Post-industrial_modern_analysis-FS

- Apart from coverage, collecting data for post-industrial and modern analysis was done manually.
    - Using AMR++ outputs for post-industrial and modern data, genes under 100 reads were filtered out and entered into an excel sheet.
    - Cmd + F function was used to count genes and were then made into percentages.
    - Percentages were converted into a bar chart (PostVsModern-Coverage_graph;Figure 4).

- Coverage was calculated using in the following steps:
    - A csv file ('postIndModern50TArgs.csv') contained the required metadata, with one column naming the period (i.e. modern or post-industrial), MEG number, ERR accession (sample), and read count.
    - Two directories 'modern' and 'post-Industrial' containing sample fastq files were created.
    - A python script was made to create multiple scripts that used the information from the metadata to map individual BAM files to their MEGARes reference genes and calculate coverage:

```
amr_metadata = '/storage02/or-
microbio/coverage_mapping_new/sbatch_job_scripts/postIndModern50TArgs.csv'
fixed_location = '/storage02/or-microbio/'
AMR_directory = '/storage02/or-microbio/AMR_download_genes/'

location_dict = {'post-Industrial':'/storage02/or-microbio/coverage_mapping_new/post-
Industrial',
'modern':'/storage02/or-microbio/coverage_mapping_new/modern'}

with open(amr_metadata, "r") as meta_in:
    for line in meta_in:
        if not line.startswith('grouping'):
            line = line.split(",")
            if line[0] in location_dict:
                with
open("{0}_{1}_{2}.sbatch".format(line[0].strip(),line[1].strip(),line[2].strip()), "w") as
script_out:
                    script_out.write("""#!/bin/sh
#SBATCH --job-name=AMR_map\n
#SBATCH --account=43070_203920\n
#SBATCH --partition=compute\n
#SBATCH --cpus-per-task=10\n
#SBATCH --time=2:00:00\n


module purge\n
module load BWA\n
module load SAMtools\n
\n
bwa mem -v 3 -t $SLURM_CPUS_PER_TASK -reference {0}{1}.fasta {2}/{3}_fixed_R1.fastq.gz
{2}/{3}_fixed_R2.fastq.gz |\n
samtools sort --threads $SLURM_CPUS_PER_TASK --reference {0}{1}.fasta -o
{4}coverage_mapping_new/{5}/{1}_{3}_mapped.bam\n
cd {4}coverage_mapping_new/{5}\n
samtools coverage -o {1}_{3}_coverage_table.tab {1}_{3}_mapped.bam\n""".format(AMR_directory,
line[1].strip(), location_dict[line[0]],line[2].strip(), fixed_location, line[0].strip()))
```

- This script outputted individual coverage tables which were collated using another python script:

```
import glob

print("period,gene,sample,Nreads,coverage")
for files in glob.glob("/storage02/or-microbio/coverage_mapping_new/**/*_coverage_table.tab",
recursive = True):
    details = files.split("/")
    gene, sample = "{0}_{1}".format(details[5].split("_")[0], details[5].split("_")[1]),
details[5].split("_")[2]
    with open(files, "r") as f_in:
        for line in f_in:
            if not line.startswith("#rname"):
                line = line.split("\t")
                print("{0},{1},{2},{3},{4}".format(details[4], gene, sample,
line[3],line[5]))(base) [fjst(ba(b(((((((((((((base) [fjstande@login01(Bradford-HPC)
coverage_mapping_new]$ cat collate_coverage.py
import glob

print("period,gene,sample,Nreads,coverage")
for files in glob.glob("/storage02/or-microbio/coverage_mapping_new/**/*_coverage_table.tab",
recursive = True):
    details = files.split("/")
    gene, sample = "{0}_{1}".format(details[5].split("_")[0], details[5].split("_")[1]),
details[5].split("_")[2]
    with open(files, "r") as f_in:
        for line in f_in:
            if not line.startswith("#rname"):
                line = line.split("\t")
                print("{0},{1},{2},{3},{4}".format(details[4], gene, sample, line[3],line[5]))
```

- Data from this coverage table was copied and pasted into Excel and converted into a box and whisker plot (Figure 5).