This feature is for correcting reference patterns between different mass spectrometer tunings. The original variable names were written for 'correcting' ones own mass spectrometer's tuning to match an external tuning (NIST’s), though it can be used for the opposite tuning to bring an external pattern to match one’s own spectrometer. The feature compares some data collected on your spectromer to that with data from external data, then corrects for your spectrometer’s tuning factor during the analysis. This can improve quantitative accuracy. The tuning correction factors and their uncertainties are also exported in the log file when available. Further, the reference pattern uncertainties associated with the correction are also exported when available.

**There are two cases. Choose the case that is correct. The final reference pattern should match what data will be analyzed. So if one is analyzing experimental data collected on their own mass spectrometer, one would want to use Case 2 from below.**

**CASE 1: Changing your patterns to match the external reference pattern (e.g., NIST) Using the Tuning Factor Correction Feature:**

1. Make two directories. A directory named “\TuningCorrection\” and a directory named “\Analysis\”. Fill the two directories to be identical (with the same files you plan to use during a regular MSRESOLVE analysis).
   1. In both directories, you will have made a reference file intended for analysis (the one that goes in referenceFileNamesList). Name this file “ReferenceForAnalysisBeforeTuningCorrection.csv”
      1. It is assumed that this file will have some ‘self-measured’ reference patterns, and possibly also some ‘NIST/Standard’ reference patterns. (thus, it is some kind of mixed reference file where different molecules have patterns from different sources, where you only want to apply tuning correction to some molecules).
2. In the directory named \TuningCorrection\ , add two *more* reference files, give the csvs names as below:
   1. UserChoices['inputFiles']['referenceFileNamesList'] = [ReferenceForAnalysisBeforeTuningCorrection.csv']
   2. UserChoices['measuredReferenceYorN']['referenceFileToTune'] =['ReferenceCollected.csv','xyyy'] 🡨 this has at least one molecule (typically multiple molecules) and contains *only* patterns collected from your instrument. This file includes the molecules which will be used for ‘calibration’ and *also* the molecules which will become corrected.
   3. UserChoices['measuredReferenceYorN']['referenceFileToMatch'] =['ReferenceLiterature.csv','xyyy'] 🡨 this has at least one molecule and contains *only* patterns from NIST or other standard reference. This file should only include molecules that you will use for ‘calibration’. Actuallhy, it can include extraneous molecules, but cannot include the molecules you are trying to correct.
   4. There must be *at least one* molecule that appears in both patterns. Fragmentation patterns that span a large range (like Hexane and Heptane) are ideal.
3. In the directory named \TuningCorrection\ directory, open UserInput.py (or the GUI)
   1. Turn on the TuningCorrector feature (also called measuredReferenceYorN), to yes
   2. Make sure that the extractReferencePatternFromDataOption is off (unless you are intentionally using that feature and want a tuned version of that feature output)
   3. Run MSRESOLVE in this directory with whatever settings you want, but DataAnalysis must be on. While it is intended to work with pre-processing, at present, this feature only works with DataAnalysis on.
   4. There will be a file created named Exported…TuningCorrector.csv Open this file. (if there is more than one, then open the one with the highest number).
      1. Copy the patterns for the molecules which you are trying to correct, as their tuning has now been corrected.
         1. The tuning factors used are in LogFile.txt along with uncertainties, but you don’t need that when following these instructions.
      2. You are only copying out the molecules that you wanted adjusted. For any molecules that you didn’t want adjusted, leave them!
         1. the program may have put out corrected patterns for the other patterns also, but those one are garbage, those are like having the tuning correction applied twice.
   5. In the directory named \TuningCorrection\ open the file named ReferenceForAnalysisBeforeTuningCorrection.csv
      1. Paste in the Tuning Corrected patterns you’ve copied in the previous step
         1. You are only pasting over for the molecules you wanted corrected.
            1. You will be pasting over all or some of the fragmentation patterns – it is fine if the molecules you are pasting are standardized to 100 even if other molecules are not.
         2. For other patterns, leave them as they are!
         3. For the patterns you have pasted over, it is good to add “Tuning\_Corrected” in the row labelled Source (for example, “NIST\_Tuning\_Corrected”)
      2. Save this updated file as **ReferenceForAnalysisAfterTuningCorrection**.csv
4. In the directory named \Analysis\
   1. Copy to this directory the file **ReferenceForAnalysisAfterTuningCorrection**.csv from the \TuningCorrection\ directory.
   2. In the UserInput.py or GUI change referenceFileNamesList to now use ReferenceForAnalysisAfterTuningCorrection
      1. UserChoices['inputFiles']['referenceFileNamesList'] = [ReferenceForAnalysisAfterTuningCorrection.csv']
   3. Open UserInput.py or GUI and turn off the feature TuningCorrector feature, also called measuredReferenceYorN, by setting it to “no”. Your Tuning correction has already been done, so the feature needs to be off.
   4. Run your analysis with your new corrected or mixed reference file!

