BINF7001 Metagenomics M4 | Part 2

2.1) Intro to gene-based functional analysis and genome-centric metagenomics

This week, we will be comparing **functional profiles** from the gut microbiome of the superworms reared on bran, polystyrene (PS), and the no feed control. Functional profiles provide us with information about the encoded functional potential of a microbial community. In this case, these profiles have been generated by classifying the reads from each sample into orthologous gene families based on **KEGG Orthology (KO)**, a so-called gene-centric analysis. Reads were aligned against a subset of the UniProt database that are annotated to encode KO gene IDs using a very fast protein similarity search tool called **DIAMOND**. This tool is up to twenty thousand times faster than BLAST and therefore allows analyses that may have previously taken months to compute. Classified reads are then collated into a count table structured the same way as a taxonomic profile, except that the variables are (in this case), KO gene IDs rather than OTU IDs.

Q: Because we have used shotgun sequencing reads in this analysis, we can identify the function encoded in the community. How could we also investigate which organism contributes the identified functionality. How would we obtain this information?

2.2) Setting up the workspace, get annotation from KEGG

1. Copy all the R files from UQ Blackboard onto your own computer and unzip the compressed file. This will result in several directories.

2. Open **RStudio**, set the **Code** folder as your working directory, open the **BINF7001_M4P3_genecentric.R** script and install the necessary packages/libraries. **Note: you need to run R version 4.2.1** (https://www.r-project.org/) for this script to work. Remember: libraries require installation only once, but they will require loading each time your environment is reset (e.g. if you have to restart RStudio or you clean your workspace with the broom). Once you have installed the libraries, 'comment them out' with a hash in front of the command so that they will not be executed during the practical.

3. **Load the data.** This can be achieved by running lines **65-70**. We have used these superworm samples in the previous practicals, however, we have now added additional samples, i.e., faeces samples from the superworms in the polystyrene (PS; see picture on the right; © Enfeng Li 2022) and the bran group.

4. For each sample, the number of sequences classified to each KO are summarised in the KO_superworms.txt file and the sample details in the metadata.txt file. We will attempt to normalise this dataset with a differential analyses package that you are all familiar with, from the last practical - DESeq2. Run the code from line **72-103**.

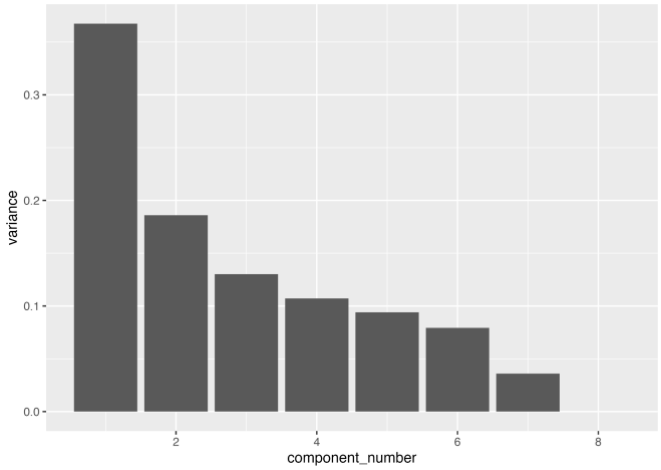
Q: Within this function, the KO table is normalised to account for sequencing depth. Why is it necessary to do this?

2.3) Exploratory analysis with PCA

As discussed in the previous practical, **PCA** is a dimension reduction technique useful in identifying the similarity between samples, especially useful when the data set contains more variables (OTUs or KO gene IDs in this case) than samples. Here we will use PCA again to show the relationship between the functional profile of each sample, then visualise the result and identify any groupings of samples in the data.

