TEnvR: MATLAB-based Toolbox for Environmental Research

Aleksandar I. Goranov[a], Rachel L. Sleighter[a,b], Dobromir A. Yordanov[c], and Patrick G. Hatcher[a,*]

[a]Department of Chemistry and Biochemistry, Old Dominion University, Norfolk, VA, USA
[b]FBSciences, Inc., Research and Development, Norfolk, VA, USA
[c]Neuralink Corp., San Francisco, CA 94110

*Corresponding authors: Dr. Aleksandar I. Goranov (aleksandar.i.goranov@gmail.com) and Dr. Patrick G. Hatcher (phatcher@odu.edu)

TEnvR Tutorial (version 2021)

Table of Contents

Section 1. Preface

This document contains a detailed tutorial for using the codes of the Toolbox for Environmental Research (TEnvR, pronounced “ten-ver”). In order to effectively follow this tutorial, it is necessary that researchers have prior familiarity with natural organic matter (NOM), as well as the analytical technique they want to use TEnvR for. TEnvR includes codes for data from ultraviolet-visible (UV-VIS) absorption spectra, excitation-emission matrix (EEM) fluorescence spectra, nuclear magnetic resonance (NMR) spectra, and ultrahigh resolution mass spectrometry peak lists from FT-ICR-MS or Orbitrap instrumentation (ultrahigh resolution mass spectrometry techniques). TEnvR also includes codes for statistical analysis of multivariate data. The scope of this tutorial is to guide researchers through the different capabilities of TEnvR for efficiently processing, visualizing, and mining different types of data. This tutorial will not provide detailed theory and background of the analytical techniques nor of the statistical approaches (e.g., principal component analysis). However, we have provided numerous useful references to show examples of the application of different techniques. The mathematics behind all codes are explained in sufficient detail, but the tutorial has been kept less technical and more novice-friendly. This tutorial is written from the perspective of applied analytical chemistry rather than a perspective of computer science. Knowledge and experience in programming with MATLAB or other languages (e.g., R, Python) is recommended but not required. Researchers that successfully follow through the entire tutorial will not only learn how to process, visualize, and evaluate data from different analytical techniques, but also will advance their knowledge and experience with MATLAB programming. For visual guidance throughout this tutorial, a PowerPoint file is provided as a second piece of supporting information (filename “2021 TEnvR Tutorial Slides.pptx”) containing screenshots with step-wise instructions of the execution of each code in the MATLAB interface.

For easier comprehension of this tutorial, we have used several different font formats: Section names as well as script names are in **Black Bold** (e.g., **UVVIS_Metrics**, **EEM_Visualize**, **FTMS_Compare3**). Slide numbers with corresponding screenshots from the 2021 TEnvR Tutorial Slides.pptx file are in **Red Bold** (e.g., **Slide 1**, **Slide 5**). *Green Italic* is used for commands that are to be keyed in MATLAB’s Command Window (e.g., *clear*, *clc*, *UVVIS_Dilution('Sample 1.csv','dilute',4,16)*) or for buttons in the MATLAB interface (e.g., *Save*, *Set Path*, etc.). Any code segments (e.g., function names such as *interp1*) are also written in *Green Italic*.

This tutorial is for the first version of the toolbox (TEnvR 2021). The codes of TEnvR and this document will be revisited annually to include modifications, improvements of algorithms, enhancement of capabilities, and/or inclusion of new codes. Researchers are welcome to contact the corresponding authors with any feedback on the present codes or with any requests for new codes or capabilities to be included in future updates. Future versions of this toolbox as well as any related announcements will be published on GitHub (https://github.com/alex-goranov/TEnvR) and ResearchGate (https://www.researchgate.net/project/TEnvR-MATLAB-based-Toolbox-for-Environmental-Research).

TEnvR is free software for non-commercial use: you can redistribute it and/or modify it under the terms of the GNU General Public License (GPL) as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version (GPL-3.0-or-later). The GNU GPL is a copyleft license allowing users to freely use, modify, and redistribute the codes within this toolbox. However, any further codes or software products derived from this product must be open-source. For further information please refer to the License.txt file located in TEnvR 2021\Supplementary files. TEnvR is also registered with the U.S. Copyright Office. We require that the copyright statement (© Old Dominion University Research Foundation) remains in the license notice section of the codes and the toolbox is cited accordingly in any produced work as following:

Goranov, A. I., Sleighter, R. L., Yordanov, D. A., and Hatcher, P. (2023): TEnvR: MATLAB-Based Toolbox for Environmental Research, *Journal TBD*, doi: XXXXXXXXXXX.

The toolbox is free for non-commercial purposes. Users wishing to use TEnvR commercially must obtain a commercial license from Old Dominion University Research Foundation – for more information please contact the corresponding authors.

Section 2. Installation of MATLAB and setting up the TEnvR toolbox on a Windows operating system

To properly operate the scripts in TEnvR, you must have MATLAB and Microsoft Excel installed on your computer. Both must be installed in English. The decimal separator in Excel must be set to a dot symbol (.)[1]. All scripts have been tested on the MATLAB 2020b, 2021a and R2022a versions. To install MATLAB, please follow the guidelines on MathWorks (https://www.mathworks.com/help/install/). You can also watch a video using the following link (https://www.mathworks.com/videos/how-to-install-matlab-1525083586145.html) or contact your local IT department. During the installation, the product installer will ask which products you would like to be installed (**Slide 2**). Please install the following packages: MATLAB, Communications Toolbox, Curve Fitting Toolbox, Deep Learning Toolbox, DSP System Toolbox, Parallel Computing Toolbox, Signal Processing Toolbox, and Statistics and Machine Learning Toolbox (total memory required ~13 GB). Users are welcome to install any additional packages they want or even install all. Installing all packages requires ~40 GB of memory. For the installation, we recommend connecting to internet using a LAN cable to allow for a faster download of packages.

The installation will create a folder named “toolbox” in its installation folder (e.g., C:\Program Files\MATLAB\R2022a\toolbox) where all internal functions and any additional packages are installed. The functions within this folder are internally accessible by MATLAB. After installation of the software, a folder named MATLAB is also created in the Documents folder (e.g., C:\Users\Administrator\Documents\MATLAB). This “MATLAB” folder is also internal, thus when a code is executed in MATLAB, the software will search for that particular code either in the “toolbox” or “MATLAB” folders.

Once MATLAB is installed, open the application. The main software interface, known as the MATLAB desktop, will open, and it is split in several windows for interactive work shown on **Slide 3** and described below. If your MATLAB interface is different, you can click *Layout* and select *Default*. If using *Default* does not show the Workspace window, please select *Three Column*. The MATLAB desktop has several components:

- → Current Folder – this is the location on your computer where MATLAB is currently able to search for files, import data from, and export data to. This folder is easily changed by the *Browse for folder* button. Changing the “Current Folder” is going to be crucial for performing many of the automation steps in this toolbox, as described in the sections below. The Current Folder currently is selected to be C:\Users\Administrator\Documents\MATLAB (**Slide 3**). Please note that using folders on some virtual spaces (e.g., OneDrive) may cause problems though in other virtual spaces (e.g., DropBox) we have not experienced issues so far.
- → Details window – this window generally displays the functions’ subsections used within scripts or can display figures if they are selected in the Current Folder. This window not be utilized in this tutorial.
- → Command Window – this is where commands from the operator are typed in.
- → Workspace – this is where all variables are displayed to the user. Please note that when a script is used as a function, the Workspace does not get populated. In order to see all variables, the function needs to be evaluated using the debugger (more

[1] To check this, go Click File > Options. Then, on the Advanced tab, under Editing options, you will see the settings for Decimal and Thousand separators. Make sure that Decimal separator is a dot (.). You may have to uncheck the “Use system separators” check box.

information found at the following link: https://www.mathworks.com/help/matlab/matlab_prog/debugging-process-and-features.html).

When using some of the MATLAB scripts of TEnvR, the “Editor” interface will also be used. In cases when graphics are produced, figure windows will pop up automatically. An example is shown on **Slide 4** with the use of the **EEM_Process_Aqualog** script.

→ Editor – this is where scripts that are being opened are displayed. Please note that you do not have to have a script open to use it, but the script must be located in an internal MATLAB directory.
→ Figure windows generally pop up and display numerous plots, graphs, tables, etc. that are generated using the various scripts. Any of the settings (fonts, styles, sizes, colors, etc.) can be changed either by altering the script or by Edit → Figure Properties...

Once MATLAB is installed, download TEnvR from any of the links below:
→ GitHub (https://github.com/alex-goranov/TEnvR)
→ ResearchGate (https://www.researchgate.net/project/TEnvR-MATLAB-based-Toolbox-for-Environmental-Research)

Please note that future versions of this toolbox will be uploaded annually on these websites. To install TEnvR, download the compressed file containing the toolbox. If you downloaded TEnvR from ResearchGate, this zip file will be named “TEnvR 2021.zip”. If you downloaded TEnvR from GitHub, this zip file will be named “TEnvR-main.zip”. Extract its contents by right-clicking on it and selecting “Extract all…”.

If you downloaded the toolbox from GitHub: 1) please rename the folder as TEnvR 2021, and 2) create two new folders in the TEnvR 2021/TestData_EEM directory named “TestData_Generic_Processed” and “TestData_Aqualog_Processed”.

Move the whole “TEnvR 2021” folder to your MATLAB directory (**Slide 5**). The toolbox can be stored anywhere on your computer and then routed to MATLAB using the *Set Path* button (**Slide 6**) even though we recommend that any user-made scripts or externally downloaded toolboxes (such as TEnvR or drEEM (Murphy et al., 2013)) are put in the “MATLAB” folder.

Once TEnvR is moved to the MATLAB directory, type the following commands (in *Green Italic*) into the Command Window (see **Slide 5**):

Slide 5: *addpath(genpath('TEnvR 2021'))*
savepath

Alternatively, use the *Set Path* button (**Slide 6**) to internally route TEnvR to MATLAB. Click *Set Path*, then select Add with Subfolders… (**Slide 6**). The TEnvR folder is greyed out, because a new dialog window is open and the MATLAB desktop is in the background. In the popped up window, find the location of TEnvR, select it, and click *Select Folder* (**Slide 7**). Then you will see that TEnvR and its subfolders has been internally routed to MATLAB (**Slide 8**) - click *Save* and then *Close*. You can double-click on the folder to open it and see its contents from the MATLAB desktop (**Slide 9**). TEnvR contains four types of files denoted by their different icons: folders with exemplary data or supplementary files, scripts, functions, and configuration files. The use of these different files will be demonstrated throughout the different sections of this tutorial.

Future versions of TEnvR will include modifications to enable its use on other operation systems such as Macintosh, which is at present not possible. The use on TEnvR via MATLAB Online is also not possible at present.

Section 3. Lists of codes and supporting files in TEnvR (version 2021)

	Script	Purpose
1	**EEM_Difference**	Calculating a differential (difference) spectrum between two EEM spectra
2	**EEM_Dilution**	Scaling up (undiluting) or down (diluting) an EEM spectrum
3	**EEM_Fold**	Converting (folding) 3D EEM spectra into 2D arrays prior to multivariate two-way statistics (HCA, PCA, etc.)
4	**EEM_Metrics**	Calculating EEM spectral metrics (peak intensities, spectral indices, etc.)
5	**EEM_Process_Aqualog**	Processing raw EEM data into processed “final” data. This code is for data acquired on HORIBA Aqualog spectrofluorometers (instrument-output data is already mostly pre-processed)
6	**EEM_Process_Generic**	Processing raw EEM data into processed “final” data from other spectrofluorometers
7	**EEM_ShimadzuReformat**	Reformatting EEM data output by Shimadzu RF-6000 spectrofluorometers into a readable format by codes in drEEM
8	**EEM_ThermoReformat**	Reformatting EEM data output by Thermo Scientific Lumina spectrofluorometers into a readable format by codes in drEEM
9	**EEM_Transposition**	Transposing EEM data (switching axes)
10	**EEM_Unfold**	Converting (unfolding) 2D arrays into 3D EEMs after PCA analysis (only for component loadings)
11	**EEM_Visualize**	Creating contour plots of one, two, or three EEM spectra
12	**EEM_WaterRamanAverage**	Averaging five replicate 2D emission scans (typically water Raman scans) from Shimadzu RF-6000 spectrofluorometers
13	**EEM_WaterRamanReformatAqualog**	Reformatting 2D water Raman spectral output by HORIBA Aqualog spectrofluorometers into a readable format by codes in drEEM
14	**FTMS_AlignmentFormulas**	Combining multiple formula lists and aligning formulas to produce an alignment matrix
15	**FTMS_Automation**	Automated processing of numerous peak lists and their formula assignment; metrics calculations and compound class categorization on numerous formula lists
16	**FTMS_Compare**	Comparison of two formula lists using the Presence/Absence approach
17	**FTMS_Compare3**	Comparison of three formula lists using the Presence/Absence approach
18	**FTMS_Compare4**	Comparison of four formula lists using the Presence/Absence approach
19	**FTMS_CompoundClass**	Categorization of molecular formulas into different compound classes
20	**FTMS_ConfigurationAssignment**	Configuration file for peak refinement and formula assignment

21	**FTMS_ConfigurationToolbox**	Configuration file for all other FTMS codes in TEnvR
22	**FTMS_Figures**	Producing various figures for visualizing formula lists
23	**FTMS_Figures_MD**	Producing figures for assessing formula lists based on mass defect
24	**FTMS_FormulaAssignment**	Assignment of formulas to peak lists
25	**FTMS_KMD**	Kendrick Mass Defect series analysis of an individual formula list
26	**FTMS_KMD_Ox**	Kendrick Mass Defect oxygenation series analysis of an individual formula list
27	**FTMS_KMD2**	Kendrick Mass Defect series analysis of two individual formula lists using the Presence/Absence approach
28	**FTMS_Metrics**	Calculating various metrics for a formula list
29	**FTMS_Peptides**	Identifying formulas within a formula list that can be small oligomeric peptides
30	**FTMS_Process**	Automated peak refinement, formula assignment, and formula refinement of an individual peak list
31	**FTMS_RefinementFormulas**	Refinement of an individual formula list
32	**FTMS_RefinementPeaks**	Refinement of an individual peak list
33	**FTMS_SpearmanCorrelation**	Spearman correlation of aligned formulas of a dataset with external parameters

34	**NMR_Automation**	Automated processing routine for 1D NMR spectra

35	**Stats_HCA**	Hierarchical cluster analysis (HCA)
36	**Stats_PCA**	Principal component analysis (PCA)
37	**Stats_CorrMatrix**	Pearson/Kendall/Spearman correlation matrix

38	**UVVIS_Automation**	Automated processing of numerous UV-VIS spectra
39	**UVVIS_Derivative**	Calculating a first order (or nth order) derivative spectrum
40	**UVVIS_Differentiation**	Calculating a differential (difference) spectrum
41	**UVVIS_Dilution**	Scaling up (undiluting) or down (diluting) a UV-VIS spectrum
42	**UVVIS_Metrics**	Calculating UV-VIS spectral metrics (E_4:E_6, slope ratio, etc.) and the Napierian absorbance spectrum
43	**UVVIS_Process**	Processing raw UV-VIS data into processed “final” data
44	**UVVIS_ReformatAqualog**	Reformatting UV-VIS spectra acquired on HORIBA Aqualog spectrofluorometers into a readable format by codes in TEnvR and drEEM

Internal codes	Purpose
combn	Internal code for **FTMS_Peptides**, creates combinations of elements
FTMS_Candidate	Internal code for **FTMS_RefinementFormulas**, creates candidates for the KMD filter
FTMS_CandidateCollection	Internal code for **FTMS_RefinementFormulas**, creates a candidate collection for the KMD filter
FTMS_KMD_Collection	Internal code for **FTMS_RefinementFormulas**, creates a KMD collection for the KMD filter
FTMS_KMD_Value	Internal code for **FTMS_RefinementFormulas**, creates KMD values for the KMD filter
natsort	Internal function for **natsortfiles**, performs natural-order/alphanumeric sorting of strings, character vectors, or categorical data
natsortfiles	Internal function for **UVVIS_Automation**, **UVVIS_Differentiation**, **EEM_Fold**, **EEM_Metrics**, **FTMS_Automation**, performs natural-order/alphanumeric sorting of filenames and folders
spear	Internal code for **FTMS_SpearmanCorrelation and Stats_CorrMatrix**, performs a Spearman correlation

Supplementary files	Purpose
EEM_Metrics.xlsx	Detailed description of all metrics in the **EEM_Metrics** code
FTMS_AminoAcids.xlsx	Molecular formulas of amino acids used in **FTMS_Peptides**
FTMS_CompoundClasses.docx	Summary of different compound classes from different studies
FTMS_PAHs.xlsx	List of polycyclic aromatic hydrocarbons (PAHs) used for developing the Ring metric in **FTMS_Metrics**
License for natsortfiles and natsort.txt	License file for **natsort** and **natsortfiles** codes
License.txt	License file for the TEnvR toolbox

Section 4. Ultraviolet-visible (UV-VIS) Spectroscopy

The codes for UV-VIS spectral data are developed to import the spectral data from comma-separated values (.csv) files. The wavelengths must be listed in the first column, and the absorbance values must be listed in the second column. Some spectrometers (e.g., Thermo Scientific Evolution 201) output data sorted by wavelength from longest to shortest. Thus, all codes are equipped with a sorting function to reorganize the data by wavelength from shortest to longest. In order to keep the codes simple, the option to import a variety of different types of files was not encoded. Instead, it is recommended that users modify the codes and tailor them to their needs. For example, if spectral data from Excel files (.xlsx or .xls) is to be loaded, *dlmread* / *dlmwrite* functions in all scripts must be substituted with *xlsread* / *xlswrite* and name extensions must be changed from *'.csv'* to *'.xlsx'* throughout the scripts.

Below is a tutorial for using the scripts pertaining to UV-VIS data. We have provided 20 test samples (Sample 1 (01) - Abs Spectra Graphs.dat to Sample 20 (01) - Abs Spectra Graphs.dat) as well as a blank test sample (Blank (01) - Abs Spectra Graphs.dat) generated using a HORIBA Aqualog spectrofluorometer.

To use the UV-VIS codes, first put all of your UV-VIS data in a separate folder and select this folder as your Current Folder using the *Browse for Folder* button. An example is shown using the TestData_UVVIS folder inside the TEnvR folder (**Slide 10**).

- **UVVIS_ReformatAqualog**

In many environmental labs, UV-VIS spectra are acquired in parallel with three-dimensional fluorescence spectra on HORIBA Aqualog spectrofluorometers. These instruments export the produced UV-VIS spectra as .dat files. Furthermore, the exported file contains much more data than the needed two arrays of wavelengths and absorbance values. The **UVVIS_ReformatAqualog** code is specifically to reformat such data into a TEnvR-friendly format. The code will automatically import all spectra (in .dat format) and reformat them into comma-separated value (.csv) files. To run this code, type *UVVIS_ReformatAqualog* in the Command Window with TestData_UVVIS being your Current Folder (**Slide 11**). You can also copy the command in *Green Italic* from here and paste it into the Command Window.

Slide 11: *UVVIS_ReformatAqualog*

After running this line in the Command Window, the code will collect all filenames in the Current Folder ending with "Abs Spectra Graphs.dat". Then, the code will individually load each of these files, reformat it, and create a corresponding .csv file. Files are also sorted by wavelength and their filenames are trimmed to include only the filename (i.e., "(01) - Abs Spectra Graphs" is removed). These new files will be used in the examples below. For this and all other codes in TEnvR, once the code is done it will output a completion message in the Command Window (e.g., *Finished reformatting Blank (01) - Abs Spectra Graphs.dat*, **Slide 11**).

