pyANI-plus

Leighton Pritchard

2025-03-10

Table of contents

About pyANI-plus

`pyANI-plus` is a Python package and software package that that calculates average nucleotide identity (ANI), provides other related measures for whole genome comparisons and renders relevant graphical and tabular summary output. It is designed to be used with draft or complete prokaryote genomes, and implements the following methods:

- ANIb (average nucleotide identity using BLAST+)
- ANIm (average nucleotide identity using MUMmer)
- dnadiff (average nucleotide identity using dnadiff)
- fastANI (average nucleotide identity using fastANI)
- sourmash (average nucleotide identity using sourmash)
- extenral-alignment (average nucletide identity using multiple-sequence-alignment)

In addition to calculating ANI for a given set of genomes, `pyANI-plus` also includes the following features:

- Plotting heatmaps and distributions for individual runs.
- Comparing multiple runs through visualization.
- Exporting any single run from the pyANI-plus SQLite3 database in tabular format.
- Classifying genomes into clusters based on ANI results.
- Resuming partial runs already logged in the database.
- Deleting any single run from the pyANI-plus SQLite3 database.

Requirements

`pyANI-plus` relies on several other programs, packages, and tools for both running and development. While many of these dependencies are installed automatically during setup, some may need to be downloaded and installed separately. This page provides a list of all required dependencies, along with explanations of their roles and why they are used.

Python3

`pyANI-plus` is designed to run on Python3, taking advantage of its latest features and improvements. It is not compatible with Python2, so using Python3 is required for installation and development.

NCBI-BLAST+

ANIb analysis, which calculates Average Nucleotide Identity using BLAST, involves comparing genome sequences through the BLAST tool provided by NCBI.

MUMer

For ANIm (Average Nucleotide Identity using MUMmer) analysis, genome sequences are compared using the nucmer tool from the MUMmer package. The same tool is applied in the dnadiff command to compare and analyze genome sequences. The key difference between the two methods lies in how the intermediate alignments are generated. `dnadiff` uses the `--maxmatch` (all anchor matches regardless of their uniqueness) parameter and `-m` (many-to-many) alignments to replicate the results reported by the `dnadiff` wrapper. In contrast, ANIm uses the `--mum` (anchor matches that are unique in both the reference and query) parameter by default, with the possibility of using the –maxmatch and -1 parameters in the delta.filter wrapper to generate 1-to-1 alignments.

sourmash

For sourmash (Average Nucletide Identity using sourmash) analysis, genome sequences are compared using the sourmash tool.

fastANI

For fastANI (Average Nucletide Identity using FastANI) analysis, genome sequences are compared using the fastANI tool.

SQLite3

The output generated by pyani analyses is stored in a local database, provided by SQLite3, for rapid querying and recovery. This allows for persistent storage of results without the need to keep the original alignment files, and for incremental addition of new analyses. SQLite is installed with Python

snakemake

By integrating snakemake, we maintain a single interface for defining and managing workflows. This allows us to standardize job execution across different environments without needing separate scheduling logic for local, cluster, or cloud execution.

Python Packages

`pyANI-plus` depends on several other Python packages, and we gratefully acknowledge their contribution:

- Matplotlib: for graphical output
- intervaltree: for identification of overlaps
- Seaborn: for graphical output
- NetworkX: for graph calculations and representation
- Numpy: for matrix calculations
- Pandas: for dataframe operations
- SQLAlchemy: for interaction with SQLite3
- Rich: provides progress bars for user interaction

Development

We rely on a number of additional packages to aid pyani development, and if you set up a development environment as recommended in Contributing to `pyANI-plus`, then the following Python packages will be installed or expected to be present:

- coverage: to generate code coverage output for the codecov.io service
- pre-commit: Manages and runs pre-commit hooks to enforce code quality and formatting before commits
- pytest: to manage and run automated testing
- pytest-cov: to integrate `pytest` with `codecov` and `coverage`
- pytest-xdist: Enables parallel test execution with `pytest`, improving test runtime efficiency.
- Ruff: Python linter that enforces coding style and helps catch potential issues.
- types-tqdm: Provides type hints for tqdm, improving type checking and IDE support.

Installation

This section describes different ways to install pyANI-plus on the most common operative systems sich as Unix/Linux and macOS.

. Currently, we support three ways to install `pyANI-plus` on your system:

1. Installation from source (i.e. download from GitHub)
2. Installation with Anaconda
3. Installation via pip

Installation from source

To install `pyANI-plus` from source, you can either download it from the Releases page or clone the repository using Git.

To get the latest version with Git, run the follwong command in a terminal:

```
git clone https://github.com/pyani-plus/pyani-plus
```

Alternatively you can visit the Relase page and click on one of the avaliable versions to get the source code.

Once the download is complete, navigat to the repository:

```
cd pyani-plus
```

Then, install `pyANI-plus` by running the appropriate script for you operating system:

```
make install_linux # For Linux
make install_macos # For macOS
```

To check if the installation was sucessful, run the following command:

```
pyani-plus --help
```

If the installation was completed correctly, this should display a list of avaliable commands and options, similar to this:

```
 Usage: pyani-plus [OPTIONS] COMMAND [ARGS]...

  Options
 --install-completion            Install completion for the current shell.
 --show-completion               Show completion for the current shell, to copy it or customiz
 --help                -h        Show this message and exit.

  Commands
 classify             Classify genomes into clusters based on ANI results.
 delete-run           Delete any single run from the given pyANI-plus SQLite3 database.
 export-run           Export any single run from the given pyANI-plus SQLite3 database.
 list-runs            List the runs defined in a given pyANI-plus SQLite3 database.
 plot-run             Plot heatmaps and distributions for any single run.
 plot-run-comp        Plot comparisons between multiple runs.
 resume               Resume any (partial) run already logged in the database.

  ANI methods
 anib                 Execute ANIb calculations, logged to a pyANI-plus SQLite3 database.
 anim                 Execute ANIm calculations, logged to a pyANI-plus SQLite3 database.
 dnadiff              Execute mumer-based dnadiff calculations, logged to a pyANI-plus SQLite3
 external-alignment   Compute pairwise ANI from given multiple-sequence-alignment (MSA) file.
 fastani              Execute fastANI calculations, logged to a pyANI-plus SQLite3 database.
 sourmash             Execute sourmash-plugin-branchwater ANI calculations, logged to a pyANI-
```

pyANI-plus walkthrough

This section walks you through how `pyANI-plus` can be applied to calculate Average Nucleotide Identity, render graphical and tabular summary output, and perform other related measures for whole genome comparisons. The general procedue for any `pyANI-plus` analysis is:

1. Collect genomes for analysis
2. Perform ANI analysis using diffrent methods such as ANIb, ANIm etc.
3. Report and visualise analysis report
4. Use the analysis results to classify input genomes and generate species hypotheses

Tip

Before using `pyANI-plus`, make sure to install it on a local machine like a laptop, desktop, server, or cluster. Please see section installation for installation instructions.

This is a command-line tool, meaning you type commands into a terminal window to run it. To view the avaliable options we type `pyani-plus` (in lower case), space, then `-h` (minus lower-case H) for the help option, and finally enter or return to run the command:

```
pyani-plus -h
```

This should output the following - hopefully in colour depending on your terminal setup:

```
 Usage: pyani-plus [OPTIONS] COMMAND [ARGS]...

  Options
 --version              -v        Show tool version (on stdout) and quit.
 --install-completion             Install completion for the current
                                  shell.
 --show-completion                Show completion for the current shell,
                                  to copy it or customize the
                                  installation.
 --help                 -h        Show this message and exit.

  Commands
 resume                     Resume any (partial) run already logged in the
                            database.
 list-runs                  List the runs defined in a given pyANI-plus SQLite3
                            database.
 delete-run                 Delete any single run from the given pyANI-plus
                            SQLite3 database.
 export-run                 Export any single run from the given pyANI-plus
                            SQLite3 database.
 plot-run                   Plot heatmaps and distributions for any single run.
 plot-run-comp              Plot comparisons between multiple runs.
 classify                   Classify genomes into clusters based on ANI
                            results.

  ANI methods
 anim                       Execute ANIm calculations, logged to a pyANI-plus
                            SQLite3 database.
 dnadiff                    Execute mumer-based dnadiff calculations, logged to
                            a pyANI-plus SQLite3 database.
 anib                       Execute ANIb calculations, logged to a pyANI-plus
                            SQLite3 database.
 fastani                    Execute fastANI calculations, logged to a
                            pyANI-plus SQLite3 database.
 sourmash                   Execute sourmash-plugin-branchwater ANI
                            calculations, logged to a pyANI-plus SQLite3
                            database.
 external-alignment         Compute pairwise ANI from given
                            multiple-sequence-alignment (MSA) file.
```

