Metagenomics: Data Analysis

> **Authors:** André Rosengaard Jørgensen & Prince Ravi Leow

This notebook is divided into 3 sections:

- **AMR gene composition study (CoDa)**, using outputs from `kma` (k-mer alignment).
- **Metagenomic bin quality study**, using outputs from `CheckM2`.
- **Microbial diversity study (TaxID)**, using outputs from `Gtdb-tk`.

The study of AMR gene composition between our 3 samples is of particular interest, as the samples are harvested from different pigs, and this data will tell us whether the AMR gene composition varies between individuals.

The metagenomic bin quality study, is to assure that we are selecting the satisfactory quality data, to conduct the taxonomical study.

Finally, the in the TaxID section, we will investigate, how the microbial diversity varies between the 3 samples, and by extesion different individuals.

All data analysis and visualisation is conducted in `python`, unless stated otherwise. For `R` implementations, please refer to the supplementary `meta.R` file.

1) AMR GENE COMPOSITION STUDY (CoDa)

AMR gene fragment abundance and length was determined by running `kma` against ResFinder. Because the composition study is only valid for bacteria, `kma` was also run against SILVA for determining the proportion of bacterial vs. non-bacterial reads in our sample.

It should be noted, that kma/SILVA is conducted on non-binned reads, making it unsuitable for true taxonomical annotation, thus necessitating the use of `Gtdb-tk` outputs later in this study.

1.1 Wrangling `kma` ResFinder and SILVA outputs

First, the `resfinder` dataframe - pre-prepared from the `*.mapstat` files - is loaded in, and isolated into their own dataframes.

Displayed: sample 24 with gene, and abundance data

```
Resfinder sample 24
```

	Abundance	Gene
0	117.0	Cfr(E)_1_NG_070225
1	30.0	VanG2XY_1_FJ872410
2	6.0	VanGXY_1_AY271782
3	0.0	VanHBX_1_AF192329
4	0.0	aac(6')-Im_1_AF337947
...	...	...
104	0.0	tet(X5)_1_CP040912
105	33.0	tetA(P)_1_AB054980
106	86.0	tetA(P)_2_L20800
107	92.0	tetB(P)_1_NC_010937
108	10.0	vat(E)_5_AJ488494

109 rows × 2 columns

Assessing the original `resfinder` dataframe, we can determine how many AMR genes there are.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3 entries, 0 to 2
Columns: 109 entries, Cfr(E)_1_NG_070225 to vat(E)_5_AJ488494
dtypes: float64(109)
```

There are >100 headers, so we want to 'amalgamate' AMR gene classes eventually.

Using the `ResFinder_classes.tsv`, we can find our how many unique AMR classes there are.

```
Index(['Phenicol', 'Glycopeptide', 'Aminoglycoside', 'Beta-Lactam',
       'Macrolide', 'Nitroimidazole', 'Sulphonamide', 'Tetracycline',
       'tet(O/32/O)_6_NG_048124'],
      dtype='object')
```

There are 8 unique classes (a 9th one that seems to fit into Tetracycline - we could merge it, or discard it later on).

We need the fragment length data (found in dataframe: `ResFinder.length`) and their corresponding gene names (`ResFinder.name`) to conduct CoDa later on. This information can be merged with `resfinder_{24,25,38}`.

The resulting dataframes contain AMR gene, class and length data.

For example, here is sample 24:

	Abundance	Gene	Class	Length
0	117.0	Cfr(E)_1_NG_070225	Phenicol	1035
1	30.0	VanG2XY_1_FJ872410	Glycopeptide	1811
2	6.0	VanGXY_1_AY271782	Glycopeptide	1811
3	0.0	VanHBX_1_AF192329	Glycopeptide	2607
4	0.0	aac(6')-Im_1_AF337947	Aminoglycoside	537
...	...	...	...	...
104	0.0	tet(X5)_1_CP040912	Tetracycline	1167
105	33.0	tetA(P)_1_AB054980	Tetracycline	1263
106	86.0	tetA(P)_2_L20800	Tetracycline	1263
107	92.0	tetB(P)_1_NC_010937	Tetracycline	1959
108	10.0	vat(E)_5_AJ488494	Macrolide	645

109 rows × 4 columns

We are almost ready to amalgamate the abundance values by gene classes.

However, we currently lack the true bacterial content, which we will use to calculate ALR values *(see CoDa subsection)*.

KMA output from the **SILVA** database is used to determine the bacterial content of the samples.

This data can be obtained directly from silva `*.mapstat` files.

Our implementation, was to select for bacterial fragmentCount rows, by an index of all rows containing 'Bacteria' instances in the TaxID column.

The total fragments was found by simply summing the entire column.

Finally, bactieral content was determined by dividing the number of bacterial fragments, by the total fragments.

Calculations result in the following bacterial proportions:

```
Sample 24 bacterial content:
0.99
Sample 25 bacterial content:
0.98
Sample 38 bacterial content:
0.96
```

This demonstrates that the amount of non-bacterial reads in our sample is effectively negligible.

However, they are scalar quantities, using them in FPKM calculations is easy, so we will still use them.

In addition, we we need the total number of total fragment for each sample.

This can also be found in the `*.mapstat` files, and is found by summing up the `fragmentCount` columns:

```
Sample 24 total fragment
102963
Sample 25 total fragment
299937
Sample 38 total fragment
59954
```

1.2 CoDa of `kma` outputs

For the CoDa section, we will compute:

- Additive-log ratios (ALR) - specifically the log(FPKM) implementation (Zhao et al. 2021) - which use a single component of the sample as a reference (in this case, bacterial fragments or 'bacterial content').
- Centre-log ratios (CLR) - which use the sample geometric mean as a reference.

While the ALR values will which tell us about composition of a SPECIFIC sample, the CLR values will reveal differences in compositions BETWEEEN samples (Quinn et al. 2019).

1.2.1 ALR a.k.a. log(FPKM) calculations & plot

In order to perform CoDa via FPKM values, we need both the data from the Silva and Resfinder databases.

The Resfinder database provides us information on the resistance genes - which ones, how many fragments we get, and how long the fragments are.

The Silva database provides us with information on how many of our genes are actually bacterial.

FPKM is fragments per kilo basepairs, per million.

Full logFPKM expression:

$$\log(\text{FPKM}) = \log\left(\frac{\text{fragments} \times 10^3 \times 10^6}{\text{gene length in bp} \times \text{total bacterial fragments})}\right)$$

We will use a dataframe containing gene, class, abundance and length, the calculated bacterial content and number of total fragments per sample.

Note: *We will conduct zero-replacements, as zeros would break the log transformation. We do this by replacing the zeros with a very small number - in this case, 1 - denoting a single gene fragment.*

