Appendix A. Creation of the Core-PFAM reference database

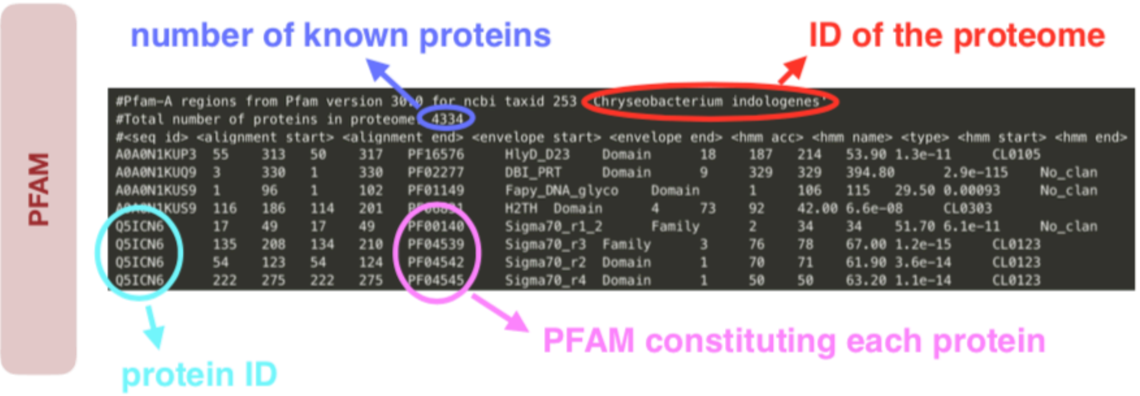
Step 1: Download the database

- 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):



```
#Pfam-A regions from Pfam version 30.0 for ncbi taxid 253 'Chryseobacterium indologenes'
#Total number of proteins in proteome 4334
#<seq id> <alignment start> <alignment end> <envelope start> <envelope end> <hmm acc> <hmm name> <type> <hmm start> <hmm end>
A0A0N1KUP3  55   313  50   317  PF16576   HlyD_D23   Domain   18   187  214  53.90 1.3e-11     CL0105
A0A0N1KUQ9  3    330  1    330  PF02277   DBI_PRT    Domain   9    329  329  394.80     2.9e-115   No_clan
A0A0N1KUS9  1    96   1    102  PF01149   Fapy_DNA_glyco    Domain   1    106  115  29.50 0.00093    No_clan
A0A0N1KUS9  116  186  114  201  [illegible]   H2TH  Domain   4    73   92   42.00 6.6e-08    CL0303
Q5ICN6      17   49   17   49   PF00140   Sigma70_r1_2    Family   2    34   34   51.70 6.1e-11    No_clan
Q5ICN6      135  208  134  210  PF04539   Sigma70_r3  Family   3    76   78   67.00 1.2e-15     CL0123
Q5ICN6      54   123  54   124  PF04542   Sigma70_r2  Domain   1    70   71   61.90 3.6e-14     CL0123
Q5ICN6      222  275  222  275  PF04545   Sigma70_r4  Domain   1    50   50   63.20 1.1e-14     CL0123
```