To see the options for a specific subcommand, use `pyani-plus <subcommand> -h`. For example, to view options for the ANIb method:

```
pyani-plus anib -h
```

Expected output:

```
 Usage: pyani-plus anib [OPTIONS] FASTA

 Execute ANIb calculations, logged to a pyANI-plus SQLite3 database.

  Arguments
 *    fasta      PATH  Directory of FASTA files (extensions .fas, .fasta,
                       .fna).
                       [required]

  Options
 *  --database   -d      FILE           Path to pyANI-plus SQLite3
                                        database.
                                        [required]
    --name               TEXT           Run name. Default is 'N genomes
                                        using METHOD'.
    --create-db                         Create database if does not
                                        exist.
    --executor           [local|slurm]  How should the internal tools be
                                        run?
                                        [default: local]
    --help       -h                     Show this message and exit.

  Method parameters
 --fragsize        INTEGER RANGE [x>=1]  Comparison method fragment size.
                                         [default: 1020]

  Debugging
 --temp         DIRECTORY  Directory to use for intermediate files, which
                           for debugging purposes will not be deleted.
                           For clusters this must be on a shared drive.
                           Default behaviour is to use a system specified
                           temporary directory (specific to the
                           compute-node when using a cluster) and remove
                           this afterwards.
 --wtemp        DIRECTORY  Directory to use for temporary workflow
                           coordination files, which for debugging
                           purposes will not be deleted. For clusters
                           this must be on a shared drive. Default
                           behaviour is to use a system specified
                           temporary directory (for the local executor)
                           or a temporary directory under the present
                           direct (for clusters), and remove this
                           afterwards.
 --log          FILE       Where to record log(s). Use '-' for no
                           logging.
                           [default: pyani-plus.log]
 --debug                   Show debugging level logging at the terminal
                           (in addition to the log file).
```

Collect genomes for analysis

While you can work with genomes placed in a local directory with `pyANI-plus`, we suggest using the genomes provided here in this walkthrough to ensure the output matches the expected results.

`pyANI-plus` accepts FASTA files with the extensions `.fasta`, `.fas`, and `.fna`, along with zipped versions like `.fasta.gz`, `.fas.gz`, and `.fna.gz`. Please make sure that your input files match these extensions to ensure that `pyANI-plus` works.

Conducting ANI analysis

`pyANI-plus` enables genome comparison using various ANI methods. In this walkthrough, we will demonstrate methods such as ANIm, ANIb, dnadiff, FastANI, and Sourmash. While running all methods is not mandatory, we recommend doing so, as we will later explore additional whole-genome comparison metrics, such as `plot-run-comp`, using `pyANI-plus`.

Running any ANI method on the donwloaded genomes requires you to first specify the directory containing the genome data (e.g., `walkthrough_data`), then the path to the pyANI-plus SQLite3 database (`walkthrough.db` for this walkthrough).

> **!** Important
>
> If this is your first analysis and the SQLite3 database does not yet exist, you must use the `--create-db` option; otherwise, you'll encounter the following error:
>
> ```
> ERROR: Database walkthrough.db does not exist, but not using --create-db
> ```

Optionally, you can provide a custom name for the analysis with the `--name` option for easier reference, and if you want to run the ANI analysis on Slurm, simply set the execution method with the `--executor` option to `slurm` (default: `local`).

Conduct ANIb analysis

In this walkthrough, we will first run the ANIb analysis on the downloaded genomes using the following command line:

```
pyani-plus anib walkthrough_data --database walkthrough.db --create-db --name "walkthrough ANI
```

If you wish you can select a different fragment size for the comparison method using the `--fragsize` option. The default size is 1020bp, which is typically used for ANIb.

Conduct ANIm analysis

Next, we will run the ANIm analysis on the same genomes using the following command line:

```
pyani-plus anim walkthrough_data --database walkthrough.db --name "walkthrough ANIm"
```

In ANIm analysis, the default setting uses anchor matches that are unique in the reference but not necessarily unique in the query (`--mode mum`). You can change this to include all anchor matches, regardless of their uniqueness, by setting the `--mode` option to `maxmatch`.

Conduct dnadiff analysis

To compare genomes in the input `walkthrough_data` directory using `dnadiff` method use the following command line:

```
pyani-plus dnadiff walkthrough_data --database walkthrough.db --name "walkthrough dnadiff"
```

Conduct fastani analysis

To run fastani analysis on the genomes in the input `walkthrough_data` directory use the following command line:

```
pyani-plus fastani walkthrough_data --database walkthrough.db --name "walkthrough fastani"
```

In `fastani` analysis, additional method parameters can be changed by the user. These include:

- `--fragsize`: Fragment length used in the analysis (default: `3000`).
- `--kmersie`: K-mer size, set to `16` by default. It can be set to any value smaller than 16.
- `--minmatch`: Minimum fraction of the genome that must be shared for ANI to be considered reliable. If the reference and query genome sizes differ, the smaller genome is used. (Default: `0.2`).

Conduct sourmash analysis

Lastly, we can run sourmash analysis with the following command line:

```
pyani-plus sourmash walkthrough_data --database walkthrough.db --name "walkthrough sourmash"
```

For `sourmash` analysis, additional method parameters can be changed by the user. These include: - `--scaled`: Compression ration (defult: `1000`) - `--kmersize`: K-mer size (default: `31`)

Reporting Analyses and Analysis Results

List all runs in the database

`pyANI-plus` enables you to view all runs defined in a SQLite3 database. To display all runs from the database (eg. `walkthrough.db` for this walkthrough), use this command:

```
pyani-plus list-runs --database walkthrough.db
```

You will see the following table, or something similar, depending on the analyses contained within the database, displayed on your screen:

```
                         5 analysis runs in walkthrough.db

 ID  Date        Method     Done  Null  Miss  Total    Status   Name

  1  2025-03-17  ANIb        100     0     0  100=10²  Done     walkthrough ANIb
  2  2025-03-17  ANIm         68    32     0  100=10²  Done     walkthrough ANIm
  3  2025-03-17  dnadiff     100     0     0  100=10²  Done     walkthrough dnadiff
  4  2025-03-17  fastANI      68    32     0  100=10²  Done     walkthrough fastani
  5  2025-03-17  sourmash     68    32     0  100=10²  Done     walkthrough sourmash
```

In this table, each row represents a single run, and the columns provide the following details:

- `ID`: Unique ID for the run
- `Date`: Date when the analysis was executed
- `Method`: ANI method used
- `Done`: Number of completed ANI comparisons
- `Null`: Number of analyses where no alignment was found (e.g., comparisons between highly divergent genomes)
- `Miss`: Number of comparisons that were not completed
- `Status`: Current status of the analysis—e.g., Done for completed analyses, Running for comparisons that have started but are still in progress
- `Name`: Run name.

Exporting ANI results in a tabular format

`pyANI-plus` allows ANI results to be exported in a tabular format, but the output directory must already be present. In this example, we create an `output` directory and use the following command to export results for the ANIb analysis:

```
mkdir output # create directory called output
pyani-plus export-run --database walkthrough.db --outdir output --run-id 1
```

Tip

If `--run-id` is not specified the latest run will be exported. To export runs for other analyses, you can specify the `--run-id` by matching the ID number provided in the table provided by list-runs subcommand.

This will report the relevant information to new files in the `output` directory. The matrix output files are named `<method>_<property>.tsv` while the long form is named `<method>_run_<run-id>.tsv` and will include the query and subject genomes and all the comparison properties as columns:

```
.
  ANIb_aln_lengths.tsv
  ANIb_hadamard.tsv
  ANIb_identity.tsv
  ANIb_query_cov.tsv
  ANIb_run_1.tsv
  ANIb_sim_errors.tsv
  ANIb_tANI.tsv
```

Important

Incomplete runs will return an error. There will be no output for empty run. For partial runs the long form table will be exported, but not the matrices.

