AMIMSpy

Release 0.1.0

Matthew Smith, Ralf Weber

Jun 18, 2021

CONTENTS

Python package for processing acoustic mist ionization mass spectrometry-based metabolomics and lipidomics data

CHAPTER

ONE

CONTENTS

1.1 Installation

1.1.1 Conda (recommended)

Install Miniconda, follow the steps described here

Start the `conda prompt`

- Windows: Open the `Anaconda Prompt` via the Start menu
- macOS or Linux: Open a `Terminal`

Create a amimspy specific `conda` environment. This will install a the dependencies required to run `amimspy`:

```
$ conda create --yes --name amimspy amimspy -c conda-forge -c bioconda -c computational-
↪metabolomics
```

Note:

- The installation process will take a few minutes.
- Feel free to use a different name for the Conda environment

You can use the following command to remove a conda environment:

```
$ conda env remove -y --name amimspy
```

This is only required if something has gone wrong in the previous step.

Activate the `amimspy` environment:

```
$ conda activate amimspy
```

To test your `amimspy` installation, in your Conda Prompt, run the command:

```
$ amimspy --help
```

or:

```
$ python
import amimspy
```

Close and deactivate the `amimspy` environment when you're done:

```
$ conda deactivate
```

1.1.2 PyPi

Install the current release of `amimspy` with `pip`:

```
$ pip install .
```

Note:

- The installation process will take a few minutes.

To upgrade to a newer release use the `--upgrade` flag:

```
$ pip install --upgrade amimspy
```

If you do not have permission to install software systemwide, you can install into your user directory using the `--user` flag:

```
$ pip install --user amimspy
```

Alternatively, you can manually download `amimspy` from GitHub or PyPI. To install one of these versions, unpack it and run the following from the top-level source directory using the Terminal:

```
$ pip install .
```

1.1.3 Testing

amimspy uses the Python `pytest` testing package. You can learn more about pytest on their homepage.

1.2 API reference

1.2.1 process

class `amimspy.process.`**`Scans`**(*run*, *well*, *well_scans*, *id_snr*, *id_tol*)

Bases: `object`

The Scans class.

This class is used to extract high quality scan data from a given sample using a user defined method.

Parameters

- **run** – Spectral data from multiple samples contained in a single *.mzML file
- **well** – Well label as provided in the corresponding metadata *.txt file
- **well_scans** – Scan IDs for all scans in a given well
- **id_snr** – User provided SNR threshold for differentiating between on and off scans

- `id_tol` – User provided number of features with SNR > id_snr to tolerate in off scans for labelling the scan type

`peaklists`(*well_scans*)

Peak lists are generated for all scan IDs provided as input. The peak lists include the spectral data (mz, intensity, snr, flags) for each scan. The peak lists havea hard SNR filter applied to diffeentiate between scan types - this is set to 15 by default.

Parameters `method` – well_scans: List of scan IDs from all scans in the given well.

Returns List of peaklist objects

`dictionary`()

A dictionary is generated using the scan IDs as keys and a binary identifier of the scan types as values (1 = 'on-scan' and 0 = 'off-scan'). The scan type is dertemined by the number of features with SNR above the applied SNR, by default >3 features needed to be labelled as 'on-scan'.

Returns Dictionary object

`padding`()

Converts the binary values in the dictionary to a string of binary values and adds padding (00) to either side. This padding enables on/off cycles to be identified at the start and end of each well.

:return String object

`extract`(*method*)

Generates a dictionary of possible on/off scan cycles (as binary patterns) from AMI-MS data as keys and the indices of the scans within each cycle to be extracted for the user defined method. The dictionary is then used to search the AMI-MS data for the provided scan cycles and extract the scan IDs required for downstream processing by calling the relavent function for the defined method. The scan IDs are returned as a list object.

Parameters `method` – Method to define which scans to extract data from. The following options are available:

- **all_scans** - Extracts data from all scans from the given well.
- **on_scans** - Extracts data from only the on scans from the given well.
- **off_scans** - Extracts data from only the off scans from the given well.
- **on_scan_no_edge** - Extracts data from only the on scans from the given well that are not immediately preceded or followed by an off-scan. For the unusual case of only two consecutive on scans, the single scan with the highest intensity is extracted. This is the default method.

Returns List object

AMIMSpy builds on top of DIMSpy. Documentation and the API reference for DIMSpy modules and functions are available from here.

