
MSPT

Release 1.0.0

Loïc Le Grégam

Apr 07, 2022

TUTORIAL:

1	Installation	3
2	Usage	5
3	References	7
	Python Module Index	9
	Index	11

Mass Spectrometry Pascal's Triangle (MSPT) is a python software to compute biases between experimental and theoretical values of isotopologue distributions during production of biological standards using controlled labeling patterns. For more information on the subject, check [this publication](#).

It is one of the routine tools used on the [MetaToul platform](#).

The code is open-source, and available on Github under a GPLv3 license.

It can be used from a command line interface or through an interactive jupyter notebook GUI. The notebook can be downloaded from the [GitHub page](#).

INSTALLATION

First, make sure that you have a python environment set up for the application with jupyter notebooks installed. For information on how to do this, please check [NmrQuant's documentation on installation](#).

Once this is done, open up a terminal in the activated environment and type in:

```
pip install MSPT
```

Next download the notebook file containing the gui interface from the [GitHub page](#).

Launch the jupyter notebook application from inside the environment and open the PascalTriangle notebook. You are now ready to use the software!

USAGE

First, launch the cell to initialize the *Graphical User Interface*.

Type in a name for the run. This name will be used to create the folder in which the results will be exported. All characters that are not allowed for naming folders should be avoided.

Tick the *Plot Mean Ratios* checkbox **if the data contains multiple samples** and the means have to be calculated.

Use the *Upload Isocor Data* button to upload your data.

Click on the *submit data* button. This will create the run folder and export the results file. A new dropdown menu will appear with the names of all the metabolites that are in the data file. **Use this dropdown to select the metabolites that should be plotted.** Once the metabolites have been selected, click the *submit metabolites* button to generate the dropdown menus for isotopologue selection.

When the metabolite dropdown menus appear, **the isotopologues for which the experimental ratios are beneath the threshold will automatically be unselected** (threshold is by default equal to 0.02). Select for each metabolite the isotopologues that must be plotted and click the *generate plots* button. This will generate the figures and you can review them before choosing or not to save them.

Once you are happy with the plots that you have, click the *save plots* button which will generate a pdf in which each requested metabolite is plotted (2 plots per page).

REFERENCES

3.1 TrianglePascal.py

class MSPT.TrianglePascal.TPascal(*verbose, threshold_value=0.02*)

Bases: object

calculate_biases()

Calculate bias between experimental ratio and theoretical values

calculate_mean_biases()

Calculate means and SDs for the calculated biases

calculate_mean_ratios()

Get the mean and standard deviation of each isotopologue ratio of a given metabolite between all samples

calculate_ratios()

Calculate Area/Total_Area ratios for each row

export_results(*run_name, home, mean_ratios*)

Export the final results table

Parameters

- **run_name** (*str*) – Name of the run (used for creating the end folder)
- **home** (*str*) – Path to root
- **mean_ratios** (*Boolean*) – Should means over all samples be calculated for the ratios

static get_abundance(*n, k, p*)

Get the abundance of an isotopologue (k) in the isotopologic space of the molecule containing n carbons from the abundance (p) of labelled precursor

Parameters

- **n** (*int*) – Number of carbon atoms in the molecule
- **k** (*int*) – Isotopologue
- **p** (*float*) – Abundance of labelled precursor

Returns Abundance of isotopologue k

static get_abundance_list(*n, p*)

Build a list of abundances for each carbon of the molecule

Parameters

- **n** (*int*) – number of carbons in the molecule
- **p** (*float*) – Abundance of labelled precursor

Returns list of abundances for each isotopologue of the molecule

get_isonumbs(*precursor_abundance=0.513*)

Calculate theoretical abundances of each isotopologue for each metabolite in each sample

import_isocor_data(*path*)

Import the data from Isocor output file

Parameters **path** (*str*) – path to Isocor file

prep_data()

Clean the data before performing calculations

3.2 notebook.py

class MSPT.notebook.TpN(*verbose*)

Bases: object

initialize_widgets()

Display initial widgets

load_events()

Prepare button click events

reset(*verbose*)

Function to reset the object in notebook (only for notebook use because without the function cell refresh doesn't reinitialize the object)

- search

PYTHON MODULE INDEX

m

MSPT.notebook, [8](#)

MSPT.TrianglePascal, [7](#)

INDEX

C

`calculate_biases()` (*MSPT.TrianglePascal.TPascal*
method), 7

`calculate_mean_biases()`
(*MSPT.TrianglePascal.TPascal* *method*),
7

`calculate_mean_ratios()`
(*MSPT.TrianglePascal.TPascal* *method*),
7

`calculate_ratios()` (*MSPT.TrianglePascal.TPascal*
method), 7

E

`export_results()` (*MSPT.TrianglePascal.TPascal*
method), 7

G

`get_abundance()` (*MSPT.TrianglePascal.TPascal* *static*
method), 7

`get_abundance_list()`
(*MSPT.TrianglePascal.TPascal* *static method*),
7

`get_isonumbs()` (*MSPT.TrianglePascal.TPascal*
method), 8

I

`import_isocor_data()`
(*MSPT.TrianglePascal.TPascal* *method*),
8

`initialize_widgets()` (*MSPT.notebook.TpN*
method), 8

L

`load_events()` (*MSPT.notebook.TpN method*), 8

M

module

MSPT.notebook, 8

MSPT.TrianglePascal, 7

MSPT.notebook
module, 8

MSPT.TrianglePascal
module, 7

P

`prep_data()` (*MSPT.TrianglePascal.TPascal method*), 8

R

`reset()` (*MSPT.notebook.TpN method*), 8

T

TPascal (*class in MSPT.TrianglePascal*), 7

TpN (*class in MSPT.notebook*), 8