Graphical output

Graphical output (`JPG`, `PDF`, `PNG` and `SVGZ` formats) is obtanined by executing the `pyani-plus plot-run` subcommand, specifying the database and ouput directory:

```
pyani-plus plot-run --database walkthrough.db --outdir output --run-id 2
```

Tip

If `--run-id` is not specified the latest run will be used to generate graphical output. To generate graphical output for other analyses, you can specify the `--run-id` by matching the ID number provided in the table provided by list-runs subcommand.
Optionally, you can label genomes using `md5` and `filename` using the `--label` option.

`plot-run` subcommand generates the following heatmaps, distribution plots, and tabular output from an ANI analysis for a specified `--run_id`, using data stored in a local SQLite3 database:

- percentage identity of aligned regions (`<method>_idenity_hea` `<method>_identity_dist.<extension>` and `<method>_iden` Figure 1)

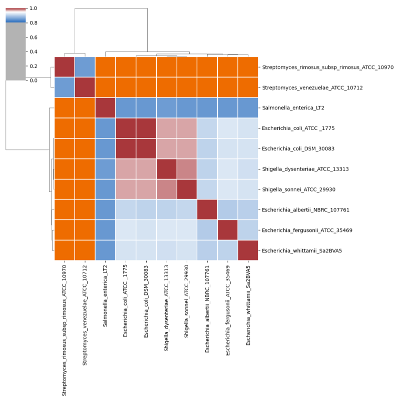

Figure 1: ANIm percentage identity heatmap for walkthrough_data

In the above heatmap, each cell represents a pairwise comparision between the genomes shown in the rows and columns, showing the pairwise identity of aligned regions. The dendrograms are single-linkage clustering trees generated from the matrix of pairwise identity results. The default color scheme assigns red to cells with identity ≥ 0.95, blue to those with identity < 0.95, and orange to cells representing comparisons with no alignment found (e.g., NULLs). This division corresponds to a widely-used convention for bacterial species boundaries.

- percentage coverage of each query genome by aligned regions (`<method>_query_cov_heatmap.<extension>`, `<method>_query_cov_dist.<extension>`, `<method>_query_cov_scatter.<extension>` and `<method>_query_cov.tsv`)

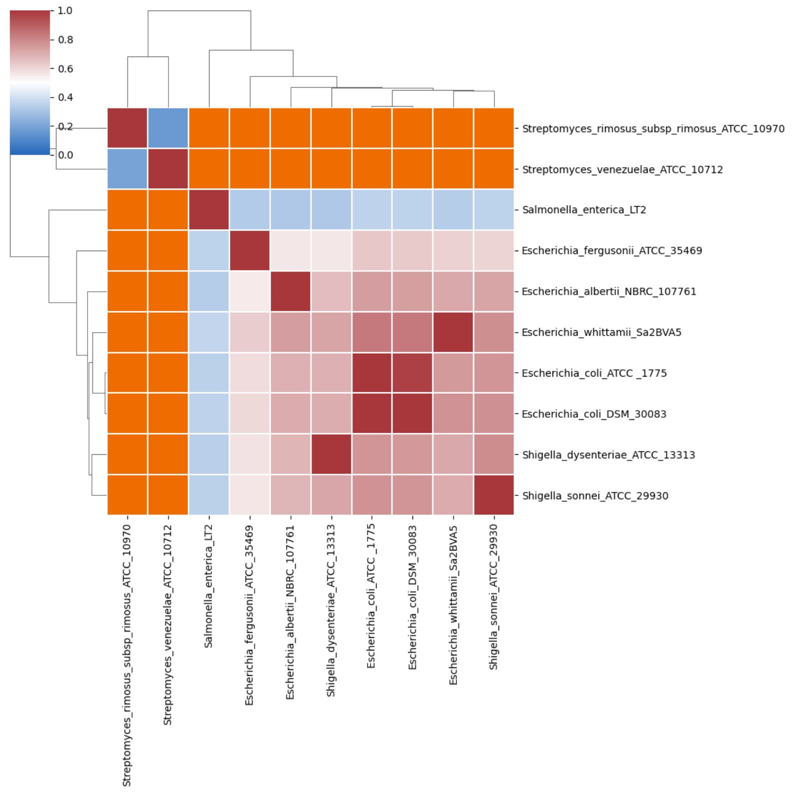

Figure 2: ANIm percentage coverage heatmap for walkthrough_data

In the above heatmap, each cell represents a pairwise comparision between the genomes shown in the rows and columns, showing the pairwise coverage of each genome by aligned regions in the comparison. The dendrograms are single-linkage clustering trees generated from the matrix of pairwise coverage results. The default color scheme assigns red to cells with coverage ≥ 0.50, blue to those with coverage < 0.50, and orange to cells representing comparisons with no alignment found (e.g., NULLs). This division corresponds to a strict majority of each genome in the comparison being alignable (a plausible minimum requirement for two sequences being considered "the same thing").

- a Hadamard matrix of percentage identity multiplied by percentage coverage for each comparison (`<method>_hadamard_heatmap.<extension>`, `<method>_hadamard_dist.<extension>` and `<method>_hadamard.tsv`)

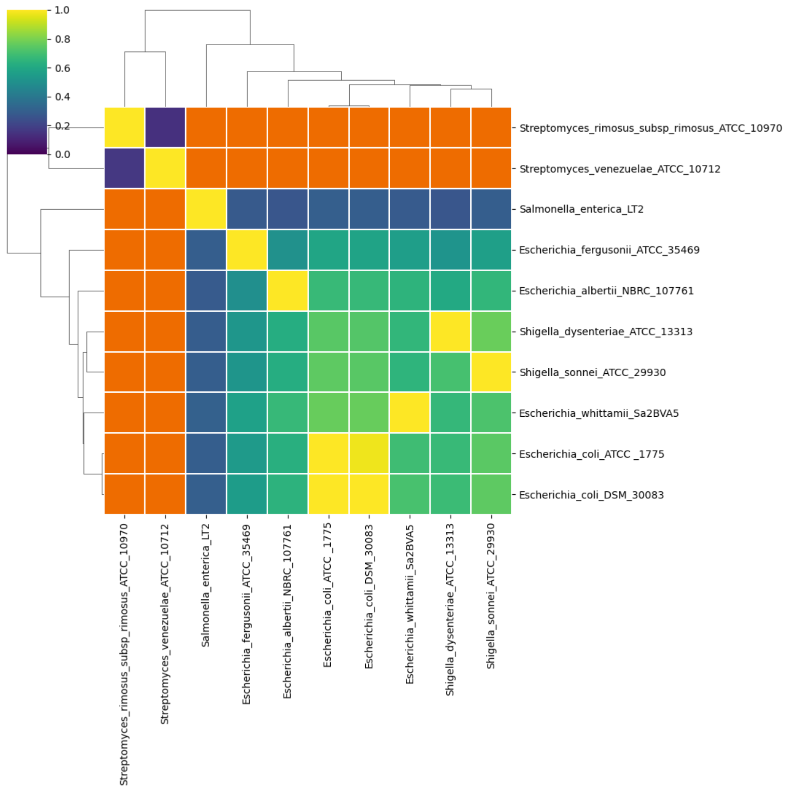

Figure 3: ANIm hadamard heatmap for walkthrough_data

- a total Avenarge Nucleotide Identity (tANI) matrix of he negative log of the coverage multiplied by identity (`<method>_tANI_heatmap.<extension>`, `<method>_tANI_dist.<extension>`, `<method>_tANI_scatter.<extension>` and `<method>_tANI.tsv`)

