Plan

Section 1 : A discussion on Affine gap [com[

* Typical alignment method, why it could be improved
* Penalty for extending gap vs. penalty for creating gap, allows us to reflect the biological nature of indel creation. [ small segment about indel sources]
* This is enacted through the 3 matrices concept
* Modify numbers if looking for more distant match

Include a discussion of alternate gap penalties that might also be used instead of Affine gaps.

* Standard gap penalties which use a fixed value for each gap introduced during alignment.
* No gap penalty? Just align the most similar parts of the sequneces and hope it’s a good alignment
* Variable gap penalties which scale with the length of the gap
* Probabilistic models, which take into account residue relationships and partially predicts the chance of a gap in the context of the existing environment
  + Profile based?
  + Especially useful for protein alignment, where the preservation of residue physicochemical properties allows for substitutions at different rates.
* End-gap penalties, which penalise end gaps more strongly than those appearing in the middle of the alignment (which are more biologically likely)
* Convex, Log

Processing time

Using these gap weighting schemes as examples, explain why would you consider using Affine gaps over these other gap weights?

* Compute time?
* Using fixed penalties as seen in affine scenario can be unrepresentative of true biology as it is too simplistic.

When would you not use Affine Gaps for alignments?

Section 2 [50% marks]

* Decide on affine gap penalties by looking at original SW paper
* Code Affine gaps as a standalone function, then work out how to merge it with Simon’s work

Using the code we have generated in BA4 (Implementing the Smith Waterman Algorithm in Python) as a foundation, outline how you would extend this code to implement support for Affine gaps.

You can choose to implement the code and present this as part of your answer.

You also answer this part of the question by just describing the modifications you would make in text in order to convert this alignment programme to support Affine Gaps.

If you present code then you can provide this as part of your answer but must still provide an overview of how the code calculated Affine Gaps