DeepHalo Documentation

Open-Source Pipeline for High-Confidence Detection of Halogenated Compounds in HRMS Data

Version 0.9.1

Table of Contents

Introduction

DeepHalo is a high-throughput computational pipeline designed for the detection and dereplication of halogenated compounds in **high-resolution** mass spectrometry (HRMS) data. It integrates cutting-edge deep learning models, robust isotope validation, and dual dereplication strategies to deliver high accuracy and efficiency. Key applications include natural product discovery and halogenated metabolite annotation.

Data Support

DeepHalo supports **high-resolution** mass spectrometry (HRMS) data, with or without tandem mass spectrometry (MS/MS) information. Note that the MS2 extraction function currently supports only data-dependent acquisition data(DDA). Additionally, DeepHalo exclusively supports the **.mzML** file format for analysis.

Core Features

1. Halogen Prediction

- **Element Prediction Model (EPM)**
  - Dual-branch Isotope Neural Network (IsoNN) architecture for Cl/Br detection.
  - Wide mass range: 50–2000 Da (resistant to interference from B, Se, Fe, and dehydro

isomers).

2. Isotope Pattern Validation

- **Dual Validation System**
  - *Mass Dimension*: Statistical rule-based correction.
  - *Intensity Dimension*: Autoencoder-based Anomaly Detection Model (ADM).

3. Multi-Level Scoring (H-score)

- A hierarchical scoring mechanism that combines predictions by leveraging isotope patterns at both the feature level and individual scan level.

4. Dereplication

- **Dual Strategy**
  - *Custom Database Matching*: Validates exact mass, halogen patterns, and isotope intensity similarity.
  - *MS2 Networking*: Integration with GNPS for spectral similarity analysis.

Technical Advantages

- **High Throughput**: process unlimited samples in <30 seconds each on standard hardware (Core i9, 16GB RAM).
- **Accuracy**: >98.6% precision in halogen detection across experimental LC-MS datasets.
- **Integration with GNPS**: Enhance Molecular Network Annotation in the Element Dimension by Embedding DeepHalo Results into GNPS Output GraphML File
- **Efficient Dereplication**: Significantly Higher Efficiency Compared to Molecular Networking Alone in GNPS

Installation

Prerequisites

- Python 3.10 (Verify with **python --version**).

Installation Methods

From PyPI

```
pip install DeepHalo
```

From Local Wheel

```
pip install path/to/DeepHalo-xxx.whl
```

From Source

```
git clone https://github.com/xieyying/DeepHalo.git
cd DeepHalo
```

```
pip install -e .
```

Quickstart

1. Detect Halogenated Compounds in mzML Files

```
halo analyze-mzml -i /path/to/mzml_files -o /output_directory -ms2
```

2. Dereplication with GNPS and/or Custom Database

```
halo dereplication -o /output_directory -g /path/GNPS_results -ud /path/custom_database.csv
```

Command-Line Usage

General Help

```
halo --help                    # List all commands
halo [command] --help          # Show options for a specific command (e.g., `halo analyze-mzml --help`)
```

Commands

1. Analyze mzML Files

```
halo analyze-mzml
   -i <input_path>             # Input .mzML file or directory (required)
   -o <project_path>           # Output directory (required)
   [-c <config_file>]          # Custom configuration (optional)
   [-b <blank_samples_dir>]    # Blank samples for subtraction (optional)
   [-ms2]                      # Enable MS2 extraction (optional)
```

2. Dereplication

```
halo dereplication
   -o <project_path>           # Output directory same as <project_path> in halo analyze-mzml
   -g <GNPS_folder>            # Unzipped GNPS results directory containing .GraphML file
   -ud <user_database.csv>     # Custom database (CSV/JSON, optional)
   -udk <formula_column>       # Column name for formula matching (optional)
```

3. Create Training Dataset

```
halo create-dataset <project_path> [-c <config_file>]
```

4. Train Model

```
halo create-model <project_path>
   [-c <config_file>]          # Custom configuration (optional)
```

```
  [-m <manual/search>]            # Training mode (default: manual)
```

Output Directory Structure

```
/output_directory
├─dereplication
│      Demo_data_1_feature.csv
│      Demo_data_2_feature.csv
└─result
    │  config.toml
    │  error_files.txt
    ├─halo
    │      Demo_data_1_feature.csv
    │      Demo_data_1_scan.csv
    │      Demo_data_2_feature.csv
    │      Demo_data_2_scan.csv
    └─ms2_output
            Demo_data_1.mzML
            Demo_data_2.mzML
```

Instruction

The whole DeepHalo analysis process including 4 steps

Step 1: Halogenate Mining

Run the command:

```
halo analyze-mzml -i INPUT_PATH -o OUTPUT_DIR -ms2
```

Parameters:

- -i INPUT_PATH
    - Input .mzML file or directory
    - Required parameter
    - Example: -i D:/data/ms_files
- -o OUTPUT_DIR
    - Output directory path
    - Required parameter
    - Example: -o D:/analysis/output
- -ms2
    - Enable MS2 data extraction
    - Optional parameter (required for steps 2-4)

