

## **Instruction for running DeepIso V1**

All you have to do is unzip the provided folder and put the raw files inside the directory:  
DeepIsoV1/rawdata/

The raw files are supposed to have .ms1 extension. You can use ProteoWizerd to convert your files to ms1 format if your files are in .mzml / .raw / other format. If you face trouble doing this, let me know, I will send you the screenshots for doing that.

In the folder DeepIsoV1, there are three more directories and python scripts.

DeepIsoV1/rawdata/ → should have all the .ms1 files

DeepIsoV1/data/ → will keep all the intermediate files and also the final feature table file.

DeepIsoV1/model/ → has all the trained models to be loaded by the python scripts.

### **Explanation of Codes:**

It is said in the manuscript that the .ms1 file is read and converted to a 2D matrix/array (np.array) where the rows represent the RT value and columns represent the m/z value. Pixels are 0.01 minute apart. Therefore these two points (15.01,400.2567) and (15.01,400.2621) becomes a single point (15.01, 400.26) and intensity of that point is the maximum of them. That intensity is converted to a grey scale value between 0 to 255 (but not discrete, its continuous).

We actually started with building a dictionary saving the same information as above. Therefore the key values are the RT values and for each RT, it has a list of (m/z, intensity) pairs. So its like: dictionary of RT(list of (m/z,intensity))). Or in python, dictionary(list).

If we have a raw file ABC.ms1 in the DeepIsoV1/rawdata/ directory, then we have to run following scripts:

### **Sequence of running the scripts:**

1. DeepIso\_v1\_preprocess\_makeDictionary.py (this one makes the dictionary)
2. DeepIso\_v1\_preprocess\_makeMS1.py (this one makes the 2D matrix)
3. DeepIso\_v1\_scanMS1\_isoDetect.py (this one runs the isodetect module on that 2D matrix)
4. DeepIso\_v1\_makeCluster.py (this one prepares a cluster list or prepares the batches to be passed together in the next step)
5. DeepIso\_v1\_reportFeature\_isoGroup.py (this is the final step producing the feature table)

In the beginning of each script, we can find the sample running instruction. That is, what command we have to put in order to run the scripts. We used command prompt of ubuntu for running the scripts. Since we have used python in ubuntu, so the file paths are set to be compatible with Linux environment. If someone has to run in Windows, then he might need to bring little changes in the file path which can be found in the beginning of each python script. In general, all should follow this format:

nohup python -u <script\_name> <parameter 1> ... <parameter n> > output.log &

This will put the logs in output.log file. We recommend to cd to that DeepIsoV1 directory and then run those scripts.

However, if any of these scripts generate minor errors, you can go through the scripts to bring little changes to solve the errors. Because different versions of python libraries sometimes generate minor errors.

Additional Comments:

1. For running the 3rd script, isoDetect, we set: parallel\_section=3 and batch\_size=5000. These can be changed based on your available GPU memory. Details are provided in that script.
2. Our dataset is generated with Orbitrap MS. In our dataset the scans were at least 0.01 RT apart. That means, the datapoints were at least 0.01 minute apart from each other. So they were like 10.02 min, 10.05 min, 10.7 min... along RT axis. So we actually hardcoded the minimum RT distance as RT\_unit=0.01. In your case, if the points are closer along RT axis, like: 10.0234, 10.024, 10.027 ... Then that 'RT\_unit' value should be changed in the scripts accordingly.