Note: During this tutorial, the Command Window and Workspace should be cleaned up after each code is presented. Any auxiliary windows (e.g., Figure windows) should also be also closed. To clean these spaces, sequentially (in different lines) type *close all*, *clear* and then *clc* in the Command Window.

- **UVVIS_Dilution**

The dilution of samples prior to UV-VIS spectrophotometric analysis is common as direct UV-VIS measurements must not exceed absorbance of 1.0 in order to keep the measured absorbance within the linearity limits of Beer-Lambert's law. The UV-VIS code is used to rescale a UV-VIS spectrum to account for the dilution. This is done based on a dilution factor (DF), which is calculated as the volume of concentrated sample that was used ($\mathrm{V_{initial}}$) to be diluted into a final volume $\mathrm{V_{final}}$. In analytical textbooks (Harris, 2015), these volumes are referred to as $\mathrm{V_1}$ and $\mathrm{V_2}$, respectively. The first example sample of the provided dataset was in fact diluted 11 times (an 11-fold dilution, or diluted with a DF of 0.0909), and that is shown in the calculation below.

$$\mathrm{DF} = \frac{\mathrm{V_{initial}}}{\mathrm{V_{final}}} = \frac{\mathrm{V_1}}{\mathrm{V_2}} = \frac{1\ \mathrm{mL\ sample}}{1\ \mathrm{mL\ sample} + 10\ \mathrm{mL\ water}} = \frac{1}{11} = 0.0909$$

The **UVVIS_Dilution** script can be used to scale up (undilutes) or scale down (dilutes) a spectrum by using a different argument in the function as shown below. This script can be also used to normalize/denormalize a spectrum to carbon content or another external parameter. Below shows an application of the code using “Sample 1.csv”.

Slide 12: *UVVIS_Dilution('Sample 1.csv','dilute',1,11)*

After running this line in the Command Window, a new .csv file is created in the Current Folder directory – “Sample 1_dil_0.0909.csv”. The absorbance values in this file are scaled down 11 times, which would have resulted if the sample was 11-fold diluted (i.e., diluted with DF = 0.0909).

Slide 13: *UVVIS_Dilution('Sample 1_dil_0.0909.csv','undilute',1,11)*

Here, this function was applied on the file that was produced in the previous example (“Sample 1_dil_0.0909.csv”). By running the script using the *'undilute'* argument, the code scales up the data 11 times. Thus, this reverses the previous action and the data in the new output file (Sample 1_dil_0.0909_undil_0.0909) now matches the data in the original file (“Sample 1.csv”).

Slide 14: *UVVIS_Dilution('Sample 2.csv','undilute',23,1)*

Here, this script is used to normalize the second example sample to dissolved organic carbon content, which is 23 mg/L (or 23 ppm). Mathematically, all absorbance values in the “Sample 2.csv” are divided by 23 to produce C-normalized decadic[2] absorbance.

[2] Decadic absorbance = absorbance computed on a $\log_{10}$ basis

Slide 15: *UVVIS_Dilution('Sample 2_undil_23.csv','dilute',23,1)*

By applying the script using the *'dilute'* argument, the data in the "Sample 2_undil_23.csv" is de-normalized, thus the data in "Sample 2_undil_23_dil_23.csv" matches the original data in "Sample 2.csv".

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

- **UVVIS_Process**

This code is a processing script that takes raw instrumental output (absorbance values versus wavelength) and transforms the data into its final (processed) form. The code can be easily altered by the user to enable or disable any of these steps.

Slide 16: *UVVIS_Process('Sample 1.csv','Blank.csv',0.0909)*

By applying this script, the sample file ("Sample 1.csv") and blank file ("Blank.csv") are loaded and checked if they are within the same wavelength ranges (in the test data provided, the range is 190-1100 nm). If the absorbance data was not sampled every 1 nm (e.g., sampled at 220 nm, 222 nm, 224 nm, etc.), then the data is interpolated using the *interp1* function (to learn more about this function, type *help interp1* in the Command Window). The code then undilutes the data based on a dilution factor for the sample of 0.0909 (where the sample was diluted 11-fold), and it is assumed that the dilution factor for the blank data is 1. After that, the blank spectrum is subtracted from the sample spectrum. This is particularly useful in analyses where a procedural blank may have significantly contributing absorbance (Lu and Wu, 2001; Peacock et al., 2014).The obtained spectrum is then corrected for scattering and deviations in baseline, temperature, and refractive index effects by averaging the absorbance values from 701 to 800 nm (Green and Blough, 1994; Helms et al., 2008). It must be noted that this range is instrument-specific and may need optimization if organic matter absorbs at wavelengths longer than 700 (edit code in lines 119-129). Lastly, the spectrum is normalized to the cuvette pathlength. Here, the value is set as 1 cm, but this can be changed in line 43 of the script. The processed "final" UV-VIS spectrum is exported as a "_Final.csv" file. For the example shown here, the file "Sample 1_Final.csv" is produced (**Slide 16**).

Note: this script will only work using versions of MATLAB that are 2019 and newer (the *readmatrix* function does not work on 2018 and older software versions).

To open the code and edit any of the processing parameters, type *edit UVVIS_Process* in the Command Window.

Slide 17: *edit UVVIS_Process*

The Editor window will pop up (or open in a new window, depending on user settings) and the code will be opened. All codes of TEnvR are generally split in several logical sections, the first one being a description section (labeled with *%% Description*). The second section contains the License Notice (*%% License Notice*), the third section (*%% Configuration*), whenever applicable, contains parameters that can be altered by the user. Different sections are separated and denoted by two percent symbols (*%%*). Comments by the user are designated with a single percent symbol (*%*) before the text. Lines with no *%* or *%%* in front are commands that MATLAB will read as code and will not skip.

To change the dilution factor for the blank, edit the *DF_blank* variable on line 42. To change the cuvette pathlength, edit the *Pathlength* variable on line 43. To disable the scattering correction, change the *Correction* variable on line 44 from *'yes'* to *'no'*. There are three possible scenarios regarding the scattering correction:

→ If the upper wavelength limit is ≤ 700, the code will not perform the scattering correction regardless of the value of the *Correction* variable.

→ If the upper wavelength limit is ≥ 800, the code will perform the scattering correction using absorbance values from 701-800 (as long as the *Correction* variable is set to *'yes'*).

→ If the upper wavelength limit is between 701-800, the code will perform the scattering correction using absorbance values from 701 to the longest wavelength (as long as it is < 800 and as long as the *Correction* variable is set to *'yes'*). For example, if a sample was acquired at wavelengths 300-750, absorbance values at 701-750 will be used for correcting absorbance values at 300-700.

After making any changes to the code, do not forget to click *Save*. To disable the blank-correction and not use a blank sample, use *'NoBlank'* instead of the blank filename in the second argument of the **UVVIS_Process** function.

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

- **UVVIS_Metrics**

This code takes a UV-VIS spectrum and calculates a variety of metrics listed in the table below. It also converts the decadic spectrum (based on a $\log_{10}$ basis) into a Napierian spectrum ($\log_e$, or ln, basis).

Metric	Type	Variable label in MATLAB	Mathematical definition	Description	Reference
A_{230}	Aromaticity proxy	METRIC_Abs230Dec	Decadic absorbance at 230 nm	Metric is directly proportional to aromatic content	
A_{254}	Aromaticity proxy	METRIC_Abs254Dec	Decadic absorbance at 254 nm	Metric is directly proportional to aromatic content[3]	Summers et al. (1987); Weishaar et al. (2003)

[3] Use A_{254} and A_{280} to calculate $SUVA_{254}$ and $SUVA_{280}$ by dividing these metrics by externally measured DOC content (Weishaar et al., 2003) and multiplying by 100.

A_{280}	Aromaticity proxy	METRIC_Abs280Dec	Decadic absorbance at 280 nm	Metric is directly proportional to aromatic content[1]	Peuravuori and Pihlaja (1997)
CDOM[4]	Aromaticity proxy	METRIC_TotalCDOM	Integrated area from 250-450 nm	Metric is directly proportional to aromatic content	Helms et al. (2008)
α_{350}	Aromaticity proxy	METRIC_alpha350	Napierian absorbance at 350 nm	Metric is directly proportional to aromatic content	Hu et al. (2002); Stubbins et al. (2012)
α_{325}	Aromaticity proxy	METRIC_alpha325	Napierian absorbance at 325 nm	Metric is directly proportional to aromatic content	Hu et al. (2002); Stubbins et al. (2012)
α_{300}	Aromaticity proxy	METRIC_alpha300	Napierian absorbance at 300 nm	Metric is directly proportional to aromatic content	Hu et al. (2002); Stubbins et al. (2012)
α_{250}	Aromaticity proxy	METRIC_alpha250	Napierian absorbance at 250 nm	Metric is directly proportional to aromatic content	Hu et al. (2002); Stubbins et al. (2012)
α_{254}	Aromaticity proxy	METRIC_alpha254	Napierian absorbance at 254 nm	Metric is directly proportional to aromatic content	Hu et al. (2002); Stubbins et al. (2012)
$E_2: E_3$	Spectral ratio	Metric_E2E3	$\frac{A_{250}}{A_{356}}$	Metric is inversely proportional to molecular size	De Haan and De Boer (1987)
$E_4: E_6$	Spectral ratio	Metric_E4E6	$\frac{A_{465}}{A_{665}}$	Metric is inversely proportional to molecular size	Chen et al. (1977); Senesi et al. (1989)
$S_{275-295}$	Spectral slope	Metric_Slope275295	Slope of linearized spectrum in the 275-295 range	Metric related to CDOM quality (source, previous photochemical or microbial exposure, molecular weight)	Helms et al. (2008)
$S_{290-320}$	Spectral slope	Metric_Slope290320	Slope of linearized spectrum in the 290-320 range	Metric related to CDOM quality (source, previous photochemical or microbial exposure, molecular weight)	Helms et al. (2008)
$S_{350-400}$	Spectral slope	Metric_Slope350400	Slope of linearized spectrum in the 350-400 range	Metric related to CDOM quality (source, previous photochemical or microbial exposure, molecular weight)	Helms et al. (2008)

[4] CDOM = Chromophoric (chromophore-containing) dissolved organic matter.

$S_{350-450}$	Spectral slope	Metric_Slope350450	Slope of linearized spectrum in the 350-450 range	Metric related to CDOM quality (source, previous photochemical or microbial exposure, molecular weight)	Helms et al. (2008)
S_R	Spectral slope	Metric_SlopeRatio	$\frac{S_{275-295}}{S_{350-450}}$	Metric related to CDOM quality (source, previous photochemical or microbial exposure, molecular weight)	Helms et al. (2008)

This code must be applied on a UV-VIS spectrum in its final form. An example is shown with the previously processed Sample 1:

Slide 18: *UVVIS_Metrics('Sample 1_Final.csv')*

The code exports an Excel spreadsheet (.xlsx) with the decadic spectrum, the Napierian spectrum, and a table with the metrics (**Slide 19**). For the example shown above, the new file is “Sample 1_Final_Metrics.xlsx”.

Please note that in the produced Excel file, the E_2:E_3 and E_4:E_6 ratios are labeled as “E2toE3” and “E4toE6”, respectively (**Slide 19**). Using symbols such as / ? * [ ] is not recommended as they can cause errors in the application of statistical codes later in this tutorial.

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

- **UVVIS_Derivative**

This code takes a spectrum and calculates its derivative. An example is shown with the previously processed Sample 1:

Slide 20: *UVVIS_Derivative('Sample 1_Final.csv')*

The newly produced file (“Sample 1_Final_deriv.csv”) contains the first-order derivative spectrum. To calculate a second-order (or nth order) derivative spectrum, apply this code again on the .csv file containing the n^{-1}-order derivative spectrum, as shown below for second- and third-order derivative spectra.

Obtaining second-order derivative spectrum:

Slide 21: *UVVIS_Derivative('Sample 1_Final_deriv.csv')*

File produced: “Sample 1_Final_deriv_deriv.csv”

Obtaining third-order derivative spectrum:

Slide 22: *UVVIS_Derivative('Sample 1_Final_deriv_deriv.csv')*

File produced: “Sample 1_Final_deriv_deriv_deriv.csv”

This code can be modified to immediately produce an n^th order spectrum, and we urge users to do so if higher order spectra are desired.

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

- **UVVIS_Automation**

While the codes listed above can be individually applied on samples, this is inconvenient for datasets containing a large number of samples. Thus, these codes are incorporated into an automation routine. The **UVVIS_Automation** script is capable of loading all files that are in the current directory (thus, do not keep any other .csv files in that folder), and applying the scripts described above in a sequential order to all samples. In the end, this script prepares a Master Report Excel file with all data conveniently combined together. The code is run by typing the following in the Command Window: *UVVIS_Automation*

Please note: Before running this script, make sure that the only .csv files in the Current Folder are the unprocessed UV-VIS spectra of interest. For example, if you are following this tutorial, remove the .csv files created by the previously shown codes above (either delete or place elsewhere) and only leave the 20 csv files corresponding to the test samples in this folder. Files of other types (e.g., .dat) can be kept in this folder without interfering with the **UVVIS_Automation** code. To delete the files of interest, hold down the Ctrl button on your keyboard and select the files to delete and then press the Delete button on your keyboard (**Slide 23**).

Slide 24: *UVVIS_Automation*

Immediately, a pop-up dialog box shows up requesting the filename of the blank (**Slide 24**). Input the full filename of the blank (*Blank.csv*), including the extension, with no apostrophes. Please double-check the name of the blank! If no blank is to be used, use *NoBlank* with no apostrophes.

Blank.csv

Immediately after the blank name is keyed in, another dialog box will pop us asking if the samples have been diluted (**Slide 24**). If you click No, the code will use a dilution factor of 1 for all samples. If you click yes, a third dialog box will pop up (**Slide 24**) requesting dilution factors for all samples. The dialog box window is coded to display the filename of each sample allowing the user to cross-reference the filename from a spreadsheet or laboratory notebook and transfer the dilution factor to the dialog box. These dilution factors will also be provided in the Master Report file. For the 20 samples of the example dataset, please use the following dilution factors:

Filename	Dilution Factor
Sample 1.csv	*0.0909*
Sample 2.csv	*0.3333*
Sample 3.csv	*0.0909*
Sample 4.csv	*0.3333*
Sample 5.csv	*0.3333*
Sample 6.csv	*0.1667*
Sample 7.csv	*1.0000*
Sample 8.csv	*0.0476*
Sample 9.csv	*1.0000*
Sample 10.csv	*0.0909*
Sample 11.csv	*0.0909*
Sample 12.csv	*1.0000*
Sample 13.csv	*0.3333*
Sample 14.csv	*0.3333*
Sample 15.csv	*0.1667*
Sample 16.csv	*1.0000*
Sample 17.csv	*0.1667*
Sample 18.csv	*1.0000*
Sample 19.csv	*0.0909*
Sample 20.csv	*0.0909*

Once the dilution factors are keyed in, the code will sequentially apply **UVVIS_Process**, **UVVIS_Derivative**, and **UVVIS_Metrics** on all sample files in the current directory (**Slide 25**). The code will also collect all of the data and export it in a master file named "UVVIS_MasterReport.xlsx" which contains four sheets:

→ Sheet1 = All decadic spectra of the samples in a matrix format (**Slide 26**)

→ Normalized = Each decadic spectrum is normalized to total spectral intensity (**Slide 27**). The resultant normalized matrix is further treated to prepare it for statistical analysis: Negative absorbance values and NaN[5] values are set to zero. Wavelengths that have the same absorbance value for all samples are removed from the dataset to avoid creating covariance values with missing values if principal component analysis is employed. Additionally, wavelengths above 450 nm are removed as usually CDOM is insignificantly absorptive at such long wavelengths (Helms et al., 2008). Researchers are welcome to modify this cut-off threshold for their needs. This matrix can be then loaded into the codes for multivariate statistics as shown later.

[5] NaN = Not a Number. Represents values that are not real. Could be complex numbers, categorical values (e.g., text), etc.

→ Napierian = All Napierian spectra of the samples in a matrix format (**Slide 28**)
→ 1st Deriv = All first-order derivative spectra of the samples in a matrix format (**Slide 29**)
→ Metrics = All metrics and dilution factors of the samples in a matrix format (**Slide 30**)

The data from this report can be then easily visualized and evaluated for trends. The dilution factors that were keyed in are also listed in column B of the metrics sheet (**Slide 30**) – please double-check them here with your spreadsheet or laboratory notebook for possible mistypes. Additional measurements (such as dissolved organic carbon content), can be easily added to the spreadsheet in order to calculate other variables such as $SUVA_{254}$ and $SUVA_{280}$ (Weishaar et al., 2003). Please do not delete the file “UVVIS_MasterReport.xlsx” and do not change the sheet name “Normalized” - the matrix in the “Normalized” sheet will be directly used in several multivariate statistical routines (e.g., principal component analysis) as shown later in this tutorial.

Please note that in the produced Excel file, the E_2:E_3 and E_4:E_6 ratios are labeled as “E2toE3” and “E4toE6”, respectively (**Slide 30**). Using symbols such as / ? * [ ] is not recommended as they can cause errors in the application of statistical codes later in this tutorial.

The **UVVIS_Automation** code utilizes external functions for alphanumeric sorting of filenames (*natsortfiles* and *natsort*) developed by Stephen Cobeldick (https://www.mathworks.com/matlabcentral/fileexchange/47434-natural-order-filename-sort). These codes and their license are found in TEnvR\Supplementary files\Internal codes.

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

- **UVVIS_Differentiation**

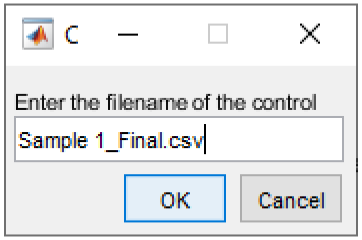
This script imports numerous UV-VIS spectra (in their processed form) and in an automated manner calculates the differential (also known as “difference”) spectra relative to one particular sample. Typically, this reference sample is a control in an experiment where one is looking at the changes in absorbance. For example, differential spectra can be useful in photochemical experiments where the absorbance of the control sample (e.g., “dark” control) is subtracted from all photo-irradiated samples (Hemmler et al., 2019). In the end, this script prepares a Master Report Excel file with all data conveniently combined together. The code is run by typing the following in the Command Window: *UVVIS_Differentiation*

Please note: The **UVVIS_Differentiation** will load all files that end with “_Final.csv” and thus, before running this script, make sure that the only “_Final.csv” files in the Current Folder are the processed UV-VIS spectra of interest. If you are following this tutorial to this point, you should have 20 “_Final.csv” spectra in your Current Folder. Files of other types (.dat, .xlsx, etc.) or other .csv files not containing the _Final suffix can be kept in this folder without interfering with the **UVVIS_Differentiation** code.

Slide 31: *UVVIS_Differentiation*

Immediately, a pop-up window will request the filename of the control sample (the sample that will be subtracted from all others). Input the full filename, including the extension, with no apostrophes, as shown here:

Sample 1_Final.csv



Enter the filename of the control

Sample 1_Final.csv

The code will then proceed to perform the calculations and will compile all resultant differential spectra in an Excel file named “UVVIS_Differentiation.xlsx”. The original spectra are compiled in “Sheet1” (**Slide 32**). The differential spectra are compiled in sheet “Differential” (**Slide 33**).

Note: this script will only work using versions of MATLAB that are 2019 and newer (the *readmatrix* function does not work on 2018 and older software versions).

The **UVVIS_Differentiation** code utilizes external functions for alphanumeric sorting of filenames (*natsortfiles* and *natsort*) developed by Stephen Cobeldick (https://www.mathworks.com/matlabcentral/fileexchange/47434-natural-order-filename-sort). These codes and their license are found in TEnvR\Supplementary files\Internal codes.