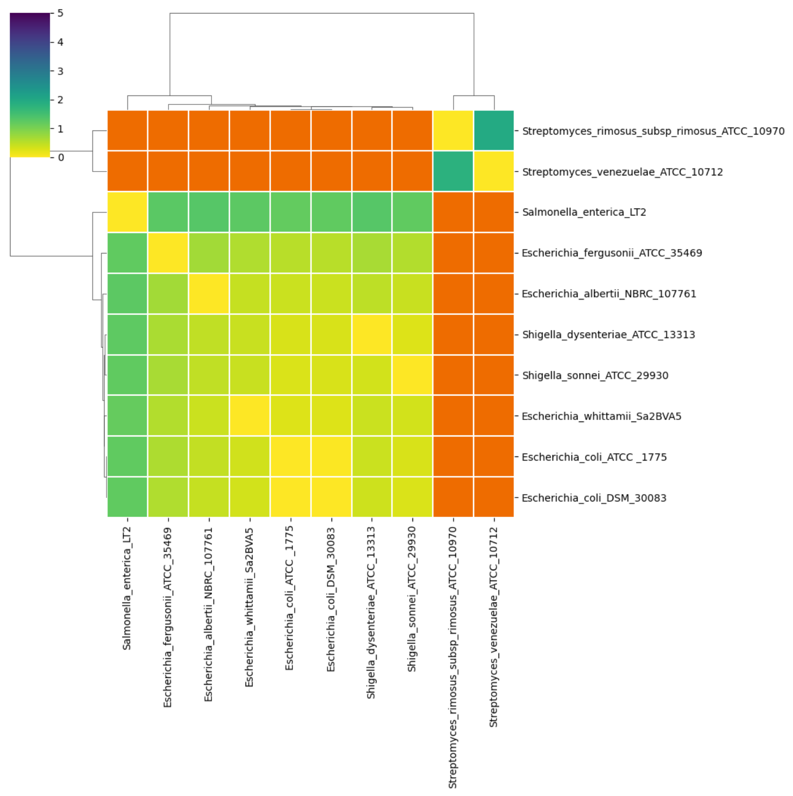

Figure 4: ANIm tANI heatmap for walkthrough_data

- number of "similarity errors" on each genome (`<method>_sim_errors.tsv`)
- Long form of ANI results which include the query and subject genomes and all the comparison properties as columns (`<method>_run_<ID>.tsv`)

Plotting comparisons between runs

ANI results can vary depending on the method used. `pyANI-plus` allows you to compare the ANI results from multiple runs. In this example, we show how to use the `pyani-plus plot-run-comp` subcommand to visualise and compare these results. Running `pyani-plus plot-run-comp` requires specifying the output directory (e.g., `output`), the path to the pyANI-plus SQLite3 database (`walkthrough.db` for this walkthrough), and a comma-separated list of run IDs for comparison.

❗ Important

The first run ID will be treated as the reference, and all subsequent runs will be compared to it.
In this example, we use ANIb as the reference method, with other methods compared against it.

```
pyani-plus plot-run-comp --database walkthrough.db --outdir output --run-ids 1,2,3,4,5
```

This command generates the following outputs:

- A set of scatterplots where the X-axis represents genome identity from the reference method (here, ANIb), and the Y-axis represents genome identity from the compared methods/runs (`<reference_method>_identity_<run_ID>_scatter_vs_others.<extension>`).

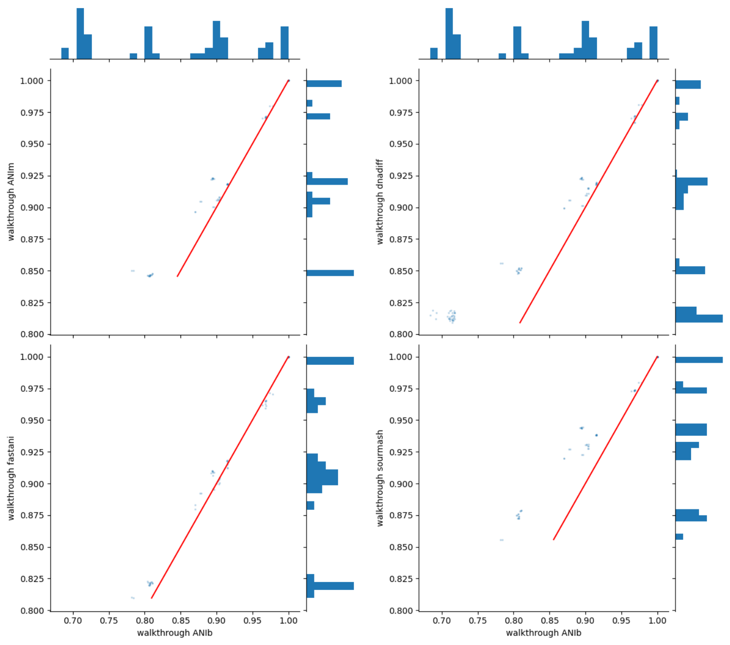

Figure 5: Scatterplot output for `pyani-plus plot-run-comp` subcommand

Tip

The red vertical line serves as a reference, indicating where data points should align on the scatterplot if the pairwise comparison results match between methods.

- A set of scatterplots showing absolute differences between pairwise comparisons, with the X-axis representing genome identity from the reference method (here, ANIb) and the Y-axis showing the difference in genome identity for the compared methods/runs.

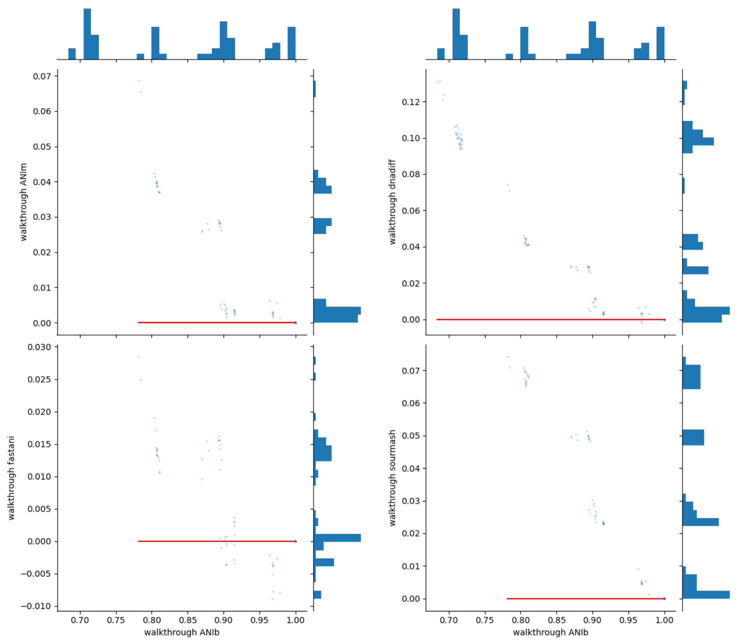

Figure 6: Scatterplot output for `pyani-plus plot-run-comp` subcommand

Tip

The red horizontal line at 0 indicates no difference between the pairwise comparison results across methods.

- Tabular summary of the comparison output (`<reference_method>_identity_<run_ID>_vs_<run_ID>.tsv`)

Part I

pyANI-plus subcommands

`pyANI-plus` follows a subcommand-based approach, where the primary command is followed by a subcommand that defines the operation to be performed. The format is: `pyani-plus <subcommand>`. For example, to obtain the list of runs in a give `pyANI-plus` SQLite3 database, you would use `list-runs` subcommand:

```
pyani-plus list-runs <path_to_database>
```

This section lists all available subcommands for `pyANI-plus`, describing their usage and functionality within the software.

1 anib

The `anib` subcommand processes genome files from the `indir` directory to perform ANIb analysis, and logs comparison and run data in a local SQLite3 database.

`pyani-plus anib` Usage

```
pyani-plus anib [OPTIONS] FASTA
```

1.1 Arguments

fasta: Directory of FASTA files (extensions `.fas`, `.fasta`, `.fna`). (PATH) [REQUIRED]

1.2 Options

`--database`: Path to `pyANI-plus` SQLite3 database. (FILE) [REQUIRED]

`--create-db`: Create database if does not exist.

`--name`: Run name. [Default: "N genomes using METHOD"] (TEXT)

`--executor`: How should the internal tools be run? [Default: local] (`local`|`slurm`)

`--help`, `-h`: Display usage information for `pyani-plus anib`.

1.3 Method parameters

`--fragsize`: Comparison method fragment size. [Default: 1020] (Integer range: $X \geq 1$)

1.4 Debugging

`--temp`: Directory to use for intermediate files, which for debugging purposes will not be deleted. For clusters this must be on a shared drive. Default behaviour is to use a system specified temporary directory (specific to the compute-node when using a cluster) and remove this afterwards. (Directory)

`--wtemp`: Directory to use for temporary workflow coordination files, which for debugging purposes will not be deleted. For clusters this must be on shared drive. Default behaviour is to use a system specified temporary directory (for the local executor) or a temporary directory under the present direct (for clusters), and remove this afterwards. (Directory)

2 `anim`

The `anim` subcommand processes genome files from the `indir` directory to perform ANIm analysis, and logs comparison and run data in a local SQLite3 database.

