User manual: MSIParser (Updated 7/21/2023)

About MSIParser:

MSIParser is an open-source Python-based algorithm designed for analysis of single-cell mass spectrometry imaging data.

License:

MSIParser is freely available for download from GitHub (https://github.com/lingjunli-research/Automatic-MSI-Spectra-Extraction) and has an included user interface for increased accessibility.

Computational requirements:

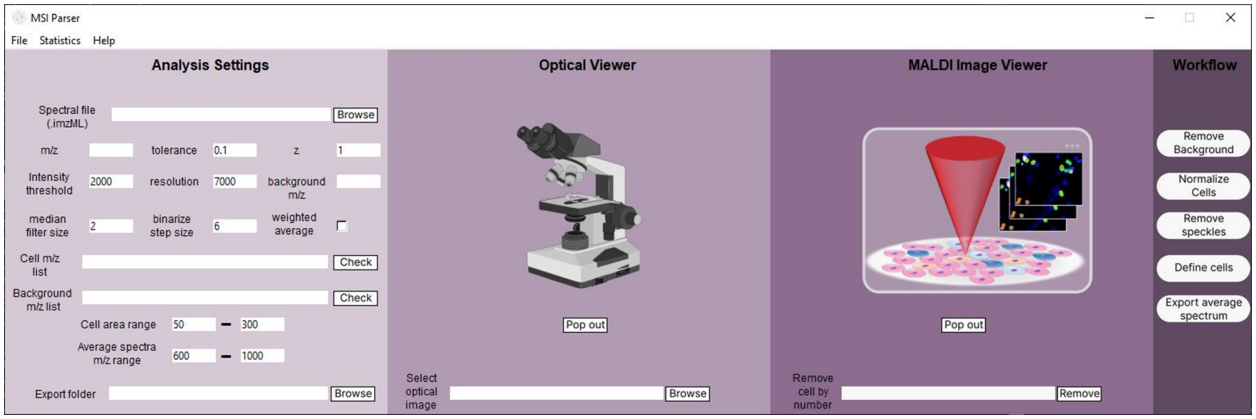
MSIParser is only supported for use with Windows OS, though limited functionality may be available with MacOS and Linux/UNIX. MSIParser is designed for functionality with .imzML mass spectra files.

Additional dependencies are necessary, which are outlined in the requirements.txt file on the home page of the MSI Parser repository. These can all be installed using pip.

Installation and start up (Command-line operation):

1. Download MSI Parser from GitHub (https://github.com/lingjunli-research/Automatic-MSI-Spectra-Extraction), save to any location in the C: drive.
2. Open command prompt of choice (either Anaconda or the default).
3. Navigate to the downloaded MSI Parser folder from within the command prompt.
4. Navigate to the "bin" folder within the MSI Parser directory using the command: `cd bin`.
5. Launch the GUI by the command: `python gui_v3.py`.

Home Screen functionalities:



1. **Selecting spectral file (.imzML):** The corresponding .ibd file should be located in the same folder
2. **m/z:** m/z of interest

3. **Tolerance:** error tolerance. Suggested value: 0.1
4. **Charge (z):** suggested value for MALDI: 1
5. **Intensity threshold:** Minimum intensity corresponding to a cell (global)
6. **Resolution:** suggested value: 7000
7. **Background m/z:** m/z where no cells are detected
8. **Median filter size:** Regions of this size will be removed as artifacts. Suggested value: 2.
9. **Binarize step size:** The fold difference of intensity required between the background and cell (local). Suggested value: 6.
10. **Weighted average:** Used for average spectra calculation.
11. **Background and cell m/z lists:** These lists must be the same number of m/zs long. Cell m/z are m/z values that you have individually determined display cells. Background m/z are to display no cells.
12. **Minimum/maximum cell size:** Cells outside of this size range will be automatically eliminated.
13. **Spectra range:** For the average spectra, these will be the m/z bounds displayed
14. **Export folder:** Folder where all results will be stored

Image analysis:

1. **Remove background:** Cells are binarized by taking the average cell pixel and subtracting the average background pixel. Spectra is parsed using pyimzML module: https://pyimzml.readthedocs.io/en/latest/
2. **Normalize cells:** This steps assigns pixels as a cell if they are surrounded by cell-pertaining pixels. This is conducted using the scipy.ndimage.binary_fill_holes module: https://docs.scipy.org/doc/scipy/reference/generated/scipy.ndimage.binary_fill_holes.html
3. **Remove speckles:** A median filter is applied to remove any small artifacts within the image. This is conducted using the scipy.ndimage.median_filter module: https://docs.scipy.org/doc/scipy/reference/generated/scipy.ndimage.median_filter.html
4. **Define cells:** Here, clusters of pixels are made and defined as cells, and filtered according to the number of pixels within the cell based on the assigned size boundaries.
5. **Remove cell by number:** If any cells are inaccurate, this is likely due to two cells overlapping in the analysis region. Here, the user can impart their own knowledge based on samples and by referring to the optical image to selectively remove cells based on their assigned number.
6. **Export results:** Exports a report of the average intensities for each m/z, a report for statistical analysis, and the average spectrum for each cell.

Statistical analyses:

Statistical analysis (general): Before doing any statistical analysis, be sure to open your statistics_report.csv file. You must then add a column titled “CellType” where each row includes the cell type before you can conduct statistical analysis.

For each analysis type, simply input the path to the statistical results, and the desired algorithm will be conducted and the associated figure will populate. Each of these algorithms are based

on well-established Python packages. Please refer to the documentation for each of these (linked below) for further details about the algorithm and necessary parameters.

PCA: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html
TSNE: https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html
UMAP: https://umap-learn.readthedocs.io/en/latest/basic_usage.html
Random Forest:
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
MLP Classifier:
https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html
Logistic Regression:
https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Tips:

1. **Checking a single m/z:** In the m/z list, if you would like to look at the mass spec image of just one m/z, simply put one m/z in the search field and click “Check”.
2. **Pop-out buttons:** Click these buttons and the current figure displayed will become interactive, allowing the ability to zoom/pan/save, etc.

Troubleshooting:

The program displays “not responding” when I click a command:

This occurs when the program is conducting a time-consuming task, the program will begin to respond again when the task is complete.

Other issues:
This document will be routinely updated as user issues arise. If your issue is not displayed in this document, feel free to either contact the current maintainer (lawashburn@wisc.edu), or open an issue on the GitHub page for this program (https://github.com/lingjunli-research/MSI-Parser/issues).