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

Section 5. Fluorescence Spectroscopy

In earlier years in the application of this technique to environmental research, spectrofluorometry was mainly used to acquire two-dimensional data (Coble et al., 2014). A common approach was to excite the sample at a single excitation wavelength (λ_{EX}), e.g., 370 nm, and then measure the emitted fluorescence over a range of emission wavelengths (λ_{EM}) to obtain a two-dimensional fluorescence spectrum (McKnight et al., 2001). Synchronous spectrofluorometry, where the excitation and emission monochromators were off-set with a certain gap (e.g., 25 nm), and the emission was detected, was also popular (Cabaniss and Shuman, 1987). Currently, it is preferred to acquire excitation-emission matrices (EEMs), which provide a representative three-dimensional spectrum of the FDOM content of samples. In earlier spectrofluorometric designs, in order to acquire an EEM spectrum both monochromators would have to sequentially scan all wavelengths, thus acquisition of an EEM could take 30-50 min/sample. After the implementation of faster detectors (photo-diode arrays or charge-coupled devices) capable of measuring the fluorescence intensity at all emission wavelengths within ~5 s for each excitation wavelength, the time to acquire an EEM decreased to ~5 min/sample, making EEM data acquisition much more efficient (Coble et al., 2014) and increasingly employed on a routine basis.

The codes in TEnvR for working with fluorescence EEM data were developed to supplement the decomposition routines for the Excitation Emission Matrices (drEEM) toolbox by Murphy et al. (2013). While most of the codes in TEnvR can be used independently of drEEM, they must be applied on processed “final” EEM spectra. There are several mandatory data correction procedures, as described by Murphy et al. (2010), and these are performed by several scripts of the drEEM toolbox. In TEnvR, two scripts are provided to serve as examples of how to process raw EEM data into “final” data (**EEM_Process_Generic** and **EEM_Process_Aqualog**). As these scripts utilize functions of the drEEM toolbox, installation of drEEM is necessary. To install drEEM, download the toolbox from its website (http://dreem.openfluor.org/). For the most recent version (drEEM-0.6.5), after downloading and extracting all files of the compressed (zipped) folder, follow the instructions below for installation (**Slide 34**) or watch an instructional video here: https://www.youtube.com/watch?v=9OyxSKEWmbE.

Briefly, set the drEEM folder as your Current Folder and use the following command:

Slide 34: *dreeminstall*

The installation routine executed by the *dreeminstall* command will internally route the codes of drEEM and set it up for use. Once installation is complete you may clear up the Command Windows using *clc*.

While the use of drEEM is beyond the scope of this manuscript and tutorial, it is recommended that users of TEnvR also explore drEEM and familiarize themselves with the tools for performing parallel factor analysis (PARAFAC). There are several useful tutorials for importing and correcting EEM data, as well as for PARAFAC modeling, that can be found here: http://dreem.openfluor.org/tutorials/. Additionally, there are tutorial scripts located in the tutorials folder of drEEM (folder named “tutorials” in versions 0.1.0 to 0.4.0, folder named tutorials_demos in the more recent versions 0.5.0 to 0.6.5) that can also be used for self-teaching purposes.

The drEEM toolbox is highly versatile and can import EEM files output from a variety of instruments (HORIBA Aqualog and Fluoromax, Hitachi, Varian, Shimadzu, etc.). In this tutorial, we have provided two different datasets of EEM spectra, one acquired on a HORIBA Aqualog and one acquired on a Shimadzu RF-6000 spectrofluorometer. Both datasets will be used as examples of how to use raw data (output from the instrument) to acquire processed data using the functions within drEEM. Then the obtained processed spectra will be subsequently used to show the application of the TEnvR codes for EEM spectra.

- **EEM_Process_Aqualog**

HORIBA Aqualog spectrofluorometers appear to be the most common instruments utilized in the aquatic fluorescence field. These instruments are not only very fast in acquiring EEMs (~5 min per spectrum) but are also capable of performing several of the data processing steps. Blank-subtraction can be enabled in the very beginning prior to acquiring the EEMs (in the "Experimental Setup" menu after selecting "EEM 3D + Absorbance" in the 3D menu). Once each EEM is acquired, the instrument software automatically performs the correction for removal of instrument-specific responses (Cory et al., 2010). Then, after acquisition of all samples, the user has the options to perform: 1) Inner filter effect correction using the absorbance method (Kothawala et al., 2013) by clicking the following button: ▣; 2) Remove 1st and 2nd order Rayleigh and Raman scattering using the Rayleigh Masking tool (button: ◩); and normalize the sample, typically to Raman Units (RU) (Lawaetz and Stedmon, 2009) or Quinine Sulfate Units (QSU) (Coble et al., 1993), using the normalization tool (button: ⛶). Note that the instrument software is also capable of applying these correction steps automatically to a whole dataset of samples (e.g., see "Automatic sample queueing" in the manual). Once these steps are applied, the exported spectra are nearly in their processed "final" form. These spectra only need to be rescaled to account for dilution and possibly trimmed to exclude wavelengths of higher noise. Typically, the Aqualog software exports the spectra in generic data files (.dat) and the drEEM toolbox is capable of reading those files and importing them in MATLAB. The instrument can also export data in .csv files that are also compatible with drEEM.

The **EEM_Process_Aqualog** script serves as an example of how to load multiple Aqualog-acquired EEM spectra (exported as .dat files) in MATLAB, rescale them to account for any dilution, evaluate and clean up the spectra from noise, and finally export the final EEMs as .csv files that can be used in the rest of the codes of TEnvR. First, set the folder with example data (TEnvR/TestData_EEM/TestData_Aqualog) as your Current Folder. Then, open the **EEM_Process_Aqualog** script either using the *Open* button in the "Home" tab or by executing the following command:

Slide 35: *edit EEM_Process_Aqualog*

This will open the **EEM_Process_Aqualog** script in the Editor window, which may open as a new, separate window or be docked within your current layout. Please note that three new tabs will show up once a script is open in the "Editor" window: Editor, Publish, and View. These have various capabilities for editing scripts and only two of their buttons will be used in this tutorial: *Save* and *Run Section* (noted in red in **Slide 35**). In difference with all previously shown scripts for UV-VIS codes, here this script is executed in several steps, also known as sections (separated by *%%*). Please note that MATLAB may suggest opening **EEM_Process_Aqualog** as a live script. As this feature is beyond the scope of this tutorial, we suggest closing this suggestion pop-up using the *X* button on the top right corner.

The first section of all codes within TEnvR are descriptive of the code and its capabilities (labeled *%% Description* inside the script). They also include any necessary citations. The second section (*%% Configuration*) is where any input by the user is required. Here, the user needs to specify the location of the data and the sample log, as well as where the processed data will be exported. Please note that there is an empty folder named TestData_Aqualog_Processed in the TestData_EEM folder – that is where EEM spectra will be exported once processed with the **EEM_Process_Aqualog** script. This script also requires a sample log formatted in a specific way (see Appendix A of Murphy et al. (2013) for more details), and we have provided one in the TestData_EEM folder ("SampleLog_Aqualog.csv"). Once the location of the raw data (line 34, variable: *Location_RawData*), the location of the sample log (line 35, variable: *Location_SampleLog*) and the location of the folder for exported data (line 40, variable: *Location_ProcessedData*) are correctly defined for your specific computer, click *Save* and then click *Run Section*. MATLAB will execute all lines from 24 to 40 and will skip lines starting with *%* or *%%*. Once *Run Section* is executed, the Workspace becomes populated with the three new variables that were just created (see **Slide 36**).

Then, click in the next stage (*%% Stage 1: Import EEM files, creates a cube "X" with all the data*). Modify the arguments in *readineems* function accordingly (line 44):

→ The first argument (*3*) is specific for the type of data format. This argument specifies how the data is organized, and the argument of *3* is specific for data from HORIBA Aqualog spectrofluorometers (excitation wavelengths in columns, emission wavelengths in rows, and excitation wavelengths are in descending order).

→ The second argument specifies the type of file. The data provided here are in generic data file format (.dat) corresponding to the argument of *'dat'*.

→ The third argument is the range of cells of the data. This is dependent on the excitation and emission ranges that were set up before sample analysis. To determine this range, open one of the provided files. Double-click on the file via the Current Folder in MATLAB, which will open the .dat in a separate MATLAB window called "Import". The range of data in the file is automatically shown in the menu bar (here, the range is A1:IC128). However, because of the way HORIBA Aqualog spectrofluorometers export data, the second and third rows do not contain EEM data, and the numeric EEM data begins from row 4 (**Slide 37**). Therefore, the EEM data of this dataset starts in cell A4 and ends in cell IC128, thus the third argument of the *readineems* function is *'A4..IC128'*.

→ The fourth argument (*[0 1]*) specifies if the first row or column of the imported data matrix contain the excitation or emission wavelengths. Here, the excitation wavelengths were omitted but the emission wavelengths are included – this situation corresponds to an argument of *[0 1]*.

→ The fifth argument is for specifying if the user desires the code to display each EEM while importing them. We prefer to not to visualize them at this stage (thus we use *0* equivalent to FALSE), but users may choose to see the EEMs by enabling this function by changing the argument to *1* (equivalent to TRUE).

→ The sixth argument is about export, which we also choose to disable at this stage (using the argument of *0*). We choose not to use these two capabilities of this function, because EEM data being imported at this stage often need extensive further correction (accounting for dilution, denoising, etc.). EEMs will be visualized and exported later in the code (see below).

For more information about any of these arguments or how this function works, please type *help readineems* in the Command Window.

Once these arguments are edited accordingly, click *Save* and click *Run Section*. Data files being imported will be listed in the Command Window. Once the code finishes importing data, the Workspace will populate with the new variables (see ***Slide 38***).

The next part of the code (*%% Stage 2: Align the imported data with the sample log*) is for extracting data from the sample log and aligning the already imported EEM spectra with those data. In this example, the sample log is used to import dilution factors and assign a dilution factor for each EEM spectrum. Typically, this section is run as-is, as long as there have been no format changes to the sample log file (see Appendix A of Murphy et al. (2013) for more details). Simply click on this section and then *Run Section* (**Slide 39**). The alignment of EEM spectra with the data from the sample log will be listed in the Command Window. Please note that the Current Folder directory will automatically change to the directory defined by the variable *Location_ProcessedData* (line 40, here: TestData_Aqualog_Processed) and this directory will be populated with a file named "EEM_DataCorrectionLog.txt". This is a "diary" file that will record anything output or typed in the Command Window from here on and serves as a digital record for the following data processing.

The next stage of the code (*%% Stage 3: Correct the EEMs*) is for any corrections of the data. Because most of them were already performed by the HORIBA Aqualog's software, the EEMs only need to be rescaled to account for dilution prior analysis. In the second EEM data processing code of TEnvR (**EEM_Process_Generic** described below), this section of the code is much more populated. Run this section after selecting it and clicking *Run Section* (**Slide 40**).

At this point, the EEM spectra are generally representative of the analyzed samples – all mandatory corrections (see Murphy et al. (2010)) were performed and spectral intensities were rescaled to take into account dilution factors. The next stage of the code is for evaluating the EEMs and determining if they need to be further refined. Note that this section will be executed line-by-line instead of running the whole section at-once (i.e., the *Run Section* button will not be used).

First, EEMs need to be visually inspected. Copy the code from line 85 and paste it in the Command Window. If no figure pops up, type 1 and enter to manually request the code to show sample 1.

Slide 41: *eemview(DS_undiluted,'X',[],[],[],[],[],'rotate','colorbar',[],[])*

Please note that we prefer to view the EEMs with the emission wavelengths as x-axis. To view the EEMs with emission wavelengths as y-axis, simply replace *'rotate'* with *[]* in the line of code (i.e., *eemview(DS_undiluted,'X',[],[],[],[],[],[],'colorbar',[],[])*)

A figure will pop up with the first EEM spectrum (EEM names are not displayed, rather their indices are shown as defined by column 1 in the sample log). Use the Enter button on your keyboard to go through and visualize all EEMs and note any issues with them. On **Slide 41**, two issues were identified and labeled on the first EEM. After clicking Enter and going through the whole dataset, these issues were found to be in all other spectra as well. Close the figure window once all spectra were viewed. As the Rayleigh Masking function of the HORIBA Aqualog spectrofluorometer is only a crude approach to remove 1st and 2nd order Rayleigh and Raman scattering, residual scatter signals were apparent in the EEMs. Thus, the data needs to be further corrected to fully eliminate them. Copy the code of line 91 and paste it in the Command Window (**Slide 42**).

Slide 42: *DS_Smooth=smootheem(DS_undiluted,[10 10],[10 10],[20 20],[10 10],[0 0 0 0],[],3382,'pause');*

Use the Enter button on your keyboard to go through all EEMs and identify if all residual scattering is sufficiently removed. We suggest maximizing the figure to fit your whole screen. If scattering is not sufficiently removed, you may need to modify the parameters in the arguments of the *smootheem* function (type *help smootheem* in the Command Window for more details).

The second issue that was noted earlier (**Slide 41**) was that these EEMs were acquired over very large excitation and emission ranges and there appear to be primarily noise signals in these long-wavelength regions. The EEMs can be trimmed to exclude these regions by using the *subdataset* function (for more information about this function, type *help subdataset* in the Command Window). To trim the EEMs, copy the code from lines 94-95 and run it into the Command Window.

Slide 43: *DS_Denoised=subdataset(DS_Smooth,[],DS_Smooth.Em>650,DS_Smooth.Ex>600);*
DS_Denoised=subdataset(DS_Denoised,[],DS_Denoised.Em<300,DS_Denoised.Ex>500);

After these two fine refinements are done, view the EEMs again, using line 98 of the code, to determine if the previously observed issues were resolved. If no figure pops up, type 1 and enter to manually request the code to show sample 1. It is common that you will have to go back-and-forth and change the function arguments of the *smootheem* and *subdataset* functions to fully refine the spectra. The provided arguments in the functions above were optimized after running these refinement functions several times with different parameters (see Appendix A of Murphy et al. (2013) and online tutorials at http://dreem.openfluor.org/tutorials/ for more details).

Slide 44: *eemview(DS_Denoised,'X',[],[],[],[],[],'rotate','colorbar',[],[])*

Again, to view the EEMs with emission wavelengths as y-axis, simply replace *‘rotate’* in this line with *[]*. This evaluation showed that all noisy regions have been successfully removed, and the Rayleigh and Raman scattering removal was effectively refined. Please note that here the scattering regions were substituted with zeros while the Rayleigh Masking function of the instrument only interpolates the data. Interpolation can also be performed by the *smootheem* function (type *help smootheem* in the Command Window for more details). During this final evaluation, however, it was identified that the EEM with index 11 has an instrument error (a very sharp signal of unusually high intensity for its region, see **Slide 45**). Such instrument errors can be removed using the function *zap* (type *help zap* in the Command Window for more details).

Slide 46: *DS_Denoised=zap(DS_Denoised,11,[535 555],[238 242]);*

The arguments of the *zap* function require the user to identify the index of the EEM that needs to be corrected, as well as the location of the region that needs to be removed (excitation range = *[238 242]*; emission range = *[535 555]*). After applying this correction (line 101), view the EEMs one last time, using line 104 of the code, to confirm that they are in their processed (“Final”) form and do not need any further refinement. If no figure pops up, type 1 and enter to manually request the code to show sample 1.

Slide 47: *eemview(DS_Denoised,'X',[],[],[],[],[],'rotate','colorbar',[],[])*

Click through all EEMs and verify that they do not need further refinement. Note that the instrument error on EEM 11 is now removed. If no other refinements are necessary, you may proceed to the last section (*%% Stage 5: Export processed data as csv files*). Run this section selecting it and clicking *Run Section* (**Slide 48**).

This section will export all EEMs to the folder with their original file names and add a suffix "_Final" to indicate that these data are processed and can be further used for plotting, extraction of metrics (e.g., humification index), etc. The EEMs are exported in the folder that was defined in the beginning of this script (variable: *Location_ProcessedData*, see **Slide 36**). The code will also export a MATLAB file (.mat) that contains all variables from the workspace ("EEM_DataProcessing_Aqualog.mat"). These two export files are utilized for keeping a paper trail of all performed data processing steps. We thank Dr. Kathleen Murphy for providing this export code on the online support page for drEEM on ResearchGate.net (https://www.researchgate.net/project/drEEM-user-support).

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

Processing EEM data from non-Aqualog Instruments

Most research-grade spectrofluorometers are not capable of doing the correction steps that can be performed by HORIBA Aqualog spectrofluorometers, thus the EEM corrections must be performed post-acquisition in MATLAB. Here, a dataset of EEMs acquired on a Shimadzu RF-6000 spectrofluorometer is provided. It is located in TEnvR/TestData_EEM/TestData_Generic. There are five folders as described below:

→ BlankEEMs – EEM spectra of the blanks (as .txt files).

→ CorrectionFiles – files for correcting the EEMs for instrument-specific responses (Cory et al., 2010). These files are usually generated by the engineer installing the instrument. Some instruments (e.g., HORIBA Aqualog, Shimadzu RF-6000, Thermo Lumina) are capable of correcting the EEM spectrum immediately after acquisition. However, for EEMs acquired on other instruments, the user must obtain (or develop) these correction files and perform the correction in MATLAB. In this folder, a comma-separated value (.csv) file must be provided for correction of both the excitation and emission dimensions (files are named "ExcitationCorr.csv" and "EmissionCorr.csv", respectively). The instrument that was used to acquire this dataset, a Shimadzu RF-6000, is in fact able to correct the EEMs on its own. However, the correction function *fdomcorr* from the drEEM toolbox has a mandatory requirement of correction files to be imported. Thus, the two files that are provided in the CorrectionFiles folder are "ghost" files – they contain a correction factor of 1 for all wavelengths. In the Supplementary Files folder of TEnvR, we provide three .pdf files for developing correction factors on the following spectrofluorometers: Thermo Scientific Lumina, PerkinElmer, and Agilent Cary Eclipse. We thank Dr. Kathleen Murphy for providing two of these files on the online support page for drEEM on ResearchGate.net (https://www.researchgate.net/project/drEEM-user-support).

→ RawEEMs – EEM spectra of the samples (as .txt files)

→ UVVIS_IFE – UV-VIS absorbance spectra of the samples. Please note that both the EEM and UV-VIS spectra must be with the same concentration of organic matter (i.e., if the sample was diluted prior measuring the EEM, a UV-VIS spectrum of the same diluted sample must be acquired and provided in this folder).

→ WaterRaman350 – folder containing 2D emission spectra of water acquired at an excitation wavelength of 350 nm (.txt files).

Before processing these data, some of them must first be reformatted as shown below.

- **EEM_WaterRamanAverage**

EEM spectra are generally normalized to Raman Units (RU) (Lawaetz and Stedmon, 2009) or Quinine Sulfate Units (QSU) (Coble et al., 1993). RU normalization is simpler and thus appears to be more common. To normalize the EEM spectra to RU, we acquire five sequential water Raman scans at λ_{EX} = 350 nm and collect emission from 365.0 to 450.0 nm in 0.5 nm intervals. The **EEM_WaterRamanAverage** code simply takes these five replicate spectra (or any five 2D fluorescence spectra) and averages them out into a single 2D spectrum (wavelengths in column 1, emission intensity in column 2). The code loads text files that were output by the instrument (.txt) and exports the averaged spectrum in a .csv file. Twenty example spectra acquired on a Shimadzu RF-6000 spectrofluorometer are provided in the following folder: TEnvR\TestData_EEM\TestData_Generic\WaterRaman350. These spectra are designated by the time they were acquired (e.g., 3 PM, 10 PM). These averaged spectra will be used later for normalizing EEM data, but first each group of five replicates must be turned into one single averaged spectrum.