The final wrangled dataframe contains log(FPKM) values, amalgamated by AMR gene class.

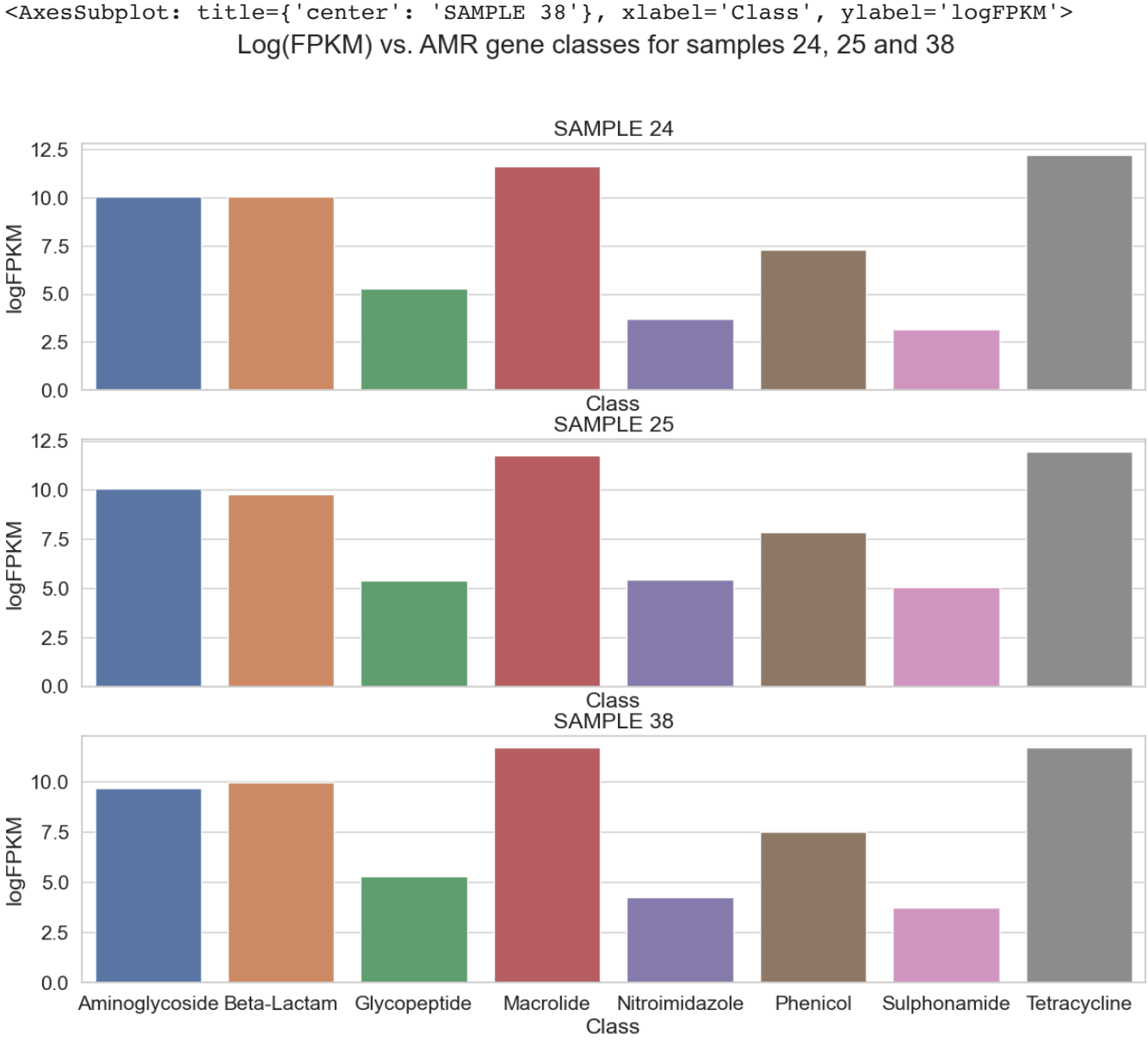
Sample 24 as an example:

`log(FPKM) values amalgamated by AMR gene class, for sample 24`

	Class	logFPKM
0	Aminoglycoside	10.070966
1	Beta-Lactam	10.070344
2	Glycopeptide	5.291849
3	Macrolide	11.602498
4	Nitroimidazole	3.676424
5	Phenicol	7.307583
6	Sulphonamide	3.158231
7	Tetracycline	12.204588

We can then generate a barplot for each dataframe.

`<AxesSubplot: title={'center': 'SAMPLE 38'}, xlabel='Class', ylabel='logFPKM'>`



The plots reveal that tetracycline and macrolide are the most abundant AMR gene classes, with aminoglycoside and beta-lactam being the second most abundant AMR gene classes.

The actual log(FPKM) quantities seem *very* similar, which would indicate a similar AMR gene class composition between all 3 samples. However, due to the nature of the ALR values, they are technically *not* comparable between different samples.

For this purpose, we will compute CLR values in the following section.

1.2.2 CLR calculations & plot

CLR values are computed by taking the original resfinder fragment abundance dataframes (from the `*.mapstat` files), and performing 'closure' to them.

Since we have used number of AMR fragments, we will use the total fragments (found in `fragmentCountAln` column).

Having performed closure, we amalgamate after AMR gene classes, as we did before with ARL.

We now perform CLR. This can be done by dividing our fragment abundances by the geometric mean for each column, and log transforming.

Resulting CLR values after applying closure and amalgamating after AMR gene classes:

	Class	CLR(24)	CLR(25)	CLR(38)
0	Aminoglycoside	1.970189	1.475739	1.523114
1	Beta-Lactam	2.096769	1.294787	1.947807
2	Glycopeptide	-2.056609	-2.408420	-2.125687
3	Macrolide	3.619219	3.283970	3.722264
4	Nitroimidazole	-4.974380	-3.694244	-4.477062
5	Phenicol	-0.567661	-0.516190	-0.434011
6	Sulphonamide	-4.974380	-3.583018	-4.477062
7	Tetracycline	4.886852	4.147376	4.320637

