One ARM Project 4 Metagenomics Pipeline

Gerald Amiel

September 18, 2024

1 Project Overview

Metagenomic analysis of ARGs in NCR hospitals, wastewaters, and surface waters

2 Legend

- ✓ Done
- ☐ Pending
- ⟳ Needs refinement
- △ Unexpected issues
- ✎ Drafted
- ➔ Moved

- Essential
- Optional
- Robust
- ~~Deprecated~~

3 HPC preparation

- ☐ Confirmation of Robustness of analysis
  - ☐ Agree upon the type of analyses
- ✓ Set up config files
  - ☐ ARG-metagenomics pipeline
- ☐ SLURM
  - ☐ Setup Docker containers for SLURM
  - ☐ SLURM container for simulations
  - ☐ Automation of SLURM requests

4 File management

- ☐ BAM file parsing
- ☐ Interconversion between SAM and BAM

5 Raw read processing

- ✓ Read quality control
  - ☐ Data visualization
  - ☐ Determination of optimal tool/s
  - ☐ Determination of optimal parameters
  - ☐ Tool combination randomization script

6 Assembly & Binning

- ☐ Testing binning pipeline
  - ✓ CD-HIT-EST clustering
  - ✓ Assembly tools
    - ✓ MEGAHIT
    - ☐ VELVET
    - ☐ meta-SPAdes
  - ☐ Benchmarking between all of them
  - ✎ Iterative assembly
  - ✎ Contig quality checking
- △ MetaWrap binning
  - ☐ Testing on datasets of higher depth
- ☐ Refinement of bins

7 Annotation

ARGs

- ✓ Identification of relevant ARG databases
  - CARD
  - NCBI AMR Database

- ⟳ Script to identify ARGs from short-reads
  - ⟳ RGI
  - ⟳ ShortBRED
  - ⟳ AMRFinder
- ⟳ Script to quantify ARGs per sampling site
  - ⟳ ShortBRED
- ✓ Script diversity calculations per site
- **Taxonomy**
- Rapid taxonomy via short reads
  - ☐ MetaPhlan4

Genomic context

- **Network analysis**
- **Associated pathogens**
-

8 Complete

General taxo-metagenomics pipeline

- ✓ Integration with config folder
- ✓ Testing with simulated datasets
- ✓ Pipeline integration
  - ✓ FastQC
  - ✓ Trimmomatic
  - ✓ Kraken2
  - ✓ Bracken
  - △ ~~QIIME2~~
  - ✓ Change QIIME2 to in-house diversity script
  - ✓ Aggregate diversity metrics

[☐]

File management

- ✓ Compression and decompression scripts
- ✓ Demultiplexing script

Read quality control

- ✓ Initial raw read QC
- ✓ Standard Trimming (Trimmomatic)
- ✓ Raw Read QC after trimming
- ✓ Parametric randomization script
- ✓ Parsing QC metrics of all iterations
- ✓ Bootstrapping script

Pipeline preparation for HPC

- ✓ Calculate using quotation from PGC
- ☐ Set up config files
- ✓ Taxo-metagenomics pipeline

ARGs

- ✓ Identification of relevant ARG databases
  - CARD
  - NCBI AMR Database
- ⟳ Script to identify ARGs from short-reads

Detection of HGT

HGT mechanisms

Script Descriptions

Pipelines

This section covers all the scripts that were created during the Project.

Listen, I'm eternally curious and I don't just want to settle for any random journal. No offense, but I'm aiming for something high-impact!

These were created during off hours or during work hours, so some scripts might seem irrelevant at first, ***but trust me, there's a conscientiousness or meticulousness to this madness.***

I have separated them into folders/repositories (shameless plug here: https://github.com/GABallena) based on their relevance to the project.

- **Project4** - Essential scripts directly related to the core analyses of the project.
- **Side** - Scripts that can potentially be used to increase the robustness of the paper.
- **Main** - Scripts related to file organization and data management, crucial for handling large datasets.

ARG-MGE.smk

Stage: Draft

General Purpose

This pipeline is designed for **comprehensive metagenomic analysis of ARGs**, it also includes placeholders for analyses of **mobile genetic elements (MGEs)**, and **plasmid detection**. It integrates tools for **read quality control, assembly, annotation, taxonomic profiling, and structural variant (SV) detection** to provide a high-resolution view of the genetic components in metagenomic samples.