First, set the WaterRaman350 folder as your Current Folder. Then, double-click on one of the files from the Current Folder window to determine the range of the actual data. The text file will open in the Editor window of MATLAB. Scroll down to determine the data range – in the provided example spectra, data begin on row 38 and end on row 208 (**Slide 49**). Because the data are in two columns, this corresponds to the range of A38 to B208 (i.e., argument of *'A38:B208'* for the import function). Once the range is determined, run the code as shown below:

Slide 49: *EEM_WaterRamanAverage('WaterRaman_1PM_1.txt','A38:B208')*

It is critical that your five replicates are labeled with a numeric suffix at the end (i.e., _1, _2, _3, _4, _5). As only the filename of the first (_1) replicate is input as a function argument, the code derives the filenames of the other four replicates. The second argument is the data range (i.e., *'A38..B208'*). The code will automatically delete the numeric notation and substitute it with "AVG" for the newly created file (i.e., new filename is "WaterRaman_1PM_AVG.csv"). Make sure that the newly produced filenames match with those in the sample log. The code also prints text in the Command Window summarizing what has been done; double-check the names of the input and output spectra.

Apply this code to all water Raman scans to obtain four averaged spectra – these will be for data normalization later.

Slide 49: *EEM_WaterRamanAverage('WaterRaman_3PM_1.txt','A38:B208')*
EEM_WaterRamanAverage('WaterRaman_5PM_1.txt','A38:B208')
EEM_WaterRamanAverage('WaterRaman_10PM_1.txt','A38:B208')

- **EEM_ShimadzuReformat**

EEM data acquired on Shimadzu RF-6000 spectrofluorometers is organized in a way that does not make it easily importable using the codes from drEEM. The **EEM_ShimadzuReformat** code takes numerous spectra exported by the instrument (_.txt format) and converts them into a readable format (**Slide 50**). The newly output spectra are denoted with the suffix _ref (in .csv format). We have provided 32 example files located in TEnvR/TestData_EEM/TestData_Generic/RawEEMs. To use this code, make the RawEEMs folder to be your Current Folder and run *EEM_ShimadzuReformat* in the command window:

Slide 51: *EEM_ShimadzuReformat*

Make sure you reformat any blank files as well. Make the BlankEEMs folder to be your Current Folder and run the code there as well:

Slide 52: *EEM_ShimadzuReformat*

The Shimadzu RF-6000 spectrofluorometer has an automatic capability to correct the acquired data for instrument-specific responses (Cory et al., 2010), but all other corrections need to be made further post-acquisition. These reformatted EEM files need to be fully corrected following the recommendations by Murphy et al. (2010) and be processed using the **EEM_Process_Generic** script as shown below.

Note: the **EEM_ShimadzuReformat** script will only work using versions of MATLAB that are 2019 and newer (the *readmatrix* function does not work on 2018 and older software versions).

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

- **EEM_Process_Generic**

This script parallels the previous script for processing EEMs acquired on HORIBA Aqualog spectrofluorometers (**EEM_Process_Aqualog**) but includes a higher degree of processing steps.

To begin processing this dataset in the TestData_Generic folder, open the **EEM_Process_Generic** script either using the *Open* button in the “Home” tab or by executing the following command: *edit EEM_Process_Generic*. Set the folder TestData_Generic as your Current Folder. Edit the first section (*%% Configuration*) to specify the location of the data and the sample log, as well as where the processed data will be exported. Please note that there is an empty folder named TestData_Generic_Processed in the TestData_EEM folder – that is where EEM spectra will be exported once processed with the **EEM_Process_Generic** script. This script also requires a sample log formatted in a specific way (see Appendix A of Murphy et al. (2013) for more details), and we have provided one in the TestData_EEM folder (“SampleLog_Generic.csv”). Once the location of the raw data (line 30, variable: *Location_RawData*), the location of the sample log (line 31, variable: *Location_SampleLog*), and the location of the folder for exported data (line 34, variable: *Location_ProcessedData*) are defined for your specific computer, click *Save* and then click *Run Section* (see **Slide 53**). Then, proceed to the next section of the code.

In this section (*%% Stage 1: Import EEM data and other auxiliary files, creates a cube "X" with all the EEM data*), all data that is provided in the five folders is imported. The *readineems* function and its arguments were described in detail earlier for the **EEM_Process_Aqualog** script. Because of the different initial format of the “raw” (sample) and blank EEMs, the *readineems* function is capable of importing both the excitation and emission wavelengths (the fourth argument now is *[1 1]* in difference with the earlier example of this function). The *readinscans* function is analogous but is designed for two-dimensional spectra (type *help readinscans* in the Command Window for more details). Make sure to double-check all data ranges. Also, the excitation wavelength to produce the water Raman scan (usually 350 nm) is also listed in line 57 (variable = *RamEx*). After any edits are made to this section, click *Save* and then *Run Section* (see **Slide 54**). Please note that the .txt files do not need to be removed from the folder containing fluorescence data (EEMs or Water Raman scans), as the import functions (*readineems* and *readinscans*) will only be importing the files that are in comma-separated value (.csv) format. After executing this code, the WaterRaman350 folder automatically becomes the Current Folder, as shown in **Slide 54**.

No necessary edits are typically needed in the next section (*%% Stage 2: Align the imported data with the sample log*), as long as there have been no format changes to the sample log file (see Appendix A of Murphy et al. (2013) for more details). Simply click on this section and then *Run Section* (**Slide 55**). The alignment of EEM spectra with the data from the sample log will be listed in the Command Window. Please note that the Current Folder directory will automatically change to the directory defined by the variable *Location_ProcessedData* (line 34, here: TestData_Generic_Processed), and this directory will be populated with a file named “EEM_DataCorrectionLog.txt”. This is a “diary” file that will record anything output or typed in the Command Window from here on and serves as a digital record for the following data processing.

The next section (*%% Stage 3: Correct the EEMs*) is very different from the previous script. Click on it and press *Run Section* (**Slide 56**). After organizing some of the imported data, water Raman spectra are integrated with the *ramanintegrationrange* function. Following integration, a figure (**Slide 56**) pops up that describes the integration, lists the integration boundaries, and shows the integrated area (shaded in black with orange lines). The different panels and figure elements are described in detail by Murphy (2011). This figure is created for each integrated water Raman spectrum (here, a total of 4) and these figures are saved in a PostScript (.ps) file (**Slide 56**). PostScript files are easily converted into .pdf files using Acrobat by double-clicking on the .ps file.

Then, the *fdomcorrect* function is used to perform corrections in the following order:

→ Spectral correction for instrument-specific response (Cory et al., 2010) using the “ExcitationCorr.csv” and “EmissionCorr.csv” files;

→ Apply inner-filter effect correction using the absorbance data in the UVVIS-IFE folder following Kothawala et al. (2013). Please note that the code assumes 1-cm cuvette pathlength, and you must modify the *fdomcorrect* script for differently sized cuvettes;
→ Normalization of the spectra using the water Raman scans (Lawaetz and Stedmon, 2009);
→ Subtracts blanks from their corresponding sample EEMs.

The function outputs information about the processing in the Command Window as shown on **Slides 57** & **58**. This output is being recorded into the diary file "EEM_DataCorrectionLog.txt".

The obtained data is then rescaled to account for the dilution factors. At this point, the EEM spectra are generally representative of the analyzed samples – all mandatory corrections (see Murphy et al. (2010)) were performed and spectral intensities were rescaled to take into account dilution factors. In the next section (*%% Stage 4: EEM evaluation and fine correction*) the EEMs will be evaluated and further refined. Note that this section will be executed line-by-line instead of running the whole section at-once (the *Run Section* button will not be used).

First, the EEMs need to be visually inspected. Copy the code from line 120 and paste it in the Command Window. If no figure pops up, type 1 and enter to manually request the code to show sample 1.

Slide 59: *eemview(DS_undiluted,'X',[],[],[],[],[],'rotate','colorbar',[],[])*

For consistency throughout this tutorial, EEMs will be displayed with the emission wavelengths as x-axis. To view the EEMs with emission wavelengths as y-axis, simply replace *'rotate'* with *[]* in the code (i.e., *eemview(DS_undiluted,'X',[],[],[],[],[],[],'colorbar',[],[])*)

A figure will pop up with the first EEM spectrum (EEM names are not displayed, rather their indices are shown as defined from column 1 in the sample log). Use the Enter button on your keyboard to go through the EEMs and note any issues with them. Significant Rayleigh and Raman scattering signals were present in all EEM spectra. It must be noted that because the previous example with the **EEM_Process_Aqualog** script, HORIBA Aqualog's software (using its "Rayleigh Masking" capability) had crudely removed the scattering signals and only residual intensities remained (see **Slide 41**). As no such correction had been made for the "Generic" dataset, the scattering signals were so intense that all fluorophoric signatures were significantly hindered. Close the figure window once all spectra were viewed. To perform scattering removal, copy the code of line 123 and paste it in the Command Window.

Slide 60: *DS_Smooth=smootheem(DS_undiluted,[15 300],[15 15],[350 20],[10 10],[0 0 0 0],[],3382,'pause');*

Use the Enter button on your keyboard to go through all EEMs and identify if all scattering is effectively removed. For this dataset, we also removed signals from wavelengths where no significant signals were observed (at very long excitation and short emission wavelengths or vice versa). As previously mentioned, you may need to run this function several times and modify the arguments to find the optimal arguments for effective scattering removal (type *help smootheem* in the Command Window for more details). During the scattering removal, it was noted that EEMs can be further trimmed from regions of noise. Thus, *subdataset* was used again. Copy the code from lines 126 and 127 and run it into the Command Window:

Slide 61: *DS_Denoised=subdataset(DS_Smooth,[],DS_Smooth.Em>600,DS_Smooth.Ex>450); DS_Denoised=subdataset(DS_Denoised,[],DS_Denoised.Em<270,[]);*

After these fine refinements are done, view the EEMs again using line 130 of the code to verify that the previously observed issues were resolved. If no figure pops up, type 1 and enter to manually request the code to show sample 1. It is common that you will have to go back-and-forth and change the function arguments of the *smootheem* and *subdataset* functions to fully refine the dataset.

Slide 62: *eemview(DS_Denoised,'X',[],[],[],[],[],'rotate','colorbar',[],[])*

This evaluation showed that all noisy regions and Rayleigh and Raman scattering signals were completely removed. Proceeding to the last section (*%% Stage 5: Export processed data as csv files*), run this section by selecting it and clicking *Run Section* (**Slide 63**).

This section will export all EEMs to the folder with their original file names and add a suffix "_Final" to indicate that these data are final and can be further used for plotting, extraction of metrics (e.g., humification index), etc. The EEMs are exported in the folder that was defined in the beginning of this script (variable: *Location_ProcessedData*, see **Slide 53**). The code will also export a MATLAB file (.mat) that contains all variables from the workspace. Lastly, the code will export a "diary" file ("EEM_DataCorrectionLog.txt"), which contains everything that was input by the user or output by the code in the Command Window("EEM_DataProcessing_Generic.mat"). These two export files are for keeping a paper trail of all performed data processing steps. We thank Dr. Kathleen Murphy for providing the export code on the online support page for drEEM on ResearchGate.net (https://www.researchgate.net/project/drEEM-user-support). The Xout variable in the workspace (created on line 133) can be further taken and used for PARAFAC modeling as explained in the drEEM tutorials.

Please note that the two processing scripts described above (**EEM_Process_Aqualog** and **EEM_Process_Generic**) are only examples that complement the drEEM tutorials and serve as examples of how processed data is obtained prior to using the codes of TEnvR. There are many more codes and diverse capabilities within drEEM, and in no way does the text within this tutorial aim to provide a comprehensive overview of the drEEM toolbox. Please refer to the drEEM support page (http://dreem.openfluor.org/) for more information, questions, or tutorials of drEEM.

- **EEM_Visualize**

Once the processed EEMs are exported by the last section of the two processing codes above, they can be visualized using **EEM_Visualize**. Please note that drEEM does contain a visualization function (*eemview*); however, it is aimed to be used on a dataset of multiple EEMs stacked together (or in MATLAB-language: a structure). **EEM_Visualize** is a simpler function that is to be used on 1-3 EEMs exported as .csv files. This code is great for producing publication-grade figures and can be easily modified to customize any of the figure elements (font sizes, color of the peak labels, etc.).

As a reminder, before using this code, sequentially (in different lines) type *clc*, *close all*, and *clear* in the Command Window to clear up the Workspace, the Command Window, and close any previously opened Figure windows. If you have been following this tutorial so far, your Current Folder should be TestData_Generic_Processed. These processed spectra will be used for the next examples.

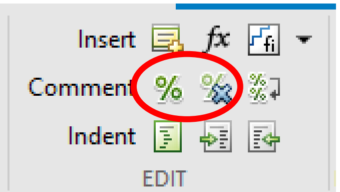
Slide 64: *EEM_Visualize('Sample 1_ref_Final.csv')*

The **EEM_Visualize** code will produce a contour plot figure (**Slide 64**). It contains standard elements (color bar, axis labels, etc.) that can be easily modified by the user. The code can be used further to visualize multiple EEM spectra (**Slides 65** and **66**):

Slide 65: *EEM_Visualize('Sample 1_ref_Final.csv','Sample 2_ref_Final.csv')*
Slide 66: *EEM_Visualize('Sample 1_ref_Final.csv','Sample 2_ref_Final.csv','Sample 3_ref_Final.csv')*

This code has four capabilities that can be easily "switched on" and "switched off" once the code is open using *edit EEM_Visualize*:

→ Peak labeling – the common peaks of NOM fluorescence (peaks A, C, B, T, etc.) are displayed on the visualized plots by default and are useful for visual referencing on changes on the plots. Peak labels can be disabled by changing the value for the *Peak_Labels* variable on line 42 from *true* to *false* (**Slide 67**). More elaborate peak labeling can be achieved by going to lines 308-338 and 388-418 of the script and enabling (by deleting the *%* symbol) or disabling (by adding a *%* symbol) the peak labels of interest (**Slide 68**). For convenience, multiple lines can be selected and the % button in the "Edit" section of the Editor tab (see to the right) can be used to delete the *%* symbol from numerous lines (thus "enabling" them). Conversely, the % can be used to add *%* symbols in front of multiple lines to "disable" them. The positions of the peak labels are obtained from Coble et al. (2014). After making any changes to the code, click *Save* before running it.



→ Switching axes to plot the excitation dimension either on the x-axis or the y-axis. To plot the excitation on the x-axis, use *true* as a variable for the *Excitation_Xaxis* variable in line 45. To plot the excitation on the y-axis, use *false* (**Slide 69**). After making any changes to the code, click *Save* before running it.

→ In cases when two or three EEMs are displayed, the default version of this code creates a color bar for each individual plot that is specific for the EEM's intensity range (e.g., color bar for the first EEM corresponds to 0-1600 RU and the color bar of the second EEM corresponds to 0-3500 RU). It is often beneficial to "normalize" the color bars and use the same intensity range for the displayed EEMs. Use *true* as a variable to the *Same_Intensity_Scale* variable on line 48 (**Slide 70**). After making any changes to the code, click *Save* before running it. The figures shown in **Slide 65** and **Slide 66** have line 48 set as *false*.

→ Normalization – the code can normalize each EEM spectrum to the sum of its intensity, allowing for qualitative comparisons that would usually be hindered due to severe differences in concentration. EEMs are displayed with their own color bar intensity ranges. To enable the normalization, use *true* as a variable to the *Normalization* variable on line 51 (**Slide 71**). After making any changes to the code, click *Save* before running it. If the user desires to have the same color bar intensity range for the three EEMs after normalization, *true* can be set as a variable for both the *Normalization* and *Same_Intensity_Scale* variables (**Slide 72**).

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

- **EEM_Difference**

Another useful approach to evaluate changes in EEM spectra is to use differential or difference spectra (analogous to what was explained earlier for UV-VIS spectra). They are very practical for cases when a sample has gone through a treatment (e.g., photochemical degradation), and the user is interested to see which fluorophores were formed and which were degraded during this treatment (e.g., (Hemmler et al., 2019). The user must define which two samples are compared, with the first sample being subtracted from the second as explained below:

Slide 73: *EEM_Difference('Sample 1_ref_Final.csv','Sample 2_ref_Final.csv')*

The produced figure is a three-panel plot. The spectrum on the left (considered a “control”) is subtracted from the EEM in the middle (considered an “experimental” sample that has changed somehow) to produce the difference spectrum on the right. A lot of the code behind this script is identical to the code from the script above (**EEM_Visualize**). This code has some of the customization capabilities discussed earlier for **EEM_Visualize**. They are briefly explained below (for more details, see the tutorial for **EEM_Visualize** above).

→ Peak labeling – common DOM fluorophoric peaks (peaks A, C, B, T, etc.) can be displayed on the visualized plots by using *true* as a value for the *Peak_Labels* variable on line 41. More elaborate peak labeling can be achieved by going to lines 198-228 and 270-300 of the script and enabling or disabling (by adding or removing *%* symbols) the peak labels of interest. The positions of the peak labels are obtained from Coble et al. (2014). After making any changes to the code, click *Save* before running it.

→ Switching axes to plot the excitation dimension either on the x-axis or the y-axis. To plot the excitation on the x-axis, use *true* as a variable for the *Excitation_Xaxis* variable in line 44. To plot the excitation on the y-axis, use *false*. After making any changes to the code, click *Save* before running it.

→ Normalization – the code can first normalize each EEM spectrum to the sum of its intensity prior subtraction. This approach can be used for qualitative presence/absence evaluation of fluorophoric signals that would usually be hindered due to differences in concentration. EEMs are displayed with their own color bar intensity ranges. To enable the normalization, use *true* as a variable to the *Normalization* variable on line 47. After making any changes to the code, click *Save* before running it. The figure shown in **Slide 73** has the *Normalization* variable on line 47 set as *true*.

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

- **EEM_Metrics**

Fluorescence EEM spectra are commonly used to extract various metrics to describe the fluorophoric (or fluorescent) dissolved organic matter (FDOM) of environmental samples. The code presented here has a high versatility and can process both individual files and a whole dataset automatically. It can calculate a total of 148 different metrics, and they are broadly classified into 3 categories: 1) General spectral descriptors; 2) Indices; and 3) Fluorophoric peaks. The full list of the 148 metrics is elaborately described in the Excel spreadsheet ("EEM_Metrics.xlsx") provided in the main Supplementary Files folder of TEnvR. The three different categories are described below:

→ General spectral descriptors: these are parameters that are derived from the whole EEM spectrum and are bulk descriptors of the composition of the evaluated fluorophoric (or fluorescent) dissolved organic matter (FDOM) mixture. These metrics are: the sum of all signals in the EEM spectrum (Total Spectral Intensity, F_{TOTAL}), the average spectral intensity calculated by dividing F_{TOTAL} by the number of data points (Average Spectral Intensity, F_{AVG}), the highest fluorescence emission value (Maximum Spectral Intensity, F_{MAX}), and the position of the most intense peak (defined by both excitation, Most Intense Peak (Ex), and emission, Most Intense Peak (Em), positions). These parameters can be very useful for doing quantitative evaluations, as well as tracking the bulk FDOM composition using the Maximum Peak Position (EX_{MAX} & EM_{MAX}) (Coble et al., 2014). Additionally, once EX_{MAX} & EM_{MAX} are calculated, the intensity in the surrounding 5 nm x 5 nm region is also summed and output (i.e., if EX_{MAX} = 300 & EM_{MAX} = 400, the intensity encompassed by excitation of 295-305 nm and emission of 395-305 nm is summed and provided as a metric). Because EEM data is often not measured in 1 nm steps (e.g., excitation is acquired every 5 nm), interpolation of the data is necessary as many of the metrics require specific wavelength positions of 1 nm spacing. The code will automatically determine if data interpolation is necessary and will perform it automatically. The code will also compute the general spectral descriptors described above for both the original and interpolated data, and they can be compared for quality control of the interpolation procedure.

