MADPP Manual

What is MADDP?

Metabolomics automated data processing platform (MADPP) is a python-based pipeline for metabolomics data processing.
It is designed to process metabolomics data from raw data to final results. It is also designed to be flexible and easy to use.
The basic goal of this tool is to use pre-processed metabolomics data as input, and through specified data post-processing steps,
finally generate integrated data that is easy to analyze, as well as related data processing results and analysis reports. Automate and simplify the data processing process.

Quick start

1. Post-processing data
   - Command line tools:
     If you want to set parameters through a `config` file:
     `python post_processing.py -c ../config/post_processing_config.txt`
     If you want to set parameters by typing values into the command line tool:
     `python post_processing.py -i ../data/example_input -o ../data/example_postprocessing_output`
   - Using `post_processing.bat` file:
     Double click `post_processing.bat` file, then the post processing would be conducted. Before running the bat file, be sure to check whether the config file is configured (the setting of parameters when running post-processing using this method can only be completed through the configuration of the config file)
2. Data analysis
   - Command line tools:
     If you want to set parameters through a `config` file:
     `python data_analysis.py -c ../config/data_analysis_config.txt`
     If you want to set parameters by typing values into the command line tool:
     `python data_analysis.py -i ../data/example_input -o ../data/data_analysis_output`
   - Using `data_analysis.bat` file:
     Double click `data_analysis.bat` file, then the post processing would be conducted. Before running the bat file, be sure to check whether the config file is configured (the

setting of parameters can only be completed through the configuration of the config file)

How to use MADDP

Basic requirements

- Python 3.8 or higher
- Python package 'openpyxl' is required
  The working directory of the tool is under `~/script.` Before using the tool, please make sure that there are two folders, `data` and `config`, in the root directory. Please see the "File preparation" section for specific format requirements.

How to post-process data?

File preparation

1. Input data
   Please put the input files into the `data` folder. The file storage format and names in the folder must strictly follow
   those in `~/data/example_input`, including:

- `concentration_table`: The folder where the excel file of the concentration table is stored, the concentration table needs
  to be strictly named `ref.xlsx`
- `injection_information`: The folder where the excel file of the concentration table is stored, the concentration table
  needs to be strictly named `injection_information.xlsx`
- `raw_batch`: The folder where the raw data files are stored, the raw data files could be named as `Batch01`, `Batch02`, etc.

2. File details

- The metabolite types in the `ref.xlsx` need to be consistent with those in the raw data
- The naming method of the same sample in `ref.xlsx` and raw batch files must be strictly the same.
- The name of the sheet in `ref.xlsx` needs to be strictly consistent with the name of the raw batch files.

Parameter Description

In the parameter file, all recorded parameter names are consistent with the parameters related to the command line.
Please enter `python post_processing.py -h` to view the parameter description.
Optional arguments:

- `-h, --help`: show this help message and exit
- `-i , --input_file` : the path to input file, which contains raw batch files,injection information and concentration reference table
- `-o , --output_file` : the path to output file
- `-r , --replace_na_method` : the method of NA replacing, options: '1k', 'half_min' and other number
- `-s , --sample_blank_ratio` : the ratio of sample to blank
- `-sp , --sample_blank_ratio_passing_rate` : proportion of batches with normal sample:blank value
- `-sf", "--sample_blank_filter",` : default is `Flase` If `True`, metabolites with unqualified s:b ratio will be removed)
- `-b , --blacklist` : list of unwanted metabolites
- `-q , --qc_rsd` : the specified rsd of the qc
- `-qp , --qc_rsd_qc_rsd_passing_rate` : proportion of batches with normal rsd
- `-qf", "--qc_rsd_filter",` : default is `Flase` If `True`, metabolites with unqualified rsd will be removed)
- `-n , --output_name` : the name of output file, default value is name of input file
- `-c , --config` : the path to config file, the program will give priority

Output files

The data processing results are integrated into the `result.xlsx`, and the report information during the data
processing is conclued in the `report.txt`.

1. Sheets in `result.xlsx`

- **raw_data_combined**: Combination of original batch data, and simple NA replacement process was performed, and samples that
did not appear in the corresponding injection information were removed.In addition, compounds that do not meet the
requirements are removed based on the set sample:blank threshold and RSD threshold.
- **QC_filtered**: After merging the data according to the concentration table, the batch effect is removed.
- **TIC**: TIC normalized data.

- **Warning**: Metabolites with abnormal TIC values.
- **RSD**: RSD values of each metabolite in different batches
- **sample-blank**: sample/blank ratio values of each metabolite in different batches
- **QCs_TIC**: TIC normalized QC samples

How to analyse data?

File preparation

1. Input data

- `post_processed_file`: The `result.xlsx` obtained after data post-processing program
- `ref_table_file`: The `ref.xlsx` matched with `result.xlsx`

Parameter Description

In the parameter file, all recorded parameter names are consistent with the parameters related to the command line.
Please enter `python data_analysis.py -h` to view the parameter description.
Optional arguments:

- `-h, --help`: show this help message and exit
- `-p, --post_processed_file`: the path to input file, which is the result '.xls' file of post-processing
- `-r, --ref_table_file`: the path to ref file
- `-o, --output_file`: the path to output file
- `-f, --FC`: set a significant threshold for fold change
- `-s, --significance_test_method`: the significance testing method
- `-fdr, --FDR`: whether to use FDR
- `-l, --labels`: list of testing subgroups
- `-c --config`: the path to config file, the program will give priority

Output files

The output results are divided into `subgroup_data` and `result` two folders.

1. `subgroup_data`
   Split the original data into different sub-datasets according to the specified label, which are saved in `subgroup_data`
   folder
2. Files in `result` folder

- **xx_TIC**: Figure of the proportion of various metabolites TIC under each sample in each subgroup and QC
- **xx-xx_volcano**: Volcano plot analysis results between all subsets
- **statistics_result**: Statistical results including fold transformation and significance tests between all subsets