**CASE 2: Changing external reference patterns (e.g., NIST) to match the measured reference pattern Using the Tuning Factor Correction Feature:**

1. Make two directories. A directory named “\TuningCorrection\” and a directory named “\Analysis\”. Fill the two directories to be identical (with the same files you plan to use during a regular MSRESOLVE analysis).
   1. In both directories, you will have made a reference file intended for analysis (the one that goes in referenceFileNamesList). Name this file “ReferenceForAnalysisBeforeTuningCorrection.csv”
      1. It is assumed that this file will have some ‘self-measured’ reference patterns, and possibly also some ‘NIST/Standard’ reference patterns. (thus, it is some kind of mixed reference file where different molecules have patterns from different sources, where you only want to apply tuning correction to some molecules).
2. In the directory named \TuningCorrection\ , add two *more* reference files, give the csvs names as below:
   1. UserChoices['inputFiles']['referenceFileNamesList'] = [ReferenceForAnalysisBeforeTuningCorrection.csv']
   2. UserChoices['measuredReferenceYorN']['referenceFileToTune'] =['ReferenceLiterature.csv','xyyy'] 🡨 this has at least one molecule (typically multiple molecules) and contains *only* patterns collected from the external pattern (e.g., NIST). This file includes the molecules which will be used for ‘calibration’ and *also* the molecules which will become corrected.
   3. UserChoices['measuredReferenceYorN']['referenceFileToMatch'] =['ReferenceCollected.csv','xyyy'] 🡨 this has at least one molecule and contains *only* patterns measured from Your pattern. This is *not* the external pattern. This file should only include molecules that you will use for ‘calibration’. Actually, it can include extraneous molecules, but cannot include the molecules you are trying to correct.
   4. There must be *at least one* molecule that appears in both patterns. Fragmentation patterns that span a large range (like Hexane and Heptane) are ideal.
3. In the directory named \TuningCorrection\ directory, open UserInput.py (or the GUI)
   1. Turn on the TuningCorrector feature (also called measuredReferenceYorN), to yes
   2. Make sure that the extractReferencePatternFromDataOption is off (unless you are intentionally using that feature and want a tuned version of that feature output)
   3. Run MSRESOLVE in this directory with whatever settings you want, but DataAnalysis must be on. While it is intended to work with pre-processing, at present, this feature only works with DataAnalysis on.
   4. There will be a file created named Exported…TuningCorrector.csv Open this file. (if there is more than one, then open the one with the highest number).
      1. Copy the patterns for the molecules which you are trying to correct, as their tuning has now been corrected.
         1. The tuning factors used are in LogFile.txt along with uncertainties, but you don’t need that when following these instructions.
      2. You are only copying out the molecules that you wanted adjusted. For any molecules that you didn’t want adjusted, leave them!
         1. the program may have put out corrected patterns for the other patterns also, but those one are garbage, those are like having the tuning correction applied twice.
   5. In the directory named \TuningCorrection\ open the file named ReferenceForAnalysisBeforeTuningCorrection.csv
      1. Paste in the Tuning Corrected patterns you’ve copied in the previous step
         1. You are only pasting over for the molecules you wanted corrected.
            1. You will be pasting over all or some of the fragmentation patterns – it is fine if the molecules you are pasting are standardized to 100 even if other molecules are not.
         2. For other patterns, leave them as they are!
         3. For the patterns you have pasted over, it is good to add “Tuning\_Corrected” in the row labelled Source (for example, “NIST\_Tuning\_Corrected”)
      2. Save this updated file as **ReferenceForAnalysisAfterTuningCorrection**.csv
4. In the directory named \Analysis\
   1. Copy to this directory the file **ReferenceForAnalysisAfterTuningCorrection**.csv from the \TuningCorrection\ directory.
   2. In the UserInput.py or GUI change referenceFileNamesList to now use ReferenceForAnalysisAfterTuningCorrection
      1. UserChoices['inputFiles']['referenceFileNamesList'] = [ReferenceForAnalysisAfterTuningCorrection.csv']
   3. Open UserInput.py or GUI and turn off the feature TuningCorrector feature, also called measuredReferenceYorN, by setting it to “no”. Your Tuning correction has already been done, so the feature needs to be off.
   4. Run your analysis with your new corrected or mixed reference file!