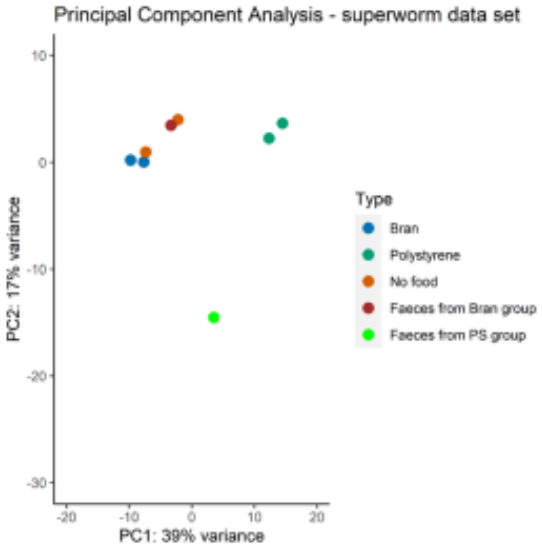
1. PCA will calculate the first, second, third, etc components, each of which explain progressively less of the variance between samples. It's important to check how much variance each component is explaining, as this helps interpret the relevance of the results. This can be done using a **scree plot**. Run the code on lines **110-126** and generate this plot. For PCA to be an informative technique, the scree plot should show **the first few components explaining a substantial proportion of the variance** and the remaining components describing a minority of the variance. If all components explain roughly the same amount of variance, it may indicate that there is not sufficient structure in the data to distinguish the samples at a reduced dimension.

Q: How much variance does the first, second and third components explain? Does this distribution look as though PCA is a useful technique for these data?



2. Next run the code on lines **129-148** to draw the PCA plot (using the first two dimensions).

Q: Do you see any patterns with the variable you have chosen?

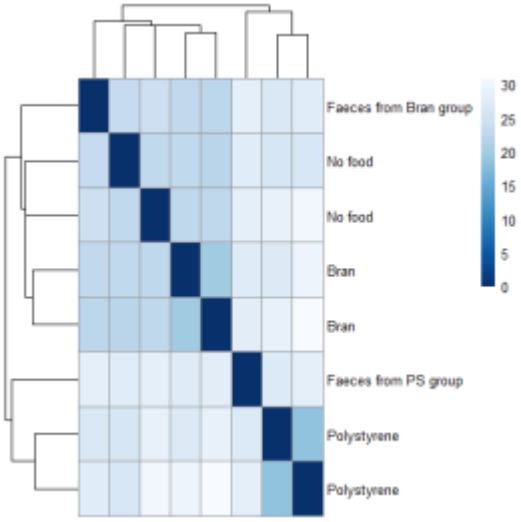


2.4) Exploring normalised abundance of dominant KOs using a heatmap

Many of the exploratory methods we used in the previous practical are applicable and useful for the functional data analysis here. This includes PCA plots, heatmaps and differential abundance (DA) analysis. Now, we will generate a **heatmap** for all sample types to determine what are the most commonly encoded KO groups and whether any KO genes are abundant but not frequently observed.

1. First, we will carry out the sample clustering. Here, we apply the dist function to the transformed count matrix to get sample-to-sample distances. Run the code from line **152-160**.

2. A heatmap of this distance matrix gives us an overview of similarities and dissimilarities between samples. We have to provide a hierarchical clustering (hc) to the heatmap function based on the sample distances, or else the heatmap function would calculate a clustering based on the distances between the rows/columns of the distance matrix.



3. For each KO you will need annotation information that contains KEGG ID, KEGG name, module and pathway. All of this information is available within the KEGGREST package in R. Within the package, there are options keggLink() and keggList() that will help with linking each KO to its annotation information and convert it to a dataframe. Execute lines **170-231** for this.