```
Sum of CLR values (they should sum to zero):
sum(CLR(24)):
1.7763568394002505e-15
sum(CLR(25)):
-3.552713678800501e-15
sum(CLR(38)):
-8.881784197001252e-16
Sums are small enough to be considered negligible.
```

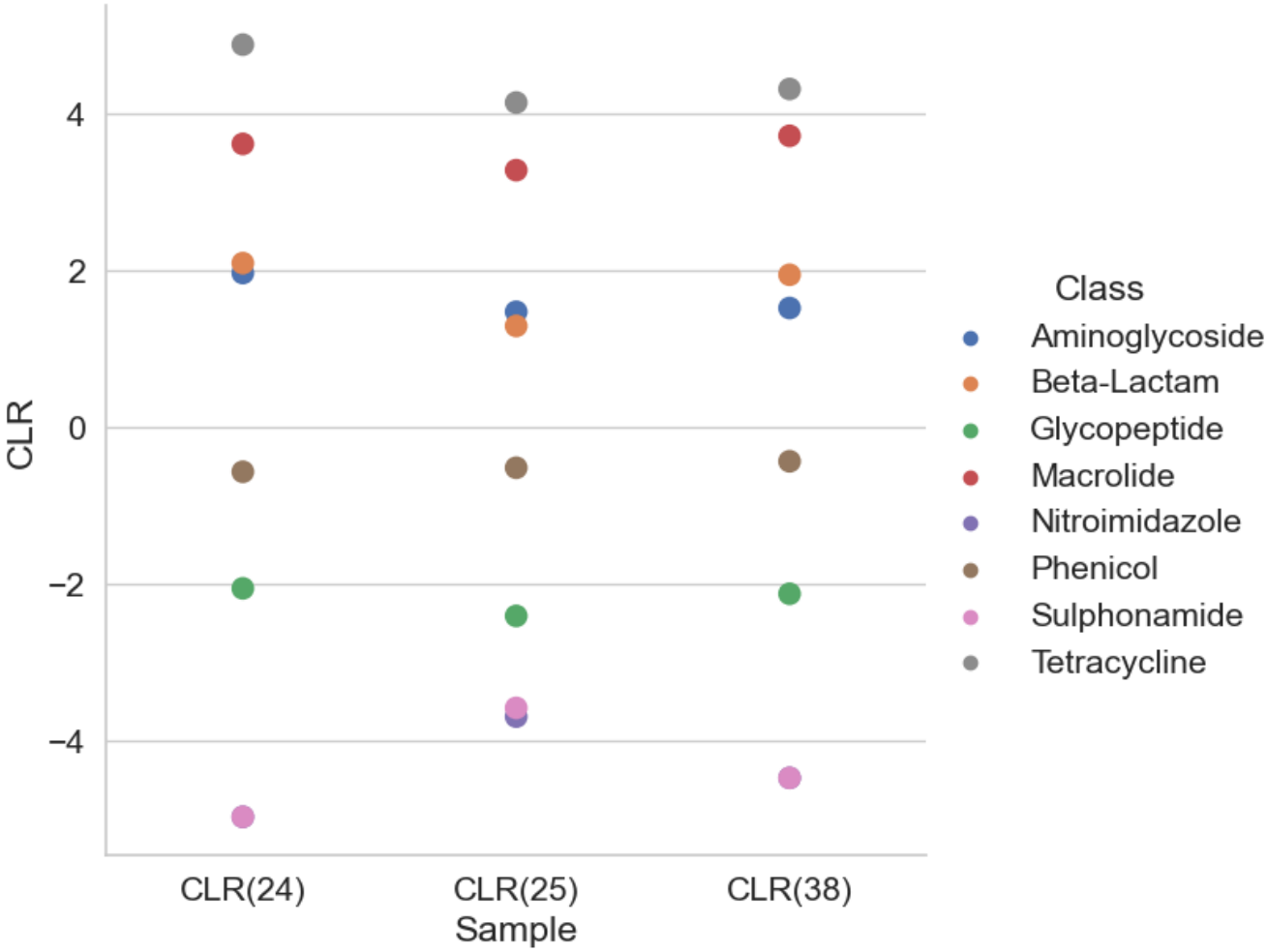
1.2.2.1 Visualising CLR values

To use the python package `seaborn`'s 'hue' function, we need to perform a 'categorisation' by Samples, or 'pivot'.

This is is done in `R`, and the resulting dataframe is exported to a csv, and loaded into the `python` workflow.

The following CLR plot is generated:

```
<seaborn.axisgrid.FacetGrid at 0x7f7bf9a56fe0>
```



This plot demonstrates that although the samples vary in size, the AMR gene *composition* between them, varies very little.

This supports the what was indicated from the log(FPKM) plots.

2) METAGENOMIC BIN QUALITY STUDY

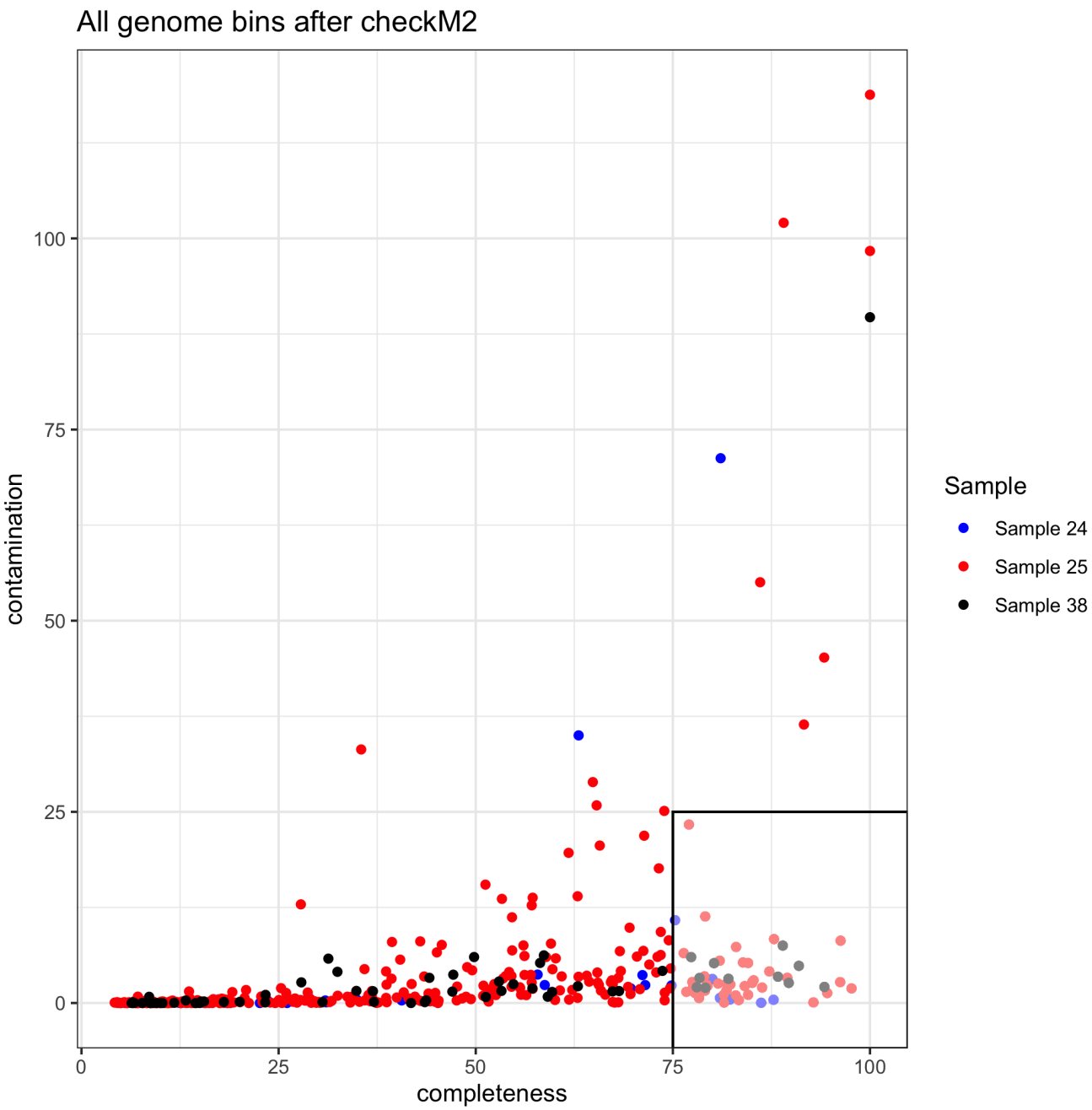
The data used was generated from `CheckM2`, which assessed the quality of the bins based on contamination and completeness. The next following step in the metagenomic pipline is de-replication (`dRep`), which by default selects bins based on completeness >75 and contamination <25.

These critera are not particularly 'high' quality. Therefore, we decided to do investigate how fitting these criteria are, relative to our samples.

The data is obtained from the pre-treated CheckM2 quality report csv files.

All data analysis for this section was conducted in `R`.

The following plot shows contamination vs. completeness for all the bins. The black rectangle, indicates all the bins selected by dRep.



