A FTICR Data Analysis Tutorial

Before we can analyze some of these data, we need a couple programs and files:

* All of the files (including this walkthrough) can be found on GitHub.
  + <https://github.com/danczakre/ICRTutorial>
* If you don’t have it installed already, please install RStudio.
  + <https://www.rstudio.com/products/rstudio/download/>
* Once RStudio is installed, please run the following commands:
  + *install.packages(c(“vegan”, “reshape2”, “ggplot2”, “devtools”, ”dplyr”, “tidyr”, “easycsv”))*
  + *devtools::install\_github("EMSL-Computing/ftmsRanalysis")*
  + **Note:** Copying and pasting these commands may not work – they might need to be directly typed into RStudio. If you wish to copy and paste these commands rather than re-type them, there is a file named “install\_packages\_commands.txt” on the repository from which the commands can be copied.
* An extra program is necessary to learn alignment/calibration
  + Formularity: <https://omics.pnl.gov/software/formularity>
  + Be sure to download the **program** and the **database**

Once you have these files installed, you can follow along with the tutorial.

The primary goal of this tutorial is to demonstrate one potential avenue through which FTICR data you receive from EMSL can be analyzed. While many of these scripts were written to be broadly applicable, I can’t guarantee that they will work with all data types – only those FTICR reports generated by Formularity are currently acceptable. Furthermore, some of the figure generation steps are hard coded for the tutorial dataset. However, as EMSL transitions to other report formats, I will be creating another GitHub repository which will contain more broadly compatible analysis scripts/functions.

**Protocol:**

1. Open the folder containing Formularity and double-click the Formularity.exe icon
   1. Drag and drop the .ref file from GitHub into the calibration file box
   2. Drag and drop the database file from omics.pnnl.gov into the database file location
   3. Change the following settings
      1. CIA: Checked
      2. Ionization: proton\_detachment
      3. Adduct: ensure this is blank
      4. Under Calibration:
         1. Start tolerance, ppm: 5
         2. Regression: linear
      5. Under Formula Assignment:
         1. Alignment: Checked
         2. Alignment tolerance, ppm: 0.5
         3. Max relationship gaps: 2
      6. User-defined filter: O>0 AND (N+S+P)<6 AND S<3 AND P<2
   4. Drag and drop all .xml files into the green box labeled “Drop Spectra Files”
2. Run the **FTMS\_Analysis.R** script by double-clicking it, and then either clicking the “Source” button in the upper left-hand corner of the script editing window or pressing *Option+Command+R* on Macs or *Ctrl+Shift+Enter* on Windows.
   1. This script uses the R package upon which FREDA is based to take a Formularity report, filter peaks based upon mass range and isotopic signature, calculate stoichiometric measurements, and output a commonly formatted dataset.
   2. You should have two files named “Processed\_HJ\_Andrews\_Tutorial\_Data.csv,” which contains the compound intensity by sample data, and “Processed\_HJ\_Andrews\_Tutorial\_Mol.csv”, which contains the molecular characteristics.
3. Using these two files, you can run the **Transformation\_Analysis.R** script in the same way detailed in step 1.
   1. This script uses a database of known biochemical transformations and their specific masses to identify compounds which might be involved in a reaction
   2. This tutorial only guides the analysis of amino acid transformations but the whole database that we typically use is also provided on GitHub
      1. Analyzing with the whole database can be time consuming depending on the number of samples
   3. This script will generate many files and two figures:
      1. There will be two folders created titled “Transformation Peak Comparisons” and “Transformations per Peak”. These folders contain peak-level information about the transformations and can be used in network generation
      2. A transformation profile will also be created – this will detail the number and types of transformations associated with each sample
      3. The figures plot 1) a comparison of AA transformations between surface and pore water, and 2) a comparison of AA transformations to themselves within the overall dataset
4. Now we will analyze the carbon use efficiency of each compound by running the **Carbon\_Efficiency.R** script to calculate. Ycs,i or “the stoichiometric coefficient of *i*th carbon source.” This metric is a measure of the number of moles carbon required to produce biomass
   1. Low values would indicate an efficient system because fewer moles are required to generate biomass; high values in turn would indicate an inefficient system
   2. This script will create two figures which will plot the Ycs,I distributions within each sample and the means by sample. Using this data, we can gain an understanding of how the carbon-use efficiency within HJ Andrews watershed changes through space and time.