**User guide**

**Dependency**

The following software and packages are required:

* Python (version 3.9 or later, Anaconda distribution is recommended)
* TensorFlow (version 2.0 or later)
* Keras (packaged with TensorFlow)

The following Python packages are required:

* numpy (version 1.20.3)
* pandas (version 1.3.4)
* sklearn (version 0.24.2)
* scipy (version 1.7.1)

**Installation**

* Install Python (Anaconda)
* Install TensorFlow (TensorFlow is utilized in deep learning for predicting iRT, ion mobility and detectability. This step can be skipped if a database is already available.)

**Getting Started**

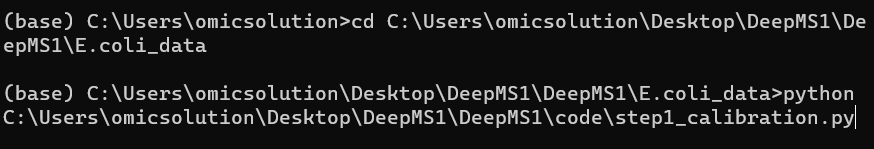
The MonoMS1 code primarily consists of three steps. It should be run in a folder containing MS2 data, MS1 features, and the corresponding database. Step 1 is executed for data preprocessing, which includes format adjustment and data calibration. Step 2 is carried out for the initial matching with a 15 ppm mass tolerance and training of the linear regression model. Finally, step 3 is performed for protein scoring and FDR filtering to obtain the final results.

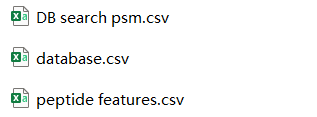
1. **Prepare a Database**

* A peptide list containing both target and decoy sequences is stored in a comma-separated values (CSV) file including columns named protein and sequence.
* The iRT, ion mobility and detectability of peptides are predicted by deep learning models (DeepDIA).
* The database should also include molecular weight and protein group information.
* Example data could be found in the “E.coli\_data” folder.

1. **Step1\_calibration**

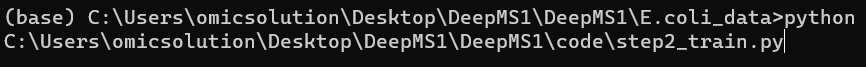
* Open the “Anaconda Prompt (py)”
* Entry to the folder including MS2 data (DB search psm.csv), MS1 features (peptide features.csv), and the corresponding database (database.csv).
* Run the step1\_calibration.py
* This step is executed for data preprocessing, which includes format adjustment and data calibration.

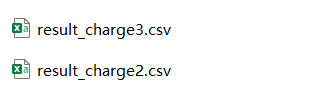




1. **Step2\_train**

* Run the step2\_train.py
* This step is carried out for the initial matching with a 15 ppm mass tolerance and training of the linear regression model.
* The output files include the result of both doubly and triply charged precursors (“result\_charge2.csv” and result\_charge3.csv”).





1. **Step3\_final**

* Run the step3\_final.py
* This step is performed for protein scoring and FDR filtering to obtain the final results.
* The output files include the final identification results with a 5% peptide-level FDR (“0.95-result\_final.csv”) and a 1% peptide-level FDR (0.99-result\_final.csv”).



