README: MS/MS Data Processing R Code

1. From Agilent Mass Hunter, run “Find by Targeted MS/MS”. Make sure that under the “Results” tab, you click “Extract MS spectra” and “Extract MS/MS spectra”, and make sure you select “Extract separate MS/MS spectrum per collision energy”.
2. Export a .CEF file for your MS/MS data.
3. Run the cef\_conversion() code to convert the .CEF file from Mass Hunter to a .MGF or .MS file that can be used with the SIRIUS software. This will generate a .MGF and a .MS file for each compound in each sample. For more information on these file formats, see SIRIUS documentation, including examples in “demo data” folder: <https://bio.informatik.uni-jena.de/software/sirius/>.
4. Run the compileforsirius() function to compile the individual compounds’ .MGF and .MS files into 1 .MGF or 1 .MS file for each sample. This makes importing into SIRIUS a bit easier.
5. Run the .MGF or .MS files through SIRIUS with CSI:FingerID. In SIRIUS, export CSI:FingerID results. This will produce a separate folder for each sample you ran through the software. Each folder will contain several sub-folders with information on each individual compound in the data.
6. Run SiriusImport(). This will pull out the top ranking structure from SIRIUS (that fits all elemental constraints set in SIRIUS, when possible) for each compound. This will produce 2 CSV files: one formatted specifically for use with the APRL-SSP Python script (published by Ruggeri and Takahama, manuscript: <https://www.atmos-chem-phys.net/16/4401/2016/>, code: <https://github.com/stakahama/aprl-ssp>), and one that contains information on molecular formulas, compound class, etc. (this will be used in later processing).
7. Enumerate atmospherically relevant functional groups. Here, we run the Ruggeri and Takahama Python script (details above), but other tools for this also exist. Typically, for these tools, you start with a list of SMILES identifiers and you search for the presence of different functional groups with SMARTS patterns.
8. Finally, run PythonImport(). This will combine the results from the Ruggeri and Takahama Python script (that enumerates functional groups) with molecular formula information (from step 6), SMILES information (step 6), and information about neutral mass and ionization (from step 3).