`pyani-plus anim` Usage

```
pyani-plus anim [OPTIONS] FASTA
```

2.1 Arguments

fasta: Directory of FASTA files (extensions `.fas`, `.fasta`, `.fna`). (PATH) [REQUIRED]

2.2 Options

`--database`: Path to `pyANI-plus` SQLite3 database. (FILE) [REQUIRED]

`--create-db`: Create database if does not exist.

`--name`: Run name. [Default: "N genomes using METHOD"] (TEXT)

`--executor`: How should the internal tools be run? [Default: local] (`local`|`slurm`)

`--help`, `-h`: Display usgae information for `pyani-plus anib`.

2.3 Method parameters

`--mode`: Nucmer mode for ANIm. [Default: `mum`] (`mum`|`maxmatch`)

2.4 Debugging

`--temp`: Directory to use for intermediate files, which for debugging purposes will not be deleted. For clusters this must be on a shared drive. Default behaviour is to use a system specified temporary directory (specific to the compute-node when using a cluster) and remove this afterwards. (Directory)

`--wtemp`: Directory to use for temporary workflow coordination files, which for debugging purposes will not be deleted. For clusters this must be on shared drive. Default behaviour is to use a system specified temporary directory (for the local executor) or a temporary directory under the present direct (for clusters), and remove this afterwards. (Directory)

3 `dnadiff`

The `dnadiff` subcommand processes genome files from the `indir` directory to perform dnadiff analysis, and logs comparison and run data in a local SQLite3 database.

`pyani-plus dnadiff` Usage

```
pyani-plus dnadiff [OPTIONS] FASTA
```

3.1 Arguments

fasta: Directory of FASTA files (extensions `.fas`, `.fasta`, `.fna`). (PATH) [REQUIRED]

3.2 Options

`--database`: Path to `pyANI-plus` SQLite3 database. (FILE) [REQUIRED]

`--create-db`: Create database if does not exist.

`--name`: Run name. [Default: “N genomes using METHOD”] (TEXT)

`--executor`: How should the internal tools be run? [Default: local] (`local`|`slurm`)

`--help`, `-h`: Display usgae information for `pyani-plus anib`.

3.3 Debugging

`--temp`: Directory to use for intermediate files, which for debugging purposes will not be deleted. For clusters this must be on a shared drive. Default behaviour is to use a system specified temporary directory (specific to the compute-node when using a cluster) and remove this afterwards. (Directory)

`--wtemp`: Directory to use for temporary workflow coordination files, which for debugging purposes will not be deleted. For clusters this must be on shared drive. Default behaviour is to use a system specified temporary directory (for the local executor) or a temporary directory under the present direct (for clusters), and remove this afterwards. (Directory)

4 `external-alignment`

The `external-alignment` subcommand compute pairwise ANI from given multiple-sequence-alignment (MSA) file and genome files from the `indir` directory, and logs comparison and run data in a local SQLite3 database.

`pyani-plus external-alignment` Usage

```
pyani-plus external-alignment [OPTIONS] FASTA
```

4.1 Arguments

fasta: Directory of FASTA files (extensions `.fas`, `.fasta`, `.fna`). (PATH) [REQUIRED]

4.2 Options

`--database`: Path to `pyANI-plus` SQLite3 database. (FILE) [REQUIRED]

`--create-db`: Create database if does not exist.

`--name`: Run name. [Default: "N genomes using METHOD"] (TEXT)

`--executor`: How should the internal tools be run? [Default: local] (`local`|`slurm`)

`--help`, `-h`: Display usgae information for `pyani-plus anib`.

4.3 Method parameters

`--alignment`: FASTA format MSA of the same genomes (one sequence per genome). (PATH) [REQUIRED]

`--label`: How are the sequences in the MSA labelled vs the FASTA genomes? [Default: stem] (`md5`|`filename`|`stem`)

4.4 Debugging

`--temp`: Directory to use for intermediate files, which for debugging purposes will not be deleted. For clusters this must be on a shared drive. Default behaviour is to use a system specified temporary directory (specific to the compute-node when using a cluster) and remove this afterwards. (Directory)

`--wtemp`: Directory to use for temporary workflow coordination files, which for debugging purposes will not be deleted. For clusters this must be on shared drive. Default behaviour is to use a system specified temporary directory (for the local executor) or a temporary directory under the present direct (for clusters), and remove this afterwards. (Directory)

5 fastani

The `fastani` subcommand processes genome files from the `indir` directory to perform fastANI analysis, and logs comparison and run data in a local SQLite3 database.

`pyani-plus fastani` Usage

```
pyani-plus fastani [OPTIONS] FASTA
```

5.1 Arguments

fasta: Directory of FASTA files (extensions `.fas`, `.fasta`, `.fna`). (PATH) [REQUIRED]

5.2 Options

`--database`: Path to `pyANI-plus` SQLite3 database. (FILE) [REQUIRED]

`--create-db`: Create database if does not exist.

`--name`: Run name. [Default: "N genomes using METHOD"] (TEXT)

`--executor`: How should the internal tools be run? [Default: local] (`local`|`slurm`)

`--help`, `-h`: Display usgae information for `pyani-plus anib`.

5.3 Method parameters

`--fragsize`: Comparison method fragment size. [Default: 1020] (Integer range: $x \geq 1$)

`--kmersize`: Comparison method k-mer size. [Default: 16] ($1 \leq x \leq 16$)

`--minmatch`: Comparison method min-match. [Default: 0.2] ($0.0 \leq x \leq 1.0$)

5.4 Debugging

`--temp`: Directory to use for intermediate files, which for debugging purposes will not be deleted. For clusters this must be on a shared drive. Default behaviour is to use a system specified temporary directory (specific to the compute-node when using a cluster) and remove this afterwards. (Directory)

`--wtemp`: Directory to use for temporary workflow coordination files, which for debugging purposes will not be deleted. For clusters this must be on shared drive. Default behaviour is to use a system specified temporary directory (for the local executor) or a temporary directory under the present direct (for clusters), and remove this afterwards. (Directory)

6 `sourmash`

The `sourmash` subcommand processes genome files from the `indir` directory to perform sourmash analysis, and logs comparison and run data in a local SQLite3 database.

`pyani-plus sourmash` Usage

```
pyani-plus sourmash [OPTIONS] FASTA
```

6.1 Arguments

fasta: Directory of FASTA files (extensions `.fas`, `.fasta`, `.fna`). (PATH) [REQUIRED]

6.2 Options

`--database`: Path to `pyANI-plus` SQLite3 database. (FILE) [REQUIRED]

`--create-db`: Create database if does not exist.

`--name`: Run name. [Default: "N genomes using METHOD"] (TEXT)

`--executor`: How should the internal tools be run? [Default: local] (`local`|`slurm`)

`--help`, `-h`: Display usgae information for `pyani-plus anib`.

6.3 Method parameters

`--scaled`: Sets the compression ratio. [Default: 1000] (Integer range: $x \geq 1$)

`--kmersize`: Comparison method k-mer size. [Default: 31] (Integer range: $x \geq 1$)

6.4 Debugging

`--temp`: Directory to use for intermediate files, which for debugging purposes will not be deleted. For clusters this must be on a shared drive. Default behaviour is to use a system specified temporary directory (specific to the compute-node when using a cluster) and remove this afterwards. (Directory)

`--wtemp`: Directory to use for temporary workflow coordination files, which for debugging purposes will not be deleted. For clusters this must be on shared drive. Default behaviour is to use a system specified temporary directory (for the local executor) or a temporary directory under the present direct (for clusters), and remove this afterwards. (Directory)

7 `plot-run`

The `plot-run` subcommand generates heatmaps, distribution plots, and tabular output from an ANI analysis for a specified `--run_id`, using data stored in a local SQLite3 database. All plots, including formats such as `JPG`, `PDF`, `PNG` and `SVGZ`, as well as the tabular data, will be saved in the `outdir` directory.

! Important

The output directory must already exist. The heatmap files will be named `<method>_<property>.<extension>` and any pre-existing files will be overwritten.