Specifics:

1.1 Preprocessing

- **It first uses PEAR to merge paired-end reads.**
  Technical Notes: PEAR (Paired-End reAd mergeR) compares paired-reads to correct infer the likely bases on its associated pair
  Rationale: The main goal here is to ensure good read quality for downstream analyses and to maximize the amount of data by reducing gaps in the sequence and improving confidence of base calls within that region.
  Note: Another tool, PandaSeq which does the same thing is used later, this usage of PEAR here is because it is more optimized for larger datasets - which in this case are trimmed reads.

- **Followed by conversion from FASTQ to FASTA using `seqtk`.**
  Technical Notes: `seqtk` converts compressed FASTQ files to FASTA.
  Rationale: Some tools cannot work on FASTQ formats.

- **Translation of the CARD-protein homologues database `transeq`, and then reverse-translates these proteins to nucleotides for alignment.**
  Technical Notes: `transeq` from EMBOSS translates nucleotide sequences to protein sequences based on **standard genetic code**. `backtranseq` reverses this to allow iterative alignments with nucleotide sequences.
  Rationale: This leverages conserved protein-level information, which is lost at the nucleotide level due to synonymous mutations - while also increasing the sensitivity to potential ARG proteins from k-mer alignment. See Nature Methods, doi: doi.org/10.1038/s41592-019-0437-4 (2019) for more details.

1.2 Metagenomic assembly of ARG-contigs

- **Iterative alignment is performed using `KMA` to align reads against reverse-translated CARD sequences for high-sensitivity identification.**
  Technical Notes: `KMA` (K-mer Alignment) is used find the reverse-translated ARGs with the proximity filtering option to determine the surrounding regions of ARGs. This is done iteratively for each gene increasing the ARG-associated database.
  Rationale: Firstly, `KMA` is used because, unlike `Bowtie2` and `BWA-MEM`, which were created specifically for Human metagenomics, KMA does not suffer (or suffers less) from multi-allelic databases. Secondly, iterative alignment using this process allows us to contextualize the region wherein ARGs reside - thereby narrowing our focus onto these local

regions instead of looking at the global genomic context. **Notes:** The script is designed to have a cap on the number of iterations KMA creates, increasing the database size.

- **Merging ARG-related reads database with `PANDASeq`.**
  Technical Notes: `PANDASeq` is then used to further refine the paired-reads collated in the ARG-related genes database.
  Rationale: `PANDASeq` was chosen as, while being slower than `PEAR`, it is more accurate. This mergins step is included to ensure that only high-confidence reads are assembled.
  Note: We leverage the fact that the ARG-related genes database is smaller compared to the raw metagenomic reads database.

- **Merged reads are assembled into contigs using `metaSPAdes`.**
  Technical Notes: `metaSPAdes` is then used to create contiguous sequences from these local regions by extending them using reads from the whole metagenomic pool. Additionally, Contigs are filtered by length here to remove possible artefacts.
  Rationale: `metaSPAdes` was chosen as, while being slower than `MEGAHIT`, it is optimized in handling highly diverse and mixed microbial populations. CARD is used here because it is manually curated and updated regularly - in order to be included in the database, there must be clinical data (ASTs) involved in the study.
  Notably, this would decrease the sensitivity of our ARGs - and would mostly be biased towards those reported in the clinical setting. To counter this, we could also incorporate other tools such as `ResFinder`, the `NCBI AMR Database`, and `ARG-ANNOT`
  Notes: Filtering of contig length is handled by a python script that determines the minimum ARG sequence dictated by `CARD`.
  Notes: Contigs are further extended using contigextender to form scaffolds
  Notes: Might add other contig extendending programs like `GapFiller` - which leverages mate-pair information

- `Python` **script is created to do another round of checking contig quality for downstream analysis,** `R` **scripts are used to visualize the data.**
  Technical Notes: The `Python` script will measure standard contig quality metrics e.g. N50, L50 etc.
  Rationale: Should be obvious why

- **Confirmation of contigs with ARGs using `RGI`.**
  Technical Notes: `RGI` scan the contigs and check whether which contigs created by `metaSPAdes` have ARGs in them.
  Rationale: `metaSPAdes` may have created contigs that DO NOT contain ARGs, and have instead assembled them into a more matching contig (a false-positive misassembly) - this can happen because of the different databases being used; also parallelization of methods like this increases robustness because it has been confirmed independently from different starting points (bottom-up vs top-down approach). This allows us to filter ARG-containing contigs.
  Notes: RGI is the official scanner of CARD.

