Appendix A. Creation of the Core-PFAM reference database

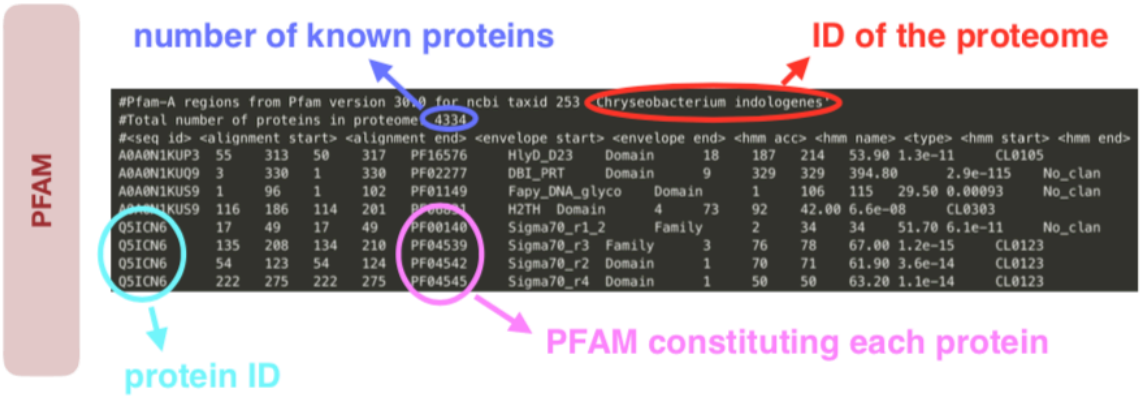
Step 1: Download the database

- Go to https://pfam.xfam.org/ > FTP > releases and choose the preferred version (we used the 32.0). Then go to the proteomes folder.

- After decompressing all the files, select those of bacteria (looking at the file rpg-55bac_arch-75euk, available at ftp://ftp.pir.georgetown.edu/databases/rps/)

- Create a folder with only bacteria files (we called it fold_bacteria).

Example of a file in PFAM database (you can see the **tax ID** on the first line):



Step 2: Create the PFAMs vs Proteomes matrix for Bacteria:

- Create a list of all the proteomes appearing in all the bacteria files (first column):

```
cd fold_bacteria
list_file=$(ls *.tsv)
echo $list_file >> ../list_proteomes_bacteria.txt
```

- Extract from each proteome file the list of its PFAMs (sixth column) and paste it to a file list_all_Pfam.txt:

```
for file in ${list_file[*]}; do awk 'NR>3{print $6}'  $file >> ../list_all_Pfam.txt; done
```

- Eliminate redundant PFAMs:

```
awk '{print $1}' ../list_all_Pfam.txt | sort | uniq -c | awk '{print $2}' >> ../
list_Pfam_bac.txt
```

- Create the folder where to put, for each proteome, a file with two columns, the first with the list of its PFAMs and the second with the associated abundance:

```
cd ..
mkdir abundances_proteomes_bacteria
cd fold_bacteria
```

