Attached is the newest MSRESOLVE and some instructions. This software is not to be distributed to others as this is not yet publicly releasable. Use of the uncertainties feature (error bars) is the new feature. For now, papers using that feature require a collaboration with me.

The below instructions are sortof “reverse” order, because I think it is easier to first understand how to do an analysis, and then understand how to make a reference file.

* **Attached are 3 zip files.** One has the complete MSRESOLVE package (without GUI), the other has a working example (without GUI). These are provided without GUI because the uncertainties feature does not yet work with the GUI.
* **For normal analysis going forward:** For you, it makes sense that when you are starting a new analysis that you will start with the “minimum working example” as a new directory. So you won’t normally use the complete package, except to open it to look at the guide or something like that.
  + The guide is very far out of date. If you decide to edit the guide Please use track changes and let me know, so that I can incorporate changes into the master copy.
* **Info on the uncertainties feature:** Right now the uncertainties feature (with error bars) is not yet compatible with iterative. It is also not compatible with the GUI. In fact, the GUI is several features behind. Further information about uncertainties will be provided below. Right now, the most important new feature that people will want to use is:
  + UserChoices['dataAnalysisMethods']['implicitSLScorrection'] = True
  + **What does this feature do?**
  + Imagine if you have two molecules with patterns and intensities like this:
    - Molecule A: m15, 2
    - Molecule A: m17, 50
    - Molecule B: m15, 20
    - Molecule B: m17, 3
  + A simple way of (approximately) solving such a problem is to set the small peaks values to 0. That feature exists in MSRESOLVE and is a very important feature. Any standardized reference pattern with value below this variable will be set to 0: UserChoices['minimalReferenceValue']['referenceValueThreshold'] = [3.1]
  + That feature also introduced errors.
  + This new feature, 'implicitSLScorrection', after the analysis is done, adds \*back\* the small features, correcting the concentrations and including some uncertainty into the error bars from this.
* **Steps for a typical analysis (when reference file is already perfectly made).**

1. Start with minimal working directory (just copy the ExampleAnalysis directory). Stick your reference file in there along with your collected data file.
   1. Make sure they are in the right format. Make sure there are no “empty rows” or “empty columns”. Open in notepad to check (excel can hide empty rows/columns).
2. Plot your collected data file in excel. Figure out if you need individual linear baseline corrections for each mass. Record *regions* of time for “early” and “Late” baselines. Like [10.0,15.0] might be an early baseline and [100.0,120.0] might be a late one. You can also just use a single region (just early baseline, no late one – “early” one can even be at the end of data). Zoom in and make as many graphs as needed. Sometimes you need to make one graph for each mass since they might need to have separate background correction factors.
   1. In UserInput.py, turn on UserChoices['linearBaselineCorrectionSemiAutomatic']
   2. Fill in the early baseline times and late baseline times.
3. Go to UserChoices['minimalReferenceValue']['on'] = 'yes'. A good initial choice is to make:
   1. UserChoices['minimalReferenceValue']['referenceValueThreshold'] = [1.0]
   2. UserChoices['minimalReferenceValue']['referenceSignificantFragmentThresholds'] = [5.0]
   3. The first of these settings makes values 0 in reference file [but now gets partially or completely added back with the 'implicitSLScorrection' feature].
   4. The second of these settings specifies how big a peak has to be before it is significant – this \*only\* affects the order of solving the problem. This used to be more important before the uncertainties feature and slsweighting features were added. Now, those two features should usually already solve the problem to reduce the solving errors. Regardless, this feature is still good to use as in some analyses it could make a difference.
4. Turn on uncertainties:
   1. UserChoices['uncertainties']['calculateUncertaintiesInConcentrations'] = True
   2. UserChoices['uncertainties']['referenceFileUncertainties'] = 2 #If you don’t know your reference file’s uncertainties, a good initial choice is 2 or 5.
      1. You can also use the word ‘File’, that means you have to have an uncertainty for \*each\* value, like in the example provided.
   3. UserChoices['uncertainties']['collectedFileUncertainties']
      1. This has four options:
         1. you can provide a list like [1.37E-10,0,0,0,1.24E-10] with one value for each mass (constant across the file).
         2. You can use ‘Auto’ which uses local standard deviations
         3. You can use ‘File’ and provide a file (with one value for \*each\* datapoint)
         4. You can use ‘None’.
   4. The other two features of uncertainty are not yet implemented.
5. Data Analysis Methods:
   1. UserChoices['dataAnalysisMethods']['answer'] = 'sls' #'inverse' or 'sls'; sls is suggested
   2. UserChoices['dataAnalysisMethods']['uniqueOrCommon'] = 'unique' #'unique' or 'common'; now ‘unique’ is suggested.
   3. UserChoices['dataAnalysisMethods']['slsWeighting'] = [1,0,0,0] #You should probably keep this as [1,0,0,0] or make it [2,1,1,1]
   4. UserChoices['dataAnalysisMethods']['implicitSLScorrection'] = True
6. concentrationFinder #This is only needed if you need to convert the units coming out to something like bar, torr, etc.

