Practical 9. Implementation of a sequence comparison program in Python

Mikael Bjorklund

IBI1, 2018/19

1. Learning objectives

- Explain how to compare two amino acid sequences
- Appreciate the logic behind sequence analysis algorithms
- Create and use a program that perform a simple non-gapped alignment
- Evaluate the output from a sequence alignment

2. Background

Sequence comparisons are a common bioinformatics task. In the simplest form, sequence comparison is just a *string comparison.* However, additional considerations such as the frequency of individual amino acid substitutions add some complexity to the task.

While it is possible to perform sequence comparisons without detailed knowledge about the underlying algorithms, it is useful to understand the logic of these comparisons to draw biologically meaningful conclusions. In this practical we use BLOSUM (BLOcks SUbstitution Matrix) matrix for protein alignment. BLOSUM is a substitution matrix, which can be used to quantify alignments between evolutionarily divergent proteins. The original matrix was obtained by analysis of highly conserved regions in a number of protein families. The relative frequencies of amino acids and their substitution probabilities were used to calculate a log-odds score[1] for each of the possible substitution pairs of the 20 standard amino acids.

There are many types of BLOSUM matrixes, but all are based on observed alignments. The number after BLOSUM indicates the criteria used to build the matrix. For example, BLOSUM62 is the matrix built using sequences with less than 62% similarity (sequences with ≥ 62% identity were clustered). We will use this matrix as it is the default matrix for protein BLAST. It is the default setting as benchmarking of BLOSUM matrixes has identified that BLOSUM62 is among the best for detecting weak sequence similarities. Other BLOSUM matrixes have more specialized uses, for example BLOSUM80 can be used for more related proteins and BLOSUM45 for distantly related proteins.

[1]*Log-odds score is a logarithm of the likelihood of an event relative to its likelihood under a null model. Positive log-odds scores indicate that the event is more likely than it would be under the null model. Therefore, structurally related amino acids have a positive score and more distant amino acid substitutions yield a negative score.*

3. Summary of the required files

• The aim of this tutorial is to implement a simple pairwise non-gapped global alignment[2] and use that for comparing the protein sequences shown below (also available as separate txt files).

•*Sequence for human SOD2 protein (NP_000627.2)*
>SOD2_human (NP_000627.2)
MLSRAVCGTSRQLAPVLAYLGSRQKHSLPDLPYDYGALEPHINAQIMQLHHSKHHAAYV
NNLNVTEEKYQEALAKGDVTAQIALQPALKFNGGGHINHSIFWTNLSPNGGGEPKGELL
EAIKRDFGSFDKFKEKLTAASVGVQGSGWGWLGFNKERGHLQIAACPNQDPLQGTTGLI
PLLGIDVWEHAYYLQYKNVRPDYLKAIWNVINWENVTERYMACKK

•*Sequence of a mouse SOD2 protein (NP_038699.2)*
>SOD2_mouse (NP_038699.2)
MLCRAACSTGRRLGPVAGAAGSRHKHSLPDLPYDYGALEPHINAQIMQLHHSKHHAAY
VNNLNATEEKYHEALAKGDVTTQVALQPALKFNGGGHINHTIFWTNLSPKGGGEPKGE
LLEAIKRDFGSFEKFKEKLTAVSVGVQGSGWGWLGFNKEQGRLQIAACSNQDPLQGTTG
LIPLLGIDVWEHAYYLQYKNVRPDYLKAIWNVINWENVTERYTACKK

•*A random sequence*
>RandomSeq
WNGFSEWWTHEVDYNQKLTIENNQRPKIHEHEQWGLRQSPPPPKLCCPTCQMCERM
RHQNRFAPLMEVGCRCMCWFHDWWVISVGTWLHTVIMYMMWPKRFHHNECPKACF
RTTYTRKNHHALYWMLFEMCCYDQDVVWSKTHIFTTVRDIEVYVEQVFFIWGPLCHV
AIACYEPVKTIRRRIPMYLCRHCIRGDNSYLLACCSIIYYFYHHMSYYGVLDIL

•BLOSUM62 matrix. Can you identify a reliable source for this?