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 l sort l uniq -c l 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
```

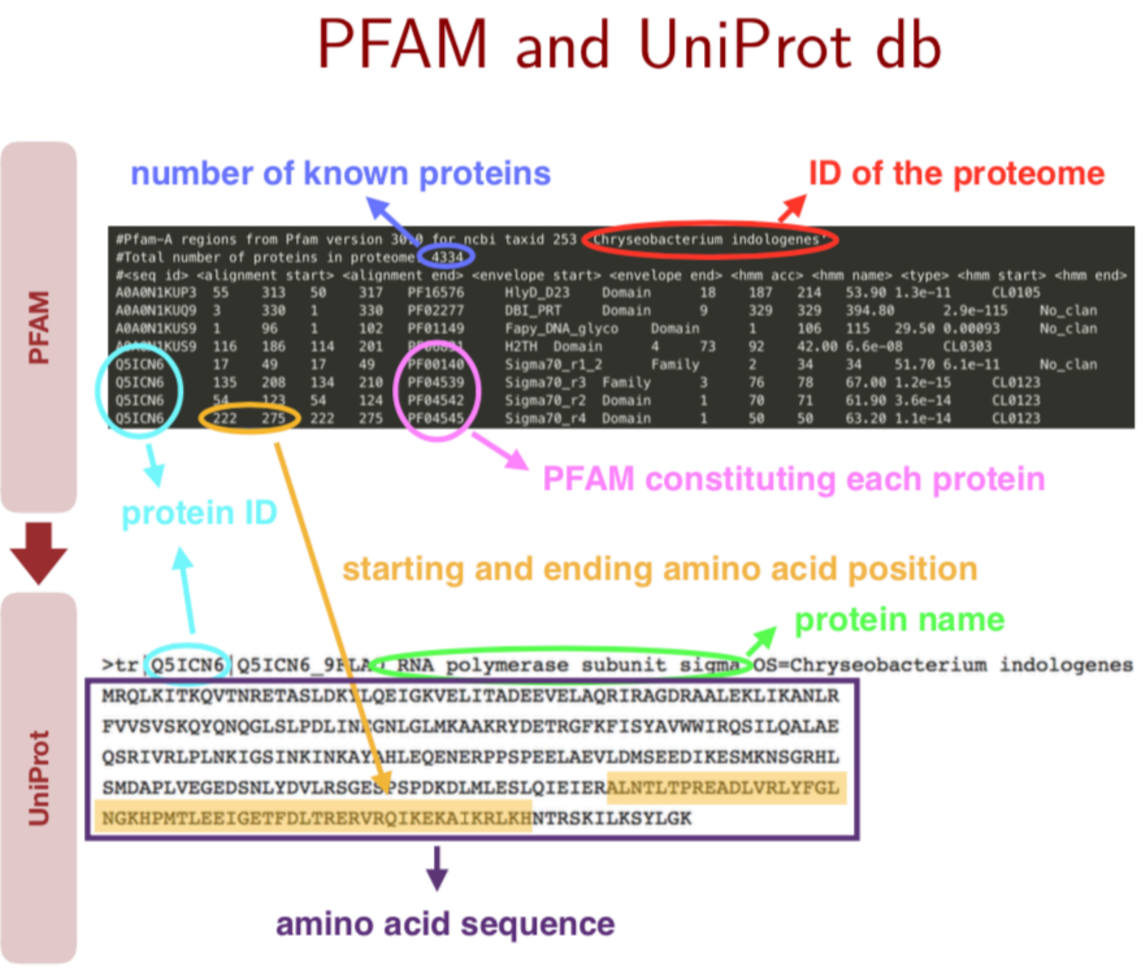
```
list_proteomes=($( ls *.tsv))
for file in ${list_proteomes[*]}; do awk 'NR>3{print $6}'  $file l sort l uniq -c l awk
'{print $2, $1}'  >> ../abundances_proteomes_bacteria/$file; done
```

- 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:



PFAM and UniProt db

```
#Pfam-A regions from Pfam version 30.0 for ncbi taxid 253 'Chryseobacterium indologenes'
#Total number of proteins in proteome 4334
#<seq id> <alignment start> <alignment end> <envelope start> <envelope end> <hmm acc> <hmm name> <type> <hmm start> <hmm end>
A0A0N1KUP3  55    313   50    317   PF16576     HlyD_D23    Domain      18    187   214   53.90 1.3e-11       CL0105
A0A0N1KUQ9  3     330   1     330   PF02277     DBI_PRT     Domain      9     329   329   394.80        2.9e-115      No_clan
A0A0N1KUS9  1     96    1     102   PF01149     Fapy_DNA_glyco    Domain      1     106   115   29.50 0.00093       No_clan
A0A0N1KUS9  116   186   114   201   PF06831     H2TH  Domain      4     73    92    42.00 6.6e-08       CL0303
Q5ICN6      17    49    17    49    PF00140     Sigma70_r1_2      Family      2     34    34    51.70 6.1e-11       No_clan
Q5ICN6      135   208   134   210   PF04539     Sigma70_r3  Family      3     76    78    67.00 1.2e-15       CL0123
Q5ICN6      54    123   54    124   PF04542     Sigma70_r2  Domain      1     70    71    61.90 3.6e-14       CL0123
Q5ICN6      222   275   222   275   PF04545     Sigma70_r4  Domain      1     50    50    63.20 1.1e-14       CL0123
```

```
>tr|Q5ICN6|Q5ICN6_9FLAO RNA polymerase subunit sigma OS=Chryseobacterium indologenes
MRQLKITKQVTNRETASLDKYLQEIGKVELITADEEVELAQRIRAGDRAALEKLIKANLR
FVVSVSKQYQNQGLSLPDLINEGNLGLMKAAKRYDETRGFKFISYAVWWIRQSILQALAE
QSRIVRLPLNKIGSINKINKAYAHLEQENERPPSPEELAEVLDMSEEDIKESMKNSGRHL
SMDAPLVEGEDSNLYDVLRSGESSPSPDKDLMLESLQIEIERALNTLTPREADLVRLYFGL
NGKHPMTLEEIGETFDLTRERVRQIKEKAIKRLKHNTRSKILKSYLGK
```

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 Mock1)

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 Mock1_1.fastq -j Mock1_2.fastq -o
Mock1_ALL.out -v
```

```
kaijuReport -t nodes.dmp -n kaijudbnames.dmp -i Mock1_ALL.out - o
Mock1_Allgenus.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 Mock1_1.fastq -j Mock1_2.fastq -o
Mock1_PFAM.out -v
```

```
kaijuReport -t nodes.dmp -n names.dmp -i  Mock1_PFAM.out -o
Mock1_PFAMgenus.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