`pyani-plus plot-run` Usage

```
pyani-plus plot-run [OPTIONS]
```

7.1 Options

`--database`: Path to `pyANI-plus` SQLite3 database. (FILE) [REQUIRED]

`--outdir`: Output directory. (DIRECTORY) [REQUIRED]

`--run-id`: Which run from the database. [Defaults to latest.] (INTEGER)

`--label`: How to label the genomes. [Default: stem.] (md5|filename|stem)

`--help`, `-h`: Display usage information for `pyani-plus plot-run` and exit.

8 `plot-run-comp`

The `plot-run-comp` subcommand compares ANI results from multiple runs, generating scatterplots and tabular summaries. All plots, including formats such as `JPG`, `PDF`, `PNG` and `SVGZ`, as well as the tabular data, will be saved in the `outdir` directory.

! Important

The output directory must already exist. The scatter plots will be named `<method>_<property>_<run-id>_vs_*.<extension>` and any pre-existing files will be overwritten.

`pyani-plus plot-run-comp` Usage

```
pyani-plus plot-run [OPTIONS]
```

8.1 Options

`--database`: Path to `pyANI-plus` SQLite3 database. (FILE) [REQUIRED]

`--outdir`: Output directory. (DIRECTORY) [REQUIRED]

`--run-ids`: Which runs (comma separated list, reference first)? (TEXT) [REQUIRED]

`--columns`: How many columns to use when tiling plots of multiple runs. Default 0 means automatically tries for square tiling. [Default: 0] (Integer range: $x \geq 0$)

`--help`, `-h`: Display usage information for `pyani-plus plot-run` and exit.

9 `list-runs`

The `list-runs` subcommand lists the runs defined in a given pyANI-plus SQLite3 database.

`pyani-plus list-runs` Usage

```
pyani-plus list-runs [OPTIONS]
```

9.1 Options

`--database`: Path to `pyANI-plus` SQLite3 database. (FILE) [REQUIRED]

`--help`, `-h`: Display usage information for `pyani-plus reasume` and exit.

10 `export-run`

The `export-run` subcommand exports ANI results in a tabular format for a specified `--run_id`, using data stored in a local SQLite3 database.

! Important

The output directory must already exist. Any pre-existing files will be overwritten. Incomplete runs will return an error. There will be no output for empty run. For partial runs the long form table will be exported, but not the matrices.
The matrix output files will be named `<method>_<property>.tsv` while the long form is named `<method>_run_<run-id>.tsv` and will include the query and subject genomes and all the comparison properties as columns.

`pyani-plus export-run` Usage

```
pyani-plus export-run [OPTIONS]
```

10.1 Options

`--database`: Path to `pyANI-plus` SQLite3 database. (FILE) [REQUIRED]

`--outdir`: Output directory. (DIRECTORY) [REQUIRED]

`--run-id`: Which run from the database. [Defaults to latest.] (INTEGER)

`--label`: How to label the genomes. [Default: stem.] (md5|filename|stem)

`--help`, `-h`: Display usage information for **`pyani-plus reasume`** and exit.

11 `resume`

The `resume` subcommand restarts any partially completed run stored in the database if it was interrupted or canceled, ensuring it continues from where it left off. Any missing pairwise comparisons will be computed, and the the old run will be marked as complete. This should have no effect on completed comparisons.

> **!** Important
>
> The output directory must already exist. The scatter plots will be named `<method>_<property>_<run-id>_vs_*.<extension>` and any pre-existing files will be overwritten.
> If the version of the underlying tool has changed, this will abort as the original run cannot be completed.

> `pyani-plus resume` Usage
>
> ```
> pyani-plus resume [OPTIONS]
> ```

11.1 Options

`--database`: Path to `pyANI-plus` SQLite3 database. (FILE) [REQUIRED]

`--run-id`: Which run from the database. [Defaults to latest.] (INTEGER)

`--executor`: How should the internal tools be run? [Default: local] (`local`|`slurm`)

`--cache`: Cache location if required for a method (must be visible to cluster workers). Default to .cache in the current directory. (DIRECTORY)

`--help`, `-h`: Display usage information for `pyani-plus reasume` and exit.

11.2 Debugging

`--temp`: Directory to use for intermediate files, which for debugging purposes will not be deleted. For clusters this must be on a shared drive. Default behaviour is to use a system specified temporary directory (specific to the compute-node when using a cluster) and remove this afterwards. (Directory)

`--wtemp`: Directory to use for temporary workflow coordination files, which for debugging purposes will not be deleted. For clusters this must be on shared drive. Default behaviour is to use a system specified temporary directory (for the local executor) or a temporary directory under the present direct (for clusters), and remove this afterwards. (Directory)

12 `delete-run`

The `delete-run` subcommand deletes any single run stored in the database. This will prompt the user for confirmation if the run has comparisons, or if the run status is “Running”, but that can be overridden.

Important

Currently this will *not* delete any linked comparisons, even if they are not currently linked to another run. They will be reused should you start a new run using an overlapping set of input FASTA files.

`pyani-plus delete-run` Usage

```
pyani-plus resume [OPTIONS]
```

12.1 Options

`--database`: Path to `pyANI-plus` SQLite3 database. (FILE) [REQUIRED]

`--run-id`: Which run from the database. [Defaults to latest.] (INTEGER)

`--force`, `-f`: Delete without confirmation.

`--help`, `-h`: Display usage information for `pyani-plus reasume` and exit.

13 `classify`

The `classify` subcommand classifies genomes into cliques (k-complete) graphs based on ANI results, generating plots and tabular summaries. This is helpful for circumscribing potentially meaningful groups of genomes that can not be described using traditional taxonomy. The output, including classify plots (`JPG`, `PDF`, `PNG`, `SVGZ`) and tabular data, is written to `outdir` directory.

`pyani-plus classify` Usage

```
pyani-plus classify [OPTIONS]
```

13.1 Options

`--database`: Path to `pyANI-plus` SQLite3 database. (FILE) [REQUIRED]

`--outdir`: Output directory. (DIRECTORY) [REQUIRED]

`--run-id`: Which run from the database. [Defaults to latest.] (INTEGER)

`--label`: How to label the genomes. [Default: stem.] (md5|filename|stem)

`--help`, `-h`: Display usage information for `pyani-plus reasume` and exit.

13.2 Method parameters

`--mode`: Classify mode intended to identify cliques within a set of genomes. [Default: identity] (identity|tANI)

`--score-edges`: How to resolve asymmetrical ANI identity/tANI results for edges in the graph (min, max or mean). [Default: min] (TEXT)

`--coverage-edges`: How to resolve asymmetrical ANI coverage results for edges in the graph (min, max or mean). [Default: min] (TEXT)

`--cov-min`: Minimum %coverage for an edge. [Default: 0.5] ($0.0 \leq x \leq 1.0$)

`--vertical-line`: Threshold for red vertical line at identity/tANI. The default is set to 0.95 if `--mode` is `identity` and -0.323 if `--mode` is `tANI`. (TEXT)

Testing

`pyANI-plus` is currently tested using the `pytest` package to ensure that it functions correctly and remains stable across updates.

Test directory structure

The `tests/` subdirectory in the `pyANI-plus` repository contains all test files, including input data, expected results, and test outputs.

- Input data for tests are provided in `tests/fixtures` directory. This `fixtures` directory contains a variety of test sets, including sequence data and intermediate files, intended for use and testing with pyANI-plus. The current test sets include:
  - `viral_example`: Three phage genomes that share similar regions, resulting in aligned regions. These viral genomes are the main test set due to the short runtime of the methods.
  - `bacterial_example`: Four bacterial genomes previously used for testing the legacy `pyANI`. This test set only being used in the test suite for the faster methods like fastANI and sourmash.
  - `bad_alignments`: Two highly divergent phage genomes with no shared regions, resulting in no alignments. This test set ensures that `pyANI-plus` correctly detects and handles such comparisons, which are recorded in the database as `NULL`.
  - `tools`: Mock tools designed for version testing of third-party tools. It also includes examples of faulty tools that are either not executable or do not provide version information.

Each test set (apart from `tools`) includes the follwoing subdirectories: - `intermediates`: Contains intermediate files expected to be generated by the ANI methods (e.g., `.filter` and `.delta` files for `ANIm`). - `matrices`: Includes the expected TSV matrix outputs for each ANI method implemented.