[2]*To appreciate what this means, see Planning section.*

4. Planning

•We will not perform a BLAST search. Instead we want to keep things simple and do a pairwise non-gapped global alignment. What this means is that we compare two sequences (*pairwise*) without considering the possibility for amino acid insertions or deletions (*non-gapped*). We perform a *global* (instead of a local) alignment starting from amino acid 1 and ending with the last amino acid.

•We do not need to perform a graphical alignment, but it would be useful to print out the analyzed sequences in addition to the final BLOSUM score. You should also print out the percentage identity (how many amino acids are identical).

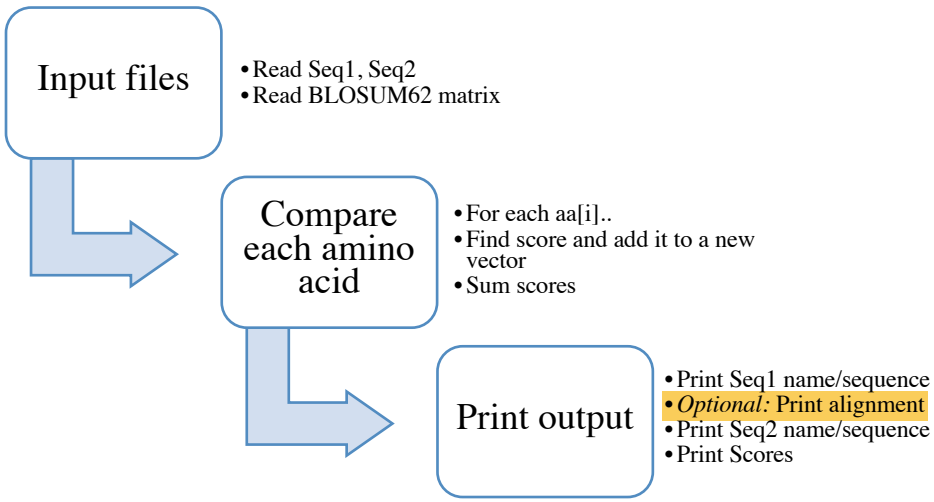

Figure. *Workflow for the program.*

•You have learned how the BLOSUM62 matrix works. Plan a workflow how you would use this information for comparing two sequences.

•Think how you can compare the output scores in a meaningful way. How would you need to adjust your scoring if you would like to compare the similarity between two 100 amino acid containing proteins and another set of 500 amino acid containing proteins? Note that "*citing a raw score alone is like citing a distance without specifying feet, meters, or light years.*" (https://www.ncbi.nlm.nih.gov/BLAST/tutorial/Altschul-1.html)

BONUS PROJECT: Print out the two sequences together with a BLAST-like visual alignment (indicate alignment with the same amino acid if there is a perfect match, and print + symbol for conservative substitutions (BLOSUM score ≥0)

5. Implementation

Start a new file called Alignment.py and use the three-step logic outlined in the figure to code your alignment script.

To help you get started this a simple script to calculate the Hamming/edit distance

```
seq1 = "MLSRAVCGT"
seq2 = "MLCRAACST"
edit_distance = 0              #set initial distance as zero
for i in range(len(seq1)): #compare each amino acid
    if seq1[i]!=seq2[i]:
        edit_distance += 1  #add a score 1 if amino acids are
                                different
print (edit_distance)
```

6. Analysis

Run all three pairwise combinations of sequences (human-mouse, human-random, mouse-random) through your analysis pipeline. How similar are mouse and human sequences compared to a random sequence (in quantitative terms)?

7. For your portfolio

Write a short summary of your findings and also include your interpretation. You can add or edit things after the Practical session. We do not look at the commit date, we just want it all to be there!