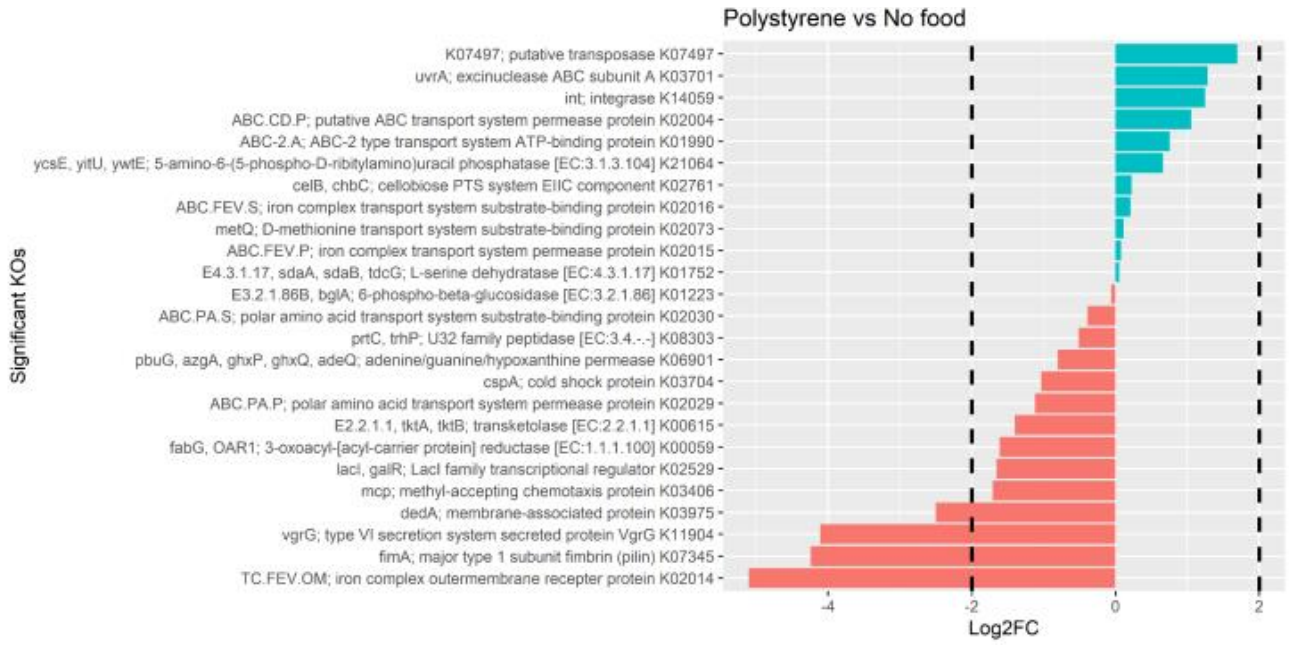
4. Execute lines **240-270**, to create a pretty heatmap, and to find the most abundant kegg. These lines will use the annotation information for your pheatmap.

Q: Which KOs are highly abundant across all samples? Can you find differential KOs that are more abundant in one of your sample types?

2.5) Differential abundance (DA) analysis

There are many KO gene IDs in this data set. Q: How many? Of all these variables, how do you determine which are more abundant in one sample type than the other? We can use variable selection methods, one of which is DA. You were introduced to this method in the previous practical.

1. Run the code on lines **320-442** to calculate the log fold changes, mean abundance, and p-value of the differences in abundance for each KO gene between the sample types, shown as a waterfall plot.



2. Investigate the plot and identify some interesting KO genes that are DA. Note, when interpreting the plot that Log2FC is +ve it will be the first group, and if it is –ve it will be the second group in that is differentially more abundant. Then create a boxplot of the abundance of each gene for all samples by each group using the code below (assign the KO ID to 'ko' to as required).

