

Disclaimer: complete accuracy is not guaranteed.

1. Collect Pfam “seed” entry:

Select the Pfam entry of your protein of interest (POI). Go to the “Alignment” tab. From “Available alignments” select “seed”. Download the seed alignment in clustal2 format. For this tutorial, Peptidase C13 was used as an example POI.

The screenshot shows the Pfam website interface for the Peptidase C13 family (PF01650). The page includes a navigation bar with links like Home, Search, Browse, Results, Release notes, Download, Help, About, and Contact us. The main content area displays the Pfam logo, the family name 'Peptidase C13 family', and a list of 33 sequences. A color-coded alignment view is shown, with a scale from 0 to 200. The sequences listed include Q852T1_TOBAC/49-320, A0A038CW15_BETV/51-321, VPEG_ARATH/59-330, VPE3_ORYS3/62-334, Q24539_VICNA/53-325, Q9LLQ4_SESN/54-326, Q95MD0_SOLLG/39-316, Q9XG75_TOBAC/40-306, VPEG_ARATH/46-316, Q6PRC7_DKORL/38-293, and LGMN_HUMAN/29-286.

2. Use the “Visual” pairwise sequencing code:

To visualize the pairwise sequence alignment between 2 homologous sequences of the POI, input the stockholm file from Pfam into the “pairwise_tool.py” module (a biopython code using the AlignIO and pairwise2 packages):

```
1 from Bio import AlignIO, pairwise2
2
3 sto_file = r"C:\Users\Tania\Downloads\PF01650.alignment.seed\PF01650.alignment.seed" #Insert your .seed stockholm/clustal2 filepath here
4
5 alignment = AlignIO.read(r"C:\Users\Tania\Downloads\PF01650.alignment.seed\PF01650.alignment.seed", "stockholm") #Insert your .seed stockholm/clustal2 filepath again
6
7 seq1 = alignment[0].seq
8 seq2 = alignment[1].seq
9
10 alignments = pairwise2.align.globalxx(seq1, seq2)
11
12 for alignment in alignments:
13     print(pairwise2.format_alignment(*alignment))
```

You should receive an output like this (Note: the bottom alignment may appear “slanted”):

```
KHVVVAGSR-GYNNYRHQDVCHAYQ-LUKK-GGLKDNITVPRYD-TAHF--ENPRPVI-INSN-----G----D-DVYK-G-VPKDYTGH-VTA-INEL-AV-ILQKA-ALS---GGSGKVE-SGPNDHIFIFYSDHGG-G--VLGP--S--G-PPLYADD-LIDV-LKR-KHASGTYSKLVF-YIEACESGSIFEG-L-
LPEGLNIYATTASNAE-EDS-NGTCPCD-----YRG-P--PPQYQ-TCLGLYA-VSM-EDSEK--HNLRR---ET-LQR-Q-YELV-KRRT--AUS-F--P-YAS-SHMQYGDLY-LMD--DP-LS-L-Y-----NGT
RHAVLTAGS-SGYNNYRHQDVCHAYQVL-KK-GGLKDNITVPRYD-IA-Y-DEENPRGV-LINSP-Y-----G-----HDVY-AG-VPKDYTG--EDVT-VNMF-FA-AILQK-DA--IT-GGSGKVV-NSGPNDHIFIFYSDHGG-AG--VLGP--T--YPVLYA-DELT--ETLK-EKHASGTYSKLV-VYIEACESGSIFEG--
ILPEGLNIYATTASNAE-VE-SSMGTCPCG-Q-----DPIWPEY-DTCLGLY-SVSM-IEDSE-R-HNL--HT--E-SL--KQYQ-VVK--TKTA-E--KPFY--GSHVQYGD-KEL--TQD-NL-YL-Y-----NGT
Score=308
```

A match score can be found below the alignment.

The Pairwise ALL tool can be used to create a pairwise visualization and find a match score between every sequence at once.

WARNING AND DISCLAIMER, PLEASE READ: You can use the “pairwise_tool_ALL_seq.py” module to conduct a sequence alignment between each sequence in the SEED file; however, this will take up a lot of memory, a lot of computer processing power, and it may even crash your computer. Running the code on VS studio may give you a bunch of random numbers in the terminal. You can simply exit VS studio, and you should still have some, if not all pairwise sequences, analyzed on the output .txt file.

```
1  from Bio import AlignIO, pairwise2
2
3  sto_file = r"C:\Users\Tania\Downloads\PF01650.alignment.seed\PF01650.alignment.seed" #Insert your .seed filepath here
4  output_file = r"C:\Users\Tania\Documents\Bio notepad reads\pepc13_read.txt" #insert your .txt filepath here
5
6  alignment = AlignIO.read(sto_file, "stockholm")
7
8  with open(output_file, "w") as outfile:
9      for i in range(len(alignment)):
10         for j in range(i + 1, len(alignment)):
11             seq1 = str(alignment[i].seq)
12             seq2 = str(alignment[j].seq)
13
14             alignments = pairwise2.align.globalxx(seq1, seq2)
15
16             for aligned_pair in alignments:
17                 outfile.write(f"Pairwise alignment between sequences {i+1} and {j+1}:\n")
18                 outfile.write(pairwise2.format_alignment(*aligned_pair))
19                 outfile.write("\n")
20
```