MSnPy

Release 0.1.0

Ralf Weber, Thomas Lawson

Sep 03, 2019

CONTENTS:

CHAPTER

ONE

INTRODUCTION

The latest Documentation was generated on: Sep 03, 2019

1.1 Installation

- MSnPy requires Python 3.7.+.
- MSnPy is published under GNU General Public License v3.0.

Get the latest version via github

https://github.com/computational-metabolomics/msnpy

or the latest package at pypi or conda

https://pypi.python.org/pypi/msnpy
hhttps://anaconda.org/bioconda/msnpy

1.2 Developers & Contact Information

Developers:

Ralf J. M. Weber - r.j.weber@bham.ac.uk
Thomas Lawson - t.n.lawson@bham.ac.uk

Contact information:

Ralf J. M. Weber

Phenome Centre Birmingham
School of Biosciences, University of Birmingham
Edgbaston, Birmingham, B15 2TT, U.K.
Tel: +44(0)121 414 3388
Email: r.j.weber@bham.ac.uk

https://github.com/computational-metabolomics

1.3 Citation

Please cite us when using MSnPy in your work.

ref

CHAPTER

TWO

MSNPY PACKAGE

2.1 Subpackages

2.2 Submodules

2.3 msnpy.msnpy module

2.4 msnpy.processing module

msnpy.processing.**assign_precursor**(*peaklist: dimspy.models.peaklist.PeakList*, *header_frag: str*, *tolerance: float = 0.5*)

Parameters

- **peaklist** –
- **header_frag** –
- **tolerance** –

Returns

Return type

msnpy.processing.**create_graphs_from_scan_ids**(*scan_dependents: list*, *scan_events: dict*, *ion_injection_times: dict*)

Create Directed Graph from scan dependent relationships

Parameters

- **scan_dependents** –
- **scan_events** –
- **ion_injection_times** –

Returns

Return type

msnpy.processing.**create_spectral_trees**(*trees: Sequence[networkx.classes.ordered.OrderedDiGraph]*, *peaklists: Sequence[dimspy.models.peaklist.PeakList]*)

Parameters

- **trees** – list of NetworkX graphs
- **peaklists** – list of PeakList objects

Returns

Return type Sequence[nx.OrderedDiGraph]

msnpy.processing.**create_templates**(*graphs: list*, *nh: int*)

Create a 'master' graph that include all the experimental trees Loop through all the subgraphs/graphs

Parameters

- **graphs** –
- **nh** –

Returns

Return type

msnpy.processing.**group_by_template**(*graphs: list*, *templates: list*)

Parameters

- **graphs** –
- **templates** –

Returns

Return type

msnpy.processing.**group_scans**(*filename: str*, *nh: int = 2*, *min_replicates: int = 1*, *report: str = None*, *max_injection_time: float = None*, *merge_ms1: bool = False*, *split: bool = False*, *remove: bool = True*)

Parameters

- **filename** –
- **nh** –
- **min_replicates** –
- **report** –
- **max_injection_time** –
- **merge_ms1** –
- **split** –
- **remove** –

Returns

msnpy.processing.**hdf5_peaklists_to_txt**(*filename: str*, *path_out: str*, *delimiter: str = '\t'*)

Parameters

- **filename** –
- **path_out** –
- **delimiter** –

msnpy.processing.**merge_ms1_scans**(*graphs: list*)

Parameters **graphs** –

Returns

Return type

msnpy.processing.**mz_tolerance**(*mz: float*, *tol: float*, *unit: str = 'ppm'*)

Parameters

- **mz** – mz value
- **tol** – tolerance
- **unit** – ppm or da

Returns

Return type float

msnpy.processing.**process_scans**(*filename: str*, *groups: list*, *function_noise: str*, *snr_thres: float*, *ppm: float*, *min_fraction: float = None*, *rsd_thres: float = None*, *normalise: bool = False*, *ringing_thres: float = None*, *exclusion_list: dict = {}*, *report: str = None*, *block_size: int = 5000*, *ncpus: int = None*)

Parameters

- **filename** –
- **groups** –
- **function_noise** –
- **snr_thres** –
- **ppm** –
- **min_fraction** –
- **rsd_thres** –
- **normalise** –
- **ringing_thres** –
- **exclusion_list** –
- **report** –
- **block_size** – number of peaks in each clustering block.
- **ncpus** – number of CPUs for parallel clustering. Default = None, indicating using as many as possible

Returns List of (average) PeakList objects (DIMSpy)

Return type Sequence[PeakList]

2.5 Module contents

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

m

INDEX