→ Index proxies from spectroscopic data (both UV-VIS absorbance and fluorescence) have been used to characterize natural organic matter for decades. The **EEM_Metrics** code has been developed to export the most common indices, as summarized by Coble et al. (2014), which are also described in the table below and in the provided spreadsheet ("EEM_Metrics.xlsx"). This list is in no way comprehensive, and we urge users of TEnvR to further modify the **EEM_Metrics** script to add, remove, and or modify existing metrics.

Index	Variable label in MATLAB	Description	Mathematical Definition*	Reference
Humification Index ($\mathrm{HIX_{SYN}}$) version a	METRIC11a_HIXsyn	Indicative of C/N ratio, aromatic content, and degree of condensation of fulvic acids.	$\frac{\mathrm{I}_{\mathrm{EM=488}}^{\mathrm{EX=470}}}{\mathrm{I}_{\mathrm{EM=378}}^{\mathrm{EX=360}}}$	Kalbitz et al. (1999)
Humification Index ($\mathrm{HIX_{SYN}}$) version b	METRIC11b_HIXsyn	Indicative of C/N ratio, aromatic content, and degree of condensation of fulvic acids.	$\frac{\mathrm{I}_{\mathrm{EM=418}}^{\mathrm{EX=400}}}{\mathrm{I}_{\mathrm{EM=378}}^{\mathrm{EX=360}}}$	Kalbitz et al. (1999)
Humification Index ($\mathrm{HIX_{SYN}}$) version c	METRIC11c_HIXsyn	Indicative of C/N ratio, aromatic content, and degree of condensation of groundwater.	$\frac{\mathrm{I}_{\mathrm{EM=408}}^{\mathrm{EX=390}}}{\mathrm{I}_{\mathrm{EM=373}}^{\mathrm{EX=355}}}$	Kalbitz et al. (2000)

Humification Index ($\mathrm{HIX_{EM}}$) version a	METRIC12a_HIXem	Indicative of degree of humification	$\frac{\mathrm{A}^{\mathrm{EX}=254}_{\mathrm{EM}=435-480}}{\mathrm{A}^{\mathrm{EX}=254}_{\mathrm{EM}=300-345}}$	Zsolnay et al. (1999)
Humification Index ($\mathrm{HIX_{EM}}$) version b	METRIC12b_HIXem	Indicative of degree of humification	$\frac{\mathrm{A}^{\mathrm{EX}=254}_{\mathrm{EM}=435-480}}{\mathrm{A}^{\mathrm{EX}=254}_{\mathrm{EM}=300-345}+\mathrm{A}^{\mathrm{EX}=254}_{\mathrm{EM}=435-480}}$	Ohno (2002)
Freshness Index (BIX) old version	METRIC15_BIXold	Indicative of ratio between recently produced vs. old organic matter.	$\frac{\mathrm{MAX}^{\mathrm{EX}=310-320}_{\mathrm{EM}=380-420}}{\mathrm{MAX}^{\mathrm{EX}=330-350}_{\mathrm{EM}=420-480}}$	Parlanti et al. (2000)
Freshness Index (BIX) new version	METRIC16_BIXnew	Indicative of ratio between recently produced vs. old organic matter.	$\frac{\mathrm{I}^{\mathrm{EX}=310}_{\mathrm{EM}=380}}{\mathrm{MAX}^{\mathrm{EX}=310}_{\mathrm{EM}=420-435}}$	Wilson and Xenopoulos (2008)
Fluorescence Index (FI) old version	METRIC17_FIold	Differentiate between terrestrial and marine composition.	$\frac{\mathrm{I}^{\mathrm{EX}=370}_{\mathrm{EM}=450}}{\mathrm{I}^{\mathrm{EX}=370}_{\mathrm{EM}=500}}$	McKnight et al. (2001)
Fluorescence Index (FI) new version	METRIC18_FInew	Differentiate between terrestrial and marine composition.	$\frac{\mathrm{I}^{\mathrm{EX}=370}_{\mathrm{EM}=470}}{\mathrm{I}^{\mathrm{EX}=370}_{\mathrm{EM}=520}}$	Cory and McKnight (2005)
T/C ratio version a	METRIC19a_TtoC	Proxy of biogeochemical oxygen demand relative to dissolved organic carbon.	$\frac{\mathrm{I}^{\mathrm{EX}=270}_{\mathrm{EM}=350}}{\mathrm{MAX}^{\mathrm{EX}=320-340}_{\mathrm{EM}=410-430}}$	Baker (2001)
T/C ratio version b	METRIC19b_TtoC	Proxy of biogeochemical oxygen demand relative to dissolved organic carbon.	$\frac{\mathrm{MAX}^{\mathrm{EX}=270-280}_{\mathrm{EM}=320-350}}{\mathrm{MAX}^{\mathrm{EX}=330-350}_{\mathrm{EM}=420-480}}$	Baker (2001)
Proctor Index	METRIC20_ProctorIndex	Indicative of degree of humification	$\frac{\mathrm{I}^{\mathrm{EX}=350}_{\mathrm{EM}=420}}{\mathrm{I}^{\mathrm{EX}=390}_{\mathrm{EM}=470}}$	Proctor et al. (2000)
Perrette Index	METRIC21_PerretteIndex	Indicative of degree of humification	$\frac{\mathrm{I}^{\mathrm{EX}=364}_{\mathrm{EM}=514}}{\mathrm{I}^{\mathrm{EX}=364}_{\mathrm{EM}=457}}$	(Perrette et al., 2005)

*Mathematical notations:
I = intensity at a given EEM location (e.g., $\mathrm{I}^{\mathrm{EX}=470}_{\mathrm{EM}=488}$ = fluorescence intensity at excitation = 470 nm and emission = 488 nm);
A = integrated area over a given range (e.g., $\mathrm{A}^{\mathrm{EX}=254}_{\mathrm{EM}=435-480}$ = integrated area of a fluorescence spectrum acquired at excitation 254 nm and emission 435-480 nm);

MAX = maximum fluorescence signal in a given range (e.g., $\mathrm{MAX}_{\mathrm{EM}=380-420}^{\mathrm{EX}=310-320}$ = the maximum intensity value found in the region enclosed by excitation of 310-320 nm and emission of 380-420 nm).

Numerous different types of fluorophores have been observed over the years, and they generally fit the operationally defined peaks summarized in the table below and in the "EEM_Metrics.xlsx" file. It must be noted that because there is variability in their definitions, there are different "versions" of these peak labels in the literature. The **EEM_Metrics** code has been developed to export parameters for the peaks summarized by Coble et al. (2014). This list is in no way comprehensive, and we urge users of TEnvR to further modify the **EEM_Metrics** script in order to add new or modify existing peaks. It must be noted that recent advances in aquatic fluorescence indicate that the fluorophores that these peaks represent can occur in environmental samples independently of sample source (Wünsch et al., 2019). Furthermore, using specific regions of the EEM spectrum can introduce artificial trends due to shifts in peak positions or changes in spectral shapes (Korak et al., 2014). Thus, the use of these operationally defined peaks or spectral indices must be done with great care. While this "peak picking" approach has been beneficial for over two decades now, users should consider PARAFAC modeling using the drEEM toolbox (Murphy et al., 2013) in order to extract real fluorophoric signatures from their spectra. Employing PARAFAC is a holistic approach that considers all of the data and is therefore less prone to the biases described by Korak et al. (2014). The obtained PARAFAC components can then be further validated using spectral matching via the OpenFluor database (Murphy et al., 2014) and related to fluorophoric signatures observed in previously published studies.

Peak	Type of fluorophores ("source")	Definition 1	Definition 2
α	Fluorophores from older, more decomposed organic matter (humic-like FDOM from soils or terrigenous/allochthonous FDOM)	λ_{EX} = 330-350 λ_{EM} = 420-480 Table 9.1	No second definition
β	Fluorophores from recently created organic matter (free amino acids, combined amino acids, proteinaceous/microbial/autochthonous FDOM)	λ_{EX} = 310-320 λ_{EM} = 380-420 Table 9.1	No second definition
B_1	Tyrosine as a free amino acid or as part of complexes or peptides; proteinaceous/microbial/autochthonous FDOM	λ_{EX} = 230 λ_{EM} = 305 Table 3.1	λ_{EX} = 225-235 λ_{EM} = 300-310
B_2 (or B)	Tyrosine as a free amino acid or as part of complexes or peptides; proteinaceous/microbial/autochthonous FDOM	λ_{EX} = 275 λ_{EM} = 305 Table 3.1	λ_{EX} = 270-280 λ_{EM} = 300-320 Table 9.2
T_1	Tryptophan as a free amino acid or as part of complexes or peptides; proteinaceous/microbial/autochthonous FDOM	λ_{EX} = 230 λ_{EM} = 340 Table 3.1	λ_{EX} = 225-235 λ_{EM} = 335-345
T_2 (or T)	Tryptophan as a free amino acid or as part of complexes or peptides; proteinaceous/microbial/autochthonous FDOM	λ_{EX} = 275 λ_{EM} = 340 Table 3.1	λ_{EX} = 270-280 λ_{EM} = 320-350 Table 9.2

N	Unknown FDOM fluorophores, likely from proteinaceous/microbial/autochthonous FDOM	λ_{EX} = 280 λ_{EM} = 370 Table 3.1	λ_{EX} = 275-285 λ_{EM} = 365-375
M_1	Proteinaceous/microbial/autochthonous FDOM, fluorophoric by-products of photochemical degradation of DOM	λ_{EX} = 240 λ_{EM} = 350-400 Table 3.1	λ_{EX} = 235-245 λ_{EM} = 350-400
M_2 (or M)	Proteinaceous/microbial/autochthonous FDOM, fluorophoric by-products of photochemical degradation of DOM	λ_{EX} = 290-310 λ_{EM} = 370-420 Table 3.1	λ_{EX} = 310-320 λ_{EM} = 380-420 Table 9.2
C_1 (or A)	Humic-like FDOM from soils or terrigenous/allochthonous FDOM	λ_{EX} = 260 λ_{EM} = 400-460 Table 3.1	λ_{EX} = 250-260 λ_{EM} = 380-480 Table 9.2
C_2 (or C)	Humic-like FDOM from soils or terrigenous/allochthonous FDOM	λ_{EX} = 320-365 λ_{EM} = 420-470 Table 3.1	λ_{EX} = 330-350 λ_{EM} = 420-480 Table 9.2
C_1^+	Humic-like FDOM from soils or terrigenous/allochthonous FDOM	λ_{EX} = 250 λ_{EM} = 470-504 Table 3.1	λ_{EX} = 245-255 λ_{EM} = 470-504
C_2^+	Humic-like FDOM from soils or terrigenous/allochthonous FDOM	λ_{EX} = 385-420 λ_{EM} = 420-504 Table 3.1	No second definition
D	Fulvic-like FDOM from soils or terrigenous/allochthonous FDOM	λ_{EX} = 390 λ_{EM} = 509 Table 2.1	λ_{EX} = 385-395 λ_{EM} = 504-514
E	Fulvic-like FDOM from soils or terrigenous/allochthonous FDOM	λ_{EX} = 455 λ_{EM} = 521 Table 2.1	λ_{EX} = 450-460 λ_{EM} = 516-526
P	Fluorophores of pigments or phytoplankton	λ_{EX} = 398 λ_{EM} = 660 Table 3.1	λ_{EX} = 393-403 λ_{EM} = 655-665
H	Fluorophoric by-products of photochemical degradation of FDOM	λ_{EX} = 230 λ_{EM} = 275-350 Table 3.1	λ_{EX} = 225-235 λ_{EM} = 275-350

In the table above, most of the definitions are taken from Tables 3.1, 9.1, and 9.2 of Coble et al. (2014). In cases when the first peak definition is based on single emission and excitation points (e.g., Peak B_1 defined at λ_{EX} = 230 nm and λ_{EM} = 305), a second alternative definition is provided where the excitation and emission ranges are expanded with ± 5 nm (e.g., Peak B_1 defined at λ_{EX} = 225-235 nm and λ_{EM} = 300-310 nm). If the peak is originally defined as a range, no alternative definition is needed.

For each peak, several parameters are computed:

→ Maximum Intensity – the highest intensity value within the defined range of the peak. Labeled as **Max Int** in the report file.
→ Summed Intensity – sum of intensity values within the defined range of the peak. Labeled as **Sum Int** in the report file.
→ Averaged Intensity – the average intensity in the defined range of the peak. Labeled as **Avg Int** in the report file.
→ Normalized Averaged Intensity – the average intensity in the defined range of the peak divided by the Average Spectral Intensity. Labeled as **Avg Int/Avg** in the report file.
→ Peak Position of Maximum (Ex) – the excitation wavelength at which the highest intensity value occurs. Labeled as **Peak (Ex)** in the report file.
→ Peak Position of Maximum (Em) – the emission wavelength at which the highest intensity value occurs. Labeled as **Peak (Em)** in the report file.

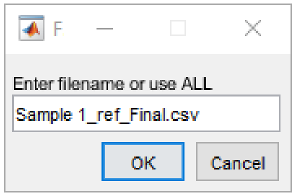
As some peaks are not defined as ranges (e.g., peak B1 at λ_{EX} = 230 nm and λ_{EM} = 305 nm), the intensity at the defined excitation and emission position is output under the “Maximum intensity” metric. The script does not output a value for the other parameters as their computation requires a range.

The **EEM_Metrics** script can be executed in two different ways: on one sample and on multiple samples. To run the code, place the samples (.csv files) containing the processed data and run the code as shown below for one of the example data files. If you have been following this tutorial so far, your Current Folder should be TestData_Generic_Processed.

Slide 74: *EEM_Metrics*

Immediately, a pop-up window will request the filename. Input the full filename, including the extension, with no apostrophes, as shown to the right.

Sample 1_ref_Final.csv


Enter filename or use ALL

Sample 1_ref_Final.csv

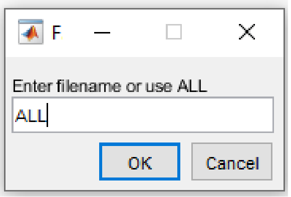
Once the sample name has been provided, the code will find the data file, import the data, calculate all metrics, and produce an Excel file (here, named “Sample 1_ref_Final_Metrics.xlsx”) in the Current Folder (**Slide 74**). This report (**Slide 75**) will contain the original EEM data in sheet “Sheet1” and the interpolated EEM data in sheet “Interpolated Data” (in cases when no interpolation has been done, the original data will be copied there). The last sheet named “Metrics” will contain three tables corresponding to the three categories of metrics described above. Please note that for peaks that are only defined at single wavelengths (e.g., Tyrosine-1, def1 in this example), no metrics besides Maximum Intensity will be computed. Peaks that are outside of the EEM range (e.g., peaks E and P in this example) will also have no computed metrics.

To apply this code on a whole dataset automatically, place all processed EEM files in a folder and make this folder to be the Current Folder. Here, the provided dataset in TestData_Generic_Processed will serve as an example. It is important that all data files have "_processed" as suffix.

Slide 76: *EEM_Metrics*

A pop-up window will request an input. In contrast with the application of this code to a single sample, here the code requires the argument ALL with no parentheses or apostrophes, as shown to the right and on **Slide 76**.

ALL



Once this argument has been provided, the code will find all .csv data files having the suffix _Final, import them, calculate all metrics, and produce an Excel file named EEM Metrics Report in the Current Folder (**Slide 76**). This report (**Slide 77**) will contain the metrics for all samples in a matrix format. Please note that for peaks that are only defined at single wavelengths, only data for the intensity at that excitation and emission position will be provided (under the label maximum intensity). Peaks that are outside of the EEM range will have no computed metrics.

The **EEM_Metrics** code utilizes external functions for alphanumeric sorting of filenames (*natsortfiles* and *natsort*) developed by Stephen Cobeldick (https://www.mathworks.com/matlabcentral/fileexchange/47434-natural-order-filename-sort). These codes and their license are found in TEnvR\Supplementary files\Internal codes.

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

- **EEM_Dilution**

While the EEM data is already corrected for the dilution factor during the processing steps, it is sometimes useful to be able to rescale an EEM spectrum that has already been processed and exported as a .csv file. This code, analogous to **UVVIS_Dilution**, can scale up (undilutes) or scale down (dilutes) the fluorescence intensity values based on the input argument. It can be also used to normalize or denormalize the intensity values to an external measurement (e.g., dissolved organic carbon content).

Slide 78: *EEM_Dilution('Sample 1_ref_Final.csv','dilute',1,5)*

In this example, the sample named “Sample 1_ref_Final.csv” was scaled down (diluted) with a dilution factor of 0.2 (or 5-fold). The diluted data is found in the newly produced file “Sample 1_ref_Final_dil_0.2.csv”. More examples of the code are shown below:

Slide 79: *EEM_Dilution('Sample 1_ref_Final.csv','undilute',1,10)*

In this example, the sample named “Sample 1_ref_Final.csv” was scaled up (undiluted) with a dilution factor of 10, as if the sample was diluted 10 times prior analysis, yielding the output file “Sample 1_ref_Final_undil_0.1.csv”.

Slide 79: *EEM_Dilution('Sample 1_ref_Final.csv','undilute',2,1)*

In this example, the sample named “Sample 1_ref_Final.csv” was normalized to a dissolved organic carbon content of 2 mg/L (or 2 ppm). Mathematically, all intensity values in the “Sample 1_ref_Final.csv” are divided by 2 to produce C-normalized fluorescence (found in the new file “Sample 1_ref_Final_undil_2.csv”).

Slide 79: *EEM_Dilution('Sample 1_ref_Final_undil_2.csv','dilute',2,1)*

In this example, the previous action is reversed – the C-normalized data in “Sample 1_ref_Final_undil_2.csv” is denormalized to result in the data in “Sample 1_ref_Final_undil_2_dil_2.csv”, which data is identical to the original data (“Sample 1_ref_Final.csv”).

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

- **EEM_Fold**

EEM data is three-dimensional: the data for each sample have independent[6] variables in two dimensions (excitation and emission wavelengths) and one dependent variable (fluorescence intensity). Data in such a 3D format cannot be evaluated using hierarchical cluster analysis (HCA) or principal component analysis (PCA), tools that have a strict requirement for only one dimension of independent and only one dimension of dependent variables (which also categorizes HCA and PCA as “two-way” methods). In other words, prior to HCA/PCA, the EEM data must be converted to two-dimensional arrays (two columns of data: one column being an independent variable, the other column being the dependent variables that change among different samples). The excitation and emission independent variables are converted into non-numeric (categorical) composite wavelength variables (EX230EM270, EX230EM275, etc.) (**Slide 80**). This conversion of 3D into 2D is referred to as folding. This enables the use of HCA and PCA on EEM data, which even though are less superior than PARAFAC (Bro, 1997), have been shown to be a useful for exploratory analysis of EEM data (Boehme et al., 2004).

Slide 81: *EEM_Fold*

[6] Variable independency is based on which variables do and do not change with different samples. Throughout an EEM dataset, excitation and emission wavelengths are kept constant (ergo, are independent variables) while fluorescence intensities are different for each sample (ergo, are dependent variables).

The **EEM_Fold** script takes multiple EEM spectra that are located in your Current Folder (ending with "_Final.csv") and folds them into 2D arrays and stacks them all into a matrix that is output in "Sheet1" of an Excel file named "EEM_AlignmentMatrix.xlsx" (**Slide 82**). This matrix is then normalized to total spectral intensity per sample so that the sum of all values equals 1. The normalized matrix is further treated to prepare the data for statistical analysis: Negative intensity values and NaN values are set to zero. Wavelengths that have the same value for all samples are removed from the dataset to avoid creating covariance values with missing values if HCA or PCA are employed. The normalized matrix is found in sheet "Normalized" of the "EEM_AlignmentMatrix.xlsx" file (**Slide 83**). This matrix can be then directly loaded into statistical codes. Please do not delete the "EEM_AlignmentMatrix.xlsx" file and do not change the sheet name "Normalized"- this file and the matrix in the "Normalized" sheet will be directly used as examples later in the tutorial.