    - Default: disabled

Example Usage:

```
halo analyze-mzml -i D:/data/ms_files -o D:/analysis/output -ms2
```

Output Files

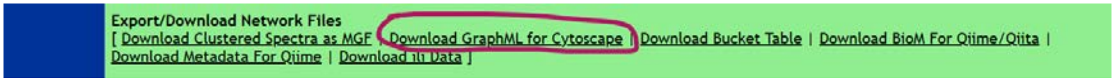
- config.toml: Analysis parameters
- error_files.txt: Failed mzML files
- halo/: MS1 information
- ms2_output/: MS2 data

Notes

- Results are filtered by H-score
- To disable filtering: Set H-score to 0 in config
- H-score visible in Cytoscape after dereplication
- For failed files:
    1. Retry analysis
    2. If error persists, reconvert raw data to mzML

Step 2: Molecular Networking analysis employing GNPS

1. Submit MS2 data in ms2_output to GNPS platform
2. Analyze molecular similarity using GNPS platform
3. Download and unzip results



Export/Download Network Files
[Download Clustered Spectra as MGF | Download GraphML for Cytoscape | Download Bucket Table | Download BioM For Qiime/Qiita | Download Metadata For Qiime | Download ili Data]

Example GNPS output structure:

```
└──Demo_GNPS_output
    │    METABOLOMICS-SNETS-V2-xxx-main.graphml
    │    params.xml
    ├──clusterinfo
    ├──clusterinfosummarygroup_attributes_withIDs
    ├──gnps_molecular_network_graphml
     └──result_specnets_DB
```

Step 3: Dereplication

Run the command:

```
halo dereplication -o PROJECT_PATH -g GNPS_DIR -ud DATABASE_FILE
```

- -o PROJECT_PATH
    - Output directory (same as used in analyze-mzml)

  - Example: -o D:/analysis/output
- -g GNPS_DIR
  - GNPS results directory containing .GraphML file
  - Example: -g D:/analysis/ Demo_GNPS_output
- -ud DATABASE_FILE
  - Custom database file (CSV/JSON)
  - Optional parameter
  - Example: -ud D:/databases/compounds.csv

Output Details

Upon completion of the analysis, the following files and directories will be generated:

1. **‘dereplication’ Directory**
   - Contains processed MS1 data in CSV format.
   - If a database was provided, compound match information will also be included in the csv file.
2. **GraphML File (generated when GNPS results are provided)**
   - The file name will have the suffix _adding_DeepHalo_results.
   - This file contains default H-score and Group prediction annotations, along with any database match information if a dereplication database was used.
3. **Processed Database File (if a database is provided)**
   - The file name will include the suffix _DeepHalo_dereplication_ready_database.
   - This file is formatted for future analyses, significantly reducing processing time in subsequent runs.

Example DATABASE_FILE:

compound_name	formula	Smiles
Compound1	C6H5Cl	ClC1=CC=CC=C1
Compound2	C6H4Cl2	ClC1=CC=C(Cl)C=C1

Important Notes

1. **File Format**
   - The input file must be in CSV format.
2. **Required Columns**
   - compound_name
   - formula
3. **Optional (but Recommended) Column**

    - Smiles (This aids in structure visualization).

4. **Additional Requirements**
    - Ensure that the formula column contains valid molecular formulas.
    - The file must be encoded in UTF-8.

Step 4: Visualization in Cytoscape

To visualize the results, open the GraphML file (with the suffix *_adding_DeepHalo_results.graphml) in Cytoscape. The file includes the following annotated variables:

- **H_scoreMean:**
    - **Range:** [0, 1]
    - **Default Threshold:** 0.4, above which the presence of halogen is inferred.
    - **Interpretation:** Higher values indicate a greater likelihood of halogen presence.
    - **Factors Affecting the Score:**
        - Sample complexity
        - Mass spectrometer resolution
- **Feature_based_prediction:**
    - **Halogenated Subclass Classification:**
        - *0:* Cl_n/Br_m ($n > 3$, $m > 1$, or the presence of both Cl and Br)
        - *1:* Br/Cl_3
        - *2:* Cl/Cl_2
- **Inty_cosine_score:**

  Measures the similarity of the isotope pattern. Higher values indicate a greater likelihood of a known compound
- **Compound_names and Adducts:**

  These fields provide the known molecular names and adduct information sourced from a user-provided database, offering essential metadata for compound identification.
- **Smiles**

  Provides the known molecular structure, sourced from either the GNPS library and a user-provided database.
- **error_ppm**

  Represents the mass error between the measured m/z value of the test molecule and that of its corresponding known compound match.

Dependencies,

- pandas == 2.0.3
- numpy == 1.22.0

- tensorflow == 2.10.1
- scikit-learn == 1.3.1
- pyopenms == 3.1.0
- Full list: See README.md

License

For methodology details and benchmarks, refer to the GitHub repository.