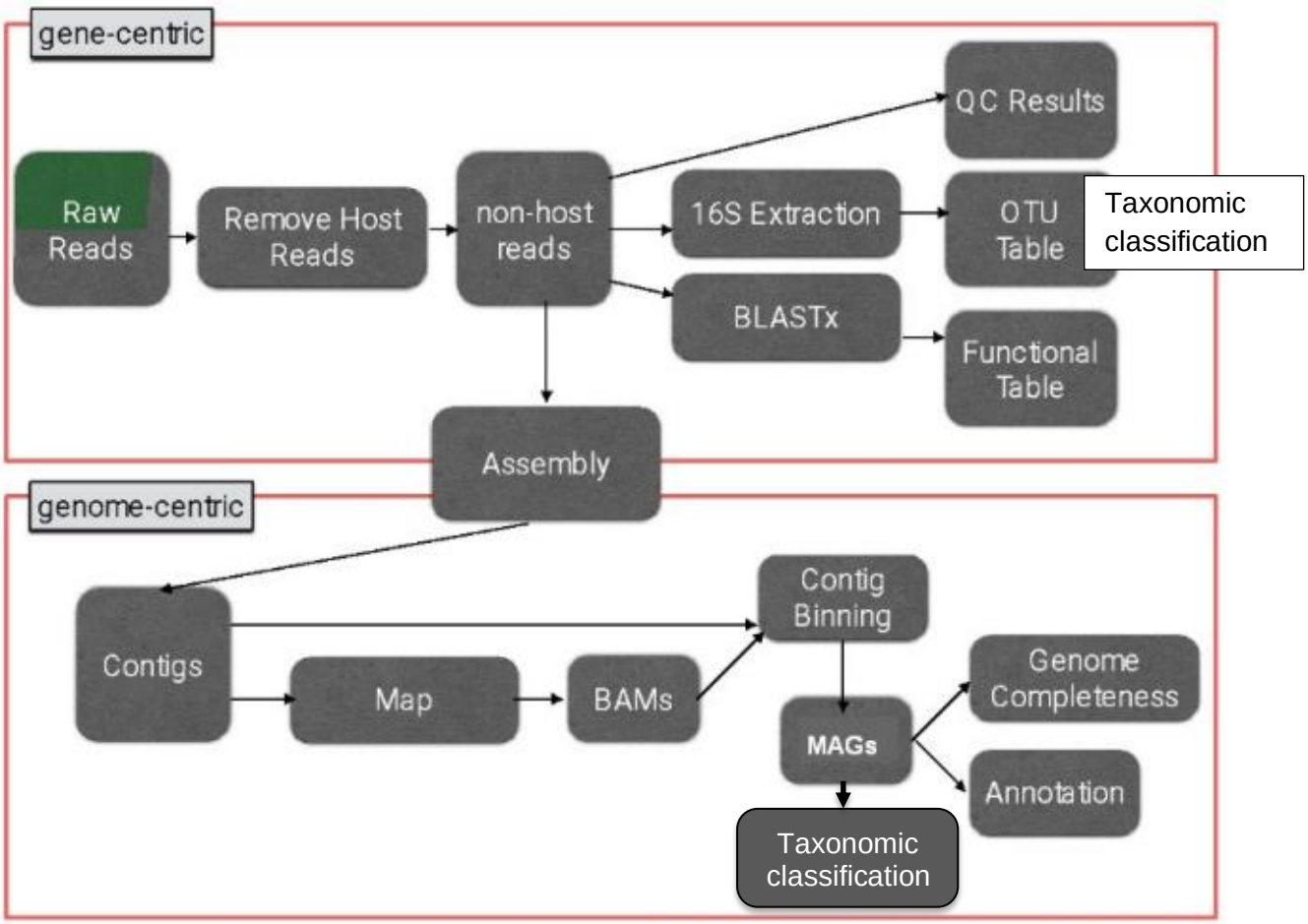
```
ko <- 'K01990'

boxplot(count ~ Type, plotCounts(dds, gene = ko, intgroup = 'Type', normalized
= T, returnData = T), ylab='Counts', main = ko)
```

2.6) Recover genomes from metagenomes (genome-centric analysis)

A major question here is how do you associate these changes in functional capacity to a specific species (or group of very closely related) bacteria?

A gene-centric analysis can tell us what's going on in the community, and how this differs between samples, but it cannot be used to assign a specific function to any organism. For this, we need to assemble the reads into **contigs**, then bin those contigs together, resulting in a "metagenome-assembled genome (**MAG**)". A MAG is also called a "**population genome**" rather than simply a "genome" because assembly of a metagenome that contains closely related lineages (high shared k-mer content, *e.g.* strains) collapses them all into a single representative genome.

We can then annotate the MAGs and determine their **functional potential**. The figure below lists these steps. The result of this process is a list of MAGs, which are then assessed for completeness and contamination (usually using the tool "CheckM"). Finally, good quality population genomes are annotated so that their functional potential can be explored. Then, for instance, we can look at the genomes more prevalent in the PS group compared to the other groups and determine which functions their genomes encode.



MAG recovery is a relatively expensive process computationally, so the following steps will already have been performed ahead of today's practical. First, the reads were assembled into contigs using a popular sequence assembler called Spades using the **metaSPAdes** option - a pipeline for metagenomic data sets (https://github.com/ablab/spades#meta). MAGs are then recovered from the assembled data using a process called "binning" that capitalises on characteristics of the contigs such as their GC content, the use of tetranucleotides (composition), or their coverage (abundance).

MAG quality was assessed using **CheckM**, which estimates completeness and contamination based on single copy marker genes (https://github.com/Ecogenomics/CheckM). You will run CheckM below on a subset of the superworm MAGs.