- Test output is written to temporary files that are automatically deleted after execution.

> Contributing and Writing Tests
>
> We welcome contributions from the community! If you would like to write new tests for `pyANI-plus`, please ensure your test data and operations follow this structure.

Running tests

To run tests with `pyest`, change directory to the root of the `pyANI-plus` repository, and invoke a `pyest` command.

Run all tests locally

To run all tests locally on your machine, issue the following command from the repository root:

```
pytest -v
```

Alternatively, if you wish to run tests and generate a coverage report, you can do so with the following command from the repository root:

```
python -m pytest -n auto --cov-report=html --cov=pyani_plus -v && open htmlcov/index.html
```

Run individual tests

Tests are organised in files with filenames matching the patter `test_*.py`. We write tests using functions, following the pytest style, as descibed in the pytest documentation.

For example, to run all ANIm-related tests, we can run:

```
pytest tests/test_anim.py
```

To run a specific test, such as the one that checks if the .delta files are parsed correctly, we can use the following command:

```
pytest tests/test_anim.py::test_delta_parsing
```

Part II

Licensing

This section provides the licensing terms for both `pyANI-plus` and its documentation. Before using `pyANI-plus` or its documentation, ensure that you have agreed to the terms of the license provided.

14 pyANI-plus

Unless otherwise indicated, all code is subjected to the following agreement:

© The University of Strathclyde 2024 Author: Leighton Pritchard

email: leighton.pritchard@strath.ac.uk

Adress:

Leighton Pritchard, Strathclyde Institute of Pharmacy and Biomedical Sciences, University of Strathclyde, 161 Cathedral Street, Glasgow, G4 0RE, Scotland, UK

The MIT License

Copyright © 2019 The Univeristy of Strathclyde

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

15 pyANI-plus documentation

Unless otherwise indicated, documentation is subjected to the following agreement:

© The University of Strathclyde 2024 Author: Leighton Pritchard

email: leighton.pritchard@strath.ac.uk

Adress:

Leighton Pritchard, Strathclyde Institute of Pharmacy and Biomedical Sciences, University of Strathclyde, 161 Cathedral Street, Glasgow, G4 0RE, Scotland, UK

Creative Commons Attribution Share Alike 4.0 International

Attribution-ShareAlike 4.0 International

===

Creative Commons Corporation ("Creative Commons") is not a law firm and does not provide legal services or legal advice. Distribution of Creative Commons public licenses does not create a lawyer-client or other relationship. Creative Commons makes its licenses and related information available on an "as-is" basis. Creative Commons gives no warranties regarding its licenses, any material licensed under their terms and conditions, or any related information. Creative Commons disclaims all liability for damages resulting from their use to the fullest extent possible.

Using Creative Commons Public Licenses

Creative Commons public licenses provide a standard set of terms and conditions that creators and other rights holders may use to share original works of authorship and other material subject to copyright and certain other rights specified in the public license below. The following considerations are for informational purposes only, are not exhaustive, and do not form part of our licenses.

```
Considerations for licensors: Our public licenses are
intended for use by those authorized to give the public
permission to use material in ways otherwise restricted by
copyright and certain other rights. Our licenses are
irrevocable. Licensors should read and understand the terms
and conditions of the license they choose before applying it.
Licensors should also secure all rights necessary before
applying our licenses so that the public can reuse the
```

```
material as expected. Licensors should clearly mark any
material not subject to the license. This includes other CC-
licensed material, or material used under an exception or
limitation to copyright. More considerations for licensors:
wiki.creativecommons.org/Considerations_for_licensors

Considerations for the public: By using one of our public
licenses, a licensor grants the public permission to use the
licensed material under specified terms and conditions. If
the licensor's permission is not necessary for any reason--for
example, because of any applicable exception or limitation to
copyright--then that use is not regulated by the license. Our
licenses grant only permissions under copyright and certain
other rights that a licensor has authority to grant. Use of
the licensed material may still be restricted for other
reasons, including because others have copyright or other
rights in the material. A licensor may make special requests,
such as asking that all changes be marked or described.
Although not required by our licenses, you are encouraged to
respect those requests where reasonable. More considerations
for the public:
wiki.creativecommons.org/Considerations_for_licensees

=======================================================================
```

Creative Commons Attribution-ShareAlike 4.0 International Public License

By exercising the Licensed Rights (defined below), You accept and agree to be bound by the terms and conditions of this Creative Commons Attribution-ShareAlike 4.0 International Public License ("Public License"). To the extent this Public License may be interpreted as a contract, You are granted the Licensed Rights in consideration of Your acceptance of these terms and conditions, and the Licensor grants You such rights in consideration of benefits the Licensor receives from making the Licensed Material available under these terms and conditions.

Section 1 – Definitions.

a. Adapted Material means material subject to Copyright and Similar Rights that is derived from or based upon the Licensed Material and in which the Licensed Material is translated, altered, arranged, transformed, or otherwise modified in a manner requiring permission under the Copyright and Similar Rights held by the Licensor. For purposes of this Public License, where the Licensed Material is a musical work, performance, or sound recording, Adapted Material is always produced where the Licensed Material is synched in timed relation with a moving image.

b. Adapter's License means the license You apply to Your Copyright and Similar Rights in Your contributions to Adapted Material in accordance with the terms and conditions of this Public License.

c. BY-SA Compatible License means a license listed at creativecommons.org/compatiblelicenses, approved by Creative Commons as essentially the equivalent of this Public License.

d. Copyright and Similar Rights means copyright and/or similar rights closely related to copyright including, without limitation, performance, broadcast, sound recording, and Sui Generis Database Rights, without regard to how the rights are labeled or categorized. For purposes of this Public License, the rights specified in Section 2(b)(1)-(2) are not Copyright and Similar Rights.

e. Effective Technological Measures means those measures that, in the absence of proper authority, may not be circumvented under laws fulfilling obligations under Article 11 of the WIPO Copyright Treaty adopted on December 20, 1996, and/or similar international agreements.

f. Exceptions and Limitations means fair use, fair dealing, and/or any other exception or limitation to Copyright and Similar Rights that applies to Your use of the Licensed Material.

g. License Elements means the license attributes listed in the name of a Creative Commons Public License. The License Elements of this Public License are Attribution and ShareAlike.

h. Licensed Material means the artistic or literary work, database, or other material to which the Licensor applied this Public License.

i. Licensed Rights means the rights granted to You subject to the terms and conditions of this Public License, which are limited to all Copyright and Similar Rights that apply to Your use of the Licensed Material and that the Licensor has authority to license.

j. Licensor means the individual(s) or entity(ies) granting rights under this Public License.

k. Share means to provide material to the public by any means or process that requires permission under the Licensed Rights, such as reproduction, public display, public performance, distribution, dissemination, communication, or importation, and to make material available to the public including in ways that members of the public may access the material from a place and at a time individually chosen by them.

l. Sui Generis Database Rights means rights other than copyright resulting from Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, as amended and/or succeeded, as well as other essentially equivalent rights anywhere in the world.

m. You means the individual or entity exercising the Licensed Rights under this Public License. Your has a corresponding meaning.

Section 2 – Scope.

a. License grant.

    1. Subject to the terms and conditions of this Public License, the Licensor hereby grants You a worldwide, royalty-free, non-sublicensable, non-exclusive, irrevocable license to exercise the Licensed Rights in the Licensed Material to:

        a. reproduce and Share the Licensed Material, in whole or in part; and

   b. produce, reproduce, and Share Adapted Material.

2. Exceptions and Limitations. For the avoidance of doubt, where Exceptions and Limitations apply to Your use, this Public License does not apply, and You do not need to comply with its terms and conditions.

3. Term. The term of this Public License is specified in Section 6(a).

4. Media and formats; technical modifications allowed. The Licensor authorizes You to exercise the Licensed Rights in all media and formats whether now known or hereafter created, and to make technical modifications necessary to do so. The Licensor waives and/or agrees not to assert any right or authority to forbid You from making technical modifications necessary to exercise the Licensed Rights, including technical modifications necessary to circumvent Effective Technological Measures. For purposes of this Public License, simply making modifications authorized by this Section 2(a)

   (4) never produces Adapted Material.