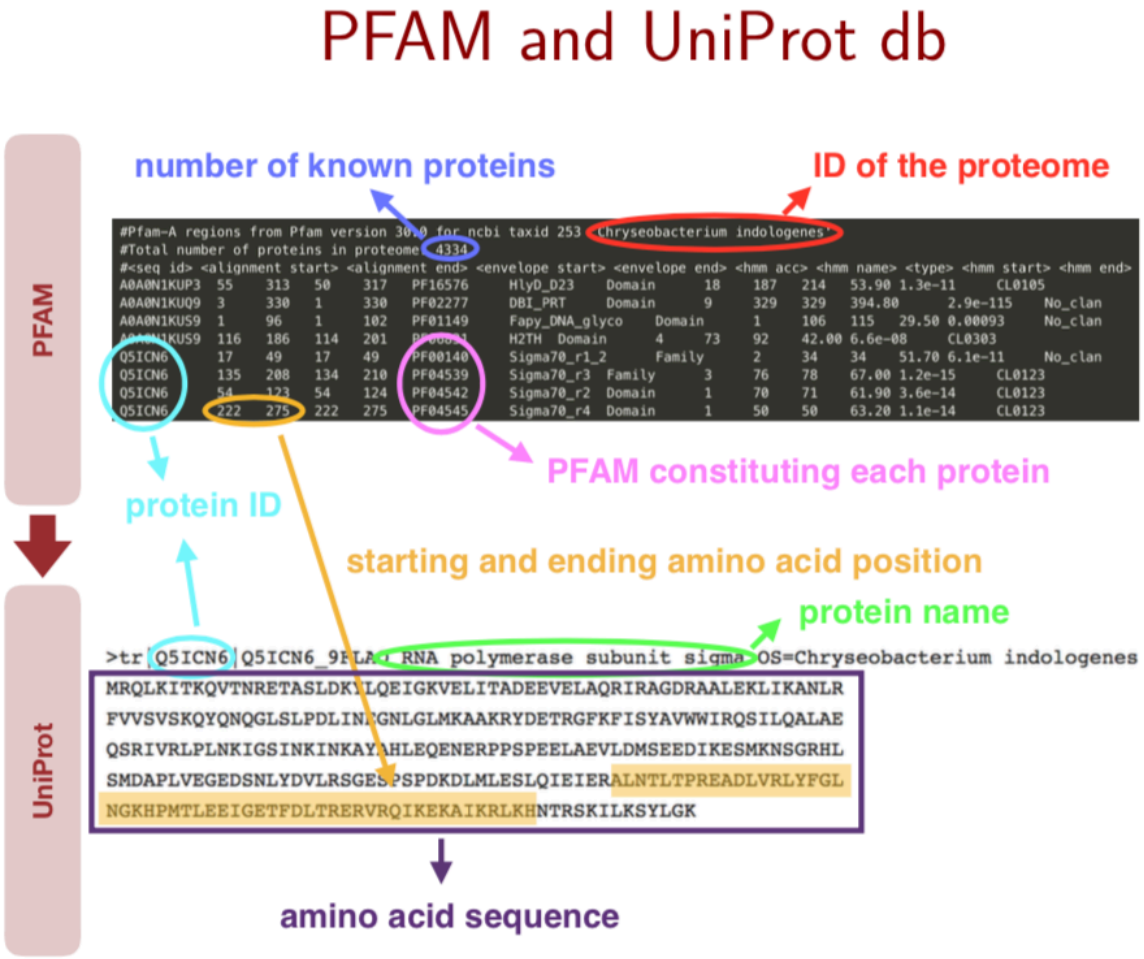
```
list_proteomes=($( ls *.tsv))
for file in ${list_proteomes[*]}; do awk 'NR>3{print $6}'  $file | sort | uniq -c | awk
'{print $2, $1}'  >> ../abundances_proteomes_bacteria/$file; done
```

- Run the script in R called Matrix_PFAMvsProteomes.R to create a FxP matrix (file matrix_PFAMvsPROTEOMES_bacteria.csv), whose entries (f,p) are the number of times PFAM f appears in proteome p.

Step 3: Find the Core-PFAMs:

- Identify the Core-PFAMs (prevalent + max occurrence = 4). In our case we found the following: PF00453, PF00572, PF01029, PF01196, PF01649, PF01795, PF03947, PF08338, PF09285, PF17136 (see R-script Analysis&CorePFAM).

- To associate the Core-PFAMs to the correspondent amino-acid sequence cross the information with UniProt:



Thus create (see R-script FindSeqCorePFAM), for each Core-PFAM, a file (called SequencesCorePFAM_PFAMname.txt) where to insert, for each proteome, the amino-acid sequence with which the PFAM appears in that proteome. Even lines are sequences, odd lines are information (in the form >proteome/protein) where the

PFAM has been found (this file can be analyzed with JalView, for example). Finally, create, for each Core-PFAM, a file (called SequencesCorePFAM_PFAMname.csv) where rows are in the form Proteome/Proten_ID/Sequence.

- Once you merge all the files into one fasta file called corepfams_reduced.fa, build the Kaiju reference index with these two commands:

```
kaiju-mkbwt -n 5 -e 3 -a ACDEFGHIKLMNPQRSTVWY -o corepfams_reduced corepfams_reduced.fa
kaiju-mkfmi corepfams_reduced
```

[At this stage we substituted two taxonomy IDs when converting the csv files to the fasta files since PFAM database contained the old IDs:
1217693 -> 70346 (Acinetobacter variabilis)
1566299 -> 1960309 (Klenkia marina)]

Appendix B. Core-Kaiju Protocol (example for CAMI high-complexity sample 1)

Step 1: Run Kaiju 1.0

- We used Kaiju version 1.6.2 with reference database ncbi2018-06-04

```
kaiju -t nodes.dmp -f kaiju_db_nr_euk.fmi -i RH_S001__insert_270.fq.gz -o CH1.out -v
```

```
kaijuReport -t nodes.dmp -n names.dmp -i CH1.out -o CH1.txt -v -r genus -l phylum,class,order,family,genus
```

Step 2: Run Kaiju with PFAM reference database (see Appendix A)

```
kaiju -t nodes.dmp -f corepfams_reduced.fmi -i RH_S001__insert_270.fq.gz -o CH1_PFAM.out -v
```

```
kaijuReport -t nodes.dmp -n names.dmp -i CH1_PFAM.out -o CH1_PFAM.txt -v -r genus -l phylum,class,order,family,genus
```

Step 3: Process the results

Create an abundance matrix where rows are genera and columns are methods (Kaiju 1.0 and Kaiju-PFAM). See R script ProcessingResults.R