1.3 Command Line Interface

```
$ amimspy --help

Executing amimspy version 0.1.0
usage: __main__.py [-h]
                   {process-scans,process-samples,hdf5-pm-to-txt,hdf5-pls-to-txt}
                   ...

Python package for processing acoustic mist ionisation-mass spectrometry
-based metabolomics and lipidomics data

positional arguments:
  {process-scans,process-samples,hdf5-pm-to-txt,hdf5-pls-to-txt}
    process-scans       Process and align scans within samples.
    process-samples     Process and align samples.
    hdf5-pm-to-txt      Write HDF5 output (peak matrix) to text format.
    hdf5-pls-to-txt     Write HDF5 output (peak lists) to text format.

optional arguments:
  -h, --help            show this help message and exit
```

```
$ amimspy process-scans --help

Executing amimspy version 0.1.0
usage: __main__.py process-scans [-h] -i source [source ...] -ms source
                                 [source ...] -o OUTPUT -f FAILED_WELLS -pr
                                 PROCESSED_SCANS
                                 [-m {all_scans,on_scans,off_scans,on_scan_no_edge}]
                                 [-d ID_SNR] [-t ID_TOL] [-s SNR_THRESHOLD]
                                 [-n MIN_SCANS] [-r RSD_THRESHOLD]
                                 [-fr MIN_FRACTION] -p PPM [-l METALIST]

optional arguments:
  -h, --help            show this help message and exit
  -i source [source ...], --input source [source ...]
                        Absolute or relative path to the *.mzml file(s). Must
                        be in same order as 'metascans *txt files'
  -ms source [source ...], --metascans source [source ...]
                        Absolute or relative path to the comma-delimited *.txt
                        metadata file. Must be in same order and 'input' *mzml
                        files. Header names must contain and be in the
                        following order names =['barcode', 'date/time', 'row',
                        'col', 'scan', 'ejection time', 'NA'] as output by MS-
                        Parser tool
  -o OUTPUT, --output OUTPUT
                        Absolute or relative path to the output file
  -f FAILED_WELLS, --failed-wells FAILED_WELLS
                        Absolute or relative path to the *.txt output of which
                        well failed
  -pr PROCESSED_SCANS, --processed_scans PROCESSED_SCANS
                        Absolute or relative path to the *.txt output of which
```

(continues on next page)

(continued from previous page)

```
                        well failed
  -m {all_scans,on_scans,off_scans,on_scan_no_edge}, --method {all_scans,on_scans,off_
↪scans,on_scan_no_edge}
                        Method to define which scans to extract data from.
                        DEFAULT = on_scans_no_edge
  -d ID_SNR, --id-snr ID_SNR
                        For identifying on/off scans: Hard SNR threshold for
                        differentiating between on/off scans. DEFAULT = 15
  -t ID_TOL, --id-tol ID_TOL
                        For identifying on/off scans: Number of features with
                        SNR > threshold to tolerate in off scans. DEFAULT = 3
  -s SNR_THRESHOLD, --snr-threshold SNR_THRESHOLD
                        SNR threshold to remove noise features. DEFAULT = 2
  -n MIN_SCANS, --min-scans MIN_SCANS
                        Minimum number of scans required to be labelled on
                        within a well for sample to be taken forward. DEFAULT
                        = 0
  -r RSD_THRESHOLD, --rsd-threshold RSD_THRESHOLD
                        RSD filter (scan level): Threshold of RSD of features
                        across scans in sample for it to be retained. DEFAULT
                        = None
  -fr MIN_FRACTION, --min-fraction MIN_FRACTION
                        Minimum fraction a peak has to be present. Use 0.0 to
                        not apply this filter.
  -p PPM, --ppm PPM     Aligning scans: m/z precision (ppm) to align scans in
                        sample - REQUIRED PARAMETER!
  -l METALIST, --metalist METALIST
                        Absolute or relative path to the tab-delimited *.txt
                        file that include the name of the data files (*.mzml)
                        and meta data. Column names: filename, replicate,
                        batch, injectionOrder, classLabel.
```

1.4 Credits

1.4.1 Developers & Contributors

- Matthew Smith (mjs708@student.bham.ac.uk) - University of Birmingham (UK)
- Ralf J. M. Weber (r.j.weber@bham.ac.uk) - University of Birmingham (UK)

1.4.2 Funding

AMIMSpy acknowledges support from the following funders:

- BBSRC and Waters Corporation for an iCASE PhD studentship.

1.5 Bugs and Issues

Please report any bugs that you find here. Or fork the repository on GitHub and create a pull request (PR). We welcome all contributions, and we will help you to make the PR if you are new to *git*.

1.6 Changelog

All notable changes to this project will be documented here. For more details changes please refer to github commit history

1.7 Citation

To cite AMIMSpy please use the following publication:

1.8 License

AMIMSpy is licensed under the GNU General Public License v3.0 (see LICENSE file for licensing information). Copyright © 2020 - 2021 Ralf Weber, Matthew Smith

CHAPTER

TWO

INDICES AND TABLES

- genindex
- search

PYTHON MODULE INDEX

INDEX