5. Downstream recipients.

   a. Offer from the Licensor – Licensed Material. Every recipient of the Licensed Material automatically receives an offer from the Licensor to exercise the Licensed Rights under the terms and conditions of this Public License.

   b. Additional offer from the Licensor – Adapted Material. Every recipient of Adapted Material from You automatically receives an offer from the Licensor to exercise the Licensed Rights in the Adapted Material under the conditions of the Adapter's License You apply.

   c. No downstream restrictions. You may not offer or impose any additional or different terms or

conditions on, or apply any Effective Technological Measures to, the Licensed Material if doing so restricts exercise of the Licensed Rights by any recipient of the Licensed Material.

6. No endorsement. Nothing in this Public License constitutes or may be construed as permission to assert or imply that You are, or that Your use of the Licensed Material is, connected with, or sponsored, endorsed, or granted official status by, the Licensor or others designated to receive attribution as provided in Section 3(a)(1)(A)(i).

b. Other rights.

   1. Moral rights, such as the right of integrity, are not licensed under this Public License, nor are publicity, privacy, and/or other similar personality rights; however, to the extent possible, the Licensor waives and/or agrees not to assert any such rights held by the Licensor to the limited extent necessary to allow You to exercise the Licensed Rights, but not otherwise.
   2. Patent and trademark rights are not licensed under this Public License.
   3. To the extent possible, the Licensor waives any right to collect royalties from You for the exercise of the Licensed Rights, whether directly or through a collecting society under any voluntary or waivable statutory or compulsory licensing scheme. In all other cases the Licensor expressly reserves any right to collect such royalties.

Section 3 – License Conditions.

Your exercise of the Licensed Rights is expressly made subject to the following conditions.

a. Attribution.

   1. If You Share the Licensed Material (including in modified form), You must:

   a. retain the following if it is supplied by the Licensor with the Licensed Material:

      i. identification of the creator(s) of the Licensed Material and any others designated to receive attribution, in any reasonable manner requested by the Licensor (including by pseudonym if designated);

      ii. a copyright notice;

      iii. a notice that refers to this Public License;

      iv. a notice that refers to the disclaimer of warranties;

      v. a URI or hyperlink to the Licensed Material to the extent reasonably practicable;

   b. indicate if You modified the Licensed Material and retain an indication of any previous modifications; and

   c. indicate the Licensed Material is licensed under this Public License, and include the text of, or the URI or hyperlink to, this Public License.

2. You may satisfy the conditions in Section 3(a)(1) in any reasonable manner based on the medium, means, and context in which You Share the Licensed Material. For example, it may be reasonable to satisfy the conditions by providing a URI or hyperlink to a resource that includes the required information.

3. If requested by the Licensor, You must remove any of the information required by Section 3(a)(1)(A) to the extent reasonably practicable.

b. ShareAlike.

In addition to the conditions in Section 3(a), if You Share Adapted Material You produce, the following conditions also apply.

1. The Adapter's License You apply must be a Creative Commons license with the same License Elements, this version or later, or a BY-SA Compatible License.
2. You must include the text of, or the URI or hyperlink to, the Adapter's License You apply. You may satisfy this condition in any reasonable manner based on the medium, means, and context in which You Share Adapted Material.
3. You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, Adapted Material that restrict exercise of the rights granted under the Adapter's License You apply.

Section 4 – Sui Generis Database Rights.

Where the Licensed Rights include Sui Generis Database Rights that apply to Your use of the Licensed Material:

a. for the avoidance of doubt, Section 2(a)(1) grants You the right to extract, reuse, reproduce, and Share all or a substantial portion of the contents of the database;
b. if You include all or a substantial portion of the database contents in a database in which You have Sui Generis Database Rights, then the database in which You have Sui Generis Database Rights (but not its individual contents) is Adapted Material,

   including for purposes of Section 3(b); and
c. You must comply with the conditions in Section 3(a) if You Share all or a substantial portion of the contents of the database.

For the avoidance of doubt, this Section 4 supplements and does not replace Your obligations under this Public License where the Licensed Rights include other Copyright and Similar Rights.

Section 5 – Disclaimer of Warranties and Limitation of Liability.

a. UNLESS OTHERWISE SEPARATELY UNDERTAKEN BY THE LICENSOR, TO THE EXTENT POSSIBLE, THE LICENSOR OFFERS THE LICENSED MATERIAL AS-IS AND AS-AVAILABLE, AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE LICENSED MATERIAL, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHER. THIS INCLUDES, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OR ABSENCE OF ERRORS, WHETHER OR NOT KNOWN OR DISCOVERABLE. WHERE DISCLAIMERS OF WARRANTIES ARE NOT ALLOWED IN FULL OR IN PART, THIS DISCLAIMER MAY NOT APPLY TO YOU.

b. TO THE EXTENT POSSIBLE, IN NO EVENT WILL THE LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY (INCLUDING, WITHOUT LIMITATION, NEGLIGENCE) OR OTHERWISE FOR ANY DIRECT, SPECIAL, INDIRECT, INCIDENTAL, CONSEQUENTIAL, PUNITIVE, EXEMPLARY, OR OTHER LOSSES, COSTS, EXPENSES, OR DAMAGES ARISING OUT OF THIS PUBLIC LICENSE OR USE OF THE LICENSED MATERIAL, EVEN IF THE LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH LOSSES, COSTS, EXPENSES, OR DAMAGES. WHERE A LIMITATION OF LIABILITY IS NOT ALLOWED IN FULL OR IN PART, THIS LIMITATION MAY NOT APPLY TO YOU.

c. The disclaimer of warranties and limitation of liability provided above shall be interpreted in a manner that, to the extent possible, most closely approximates an absolute disclaimer and waiver of all liability.

Section 6 – Term and Termination.

a. This Public License applies for the term of the Copyright and Similar Rights licensed here. However, if You fail to comply with this Public License, then Your rights under this Public License terminate automatically.

b. Where Your right to use the Licensed Material has terminated under Section 6(a), it reinstates:

   1. automatically as of the date the violation is cured, provided it is cured within 30 days of Your discovery of the violation; or

   2. upon express reinstatement by the Licensor.

   For the avoidance of doubt, this Section 6(b) does not affect any right the Licensor may have to seek remedies for Your violations of this Public License.

c. For the avoidance of doubt, the Licensor may also offer the Licensed Material under separate terms or conditions or stop distributing the Licensed Material at any time; however, doing so will not terminate this Public License.

d. Sections 1, 5, 6, 7, and 8 survive termination of this Public License.

Section 7 – Other Terms and Conditions.

a. The Licensor shall not be bound by any additional or different terms or conditions communicated by You unless expressly agreed.

b. Any arrangements, understandings, or agreements regarding the Licensed Material not stated herein are separate from and independent of the terms and conditions of this Public License.

Section 8 – Interpretation.

a. For the avoidance of doubt, this Public License does not, and shall not be interpreted to, reduce, limit, restrict, or impose conditions on any use of the Licensed Material that could lawfully be made without permission under this Public License.

b. To the extent possible, if any provision of this Public License is deemed unenforceable, it shall be automatically reformed to the minimum extent necessary to make it enforceable. If the provision cannot be reformed, it shall be severed from this Public License without affecting the enforceability of the remaining terms and conditions.

c. No term or condition of this Public License will be waived and no failure to comply consented to unless expressly agreed to by the Licensor.

d. Nothing in this Public License constitutes or may be interpreted as a limitation upon, or waiver of, any privileges and immunities that apply to the Licensor or You, including from the legal processes of any jurisdiction or authority.

==

Creative Commons is not a party to its public licenses. Notwithstanding, Creative Commons may elect to apply one of its public licenses to material it publishes and in those instances will be considered the “Licensor.” The text of the Creative Commons public licenses is dedicated to the public domain under the CC0 Public Domain Dedication. Except for the limited purpose of indicating that material is shared under a Creative Commons public license or as otherwise permitted by the Creative Commons policies published at creativecommons.org/policies, Creative Commons does not authorize the use of the trademark “Creative Commons” or any other trademark or logo of Creative Commons without its prior written consent including, without limitation, in connection with any unauthorized modifications to any of its public licenses or any other arrangements, understandings, or agreements concerning use of licensed material. For the avoidance of doubt, this paragraph does not form part of the public licenses.

Creative Commons may be contacted at creativecommons.org.