PAVE workflow documentation

Step 1. Initialization & Data preparation

* 1. Collecting all the .mzXML files converted from raw data with proper naming system. (samplename-polarity-scan#-method-replica.mzXML)
  2. Use parse tools to convert group .mzXML files into a Matlab readable file format (.mat) that containing only the data needed and removing all the redundant information. The variables stored in the .mat file is named M for [A B C D pblk] unlabeled and labeled data and M\_CID for (0eV, 2eV, 4eV) unlabeled CID data. (A=unlabeled, B=15N labeled, C=13C labeled, D=both labeled, pblk=procedure blank). Note that, when loading all the files, need to follow the correct sequence as above.
  3. Use Elmaven to generate a peaklist for unlabeled data (A), and save it as a .csv file. By default, columns 3, 5, 6 of the output csv file should correspond to groupID, mz, rt, respectively. (positive mode requires two list merge)
  4. Similarly, use Elmaven to generate another peaklist for all CID data. This list will be used to correct for rt shifts, if the CID experiment is performed at a different date. (positive mode requires two list merge)
  5. Setup the parameters for adduct list configuration file. One can add/delete/modify the list of common adducts or isotopes with name, mass difference. C/N matching criteria, Intensity ratio bounds. Note that, negative mode and positive mode should have different list and settings.
  6. Have available a database file, containing 6 columns including: ID, Name, Formula, Mass, C\_number, N\_number.
  7. A low C boundary files, containing an array of numbers m(i), defining the highest possible mass m for a given C number (the array index, i)
  8. Setup the parameters in the settings structure:

|  |  |  |
| --- | --- | --- |
| filed | description | default |
| Settings.mode | Neg=-1, pos=1 |  |
| Settings.rtm | Rt window resolution | 0.3 |
| Settings.ppm | Mass resolution | 10/1e6 |
| Settings.rt\_tol | RT tolorance | 0.1 |
| Settings.mz\_tol | Mz tolorance | 10/1e6 |
| Settings.ave | Moving average for EIC | 5 |
| Settings.prominence | Peak prominence for EIC | 1e3 |
| Settings.peakwidth | Min peakwidth for EIC | 0.02 |
| Settings.scorecutoff | Lowest corr. score for reliable atomcount | 0.75 |
| Settings.threshold | Lowest corr. Score for pattern recognition | 0.75 |
| Settings. override | Do only if feature is empty | 0 |
| Settings.verbose | Display(or not) the progress | 1 |
|  |  |  |

* 1. Edit the PAVE\_ini module file to specify input filenames, path and parameter settings and run it. The peak list (pks) should be initialized to contain 3 fields: pks.id, pks.mz, pks.rt.

Step 2. Remove duplicate peaks

Duplicate peaks are defined as peaks with similar mz and rt, both within the predefined tolerances. (see settings.mz\_tol and settings.rt\_tol). In pave\_main, Call the routine pks = pave\_find\_dup(pks,settings); and the pks list will be updated to remove the duplicated items.

Step 3. Atomcount

Solve C/N numbers, distinguish biological peaks (solvable) from background peaks (unsolvable), along with a score defining the reliability of the outcome. Call the routine [pks,idx] = pave\_atomcount(M,pks,settings,rep); to update pks with added fields of sig, N\_num, C\_num, score, scoremat and feature. The feature is filled with Backgound if not unsolvable.

Step 4. Junkremover

[pks,idx]=pave\_junkremover(M,pks,settings,rep,adduct);

4.1 find isotopes/adducts/multiplecharged. What’s common for these three categories is that, they should have putative parent peak appearing at similar rt and mass shifted by a well-defined mass difference dm. Repeatedly call the routine[pks,id,idx]=pave\_find\_adduct(M,pks,settings,rep,adduct(i))one adduct at a time.

4.2 find dimers. The putative parent for dimers are not defined by a fixed mass difference and therefore, this procedure is done separately from 4.1.

[pks,idx]=pave\_find\_dimer(M,pks,settings,rep);

4.3 find compounds that are the product of CO reaction. This is a special category which does not require rt matching. But mz matching and C/N matching are required.

[pks,idx]=pave2\_find\_CO(pks,settings)

4.4 Find parent ID. if the complete peak list is present, this step looks into it and find out if the identified putative parent peak is present in the list. If yes, record the peak ID for parent. If not, it is possibly a missing peak from peak picking and can be backfilled.

Step 5. Find low C, m is the boundary mz array. pks gets updated by assigning Low\_C in the feature field.

pks=pave\_find\_lowC(pks,settings,m);

Step 6. Identify fragment, list is the peaklist for CID data. pks gets updated by assigning fragment in the feature field.

pks=pave\_identify\_frag(M\_CID,list,pks,settings);

Step 7. Database search

pks=pave\_dbsearch(dbase,pks,settings);

.

|  |  |
| --- | --- |
| Pks.id | Peak id |
| Pks.mz | Peak Mz |
| Pks.rt | Peak Rt |
| Pks.sig | Intensity |
| Pks.C\_num | Carbon number |
| Pks.N\_num | Nitrogen number |
| Pks.score | Correlation score |
| Pks.scoremat | Correlation scores of all combination |
| Pks.feature | Peak annotation |
| Pks.description | Adduct/iso/.. name |
| Pks.parent | Parent peak in a pk structure |
| Pks.parentID | Parent peak ID |
| Pks.frag | T score for fragment id. |
|  |  |