists of 3 parts:

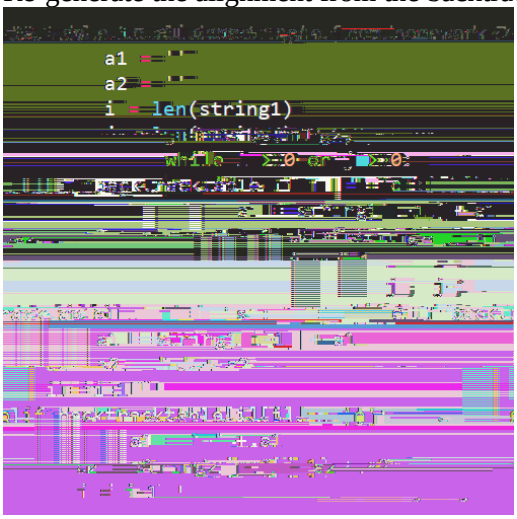
1. Initialize the tables, including three scoring table (upper, mid, lower) and a backtrack table



2. Fill the scoring tables and backtrack table



3. Re-generate the alignment from the backtrack table



Results:

This is the scoring rules:

```
matchScore = 1
mismatchPenalty = 1
indelScore = 10
openGapPenalty = 10
gapExtensionPenalty = 1
```

The result of the test strings:

Affine gap alignment has a score of -12, only one gap
Global alignment has a score of -55, three gaps
Affine gap alignment is an algorithm that favors long single gaps instead of short single ones.
Thus, it works.