COG_analysis-FS

- The proportion of functions in the core and accessory genome was calculated:
    - Individual core and accessory genome lists were created from the ggCaller output file ‘gene_presence_absence.Rtab’ (renamed 'pangenome.csv') using an R-script:

```
# Load necessary library
library(dplyr)

# Step 1: Load the data
# Replace "Pangenome.csv" with the correct path to your file
data <- read.csv("pangenome.csv", header = TRUE, row.names = 1)

# Step 2: Calculate presence/absence across genomes
# Sum across rows to count how many genomes each gene is present in
presence_counts <- rowSums(data)

# Step 3: Identify the core genome
# With 35 genomes, genes present in all genomes will have a count of 35
core_genome <- data[presence_counts == 35, ]

# Step 4: Identify the accessory genome
# Genes present in less than 35 genomes
accessory_genome <- data[presence_counts < 35, ]

# Output the results
# Writing the core genome to a file
write.table(core_genome, "core_genome.txt", sep="\t", quote=FALSE, col.names=NA)

# Writing the accessory genome to a file
write.table(accessory_genome, "accessory_genome.txt", sep="\t", quote=FALSE, col.names=NA)

# Optional: Print the number of core and accessory genes
cat("Number of core genes:", nrow(core_genome), "\n")
cat("Number of accessory genes:", nrow(accessory_genome), "\n")
```

    - Using an R script (note: the file path must be redefined for core and accessory lists), the proportion of functions in the core and accessory genome were calculated by reading the CDs in the core and accessory lists to the COG categories in the eggNOG annotations output, and new lists were created with this combined information:

```
# Load necessary library
library(dplyr)

# Load the CSV files
accOnly <- read.csv("accOnly.csv")
eGGNOGAnnotations <- read.csv("eGGNOGAnnotations.csv")

# Merge the dataframes
# Assuming 'UniProt_ID' in coreOnly corresponds to '#query' in eGGNOGAnnotations
merged_data <- accOnly %>%
  left_join(eGGNOGAnnotations, by = c("UniProt_ID" = "X.query"))

# Select only the necessary columns (assuming 'UniProt_ID' and 'COG_category')
output_data <- merged_data %>%
  select(UniProt_ID, COG_category)

# Save the new dataframe to a CSV file
write.csv(output_data, "AccGenes_with_COG.csv", row.names = FALSE)

# Print a message indicating success
cat("The new CSV file 'AccGenes_with_COG.csv' has been created successfully.")
```

- The counts and % of COG categories (represented by letters in eggNOG annotations) in core and accessory genomes were calculated and put into lists using an R script:

```
# Load necessary libraries
library(dplyr)

# Step 1: Read the CSV file
data <- read.csv("CoreGenes_with_COG.csv")

# Step 2: Count the occurrences of each COG category
cog_counts <- data %>%
  group_by(COG_category) %>%
  summarise(count = n())

# Step 3: Calculate the percentage for each category
cog_counts <- cog_counts %>%
  mutate(percentage = (count / sum(count)) * 100)

# Step 4: Display the results
print(cog_counts)
```

- These lists were copied and pasted into Excel.

- On excel, COG counts and % in core and accessory genomes were manually separated into sub and main COG categories (sub-category COG are just represented by letters and main category COGs were a combination of these letters under more generalised functions. Done in excel sheet 'Working_out' and described in the following steps):
  - ggCaller mixed together many sub-category COGs (just letters), so these letters had to be manually distributed out to single categories, e.g. mixed category KLV would count as 1 K, 1 L, and 1 V.
  - Once the sub-categories were recounted, they were put into a sub-category table 'SubCat' showing % of COG sub-categories identified in core and accessory genomes.
  - For the main COG category table, sub-categories were grouped together and put into main categories, and an overall COG percentage was calculated, e.g. J, K, and L would come under category 1 (information and storage processing), and when combined, this category would be 13.53% of the core genome and 18.37% of the accessory genome. This data was then made into a table 'MainCat' showing the % of main COG categories identified in core and accessory genome.