**Resistance Gene Association and Inference Network (ReGAIN) Program Manual**

**Installation (work in progress):**

**Prerequisites:**

Python 3.8 or higher

rpy2

**Dependencies:**

To install Python 3.8 or higher:

sudo apt-get install python3.8 (Linux)

brew install python@3.8 (macOS)

pip install rpy2 (or pip3, depending on OS)

**Brew installation:**

/bin/bash -c $(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)

**Note that to ensure ‘brew’ commands work properly, you may need to add Homebrew to your PATH:**

On Z-shell (zsh):

1. nano ~/.zshrc
2. add the following line to the end of your .zshrc file:
   1. export PATH=/usr/local/bin:$PATH

On bash:

1. nano ~/.bash\_profile (or ~/.bashrc)
2. add the following line to the end of your bash profile:
   1. export PATH=/usr/local/bin:$PATH

For both, to save the file, press ‘Ctrl + X’ to exit, then ‘Y’ to confirm changes and ‘Enter’ to fully exit the text editor.

Reload your shell so it recognizes the changes:

1. for Z-shell:
   1. source ~/.zshrc
2. for bash:
   1. source ~/.bash\_profile or ~/.bashrc

**Invoke ReGAIN:**

% regain

**General:**

-h, --help: brings up help menu

**Order of Programs**

1. AMR.py
   1. Input
      1. Genomes in FASTA format
   2. Usage: evaluate genomes in FASTA format for presence of resistance and/or virulence genes
   3. Output
      1. One CSV file of results per input FASTA file
      2. One combined CSV file containing all unique gene values identified across all input genomes
2. matrix.py
   1. Input
      1. Directory containing AMR.py results
   2. Usage: extract relevant data from AMR.py results and create the data matrix needed for all ReGAIN statistical analyses
   3. Output
      1. Presence/absence matrix of genes found across all input genomes in CSV format
      2. Metadata file containing unique gene values in CSV format
3. BN.py
   1. Input
      1. Presence/absence matrix
   2. Usage: use dataframe created from matrix.py to run Bayesian analysis
   3. Output
      1. Raw Bayesian Network (RDS file)
      2. Bayesian Network database (RData file)
4. Query.py or queryFit.py
   1. Input
      1. Bayesian Network database
      2. Metadata file
   2. Usage: query pairwise gene-gene probabilities to obtain conditional probability and relative risk values
   3. Output
      1. Probability table in CSV format
5. network.py
   1. Input
      1. Presence/absence matrix
      2. Raw Baysian Network file
      3. Metadata file
   2. Usage: visualize the Bayesian Network
   3. Output
      1. network.html file
6. Additional Programs (work in progress)
   1. PCA, PCoA, MDS
   2. Input
      1. Presence/absence matrix
   3. Output
      1. Plot in PDF, PNG, or JPEG format

**Programs and example usage:**

**For AMR analysis:**

-d, --directory: path to directory containing FASTA files to analyze

-O, --organism: specify what organism (if any) you want to analyze

-T, --threads: number of cores to dedicate

-o, --output-dir: output directory to store AMRfinder results

**Currently supported organisms and how they should be input:**Acinetobacter\_baumannii:

Burkholderia\_cepacia

Burkholderia\_pseudomallei

Campylobacter

Clostridioides\_difficile

Enterococcus\_faecalis

Enterococcus\_faecium

Escherichia

Klebsiella

Neisseria

Pseudomonas\_aeruginosa

Salmonella

Staphylococcus\_aureus

Staphylococcus\_pseudintermedius

Streptococcus\_agalactiae

Streptococcus\_pneumoniae

Streptococcus\_pyogenes

Vibrio\_cholerae

**AMR analysis example usage:**

% regain AMR -d path/to/directory -O Pseudomonas\_aruginosa -T 8 -o output\_directory

**Presence/absence matrix construction:**

-d, --directory: path to AMRfinder results in CSV format

-s, --search-strings-output: name of output file where gene names will be stored (will be stored in same directory as that specified with -d flag)

--gene-type: searches for resistance genes, virulence genes, or both

-f, --search-output: presence/absence matrix with all genes in your dataset

--min: minimum desired occurrence of genes across genomes

--max: maximum allowed occurrence of genes (should be less than number of genomes, as genes occurring across all genomes can significantly slow down Bayesian analysis.

--simplify-gene-names: gets rid of special characters in gene names, i.e., aph(3’’)-Ib becomes aph3pp\_Ib. This is not required for multidimensional analyses, but it is required for Bayesian analysis.

-o, --output: output of final cleaned up presence/absence matrix

Note: both -f and -o will default output to the current working directory

**Matrix construction example usage:**

% regain matrix -d path/to/directory -s search\_strings --simplify-gene-names --gene-type resistance -f matrix.csv --min 5 --max 500 -o matrix\_final.csv

**For Bayesian Network:**

-i, --input: input file in CSV format

-o, --output\_boot: output bootstrap file in .rds format

-T, --threads: number of cores to dedicate

-n, --number\_of\_boostraps: how many bootstraps to run

-D, --database\_output: output of fitted Bayesian Network compiled database in .RData format

**Bayesian Network example usage:**

% regain BN -i input\_file -n number\_of\_boostraps -T threads -o bootstrap\_output\_file -db database\_output\_file

Note: Bayesian analyses are very computationally expensive, so running the analysis in parallel is encouraged.

**Query the Bayesian network (grain object):**

-i, --input: input database in .RData format (database from -D output in Bayesian Network)

-M, --meta\_data: import metadata CSV file containing list of genes to query through the database

-T, --threads: number of cores to dedicate

-o, --output: output file containing odds ratio and raw probabilities

**Querying the network example usage:**

% regain query -i database.RData -M metadata.csv -T 10 -o probabilities.csv

**Query the Baysian Network (bn.fit object):**

-i, --input: input Bayesian Network in .rds format

-p, --presence\_absence\_matrix: input presence absence matrix

-M, --meta\_data: input metadata file containing genes to query

-o, --output: output probability table in CSV format

**Example usage:**

% regain queryFit -i boot.rds -p matrix.csv -M metadata.csv -o probabilities.csv

**Visualizing the Network:**

-p, --presence\_absence\_matrix: input presence absence matrix

-b, --boot\_intput: input boot file in .rds format

-M, --meta\_data: input metadata file containing genes and gene class

-o, --network\_output: output network file, must include ‘.html’ extension

**Visualize the Network example usage:**

% regain network -p matrix.csv -b boot.rds -M metadata.csv -o network.html

**Additional statistical programs:**

PCoA

PCA

MDS

AMR (AMRfinder analysis)

BN (Bayesian Network)

**For multidimensional analyses:**

-i, --input: input file in CSV format

-m, --method: jaccard, euclidean, manhattan, bray, cosine, or hamming

-c, --centers: how many centers you want for your multidimensional analysis (1-10)

-o, --output: output file format (PDF, PNG, JPEG)

-m is only for PCoA/MDS. Default is Jaccard method of distance

-c default is 1, max is 10

**Multidimensional analysis example usage:**

PCoA:

% regain PCoA -i input\_file -m method -c centers -o output\_file

PCA:

% regain PCA -i input\_file -c centers -o output\_file

MDS:

% regain MDS -i input\_file -m method -c centers -o output\_file

**Utility scripts:**

**Split multi-FASTA files into individual files example usage:**

% regain split\_fasta -i file.csv -o path/to/output/directory

**Split multi-genbank files into individual files usage:**

% regain split\_genbank -i file.csv -o path/to/output/directory