1.3 Read-mapping

- **C**hecking coverage statistics on contigs using raw reads
  Technical Notes: `Samtools` is used here to map the raw reads from the larger database back to the assembled contigs and then calculates the coverage over the entire contig.
  Rationale: Read-mapping is a quality control protocol used in metagenomics, to determine the quality of the assembly. High-coverage means that many of the k-mers align well

with that region of the contig, while low coverage is evidence of inconsistent mapping and that the contigs should be refined, split, or discarded.
Note If there are persistent (after further refinement and reassemblies) sudden differences in coverage across a contig, that contig could be chimeric, meaning, it could be from two different populations.
Extra note A `Python` script is included in the pipeline that is determine visualize and check how the coverage changes over contig regions. In general, they could be interpreted as the following:

Smooth, Uniform Coverage: Typically shown by well-assembled contigs.
Sharp Coverage Drop: May need to be split or flagged for reassembly. May also be a misassembly point (chimeric contitg) or caused by a structural variant
Coverage Gaps: Regions with little to no read support; a strong indicator of misassembly.
Gradual Drops: Overlapping reads, repetitive or duplicate regions, partial HGT, sequence heterogeneity, or coverage differences due to a mixed population. Repeats and duplications can be filtered out using tools like `RepeatMasker` or `BLAST`
(to be added to the script)
Sharp increase: May be due to repetitive or duplicated regions, the assembler decided to collapse all the reads into that one contig, amplification bias from PCR, HGT, SV, chimeras.

1.3.1 Read mapping more details

Read mapping tools
Technical Notes: Four (4) Tools will be used in parallel to do the read mapping process `BWA` `Bowtie` `KMA`, and `minimap2`, their parameters have been adjusted to map reads at 95 percent identity to the contigs.

Mapper	K-mer Length	Mismatch Penalty	Gap Opening Penalty	Gap Extension Penalty
BWA	21	5	7	2
BWA	31	4	6	2
BWA	51	3	5	1
Bowtie2	-	4,2	5,2	5,2
KMA	Default	95% identity	Automatic	Automatic
Minimap2	-	5	7,2	4,1

Table 1: Example table of mapper configurations without command example

Rationale: 95 percent identity is used to increase sensitivity - as is standard for determining homologous sequences. This adjustment was made because k-mers are either attach or don't. Parallelization is used to increase robusness.
Note K-mer extension is used to increase the accuracy of mapping. BWA k-mer lengths can be adjusted, while KMA does it by default. The others, cannot be adjusted.

Note (September 18, 2024)

I chose to change the script to not allow gaps during this phase as we already used reverse translation earlier to correct for synonymous codons, and protein sequences are more important when it comes to ARG function, the new values are below. I also added protein-based read mapping.

Tool	K-mer Length	Mismatch Penalty	Gap Opening Penalty	Gap Extension Penalty
BWA	21, 31, 51	5, 4, 3	1000	1000
Bowtie2	-	4,2	1000,1000	1000,1000
KMA	-	95% identity	Automatic	Automatic
Minimap2	-	5	1000,1000	1000,1000

Table 2: Alignment parameters for ungapped alignments across BWA, Bowtie2, KMA, and Minimap2

Tool	Input Files	Key Parameters
tblastn	– Protein sequences: `Translated protein sequences contigs` – Nucleotide database: `Cleaned sequence database`	– `-outfmt 6` – `-evalue 1e-5` – `-gapopen 5` – `-gapextend 2` – `-matrix BLOSUM62`
blastp	– Protein sequences: `Translated protein sequences contigs` – Protein database: `Translated cleaned sequence database`	– `-outfmt 6` – `-evalue 1e-5` – `-gapopen 5` – `-gapextend 2` – `-matrix BLOSUM62`

Table 3: Protein read-mapping parameters for `tblastn` and `blastp`

Rationale: The main rationale for adding a protein-based read mapping protocol is because ARGs are primarily about their ***protein-protein interactions*** (biological relevance). This method also accounts for frameshifts with higher specificity to homologous regions.

