README – Insignia Microbial Signature Pipeline (Command Line)

# Overview

This pipeline builds a unique microbial signature database using whole-genome sequences. It performs genome cleaning, alignment (via MUMmer), match detection, and unique signature generation. All scripts run from the `programs` directory using the test-data and db\_seq folders as input and output points.

# System Requirements & Dependencies

OS: Linux (Ubuntu) or macOS  
Shell: bash  
Languages: Python 3.8+

Required packages/tools:

* - MUMmer v3.x (`nucmer`, `show-coords`, `mummer`)
* - Python packages: `pandas`, `numpy`
* - Utilities: `grep`, `awk`, `cut`, `sed`, `sort`, `head`, `tail`, `basename`

# Folder Structure

You should create the following structure:

insignia/  
├── programs/ # All pipeline scripts  
├── test-data/ # Input .fna genome files (multi- or single-FASTA)  
├── db\_seq/ # Processed genomes, .ins.fna, .db.fna, and indexes  
├── db\_cov/ # Final .cov files (per target genome)  
├── projects/ # Per-target alignment results (.mums, .match)  
├── signatures/ # Final .sig files

# Genome File Naming Requirements

Raw individual Genomes must placed into test-data/ with .fna ending. The header can be any length but should follow FASTA format with an identifier and species and genus directly following (e.g. >NC\_XXXXX Bacillus subtilis). It may have any other descriptor after that.   
- Example: `NC\_009848.4.fna`  
- Header format will be auto-cleaned to: `>ACCESSION Genus\_species` (e.g., `>NC\_009848.4 Bacillus\_pumilus`)

# How to Use the Pipeline

Run the following steps from the `programs/` directory:

1. 1. Clean and build initial database:

$ ./createdb.sh

- Cleans headers  
 - Creates .clean.fna files in `test-data/`  
 - Outputs .ins.fna, insignia.db.fna, t.idx, insigniaindex.csv in `seq-db/`

1. 2. Build genome-to-genome alignments:

$ ./buildalignments.sh

- Produces .mums and .match files per target-background combination  
 - Outputs into `projects

1. 3. Generate unique signatures:

$ ./signify.sh

- Interactively lets you choose a target  
 - Outputs .sig and .ref files into `signatures/`

# Targets.txt and Backgrounds.txt

These are now auto-generated by the `createdb.sh` script from `t.idx`.

Format:

- `targets.txt`: ACCESSION  
- `backgrounds.txt`: ACCESSION START\_BYTE GENOME\_LENGTH

# Rebuilding or Adding Genomes

To add new genomes:

- Drop new `.fna` files into `test-data/`  
- Rerun `./createdb.sh` and `./buildalignments.sh`

# Output Files Description

* - `\*.clean.fna`: Header-cleaned versions of input genomes
* - `\*.ins.fna`: Indexed versions used in db construction
* - `insignia.db.fna`: Concatenated genome database
* - `t.idx`: FASTA byte index file
* - `\*.mums`, `\*.match`: Pairwise alignment outputs
* - `\*.sig`: Final signature output files

# Tips for Clean Pipeline Execution

- Run scripts from `programs/`  
- Clear old `\*.clean.fna` files if starting fresh  
- Use git to version scripts and README