The **EEM_Fold** code utilizes external functions for alphanumeric sorting of filenames (*natsortfiles* and *natsort*) developed by Stephen Cobeldick (https://www.mathworks.com/matlabcentral/fileexchange/47434-natural-order-filename-sort). These codes and their license are found in TEnvR\Supplementary files\Internal codes.

- **EEM_Unfold**

This code contains an algorithm that converts 2D folded EEM data into its typical 3D format. While the algorithm of this code may be used to reverse the action of **EEM_Fold**, this is not a useful capability as the original EEM files in 3D format are already available. However, if PCA is performed, it creates variable loadings that are of a folded 2D format. Thus, the **EEM_Unfold** code is needed and therefore, this code is exclusively used for unfolding PCA component loadings into 3D EEM loadings spectra. The application of this code will be shown later during the tutorial for PCA.

- **EEM_Transposition**

EEM data, output by the two processing codes described above (**EEM_Process_Aqualog** and **EEM_Process_Generic**), is organized as excitation wavelengths as columns and emission wavelengths as rows. While all codes in TEnvR require that the codes are organized this way, in working with collaborators or other software it has been found necessary that the EEM files are converted into a new format – excitation wavelengths as rows and emission wavelengths as columns. This flipping of axes is known as transposition (**Slide 84**). The **EEM_Transposition** code is designed to take multiple EEM spectra and transpose them into new .csv files denoted with the suffix _transposed (e.g., "Sample 1_ref_Final_transposed.csv").

To apply this code on a whole dataset automatically, place all processed EEM files in a folder and make this folder your Current Folder. Here, the provided dataset in TestData_Generic_Processed will serve as an example. If this code is to be run on only one EEM file, place the file in an empty folder and set that folder as your Current Folder.

Slide 85: *EEM_Transposition*

The code specifically identified any files in the current folder with the suffix "_Final.csv" and transposes them. Thus, the files that were previously created during the previous exercise (with **EEM_Dilution**) and do not have the "_Final" suffix directly before .csv were not transposed.

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

- **EEM_ThermoReformat**

EEM data acquired on Thermo Scientific Lumina spectrofluorometers is organized in a way that does not make it easily importable using the codes from drEEM. The **EEM_ThermoReformat** code takes numerous spectra exported by the instrument (_3DScan.xls format) and converts them in a readable format (**Slide 86**). The newly output spectra are denoted with the suffix _ref (in .xlsx format). We have provided 14 example files located in TEnvR/TestData_EEM/TestData_Thermo. To use this code, make the TestData_Thermo folder to be your Current Folder and run *EEM_ThermoReformat* in the command window:

Slide 87: *EEM_ThermoReformat*

The Thermo Scientific Lumina spectrofluorometer has an automatic capability to correct the acquired data for instrument-specific responses (Cory et al., 2010), but all other corrections need to be made further post-acquisition. This parallels the data processing pipeline for EEMs acquired on Shimadzu spectrofluorometers. The reformatted files need to be fully corrected following the recommendations by Murphy et al. (2010) and be processed using the **EEM_Process_Generic** script. To import the spectra, use the following arguments for the *readineems* function in the **EEM_Process_Generic** script at lines 39 (blanks) and 48 (samples): *readineems(1,'xlsx','A1..AU4102',[1 1],0,0)*.

Please note that the Thermo Scientific Lumina software considers each emission scan at each excitation wavelength to be a different sample. Therefore, excitation wavelengths in the _3DScan.xls files are labeled as different samples (e.g., Excitation at 230 nm = Sample 1, Excitation at 235 nm = Sample 2, etc.). Once files are reformatted these confusing labels are removed (**Slide 86**).

- **EEM_WaterRamanReformatAqualog**

HORIBA Aqualog Spectrofluorometers have the capability to analyze a water sample for its Raman spectrum with pre-defined settings. These spectra can be used for data normalization online using the instrument's software as described above or be exported and used for normalization in MATLAB. For the second scenario, the **EEM_WaterRamanReformatAqualog** is an automated approach that will reformat many water Raman spectra into a format that can be easily loaded in MATLAB using drEEM's *readinscans* function. We have provided 8 example files located in TEnvR\TestData_EEM\TestData_WaterRamanAqualog. To use this code, make the TestData_WaterRamanAqualog folder to be your Current Folder and run *EEM_WaterRamanReformatAqualog* in the command window:

Slide 88: *EEM_WaterRamanReformatAqualog*

The code loads .dat files that end with the suffix “Raman Area Graph.dat”. A .csv file containing wavelengths in column A and fluorescence intensity in column B will be created in the Current Folder for each of the loaded files.

Section 6. Ultrahigh Resolution Mass Spectrometry

The codes for processing ultrahigh resolution mass spectrometry were developed using data acquired only on Fourier transform – ion cyclotron resonance – mass spectrometers (FT-ICR-MS). In our experience, we have mainly used the 12 Tesla Bruker Daltonics Apex Qe spectrometer equipped with an Apollo II electrospray ionization (ESI) source at the College of Sciences Major Instrumentation Cluster (COSMIC) facility at Old Dominion University (Norfolk, VA). The instrument is currently operating at a magnetic field strength of 10 Tesla due to a recent magnet quenching incident. We also have limited experience with the 15 Tesla Bruker SolariXR FT-ICR-MS housed in the Ohio State University Mass Spectrometry and Proteomics Campus Chemical Instrument Center (Columbus, OH). While theoretically it is possible that these codes can be applied to spectra acquired on Orbitrap spectrometers (10x lower resolving power), further testing and validation would be required for the codes on spectral processing and formula assignment. Codes here that use assigned formulas (for plotting figures, calculating metrics, etc.) will work properly regardless of spectral resolution.

Typically, environmental samples are analyzed using soft ionization (i.e., ESI), producing spectra of several thousand peaks (Sleighter and Hatcher, 2007). While broadband excitation mode is commonly employed, sequential selective ion accumulation (SSIA), also known as continual accumulation of selected ions (CASI), can also be employed for enhancing the spectral qualities (Sleighter et al., 2009). Generally, 300-400 scans are acquired, and the software sums them into a final co-added spectrum. Technical details about the instrument have been described by Marshall et al. (1998). The instrument in COSMIC is calibrated daily with a polyethylene glycol standard and instrument blanks are analyzed throughout the day to verify that no sample carryover occurs. We have previously used in-house laboratory surrogate NOM standards (such as whole water from the Great Dismal Swamp, Virginia) for evaluating tuning parameters. Since our involvement in the recent inter-laboratory comparison study (Hawkes et al., 2020), we have implemented the use of the Suwannee River fulvic acid (SRFA) standard sample (purchased from the International Humic Substances Society (IHSS), http://humic-substances.org/) as a NOM standard to verify our tuning parameters and ensure that the instrument is properly functioning prior to analyzing unknown samples.

Once spectra are acquired, they are initially processed with the Bruker Daltonics Data Analysis software. This can be done immediately after sample analysis on the same computer that is used to control the instrument or at a separate processing station later. The raw instrumental data is in proprietary Bruker format (.d), making it necessary to utilize the Bruker software for the first steps of data processing. This is the first stage of processing of FT-ICR-MS data (see Fig. 3 in the manuscript as an overview of the entire processing pipeline). First, peak picking is performed based on the signal-to-noise (S/N) ratios. Typically, a S/N threshold of 3 is used for data acquired on the FT-ICR-MS instrument at Old Dominion University (Sleighter et al., 2012), and a S/N threshold of 7-10 is used for data acquired on the FT-ICR-MS instrument at Ohio State University. The precision of the m/z measurements is generally set to 6 for the two mass spectrometers mentioned above. They measure m/z with ultrahigh precision of up to five decimals, but the sixth decimal is also exported as that is the uncertainty of the measurement. Peak picking parameters may also be set to exclude peaks of insignificant intensity. In the Bruker Data Analysis software, by default, peaks with intensity below 0.01% of the intensity of the base peak are not picked. This value can be changed as needed, and setting it to 0.000001% generally removes the intensity threshold as a parameter for peak picking. The spectra are then evaluated visually by the user for any severe artifacts, such as peak splitting from space-charge effects, electronic noise signals, or shoulder peaks. Even with the recent developments in data processing automation, for ultrahigh resolution mass spectrometry or other types of instrumentation, visual inspection of the individual raw spectra is critical for obtaining high-quality results. This has also been recommended recently by Patriarca and Hawkes (2020) who identify that multiply charged peaks introduce significant artifacts in the molecular formula lists. Once peak picking and inspection are done, the spectrum is internally calibrated using naturally present fatty acids and other NOM compounds belonging to various CH_2 homologous series (e.g., $C_{14}H_{27}O_2^-$, $C_{15}H_{29}O_2^-$,$C_{16}H_{31}O_2^-$, etc.), following Sleighter et al. (2008). Spectra are then exported as text files (.txt), containing m/z, magnitude, and S/N values for each peak. This type of data will be hereafter referred to as a "peak list". Any further processing in performed in MATLAB using TEnvR as described below.

For this tutorial, we have provided calibrated peak lists for 20 samples and one procedural blank acquired in negative-mode ESI (found in TEnvR/TestData_FTICRMS, **Slide 89**) and five samples and one procedural blank acquired in positive-mode ESI (found in TEnvR/TestData_FTICRMS/PositiveMode, **Slide 90**). The samples are of NOM from swamps and rivers, and some of the samples have been photo-irradiated. The samples were prepared by sterile-filtration (0.1 μm filter) and then solid-phase extraction using PPL cartridges (Dittmar et al., 2008). Methanolic PPL extracts were infused directly into the 15-Tesla SolariXR FT-ICR-MS housed in the Ohio State University Mass Spectrometry and Proteomics Campus Chemical Instrument Center (Columbus, OH) and were analyzed in negative-mode ESI. While we will show examples of processing and assessing data from negative-mode ESI, all codes in TEnvR work on peak lists of spectra acquired in positive-mode ESI as well (sample data provided in TEnvR/TestData_FTICRMS/PositiveMode, **Slide 90**). The codes of TEnvR have not been tested on data acquired on other types of sources (APPI, LDI, MALDI, etc.), thus using TEnvR for such data will likely require some code modification. For such requests, users are invited to communicate with the corresponding authors.

Please note that users may export a different data property in the third column instead of S/N (e.g., resolving power, retention time in case of LC-MS applications). The codes in TEnvR currently do not depend on S/N data to function properly. If users choose to export a different property in column 3 (or not export anything at all), this will only affect the data quality control figure exported at the end of the **FTMS_RefinementFormulas** code (described in detail later).

- **FTMS_ConfigurationAssignment** and **FTMS_ConfigurationToolbox**

The FTMS side of TEnvR requires the use of two configuration files (**FTMS_ConfigurationAssignment** and **FTMS_ConfigurationToolbox**). Different from all previous codes where the user had to open the code itself and alter parameters, here all such parameters are consolidated into configuration files.

Slide 91: *edit FTMS_ConfigurationAssignment*

The **FTMS_ConfigurationAssignment** file contains all parameters required for processing FTMS data (refining peak lists, assigning formulas, refining formulas) shown on **Slides 91** and **92**. These parameters will be described in detail later. This code often requires modification by the user to tailor the FTMS data processing to their specific type of samples.

Slide 93: *edit FTMS_ConfigurationToolbox*

The **FTMS_ConfigurationToolbox** code defines the file format of the produced formula lists (lines 63-67, **Slide 93**) and informs how the rest of the codes (pertaining to data analysis, visualization, etc.) import and manipulate data (lines 44-57). We advise users to never alter the file format parameters (lines 63-67) unless they are altering the formula refinement code (**FTMS_RefinementFormulas**) and the way formulas are exported. Variables on lines 44-57 are to be altered depending on the used dataset of samples and instrumentation. These parameters will be described in detail later. This code often requires modification by the user to tailor the FTMS data processing to their specific type of samples.

We recommend that users take these default configuration codes and create new ones for each different dataset. We recommend modifying the file name to be specific for each individual project (e.g., **FTMS_ConfigurationAssignmentAmazonSoils**, **FTMS_ConfigurationToolboxOceanPhoto**, etc.) and thus configuration files can be easily distinguished. These files can be preserved to keep a paper trail for the employed parameters in formula assignment and formula evaluation using TEnvR. Please note that if you change the name of the file, you also must make an identical edit inside the code on line 1 for the variable corresponding to *classdef*.

- **FTMS_RefinementPeaks**

The next step of processing FT-ICR-MS data is the refinement of peak lists. A previous version of this code had been previously published (Obeid, 2015) but the script has undergone significant modifications since. The **FTMS_RefinementPeaks** script loads the peak lists of a sample and its corresponding blank (instrument blank, procedural blank, etc.). The code identifies which peaks of the blank are found in the sample within the defined mass accuracy range (the part per million (ppm) difference in their m/z values). Mass accuracy of 1 ppm is typical for the two FT-ICR-MS instruments we have experience with, and this parameter must be tested and adjusted when the codes of TEnvR are employed on data from other instruments. This is particularly important for instruments of lower resolving power, such as Orbitrap FT-MS. The identified blank peaks will be removed. Then, peaks of inorganic origin, such as solvent-salt or NOM-salt adducts (Brown and Rice, 2000; Stenson et al., 2002; Chen et al., 2011) are identified based on their mass defect (i.e., the distance that the peak is displaced from its nominal mass) as described in the table below. These peaks are referred to as “salt” peaks. These peaks cannot be assigned molecular formulas using the elemental criteria described later, or if are assigned, they are of abnormal elemental composition. Therefore, peaks of inorganic origin are removed from the peak list.

For peaks with m/z:	Peaks of inorganic origin are with mass defect of:
< 300	0.4 – 0.97
300-500	0.5 – 0.96
500-800	0.6 – 0.95
> 800	0.7 – 0.94

Once salt peaks are identified, then ^{13}C, ^{34}S, ^{54}Fe, ^{37}Cl, and ^{200}Hg isotopologue peaks are identified. For example for carbon, a monoisotopic peak contains only carbon-12 atoms (^{12}C). The corresponding ^{13}C isotopologue peaks are of ions containing one or more carbon-13 isotopes, with the remaining C-atoms being ^{12}C. Typically in NOM spectra, only isotopologue ions containing one ^{13}C are detected. For some peaks of very high magnitude, isotopologue ions having two ^{13}C peaks are also sometimes observed. The **FTMS_ RefinementPeaks** code identifies ^{13}C, ^{34}S, ^{54}Fe, ^{37}Cl, and ^{200}Hg isotopologue peaks based on mass differences and intensity thresholds as described in the table below. For example for carbon, ^{13}C isotopologue peaks are defined as peaks occurring with a m/z spacing of 1.003355 (^{13}C – ^{12}C), and the peak magnitude of the ^{13}C isotopologue peak(s) must be lower than 50% of the peak magnitude of the monoisotopic (^{12}C only) peak. Peak identification is done with a user-defined mass accuracy (typically ± 1 ppm). It must be noted that ^{37}Cl and ^{200}Hg isotopologues have a more complex isotopic distribution and therefore their magnitude threshold is a range. For more information about ^{37}Cl and ^{200}Hg isotopologues and their identification, readers are referred to the article by Chen et al. (2017). The magnitudes of the monoisotopic and first isotopologue peaks can be used to estimate the number of elements in the ion using the equation shown in the last column of the table below:

Isotopologue	m/z spacing	Magnitude threshold	Estimation of number of elements:
^{13}C	$^{13}C - {}^{12}C = 1.003355$	Below 50%	$\text{Number of C} = \frac{\text{Magnitude of } ^{13}\text{C}}{\text{Magnitude of } ^{12}\text{C}} \times 0.011$
^{34}S	$^{34}S - {}^{32}S = 1.995796$	Below 30%	$\text{Number of S} = \frac{\text{Magnitude of } ^{34}\text{S}}{\text{Magnitude of } ^{32}\text{S}} \times 0.045$
^{54}Fe	$^{56}Fe - {}^{54}Fe = 1.995327$	Below 20%	$\text{Number of Fe} = \frac{\text{Magnitude of } ^{54}\text{Fe}}{\text{Magnitude of } ^{56}\text{Fe}} \times 0.063$
^{37}Cl	$^{37}Cl - {}^{35}Cl = 1.99705$	Above 20% & Below 200%	$\text{Number of Cl} = \frac{\text{Magnitude of } ^{37}\text{Cl}}{\text{Magnitude of } ^{35}\text{Cl}} \times 0.32$
^{200}Hg	$^{202}Hg - {}^{200}Hg = 2.002316$	Above 50% & Below 200%	$\text{Number of Hg} = \frac{\text{Magnitude of } ^{202}\text{Hg}}{\text{Magnitude of } ^{200}\text{Hg}} \times 0.77$

Once isotopologue peaks are identified, doubly-charged peaks are identified. While molecules of NOM generally ionize as singly-charged ions due to loss (in negative ESI) or gain (in positive ESI mode) of a hydrogen ion (Kujawinski and Behn, 2006), doubly-charged ions can sometimes be observed. These ions have lost (or gained) two H^+, and if that is the case, their ^{13}C isotopologues occur at a m/z spacing of 0.501678 (1.003355/2). The formation of doubly-charged ions has been recently evaluated in detail by Patriarca and Hawkes (2020), and it has been determined that these ions must be removed prior formula assignment. Otherwise, doubly-charged ions may be misinterpreted as unique ions, and therefore, misassigned with incorrect molecular formulas. **FTMS_RefinementPeaks** includes the MATLAB code that is provided by Patriarca and Hawkes (2020), which has been slightly modified. After identification of all peaks described above (blank, salt, doubly-charged, and isotopologues), they are removed from the mass list.
To see an application of this code, set the folder TestData_FTICRMS as your Current Folder.

Slide 94: *FTMS_RefinementPeaks('Sample 1.txt','Blank PPL.txt',FTMS_ConfigurationAssignment)*

The first argument of the code (*'Sample 1.txt'*) is the sample name, while the second one is the blank name (*'Blank PPL.txt'*). The code also requires a configuration file (*FTMS_ConfigurationAssignment*) described in detail later. The code creates an Excel file ("Sample 1_Refinement.xlsx") with two sheets: "Sheet1" and "DataRefined". Sheet "Sheet1" contains all information about the refinement process (**Slide 95**). The first three columns (A, B and C) contain the data of the sample that was loaded (here, this is the peak list in "Sample 1.txt" containing m/z, magnitudes, and S/N values). The next three columns identify blank, salt, and doubly-charged peaks (columns D, E, and F). Their cells are populated with the m/z value of the peak that is identified to be blank, salt, or doubly-charged, respectively. For example (**Slide 96**), the peak with m/z 220.911872 (row 227) is identified as a salt peak and the "Salt" column (column D) is populated with its m/z value (220.911872). The following columns (G → AE) correspond to the isotopologue analysis. There are four columns for each element (C, S, Fe, Cl, Hg) as described below for carbon:

→ Column G, label "13C" – This column is populated for ^{13}C isotopologue peaks with their m/z value (e.g., row 244 on **Slide 96**).

→ Column H, label "Estim C#" – This column will provide a computation of the estimated number of carbons in a molecule that would have ionized to produce the ^{12}C isotopologue peak (e.g., row 241 on **Slide 96**). This is calculated based on the ^{13}C/^{12}C magnitude ratio threshold listed in the table above. Please note that carbon estimates are reliable only when ^{13}C isotopologues of high S/N (e.g., ≥ 25) are used (Koch et al., 2007).

→ Column I, label "13C m/z" – This column will provide the m/z value of the corresponding ^{13}C isotopologue peak (e.g., row 241 on **Slide 96**).