On a personal note: This approach may also require further exploration into whether protein-protein interactions are altered—perhaps by investigating changes in binding sites. Which is a story for another day (***why do I do this to myself?***)

1.4 Contiguous chimeras

- **On a parallel process: taxonomic profiling is done using `Kraken2` and `SprayNPray`. and aligned with the ARG-raw reads and ARG-contigs databases**
  Technical Notes: `Kraken2` uses k-mer-based classification to assign taxonomy based on raw-reads. While `SprayNPray` complements this by assigning taxonomy at contig level.
  Rationale: By assigning comparing their respective databases with our ARG-related databases we will be able to connect our reads and/or contigs to their corresponding taxa, uncovering the microbial hosts responsible for carrying and potentially spreading ARGs in the environment.

- **Detects structural variants (SVs) in contigs using `Manta` and identifies *chimeric contigs* based on SVs and taxonomic classification.**
  Technical Notes: `Manta` identifies large genomic rearrangements such as insertions, deletions, and duplications.
  Rationale: Chimeric contigs may be due to systematic error or real biological signals. These chimeric contigs can be detected by Kraken2 and SprayNPray (i.e.) when a portion of a contig is being assigned to different taxa. The rationale behind this step is to investigate whether structural variations are present - which may be evidence of horizontal gene transfer (HGT) events.

- **Or perhaps due to mobile genetic elements like transposons**
  Technical Notes: Tools such as the following can be used:

  - `HMMER3` suite
  - `Tnppred` - a transposon predictor tool

  Rationale:

- **Sketching contigs followed by calculating Bray-Curtis diversity.**
  Technical Notes: `Mash Sketch` uses a MinHash approach to generate a presence/absence profile of ARGs across contigs. This gives us a quick snapshot of what the contigs "look like" in terms of ARG content. For Bray-Curtis diversity, we calculate a dissimilarity matrix from the abundance data of ARGs, followed by a PCoA plot to visualize similarities between contigs based on their ARG profiles.
  Rationale: The `Mash Sketch` helps rapidly identify the genetic makeup of contigs in terms of ARGs, which provides a foundation for further investigation. By applying Bray-Curtis diversity and using PCoA, we can group contigs based on their ARG similarity. If contigs with the same sketch group together, we can trace them back to their taxonomic IDs to identify the microbial hosts.However, if contigs have similar ARG profiles but belong to different taxa, this could serve as **evidence for Horizontal Gene Transfer (HGT)**. This dual approach allows us to trace ARG spread and potential HGT events in a metagenomic context.
  Note Bray-Curtis similarity is most often used as a presence or absence diversity metric.

1.4.1 Plasmids

- **Determination of putative plasmids**
  Technical Notes: The tools listed below will be used in parallel. Plasmids are considered valid when all 4 tools predict plasmid signatures in the contig.

- `PlasPredict` pipeline
- `Recycler`
- `PlasmidFinder`
- `MOBSuite` plasmid marker annotator

If plasmid signatures are present, `plasmidSPAdes` along with `GapFiller` will be used to check if the contig can circularize. `oriTfinder` will then be applied to contigs with fewer than 4 fragments. A `Python` script will calculate GC skews of the chimeric contig and compare it to its taxonomic counterparts. Another `Python` script will normalize the data according to `16S rRNA` from trimmed reads. Lastly, plasmid percentage will be calculated based on reads mapped to putative plasmids over the total reads.

Equation:

$$\text{Plasmid Percentage} = \left(\frac{\text{Plasmid Reads}}{\text{Total Reads}}\right) \times 100$$

Rationale: `oriTfinder` looks for origin of transfer sites (oriT), which are characteristics of conjugative plasmid . This whole sub-pipeline is to look for evidence of conjugative plasmid transfer as the cause of these chimeric contigs. Normalization and percentage counts are used here to further check whether these "plasmids" align with our understanding of the average plasmid copy number.
Note: Will also be drafting a script to do sliding window analysis of GC-skews - as different characteristics of this curve can be interpreted in different ways.