MAGs were taxonomically classified using **GTDB-Tk** https://github.com/Ecogenomics/GTDBTk, which is based on the Genome Taxonomy Database (GTDB).

Feel free to explore bins at:

- https://www.microbiologyresearch.org/content/mgen/10.1099/mgen.0.000842.T1?fmt=ahah&fullscreen=true.
- https://ncbi.nlm.nih.gov/datasets/genome/?bioproject=PRJNA801070

2.7) Perform quality assessment on a subset of superworm MAGs

1) Three bins have been provided to you, perform a quality check using checkM.

The bins are located at: **/opt/BINF7001/2024/Prac12_2024/1_genome_recovery/bins/**

Symlink the files to your directory.

Run checkM using the following parameters:

#checkm bin_input output_dir arguments

checkm lineage_wf ./ checkm_results -x fna -t 5 --tab_table -f checkm_results.tsv

This will take 5-10 minutes, be patient.

Q1: What is the completeness of your MAGs? Is there much contamination?

Q2: Approximately how big are these MAGs?

2) Evaluate the taxonomy of the superworm MAGs

The next step in our workflow is to assign a taxonomy to the recovered bins using GTDB-Tk. This step is too computationally expensive to perform in the practical. Therefore, we have analysed all the MAGs from the superworm dataset with CheckM and then fed these MAGs and the CheckM output into GTDB-Tk.

Obtain the GTDB-Tk output file**:**

/opt/BINF7001/2024/Prac12_2024/1_genome_recovery/checkm_gtdbtk_35MAGs.xlsx

Transfer it to your desktop and open the file in Excel.

Q1: Assign your MAGs to high, medium or low-quality using the table at this link. What is the highest quality MAG and to which lineage is it assigned?

Q2: What is one of the main causes of contamination?

Q3: Do these MAGs resemble any that have already been sequenced? To answer this, take ~2000 bp regions from a contig in each bin and BLAST at NCBI against RefSeq (http://blast.ncbi.nlm.nih.gov/).

2.8) Evaluate the functional profiling of the superworm MAGs

As you learned in Module 3 with CX, genome annotation involves identifying genes (open reading frames and non-coding RNAs) and assigning function to as many of these predicted genes as possible. MAG of high enough quality can be annotated using this process as you would a standard isolate genome. Genes in the MAGs have been identified and extracted using Prodigal. Each sequence has then been annotated with the metagenomic annotation tool, DRAM (https://github.com/WrightonLabCSU/DRAM).

Obtain the DRAM output files**:**

- /opt/BINF7001/2024/Prac12_2024/2_genome_annotation/all_bins_dram_distill.xlsx

- /opt/BINF7001/2024/Prac12_2024/2_genome_annotation/all_bins_dram_product.tsv

- /opt/BINF7001/2024/Prac12_2024/2_genome_annotation/all_bins_dram_annotations.tsv

Transfer them to your desktop and open files in Excel.

1. Work through your list of genes (KO gene ids, modules, and pathways) identified as potentially interesting in previous exercises from gene-based functional profiling, and search for these genes in the MAGs. Specifically, see files **all_bins_dram_distill.xlsx** and **all_bins_dram_product.tsv**. These are the main processed results provided by DRAM.

all_bins_dram_distill.xlsx provides an overview of the metabolic functions annotated in each MAG, including gene copy number (listed as KEGG, CAZy, etc. IDs), and the corresponding functional pathways/modules that the genes belong to.
all_bins_dram_product.tsv describes, for each MAG, the pathway coverage/completion (e.g., glycolysis) and presence of specific functions (e.g., mcrA, methanogenesis).
all_bins_dram_annotations.tsv contains all annotations for all predicted open reading frames, and you will only need this file if you need more detail than that provided in the summary files **all_bins_dram_distill.xlsx** and **all_bins_dram_product.tsv**.

2. Record the MAG/gene ID of interesting MAGs/genes and the lineage or taxonomy of the MAG if this information is available.

This concludes the practical for this week. We hope you have enjoyed and learned from this foray into metagenomic sequence analysis. Next week we have set time aside to discuss any questions you have with your assignment, and to ensure you understand the concepts in these practicals.