→ Column J, label "13C Int" – This column will provide the magnitude value of the corresponding ^{13}C isotopologue peak (e.g., row 241 on **Slide 96**).

→ Column K, label "difC13" – This column will provide the difference in m/z values between the ^{13}C and ^{12}C isotopologue peaks (e.g., row 241 on **Slide 96**). This difference should be nearly identical to the one listed in the table above (for carbon = 1.003355)

These data can be very useful in detailed molecular assessments, validating formula assignments, or determining the contributions of heteroelements (e.g., Cl). We urge users to explore these evaluations and expand this code for other heteroelements (e.g., Br) if needed.

On the top right (in the range AG2:AJ13), a table is provided with a summary listing the number of peaks of each category (all peaks, blank, inorganic, etc.), including two types of percentages. The number-based percentage (% Num) is based on number of formulas. For example, in the Sample 1 given as an example here, there are 130 salt peaks and 9526 total peaks. Thus, the number-based percentage is based on these numbers (130/9526 * 100 = 1.36%). The magnitude-based percentage (% Magn) is based on the spectral magnitude of peaks. Thus, the magnitude-based percentage of salt peaks in Sample 1 is the summed magnitude of all salt peaks divided by the summed magnitude of the total peaks resulting in 0.19%.

The results found in this table are very important for quality control and must be evaluated for each sample. The absolute number of blank peaks, along with their number- and magnitude-based percentages, are generally used to assess for contamination. We generally consider blanks to be "clean" if there are 500 peaks or less, but we have also observed blanks of 1000+ peaks in cases where we have performed procedural blanks covering a whole experiment. The parameters for salt peaks can be used to evaluate the cleanliness of the sample matrix. The presence of too many salt peaks (% Num > 10%) can indicate that the sample matrix is not suitable for ESI-FT-ICR-

MS analysis and a desalting procedure may need to be employed. Commonly, such procedures include solid-phase extraction (SPE) using PPL (Dittmar et al., 2008), C18 cartridges (Louchouarn et al., 2000; Kim et al., 2003b; Sleighter and Hatcher, 2008), XAD resins (Wilson et al., 1983; Mopper et al., 2007), or reverse osmosis-electrodialysis (Chen et al., 2011; Chen et al., 2014). However, it must be noted that any of these methods can fractionate NOM, and this will affect the observed molecular formulas (Green et al., 2014). The parameters regarding the other isotopologues can be used to determine if unusual heteroelements (Cl, Hg, Fe) are present in the sample and therefore, if one of these heteroelements needs to be involved in the formula assignment process (described later). In a nutshell, the results in this table are to be evaluated to perform quality control of the peak list.

The second sheet, “DataRefined”, contains the refined mass list (**Slide 97**). It contains the m/z, magnitude, and S/N values in addition to the estimated numbers of carbon atoms for peaks which have ^{13}C isotopologues. This refined list is also exported as a text file (here, as “Sample 1_Refinement.txt”) and is used further for formula assignments in the next steps of the data processing pipelines (**Slide 98**).

The processing pipelines of TEnvR are versatile and can be fine-tuned. This is done by modifying the parameters in the **FTMS_ConfigurationAssignment** configuration file.

The **FTMS_RefinementPeaks** code has several modifiable capabilities. First, it has a mass filter allowing the user to trim the m/z range down to a specified region. This filter is configured in **FTMS_ConfigurationAssignment** (**Slide 91**). The desired range is defined using the *MZcutoff_low* and *MZcutoff_high* variables on lines 42 and 43, respectively. It is sometimes beneficial to trim the data to a specific m/z range (e.g., 300-800), especially if there are significant tuning differences among samples and the detection of ions below m/z *MZcutoff_low* and above *MZcutoff_high* is less reproducible. To disable this feature, simply put numbers outside of the range you will be working with. For example, with the provided dataset, using *MZcutoff_low = 0* and *MZcutoff_high = 2000* (**Slide 91**) is sufficient to disable this feature as mass spectra have been acquired over m/z 100-1000.

Another capability of the code is the export of a .txt file containing ^{35}Cl isotopologue peaks. These peaks contain organically bound chlorine (e.g., CHOCl, CHONCl) and have an associated ^{37}Cl isotopologue peak. The formula assignment of peaks attributed to halogenated molecules is not straight-forward, and thus it must be supervised by the user. We use this feature to manually export ^{35}Cl isotopologue peaks and separately assign them in cases where organochlorine compounds are suspected to be significantly contributing to the sample composition (Wozniak et al., 2020). This feature can be enabled using the *Export_Cl* variable on line 46 in **FTMS_ConfigurationAssignment** (use *true* to enable this feature and *false* to disable this feature). Lastly, mass accuracy and precision must be defined using the *MassAccuracy* and *Precision* variables on lines 49 and 50. Currently their values are set to *1* ppm and *5* decimals, respectively (**Slide 91**). These parameters may need tuning for instruments of lower resolving power (e.g., Orbitrap FT-MS). Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

- **FTMS_FormulaAssignment**

Once the calibrated peaks are refined using the **FTMS_RefinementPeaks** code, formulas are assigned to them using the **FTMS_FormulaAssignment** code. Fundamentally, this code creates all possible combinations of C, H, O, N, S, P, and optionally, other elements, and then refines the corresponding formulas based on their DBE (described in detail later). DBE must be zero or a positive whole number to satisfy the nitrogen rule (McLafferty and Turecek, 1993). Formulas are matched to the imported peak m/z values based on the user-determined mass accuracy in the **FTMS_ConfigurationAssignment** (line 49, variable *MassAccuracy*, **Slide 91**).

This code has been previously published by Obeid (2015). The formula assignment algorithm is configured using the **FTMS_ConfigurationAssignment** file (**Slides 91-92**).

Slide 91: *edit FTMS_ConfigurationAssignment*

There are several key parameters described below. The parameters currently set are the ones that we consider as default and are in no way serving a universal purpose. In addition to evaluating their appropriateness for each dataset, the user must consider them in the formula assignments assessment and data interpretation. We recommend taking a screenshot and saving it in the folder with the dataset or recording these parameters in your laboratory notebook.

→ Ionization mode: This is dependent on the analysis and the user must choose *'negative'* or *'positive'* (line 54, variable *Ionization_Mode*). Please note that if positive mode is used, sodium (^{23}Na) and potassium (^{40}K) must be also included in the elemental criteria (described below).

→ Default elemental criteria for negative-mode ESI: $^{12}C_{5-\infty}$, $^{1}H_{5-100}$, $^{16}O_{1-30}$, $^{14}N_{0-5}$, $^{32}S_{0-4}$, $^{31}P_{0-2}$. These notations mean that, for example for oxygen (its ^{16}O isotope), the number of oxygens in a molecule must be between 1 and 30. It must be noted that these elemental criteria ranges are variable among studies. The ranges we provide here are broadly applicable for NOM samples but must be verified on a per-project basis to ensure for their appropriateness. The masses of the elements used are listed in lines 59-61 and correspond to the mass of the element of the highest isotopic abundance (^{12}C, ^{1}H, etc.).

→ Heteroelements: the user is able to assign one additional element other than C H O N S P K Na (referred to as heteroelement hereafter and labeled as E). This can be ^{35}Cl (Wozniak et al., 2020), ^{202}Hg (Chen et al., 2017), or others. Enable this feature by changing the *Heteroelement_Halogen* variable (line 55) to *true*. If the heteroelement happens to be ^{15}N, change the *Heteroelement_N15* variable (line 56) to *true* and keep the *Heteroelement_Halogen* variable (line 55) as *false*. To indicate the mass of the heteroelement, use the *Mass_Heteroelement* variable on line 57. Indicate the range of heteroelements using the *E_min* and *E_max* variables on lines 67 and 68, respectively. Please note that adding different heteroelements may require adjusting the codes as well as parameters in the configuration files as certain mathematics (e.g., DBE and AI_{MOD} computations) will be affected by the variable valences of different heteroelements. We urge users to be very cautious and carefully assess the assignment of formulas and their refinement throughout the different stages of the **FTMS_FormulaAssignment** and **FTMS_RefinementFormulas** codes.

→ Default elemental criteria for positive-mode ESI: $^{12}C_{5-\infty}$, $^{1}H_{5-100}$, $^{16}O_{0-30}$, $^{14}N_{0-5}$, $^{32}S_{0-4}$, $^{31}P_{0-2}$, $^{23}Na_{0-1}$, $^{40}K_{0-1}$. When samples are analyzed in positive-mode ESI, many ions can form adducts with naturally present sodium ($[M+Na]^+$) or potassium ($[M+K]^+$) ions (Konermann et al., 2013). These ions are also known as being “sodiated” or “potassiated”. When assigning peak lists from mass spectra acquired in positive-mode ESI, the codes in TEnvR require the inclusion of ^{23}Na and ^{40}K in the elemental criteria. Do this by setting both *K_max* and *Na_max* variables to *1* (line 68).

Once the settings in the **FTMS_ConfigurationAssignment** file are set, you can run the **FTMS_FormulaAssignment** code on the "Sample 1_Refinement.txt" file.

Slide 99: *FTMS_FormulaAssignment('Sample 1_Refinement.txt',FTMS_ConfigurationAssignment)*

The code produces a file named "Sample 1_Refinement_F.txt" which contains all formulas. Given that the mass spectrometric data represents m/z values for singly-charged ions (multiply-charged ions, if such existed, had been removed), the assigned formulas here are for ions and we refer to them as ion formulas. Later, these formula data will be corrected to represent the initial molecules (i.e., obtain the molecular formulas). In this file (**Slide 100**), it can be seen that there are two different m/z values for each formula, labeled m/z and m/z2. For example, the first peak at m/z = 177.055598 has an m/z2 = 177.0557173799. Here, the m/z value of 177.055598 is the m/z value of the ion that was measured by the instrument, calibrated, and loaded by the user into the code. The m/z2 value corresponds to the calculated molecular weight of the ion plus the mass of one electron. Therefore, the m/z2 for the first peak is equal to $(10 \times 12.000)_C + (9 \times 1.007825)_H + (3 \times 15.9949146)_O + (1 \times 0.0005485799)_{electron}$. The m/z2 values (the molecular weight of the ions) are not used anywhere further in TEnvR as the molecular weight of the neutral molecule (i.e., the ExactMass) is computed later by the **FTMS_RefinementFormulas** code.

Upon further inspection of this file (scrolled down in "Sample 1_Refinement_F.txt", **Slide 101**), it can be seen that there are two possible assignment scenarios for each mass peak:

→ There is only one formula generated for one mass peak: this means that, with the assignment parameters described above, only one ion formula could be generated for the m/z value of the mass peak (i.e., the peak is assigned unambiguously). For example (**Slide 101**), the peak at m/z = 241.071765 has been assigned unambiguously (only one possible formula: $C_{11}H_{13}O_6$). Such formulas are often referred to as "unique" or "unambiguous" assignments. The value in the first column of the "Sample 1_Refinement_F.txt" file for such formulas is 1, as there is only one formula option for this peak.

→ There are two or more formulas generated for one mass peak: this means that, with the assignment parameters described above, multiple ion formulas could be generated for the m/z value of the mass peak (i.e., the peak is assigned ambiguously). For example (**Slide 101**), the peak at m/z = 237.004066 has been assigned ambiguously (two possible formulas: $C_5H_{10}ON_4S_2P$ and $C_{10}H_5O_7$). Such formulas are often referred to as "ambiguous" assignments. The value in the first column of the "Sample 1_Refinement_F.txt" file for such formulas is 2 or more, as there are multiple formula options for this peak.

This list of formula assignments, however, is far from final. Ambiguous assignments need to be refined. Furthermore, some formulas, even though they are assigned unambiguously, need to be removed from the dataset as they are chemically unrealistic (Stubbins et al., 2010). The list of ion formulas will be further processed and refined using the next code of the toolbox.

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

- **FTMS_RefinementFormulas**

The final step of processing ultrahigh resolution mass spectrometry data is refining the list of assigned formulas. This is necessary, because many of the mass spectral peaks have more than one option of assignable formulas (i.e., most peaks are assigned ambiguously, see **Slide 101**). It is critical that the final formula list contains only one molecular formula per mass peak. Thus, the major goal of the refinement process using **FTMS_RefinementFormulas** is to evaluate all ambiguous formula assignments and determine the ones that have the highest probability to be the most accurate molecular formula. The code will also filter formulas that are abnormal for NOM in dependence on user-defined constraints (Stubbins et al., 2010). Due to the associated mathematics and high dimensionality of data, the **FTMS_RefinementFormulas** script is likely the most complex code of TEnvR. We advise users to be highly cautious when performing modifications to this script. To apply this code, please keep the folder TestData_FTICRMS as your Current Folder. The configuration file **FTMS_ConfigurationAssignment** contains the processing parameters, and they must be edited if necessary. For the example below, keep the settings in **FTMS_ConfigurationAssignment** as set by default.

Slide 102: *FTMS_RefinementFormulas('Sample 1_Refinement_F.txt',FTMS_ConfigurationAssignment)*

While the script is running, in the bottom left corner of MATLAB will show “Busy”. If at any time the user wants MATLAB to stop an execution, use Ctrl+C on Windows computers. The formula refinement algorithm is generally fast and the whole code takes a couple of minutes to complete per sample.

Once the code finishes refining the formula list in “Sample 1_Refinement_F.txt”, it will output two Excel files (“Sample 1_Refinement_Processing.xlsx” and “Sample 1_Final.xlsx”), as well as a figure in a portable network graphic (PNG) format (“FTMS Processing_Sample 1.png”) (**Slide 102**). The figure will be shown as a pop-up MATLAB window that will maximize on the whole screen (**Slide 103**). All processing steps, figure, and output files are described in detail below. Conceptually, the formula refinement process is described on **Slide 104**.

The **FTMS_RefinementFormulas** code is split into eight processing stages. During the first stage of the code (*%% Stage 1: Import Data & Reorganize file from Molecular Formula Calculators*), the assigned formulas are loaded into MATLAB. Immediately after import, the data is reorganized to a consistent format. Then, an index parameter is computed to identify ambiguous assignments – formulas having the same index value indicate two or more ambiguous formulas per peak that need refinement (e.g., the ambiguous formulas $C_{10}H_5O_7$ and $C_5H_{10}ON_4S_2P$ assigned to m/z 237.004066, **Slide 101**). This index value is used later in the formula refinement steps. In this stage of the code, the results from the **FTMS_RefinementPeaks** code found in the “_Refinement.xlsx” are also loaded (here, from “Sample 1_Refinement.xlsx”). Therefore, it is critical that all processing files are kept in the same folder. The data before (from “_Refinement.xlsx”) and after formula assignment (from “_F.txt”) are aligned together. The S/N and estimated C-number data from the “_Refinement.xlsx” file are transcribed into the new file containing the assigned formulas. Once this is done, the code creates the “Sample 1_Refinement_Processing.xlsx” file and writes the data at the current stage in “Sheet1” (**Slide 105**). This processing file will serve as paper trail for the whole formula refinement process. This formula list in “Sheet1” at this stage of the code has had no refinements and this exported list is solely the reformatted data from the “_F.txt” file (columns B-O) with the calculated index parameter (column A) and the imported S/N and estimated C-numbers from the "_Refinement.xlsx" file (in columns P and Q, respectively). It must be noted that in this list, the number of assigned H-atoms (column F, named Hion) are for the detected ion, and thus these are ion formulas and not molecular formulas. Column K is where the number of assigned heteroelements (if any) are listed under the label "E". If no such element is assigned, this column is populated with zeros. If K and Na are assigned for samples analyzed in positive ionization mode, columns L and M will be populated with their numbers, respectively. Otherwise for negative ionization mode, columns L and M are populated with zeros.

The next stage of the code (*%% Stage 2: H-correction, Reformatting, and Initial Filtering using Elemental Constraints*) takes this list of formulas and performs the first refinement steps. Remember, the imported formulas are for the detected ions by the ICR cell. Here, the number of hydrogens (H-number, denoted as "H" in calculations below) is computed for the molecule that would have been ionized to produce the singly-charged ion. This is critical for obtaining the molecular formulas corresponding to the original molecules. This computation is dependent on the ionization mode. Molecules analyzed using negative-mode ESI lose one hydrogen atom to become singly-changed ions ($[M\text{-}H]^-$). Thus, the H-number of the molecule simply needs one additional hydrogen atom. This requires that all ions are singly-charged, which is why removal of multiply-charged ions is critical prior to formula assignment (Patriarca and Hawkes, 2020). Molecules analyzed using positive-mode ESI gain either a hydrogen ($[M+H]^+$), a sodium ($[M+Na]^+$), or a potassium ($[M+K]^+$) ion. In cases when they have gained a hydrogen ion, the computed H-number of the ion already contains it and therefore it must be subtracted. In cases when they have gained a K^+ or Na^+, the H-number for the molecule is identical to that of the potassiated or sodiated ion. Therefore, the equation for positive-mode ESI below shows the subtraction of 1 plus the gained K^+ or Na^+ to obtain the H-number of the molecule. Please note that in all equations, the number of elements (e.g., number of hydrogen atoms) in a molecule/ion are expressed as the atom label with the species type, where applicable, as a subscript (e.g., $H_{molecule}$, K_{ion}). These computations are done automatically in the second stage of the **FTMS_RefinementPeaks** code, but the user must define the ionization mode in the **FTMS_ConfigurationAssignment** (line 54, variable: *Ionization_Mode*, **Slide 91**). The user must choose between *'negative'* or *'positive'*. Usage of other wording will produce an error message.

$$\text{If } \mathit{Ionization\_Mode} = \mathit{'negative'}: \; H_{molecule} = H_{ion} + 1$$

$$\text{If } \mathit{Ionization\_Mode} = \mathit{'positive'}: \; H_{molecule} = H_{ion} - 1 + K_{ion} + Na_{ion}$$

Once the H-number is corrected, the obtained set of elements now represents the formulas of the original molecules (i.e., molecular formulas). For example for negative-mode ESI, a peak with m/z = 177.019261 is assigned with a formula $C_9H_5O_4$. This formula represents a $[C_9H_6O_4\text{-}H]^-$ ion. The code will correct the H-number using the equation above, and therefore, the obtained molecular formula will be $C_9H_6O_4$. All further calculations described below will be made using elemental values of molecular formulas and not ion formulas. For example for positive-mode ESI for a hydrogenated ion ($[M+H]^+$), a peak with m/z = 189.091012 is assigned with a formula $C_{12}H_{13}O_2$. This formula represents a $[C_{12}H_{12}O_2+H]^+$ ion. The code will correct the H-number using the equation above, and therefore, the obtained molecular formula will be $C_{12}H_{12}O_2$. For example for positive-mode ESI for a sodiated ($[M+Na]^+$) or a potassiated ion ($[M+K]^+$), a peak with m/z = 211.05763 is assigned with a formula $C_8H_{12}O_5Na$. This formula represents a $[C_8H_{12}O_5+Na]^+$ ion. The code will correct the H-number using the equation above and therefore the obtained molecular formula will be $C_8H_{12}O_5$ (in this case the H-number does not change). All further calculations described below will be made using elemental values of the molecular formulas and not ion formulas. The exact molecular weight (ExactMass) of molecular formulas is then computed using the user-defined atomic weights in the **FTMS_ConfigurationAssignment** (lines 59-61, **Slide 91**). Again, this ExactMass is for the neutral molecule. Therefore, if a molecule was ionized and detected as both $[M+H]^+$ and $[M+Na]^+$ species, the resultant ExactMass will be duplicated and the same for the two different m/z measurements.

$$ExactMass_{molecule} = C * Mass\_12C + H * Mass\_1H + O * Mass\_16O + N * Mass\_14N + S * Mass\_32S \; + P * Mass\_31P \\ + E * Mass\_Heteroelement$$

If an additional heteroelement is specified in line 55 or 56 of **FTMS_ConfigurationAssignment** (variable: *Heteroelement_Halogen*, Cl, Br, Fe, Hg, etc. or variable: *Heteroelement_N15* for ^{15}N isotopic labeling), the atomic weight of the heteroelement must also be listed (line 57, variable: *Mass_Heteroelement*, **Slide 91**). The value given to this variable is irrelevant if no heteroelements are assigned (we prefer to keep 0 if no heteroelement is used in the assignment).