1.5 Phages

- **Phage influence signatures**
  Technical Notes: They will be determined using a variety of tools:
  - `VirSorter`: Identifies viral signatures within microbial genomes and separates prophages from bacterial sequences.
  - `PHASTER`: A web-based tool for phage search and annotation, identifying integrated prophages.
  - `VIBRANT`: A tool that combines several approaches to identify and annotate phage elements in metagenomic sequences.

  Rationale:This analysis aims to detect potential phage signatures in the chimeric contigs. Since phages are mobile genetic elements, their involvement in transferring ARGs through transduction is highly relevant.Phages, especially temperate phages, can integrate into bacterial genomes and excise themselves, sometimes carrying host genetic material, such as ARGs, with them.The integration and excision signatures detected in contigs will provide evidence of possible transduction events in our datasets, supporting the hypothesis of ARG dissemination via phages.

- **Phage Signature Extraction and Phylogenetic Analysis**
  Technical Notes: Phage-associated genes will be extracted from the chimeric contigs, followed by phylogenetic analysis to uncover evolutionary relationships.`FastTree` will be used to build a phylogenetic tree based on the extracted phage genes.For visualization, tools like `iTOL` or `FigTree` can be used to generate an interpretable phylogenetic tree.

  Rationale:Phage genes embedded in chimeric contigs (their taxonomy) may serve as strong evidence of horizontal gene transfer (HGT) events. The aim here is to check the

evolutionary origins of the phage genes found in our dataset and their potential involvement in the dissemination of ARGs.

Future Considerations: Will continue improving this section (everything regarding HGT) by evaluating the results from these tools, aligning them with ARG presence, and refining the approach for identifying conjugation, transposon, and transduction events within chimeric contigs. This may also involve validating phage activity—*again, why do I do this to myself?*

metagenomics_general.smk

Stage: Done

General Purpose taxo-metagenomics

This pipeline is designed for **The essentials in metagenomics**, which includes **quality checking, filtering, and trimming of raw reads to clean reads**. As well as the usual **taxo-metagenomic analysis**.

Specifics:

1.1 Quality control (Pre-processing)

Raw trimming of raw metagenomic data
Technical Notes: `FastQC` is used on **raw reads**
Rationale: This is mainly used as a point of comparison - determine whether the next step (trimming) was effective. This pre-processing step is the starting point in any and all metagenomics pipelines.
Notes: This whole quality control steps are interconnected with each other.

Trimming
Technical Notes: `Trimmomatic` is used on **raw reads**
Rationale: Trimming involves removing low quality bases (often depending on something called the Phred Score - which is just a measure of how "confident" we are that the base on that site was accurate), adapters, and filtering reads that go below a specific length threshold.
Notes: Journals often report the parameters on `Trimmomatic` (or any trimmer they decide to use); this is often so that the study is reproducible, should one decide to actually reproduce the study starting from scratch (raw reads).
Notes: There are many different trimmers each with their own strengths and weaknesses, `Trimmomatic` is just the most popular trimmer and is thus used here, though studies differ in the parameters they used for trimming - which often dictates how strict they are with what they define as "good enough".

Perspective: Why different people choose different trimmers depend on the **strengths and weaknesses** of the trimmer e.g. trimmers like "fastp" is used because it's fast making it suitable for very large datasets like deep sequencing. While some trimmers like `Sickle` has automatic adjustment over the entire sequence - which makes it useful for very ancient datasets where DNA is often highly-degraded. Other times, it's for **convenience** like `Trim-Galore` which combines `FastQC` and `Cutadapt` trimmer in a single command. Another good example is `BBDuk` which is part of a larger package called `BBtools`, `BBDuk` also has an built-in contamination detection - so it's particularly good at filtering out usual contaminants like sequences known to be from the human genome. so you can simply just use all the modules in that package for all-in-one processing. Other times, it's just **familiarity**.