The plot reveals that the vast majority of bins fall bellow dRep's already 'low' criteria.

For clarity, we re-plotted the bins, only including those which passed the quality screening.

The light green section highlights literature-based suggestion for what a 'medium' quality thresholds and the dark green section highlights 'high' quality.

As can be seen, the vast majority of bins - save for 3 outiers - fall comfortably within a literature-suggested 'medium' quality bin. On the contrary, only 2 bins for sample 25, qualified for 'high' quality bins.

This indicates, that while majority of bins used in the final TaxID process do not quality as 'high' quality, all - except 3 - are within the 'medium' threshold, indicating that the bins are certainly not 'low' quality.

3) MICROBIAL DIVERSITY STUDY (TaxID)

Having found investigated the AMR gene class distribution - and assured that our bins are of an acceptable quality - we will now investigate bacterial taxonomical distribution.

This will be done by extracting the taxonomica annotations (or TaxID's), generated from `Gtdb-tk`. While there are different algorithms used for determining TaxID's, for simplicity we have selected one - namely: 'Closest Placement Taxonomy'. These ID's can be accessed from its respective column in the respective `summary.tsv`'s, and separated by a string split into taxonomical classifications (i.e. family, genus, species etc.).

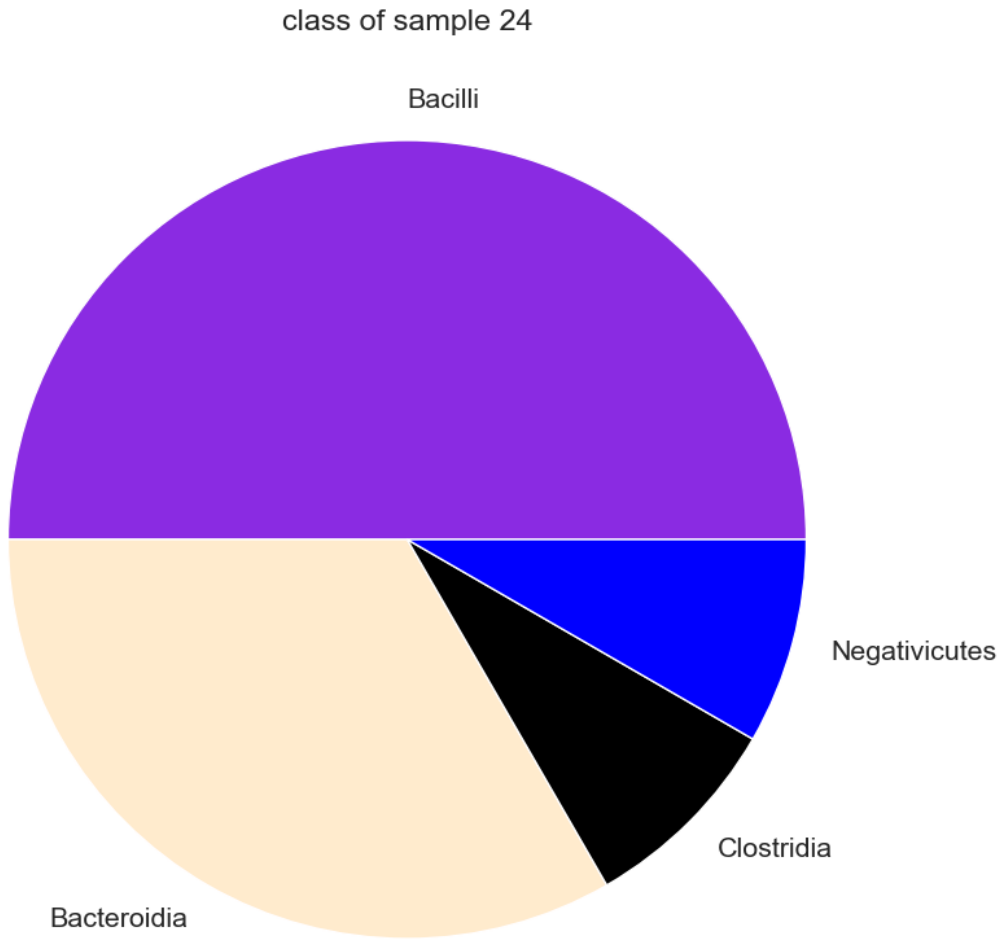
This is done in `R`, and the csv's are imported for the final microbial diversity study.

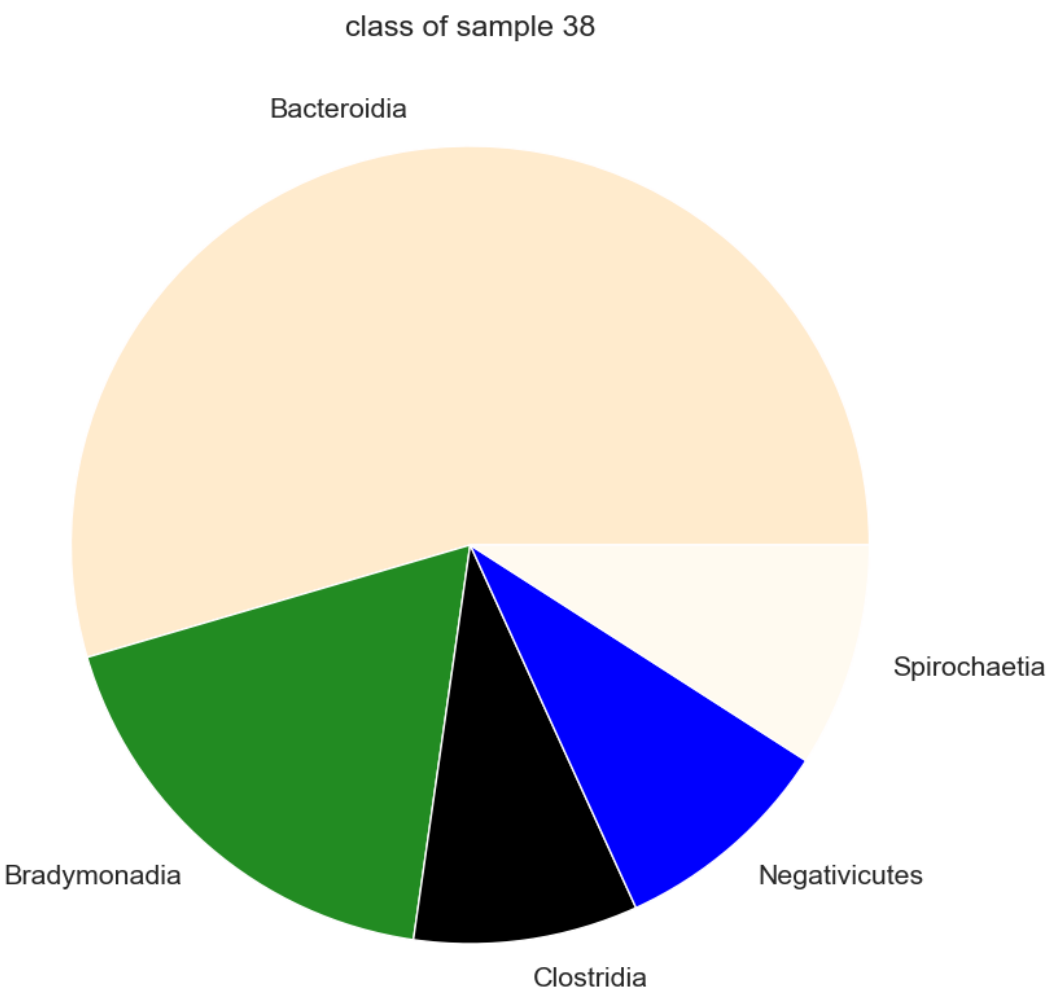
We use the value_counts function to count instances of each element at each classification level.

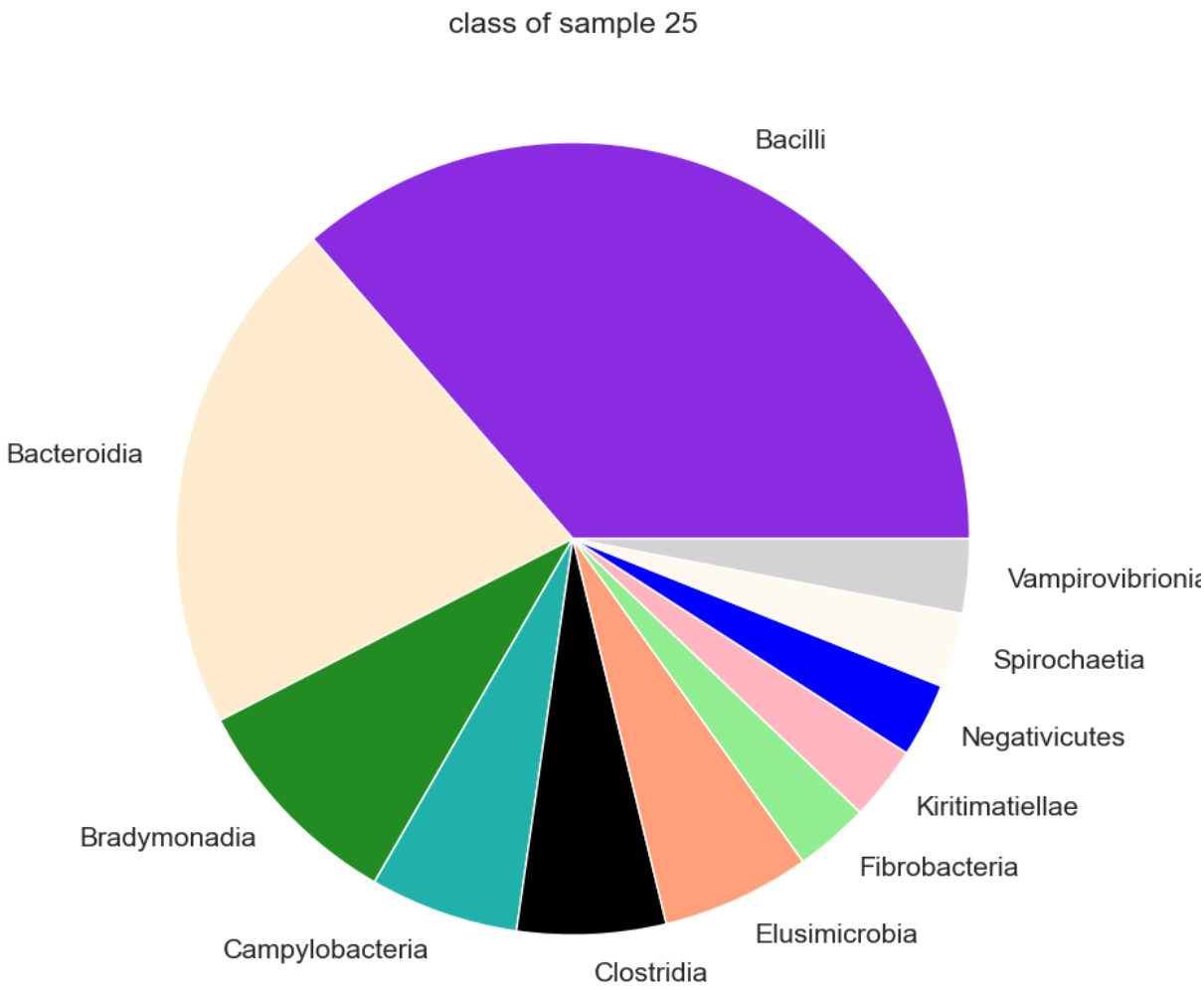
We make a dictionary with all names in all samples, so we can consistently color across samples.

We can now make a pie chart for each sample for the different classification levels.

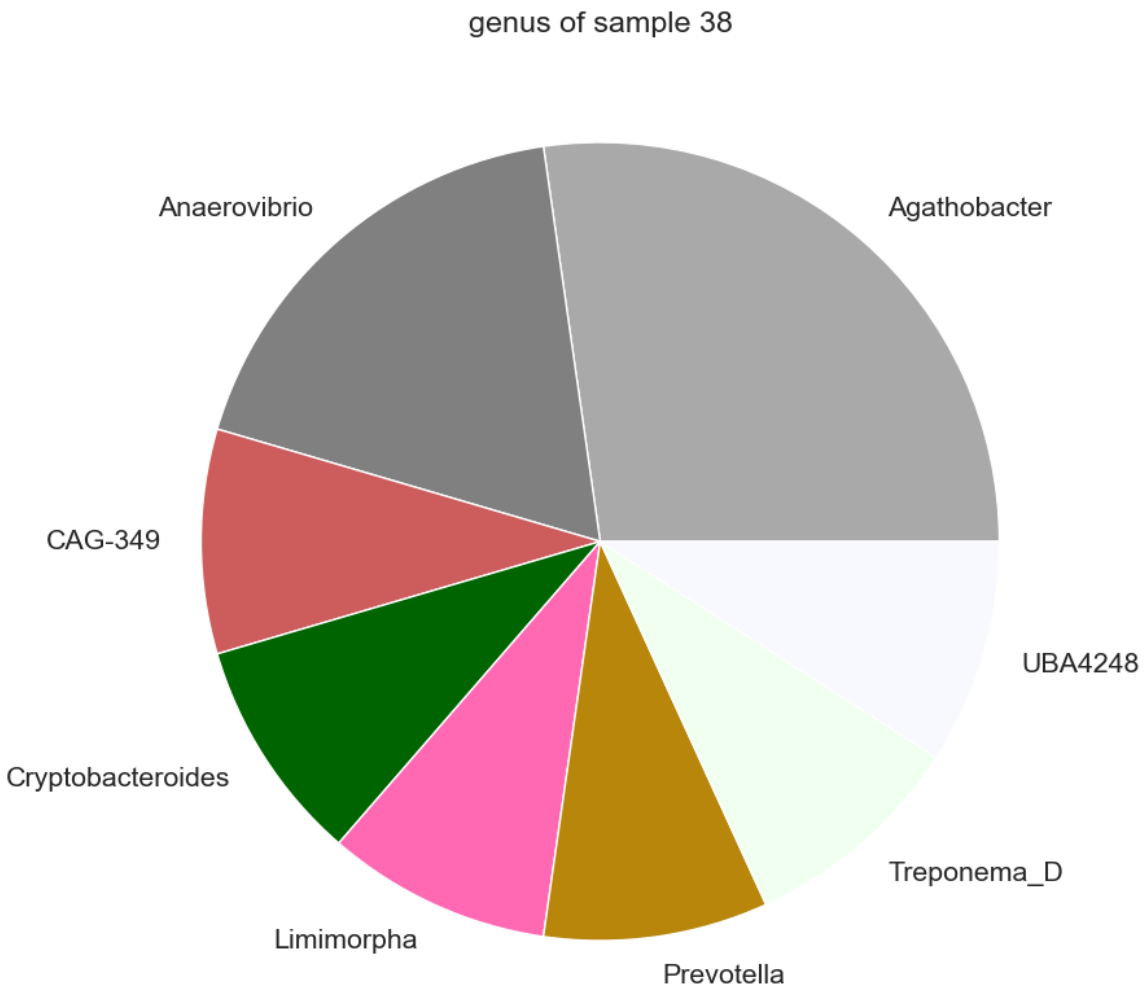
By looking at the pie-charts we can see that the classes represented in sample 38 are different than the other two samples. Samples 24 and 25 has a significant portion of Bacilli present, but Bacilli is not detected in 38.


class of sample 24
Bacilli
Negativicutes
Clostridia
Bacteroidia


class of sample 38
Bacteroidia
Spirochaetia
Negativicutes
Clostridia
Bradymonadia



Additionally, we can inspect the distribution for any classification level for any sample. We can for example see that sample likely contains multiple bacteria from the Agathobacter genus.



References

- Quinn, T. P., Erb, I., Gloor, G., Notredame, C., Richardson, M. F., & Crowley, T. M. (2019). A field guide for the compositional analysis of any-omics data. 8, 1–14. https://doi.org/10.1093/gigascience/giz107
- Zhao, Y., Li, MC., Konaté, M.M. et al. TPM, FPKM, or Normalized Counts? A Comparative Study of Quantification Measures for the Analysis of RNA-seq Data from the NCI Patient-Derived Models Repository. J Transl Med 19, 269 (2021). https://doi.org/10.1186/s12967-021-02936-w