* **Steps for making a typical reference file *from* reference patterns. This step is not well described here, not the purpose of this document.**

1. download JDX files from nist webbook for each molecule.
2. Use JDX Converter to convert them into the right format. <https://github.com/AdityaSavara/JDX_Converter>
3. When possible, make reference patterns from your own instrument. Gather some reference data from your own instrument, then extract the pattern (see next section). Under normal circumstances, you will have some reference patterns collected on your own instrument and some that you could not calibrate and had to get from NIST webbook.
4. follow the example inside the full package documentation for 190930TuningCorrectorInstructions.docx (this allows to make a reference file consisting of patterns collected from your instrument and also from NIST).

* **Steps for making reference pattern from collected data.**

1. Collect a reference pattern from your own instrument.
   1. Advice: Usually you will want to take a scan where there are 10 minutes or longer with just inert gas or vacuum, and then you will introduce your chemical continuously for 10 minutes or longer, followed by another 10 minutes or longer of background again. For some setups 10 minutes is sufficient for each region, for other setups 1 hour is required for each region.
      1. It is okay to collect multiple calibrations in the same file.
2. Create a directory as if you will be making an analysis.
   1. Your collected data will be the “collected data” file.
   2. You will \*still\* need a real or fake reference pattern file. You can make a reference file that has a column for this chemical and then just put all the intensities as 1, for example.
3. Plot the data in excel and get the baseline correction information, as you would for a regular analysis.
4. Fill in the UserInput file with the reference extraction feature on and with export at each step turned on. Run the analysis. You will find that the first “Exported” file is the extracted reference pattern along with the uncertainties for each mass signal (as determined by the standard error of the mean from the region where the extraction occurred).
   1. Be advised that there may be other sources of error like baselines etc., so the true errors can be larger than the uncertainty provided.

* **Concentration Units & Ionization Factors**

1. In general, one can convert to concentration units using UserChoices['concentrationFinder']
   1. the concentrations can be for a single molecule or for a list of molecules. A list of molecules requires using “SeparateMoleculesFactors”
   2. one specifies a mass and a particular molecule and the intensity at that mass, and what concentration it corresponds to. Then all molecules get converted into that unit.
      1. One can also specify multiple molecules using lists for everything (other than the unit name). In that case any molecules that are not specified will get normalized according to the 1st molecule specified.
2. I had been keeping this a secret for a while, but I suppose I should stop doing so:
   1. the Madix & Ko correction factors are not sufficiently accurate. In general, different molecules in different classes of molecules have different ionization factors. In the package, there is a feature for this and I have been collecting ionization factors for various molecules. So to have real accuracy, one should use molecules in that list and/or identify realistic guesses.
      1. Note: the ionization factors are not dependent ‘only’ on molecule type. It’s \*mostly\* on number of electrons and \*partially\* on molecule type.