Quality checks of post-trimmed data
Technical Notes: `FastQC` is used on **trimmed reads** to determine how effective the trimming process was.
Rationale: If the trimming process was effective, we should notice a better quality reads here, otherwise, we might have to adjust the trimming parameters.

Notes: Determination of whether the trimmed reads are "clean enough" is more of an art rather than actual science, though thresholds exist like Phred > 20 or Phred > 30 depending on how strict you are as a researcher.

September 18, 2024

1.1 To-do List

- ☐ Call for a group meeting regarding logistics (or when they will be available via Zoom call)
  - ☐ Discuss the issue about VMMC review fee
  - ☐ Raise concerns about data storage and management
  - ☐ Inquire if there are scripts I can develop to streamline the bureaucratic process
- ☐ Contact engineering/sanitation departments via landline
  - Talk about possible sampling sites and that we will present out authorization upon arrival
  - ☐ VMMC
    - Tell them that we are have a go-signal from the admin, but we have yet to pay for the fee - which can be held off for later
  - ☐ Mary Johnston
    -
  - ☐ ManilaMed
    - Tell them we already have a go signal from Ma'am Eula
  - ☐ St. Lukes
    - Inquire where Engr. Valenzuela is there
    - Inquire the status of their renovation and if they are available for sampling this October already as talked about last year

1.2 Completed Tasks

- ✓ raw API results now readable stored in TSV file
- ✓ Created symlinks in private (repo) to .git folders in public repos to enable tracking of Git and Github activities
- ✓ Resolved residual time-tracking activities within the local computer

1.3 Scripts Worked On

- ✓ checkingAPI.sh (now updated to make the raw API results readable)

1.4 Issues Encountered

-

1.5 Need to Troubleshoot

-

September 17, 2024

1.6 To-do List

- ➔ Call for a group meeting regarding logistics
  - ✓ Offer to use landline to minimize on-site visits and reduce transportation costs; as point-persons have not been replying via email as quickly lately
  - ✓ Discuss PGC calculations and talk about allocatable budget
  - ✓ On-site orientation with PGC engineering department (OETS) regarding sampling sites **taken care of by James and Roch**
  - ➔ Raise concerns about data storage and management
  - ➔ Inquire if there are scripts I can develop to streamline the bureaucratic process

1.7 Completed Tasks

- Setup wakatime in VS Code and Ubuntu to track my coding productivity
- Helped in receiving and moving the autoclave and fridge needed by the Program

1.8 Scripts Worked On

- ~~rawAPIformatter.sh~~
- ✓ checkingAPI.sh
- ~~wakatime.sh~~

1.9 Issues Encountered

- ✓ Connection blockage from wakatime API
- ➔ raw API results is unreadable

1.10 Need to Troubleshoot

- conky display of wakatime API for desktop

September 16, 2024

1.11 To-do List

- ➔ Call for a group meeting regarding logistics
  - ➔ Offer to use landline to minimize on-site visits and reduce transportation costs; as point-persons have not been replying via email as quickly lately
  - ➔ Discuss PGC calculations and talk about allocatable budget
  - ➔ Raise concerns about data storage and management
  - ➔ Inquire if there are scripts I can develop to streamline the bureaucratic process

1.12 Completed Tasks

- ✓ Updated GitHub repositories; created new repositories: `Documentation` and `Confidential`
- ✓ Created a new bash script to track progress across all repositories
- ✓ Set up this research diary to document daily progress and tasks
- ✓ Python script that calculates contig quality metrics L50, N50, but also L90, N90, GC skew etc.
- ✓ Python script that detects and calculates changes in coverage over contigs for read mapping
- ✓ Typesetting: explaining the process and rationale inside the ARG-MGE.smk pipeline
- ✓ Integration of the two scripts to the ARG-MGE pipeline

1.13 Scripts Worked On

- `Project4/ARG_MGE.smk` - updated
- ✓ `Project4/calculate_contig_quality.py`
- ✓ `Project4/plot_and_detect_intermediate_coverage.py`

1.14 Issues Encountered

- ✓ Could not access GitHub, needed to reset the SSH keys and update my Git histories for version control

1.15 Need to Troubleshoot

- None identified at this time