The code then calculates five additional metrics from the molecular formulas: hydrogen-to-carbon (H/C) and oxygen-to-carbon (O/C) molar ratios as well as double-bond equivalencies (DBE), C-normalized DBE (DBE/C), and modified aromaticity index (AI_{MOD}). H/C and O/C are critical parameters for evaluating molecular composition and viewing (visualizing) FT-ICR-MS data (Van Krevelen, 1950; Kim et al., 2003a). They are described in detail later in this tutorial.

$$H/C = \frac{Number\ of\ H}{Number\ of\ C}$$

$$O/C = \frac{Number\ of\ O}{Number\ of\ C}$$

DBE is a measure of the total number of π-bonds (usually double C-bonds, C=C) and alicyclic rings in a molecule (Bae et al., 2011). It is also known under the names “degree of unsaturation” (DoU or DU), “degree of hydrogenation” (DoH or DH), “index of hydrogen deficiency” (IHD), and “unsaturation index”. When normalized to the number of carbons in the molecule, the DBE/C measure can serve as a proxy for the aromaticity of molecules (Hockaday et al., 2006). In difference with DBE, which only quantifies π-bonds + alicyclic rings in the total molecular structure, the modified aromaticity index (AI_{MOD}) measures the double-bond density in a molecule (Koch and Dittmar, 2006, 2016). Molecules with $AI_{MOD} \geq 0.67$ have the highest probability of having many double-bonds concentrated in one region of their structure, and thus are operationally defined as “condensed aromatic”. Molecules with $0.5 \leq AI_{MOD} < 0.67$ are classified as aromatic. Molecules with $0 < AI_{MOD} < 0.5$ have the probability of having either a smaller aromatic moiety that is highly functionalized with aliphatic groups or have many olefinic/alicyclic bonds spread out throughout their structure. Compounds with $AI_{MOD} = 0$ are classified as aliphatic.

The computation of DBE and AI_{MOD} is dependent on the assignment of a heteroelement and its valence. The current version of the **FTMS_ConfigurationAssignment** code can consider how the presence of two types of heteroelements alter the way DBE and AI_{MOD} are calculated. Halogens in NOM (Cl, Br, etc.) are of valence = 1, are highly electronegative (enhancing molecular ionization), and they behave similarly like hydrogen in terms of bonding with organic molecules (they substitute hydrogen). Thus, in terms of the DBE and AI_{MOD} computations, they are involved in the calculation in the same way as hydrogen is. Deuterium-labeling has recently become a popular approach (Bianca et al., 2020; Zherebker et al., 2020a; Zherebker et al., 2020b) and if ^2H (or D) is assigned, the user must define its mass in the **FTMS_ConfigurationAssignment** (line 57, variable: *Mass_Heteroelement*, **Slide 91**) and inform the code that the heteroelement behaves like a halogen. To define that the assigned heteroelement is a halogen or behaves in a similar way regarding DBE and AI_{MOD} computations, change the variable *Heteroelement_Halogen* in the **FTMS_ConfigurationAssignment** (line 55, **Slide 91**) from *'false'* to *'true'*. In our experience, we have previously used ^{15}N-labeling and had to assign ^{15}N as a heteroelement. If ^{15}N is assigned as the heteroelement, change the variable *Heteroelement_N15* in the **FTMS_ConfigurationAssignment** (line 56, **Slide 91**) from *'false'* to *'true'*. Future versions of TEnvR will incorporate mathematics inclusive to other elements as well as groups (e.g., UO_2^{2+}).

In the calculations below and in the rest of this tutorial, the number of elements (e.g., C-number, H-number) in formulas is going to be only defined by the element label (e.g., C, H). The label "E" is used to denote the number of heteroelements (if none assigned, all E values = 0).

$$\text{If } Heteroelement\_Halogen =' true' \,\&\, Heteroelement\_N15 =' false' \begin{cases} \mathrm{DBE} = 1 + \dfrac{2 * \mathrm{C} - \mathrm{H} + \mathrm{N} + \mathrm{P} - \mathrm{E}}{2} \\[2ex] \mathrm{AI_{MOD}} = \dfrac{1 + C - \frac{O}{2} - S - \frac{N + P + H + E}{2}}{C - \frac{O}{2} - N - S - P} \end{cases}$$

$$\text{If } Heteroelement\_Halogen =' false' \,\&\, Heteroelement\_N15 =' true' \begin{cases} \mathrm{DBE} = 1 + \dfrac{2 * \mathrm{C} - \mathrm{H} + \mathrm{N} + \mathrm{P} + \mathrm{E}}{2} \\[2ex] \mathrm{AI_{MOD}} = \dfrac{1 + C - \frac{O}{2} - S - \frac{N + P + H + E}{2}}{C - \frac{O}{2} - N - S - P} \end{cases}$$

$$\text{If } \mathit{Heteroelement\_Halogen} = '\mathit{false}' \ \& \ \mathit{Heteroelement\_15N} = '\mathit{false}' \begin{cases} \mathrm{DBE} = 1 + \dfrac{2 * \mathrm{C} - \mathrm{H} + \mathrm{N} + \mathrm{P}}{2} \\ \mathrm{AI_{MOD}} = \dfrac{1 + C - \frac{O}{2} - S - \frac{N+P+H}{2}}{C - \frac{O}{2} - N - S - P} \end{cases}$$

It must be noted that if the computed $\mathrm{AI_{MOD}}$ value ends up being negative, the $\mathrm{AI_{MOD}}$ value is then set to 0 as $\mathrm{AI_{MOD}}$ cannot be negative (Koch and Dittmar, 2006, 2016). Formulas with $\mathrm{AI_{MOD}} \geq 1$ are considered abnormal for NOM and are therefore excluded (see table below).

After H-correction and the calculation of these metrics is completed, the first level of formula refinement begins. Formulas are first filtered using 18 elemental constraints summarized in the table below. Most of them were summarized by Stubbins et al. (2010). The formulas that fall outside of the defined elemental constraints are either chemically impossible to exist or are unlikely to occur as part of NOM. Hence, formulas that do not adhere to these 18 elemental constraints are rejected from the formula list. However, it must be noted that these constraints may need adjustment for specific datasets and projects. The recommended values in the table below are only acceptable for non-problematic NOM samples and in our experience, we have had to adjust some of them to better fit specific project needs. The values for constraints 1 – 15 are changeable and are defined in the **FTMS_ConfigurationAssignment**, lines 77-85 (**Slide 92**).

	Elemental Constraint	**Description**	**Recommended value**	**Reference**
1	Minimum O/C	No minimum number of O in a molecule	0	Stubbins et al. (2010)
2	Maximum O/C	No more than 1 oxygen per 6 carbons	1.2	Stubbins et al. (2010)
3	O-C	At least 2 more carbons than oxygens	2	Stubbins et al. (2010)
4	Minimum H/C	Minimum of 1 hydrogen per 3 carbons	0.33	Stubbins et al. (2010)
5	Maximum H/C	Maximum of 9 hydrogens per 4 carbons	2.25	Stubbins et al. (2010)
6	Minimum N/C	No minimum number of N in a molecule	0	Stubbins et al. (2010)
7	Maximum N/C	Maximum of 1 nitrogen per 2 carbons	0.5[7]	Stubbins et al. (2010)
8	Minimum S/C	No minimum number of S in a molecule	0	Stubbins et al. (2010)
9	Maximum S/C	Maximum of 1 sulfur per 5 carbons	0.2	Stubbins et al. (2010)
10	Minimum P/C	No minimum number of P in a molecule	0	Stubbins et al. (2010)
11	Maximum P/C	Maximum of 1 phosphorus per 10 carbons	0.1	Stubbins et al. (2010)

[7] While the recommended value per Stubbins et al. (2010) is 0.5, for some of our research involving samples of high proteinaceous content, we had to increase this value to 1 to allow the assignment of some unusual N-rich molecules.

12	Minimum O/P	At least 3 oxygens per 1 phosphorus	3[8]	
13	Maximum O/P	No limitations on maximum oxygens relative to P	Inf[9]	
14	Minimum DBE	DBE must be a positive number	0[10]	McLafferty and Turecek (1993)
15	Maximum DBE	Molecules with DBE > 50 will be very large and likely not be soluble nor ionizable by ESI	50	Wozniak et al. (2020)
16	Maximum AI_{MOD}	AI_{MOD} ranges from 0 – 1	1	Koch and Dittmar (2006), Koch and Dittmar (2016)
17	Maximum K+Na	Only singly-charged peaks exist in the mass list. Therefore, multiply-charged formulas (e.g., $[M+K+Na]^{2+}$) must be removed.	1	
18	Unionizable molecules	See below.	n/a	

The 18th elemental constraint refines molecules based on their ionizability in electrospray ionization. This 18th criterion considers the ionization mode (defined in the **FTMS_ConfigurationAssignment**, line 54, variable: *Ionization_Mode*, **Slide 91**) and the electronegativity of the assigned heteroelement (defined in the **FTMS_ConfigurationAssignment**, line 55, variable: *Heteroelement_Halogen*, **Slide 91**). In negative ionization mode, only molecules having at least one oxygen, one sulfur, or one halogen can ionize. Therefore, formulas of elemental composition of CH, CHN, CHP are rejected from the formula lists. If the assigned heteroelement is not a halogen (or another highly electronegative element enhancing ionization), CHE formulas (e.g., CHHg, CHFe) are rejected as well. In positive ionization mode, only molecules having at least one oxygen, one sulfur, or one nitrogen can ionize. Therefore, formulas of elemental composition of CH and CHP are removed. The presence of a halogen here is irrelevant and therefore, CHE formulas (CHCl, CHBr, CHFe, CHHg) are rejected as well. It must be noted that these rules are broad and may not be applicable to all samples/datasets and thus, the values for the 18 constraints in the table above are only recommended. For example, P-containing formulas with no oxygen (CHP) will likely be not ionizable in either negative or positive mode, but knowledge about such compounds in NOM is highly limited. Therefore, users are urged to evaluate these different elemental constraints and tailor the codes to their samples and datasets. The 18th elemental constraint is summarized below:

$$\text{If } \mathit{Ionization\_Mode} = \mathit{'negative'} \text{ and } \mathit{Heteroelement\_Halogen} = \mathit{'true'}\text{: Reject formulas with } O = 0\ \&\ S = 0\ \&\ E = 0\ (\text{CH, CHN, and CHP})$$

[8] This rule assumes that the molecular phosphorus in NOM is found exclusively as PO_3 groups (Sleighter and Hatcher, 2008; Maizel and Remucal, 2017; Kurek et al., 2021). Limited knowledge about the speciation of organic P in NOM exists (Reemtsma, 2009). Furthermore, organic P exists at such low concentrations (Cooper et al., 2005) that special isolation/concentration procedures are needed for its evaluation.

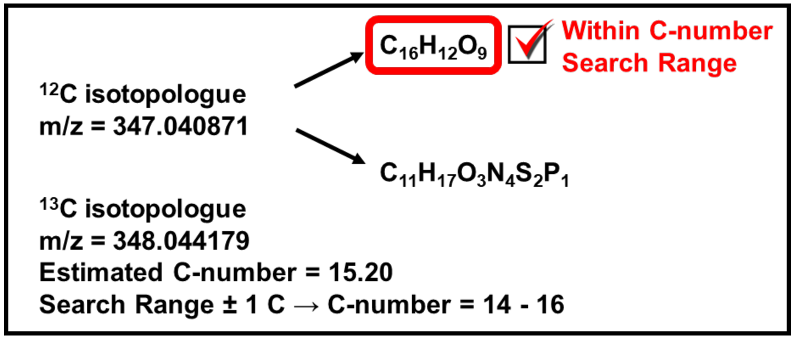
[9] The notation for “infinity” in MATLAB is *Inf*.

[10] Please note that DBE must be zero or a positive whole number (McLafferty and Turacek 1993). While this requirement is not explicitly listed in the Elemental Constraints in the **FTMS_RefinementFormulas** code, the formula assignment algorithm in **FTMS_FormulaAssignment** uses this rule to generate molecules with valid (whole) DBE values resulting in formulas adhering to the nitrogen rule (McLafferty and Turacek 1993).

$$\text{If } \mathit{Ionization\_Mode} = \mathit{'negative'} \text{ and } \mathit{Heteroelement\_Halogen} = \mathit{'false'}\text{: Reject formulas with } O = 0 \;\&\; S = 0 \text{ (CH, CHN, CHP, and CHE)}$$

$$\text{If } \mathit{Ionization\_Mode} = \mathit{'positive'}\text{: Reject formulas with } O = 0 \;\&\; S = 0 \;\&\; N = 0 \text{ (CH, CHP, and CHE)}$$

After the formula list is refined using these constraints, the list of refined formulas and list of rejected formulas are output to the Excel file "_Refinement_Processing.xlsx" (here, "Sample 1_Refinement_Processing.xlsx") in the sheets "Refined with Elem.Constraints" and "Rejected with Elem.Constraints" (**Slide 106**). It must be noted that the formula lists in these sheets are formatted differently than in "Sheet1". This new format includes the type of molecular formula (CHO, CHON, etc. as shown in Column E) as well as the calculated O/C, H/C, DBE, DBE/C, and AI_{MOD}. This will be the standard formula list format for TEnvR.


$C_{16}H_{12}O_9$
☑
Within C-number Search Range
^{12}C isotopologue
m/z = 347.040871
$C_{11}H_{17}O_3N_4S_2P_1$
^{13}C isotopologue
m/z = 348.044179
Estimated C-number = 15.20
Search Range ± 1 C → C-number = 14 - 16

The third stage of the formula refinement (*%% Stage 3: Formula Refinement using Isotopic Filter and selection of Unique Formulas*) is focused on selecting Formulas of High Confidence. These formulas are most likely to match the elemental composition of the actual molecule that was ionized to produce the measured ion peak. The Formulas of High Confidence will be critical for the following formula refinement in the later stages of the code. The first approach for selecting Formulas of High Confidence is by using the Isotopic Filter capability of this code. This filter evaluates the assigned number of carbons in the molecules and compares it to the estimated number of carbons using the $^{13}C/^{12}C$ magnitude ratio. If any of the assigned formulas' C-number is close to the estimated C-number, the code will select that particular molecular formula and will label it as a Formula of High Confidence[11]. This is conceptually explained on the figure above. This filter is enabled or disabled using in the **FTMS_ConfigurationAssignment**, line 90, variable: *Filter_Isotopes*, **Slide 92**). By default, the Isotopic Filter looks for a carbon match within a specified range. The default range is two carbons (± 1 C), but this range can be modified by the user in the **FTMS_ConfigurationAssignment**, line 91, variable: *Filter_Isotopes_Range*, **Slide 92**). This range will also need adjustment if instruments of different resolving power are used, as the complete resolution and precise measurement of ^{13}C isotopologues here is critical. The ^{13}C isotopologue peak must have a high S/N in order to be used in reliably predicting the carbon number (Koch et al., 2007). Carbon-estimates only from ^{13}C isotopologue peaks with S/N ratio ≥ 25 are used in the isotopic filter. The S/N ratio threshold of 25 is used as an example here and can be modified by the user in the **FTMS_ConfigurationAssignment**, line 92, variable: *Filter_Isotopes_SNthreshold*, **Slide 92**). As this threshold is likely instrument specific, we recommend analyzing a standardized sample, such as SRFA, which has been studied for years and has its molecules confidently assigned (Stenson et al., 2003; Herzsprung et al., 2016; Hawkes et al., 2020), to develop such threshold as described by Koch et al. (2007). Formulas selected using the Isotopic Filter are listed in a newly created sheet "Isotopic Filter" in the "_Refinement_Processing.xlsx" file (here, "Sample 1_Refinement_Processing.xlsx") (**Slide 107**). In our experience, we usually do not use this filter as the estimated C-numbers from the $^{13}C/^{12}C$ magnitude ratios can sometimes be unreliable, as discussed previously (Mitchell and Smith, 1995; Stenson et al., 2003; Koch et al., 2007). However, usage of

[11] If multiple formulas contain a number of carbons within the search range, the code will accept all matching formulas into the pool of Formulas of High Confidence.

this isotopic filter can be very beneficial in validating molecular formulas (Wagner et al., 2015a; Khatami et al., 2019; Merder et al., 2020), and users are urged to explore this capability for their samples.

The second approach for selecting Formulas of High Confidence is to evaluate their ambiguity and identify only unambiguously assigned (unique) molecular formulas. The second filter of this code, the Uniqueness Filter, selects formulas with unique index values and they are separated out. We most commonly use this filter and in our experience the formulas that are assigned unambiguously are representative of the whole sample. The Uniqueness Filter is enabled or disabled in the **FTMS_ConfigurationAssignment** (line 95, variable *Filter_Unique*, **Slide 92**). Formulas selected using the Uniqueness Filter are listed in a newly created sheet "Unique" in the "_Refinement_Processing.xlsx" file (here, "Sample 1_Refinement_Processing.xlsx") (**Slide 108**). The **FTMS_RefinementFormulas** code requires that at least one of the two filters, or both, be enabled. It is critical that some Formulas of High Confidence are selected prior the next refinement step.

The next stage of the code *(%% Stage 4: Formula refinement using KMD Filter*) refines the rest of the molecular formulas using Kendrick Mass Defect series analysis (Kendrick, 1963; Kujawinski and Behn, 2006; Koch et al., 2007). This filter is enabled or disabled in the **FTMS_ConfigurationAssignment** (line 98, variable *Filter_KMD*) using *'true'* or *'false'* (**Slide 92**). First, the code identifies which formulas have to be refined. It takes all NOM-like molecular formulas that were sorted using elemental constraints and picks formulas that are not Formulas of High Confidence. The selected formulas are the Refinable Formulas that the KMD filter will work on (**Slide 104**). The KMD filter establishes KMD series using the Formulas of High Confidence. Formulas that sit in these series have formulas with the same KMD value within the user-defined precision in the **FTMS_ConfigurationAssignment** (line 50, variable *Precision*, **Slide 91**). Series are established using user-defined molecular building blocks in the **FTMS_ConfigurationAssignment** (line 99, variable *Filter_KMD_Series*, **Slide 92**):

```
Filter_KMD_Series = double([14/14.015650, 2/2.015650, 44/43.989829, ...
        30/30.010565, 44/44.026215, 16/15.994915, 18/18.010565, 17/17.026549, ...
        36/35.976678]);
```

These ratios correspond to the nominal mass (rounded value to the closest whole number) over the whole mass of the following building blocks: CH_2, H_2, COO, CH_2O, C_2H_4O, O, H_2O, NH_3, HCl. Users are welcome to adjust this parameter (*Filter_KMD_Series*) for their needs. Another critical parameter is that the user may define the minimum number of points on a KMD series in order for the series to be validly used for a formula refinement. We usually require a minimum of 3 points to exist on the series using Formulas of High Confidence. This minimum threshold can be modified in the **FTMS_ConfigurationAssignment** (line 102, variable *Filter_KMD_Series_Threshold*, **Slide 92**). Once KMD series with 3 or more points within them are established, the formulas in the pool of Refinable Formulas are tested to determine if they have the same KMD value (within the user-defined precision, variable *Precision*) as the KMD values of the established series. Refinable Formulas that fall into any of the series (i.e., are the 4th or further data point) are extracted from the Refinable Formulas pool and moved into the matrix of Formulas of High Confidence. Then, new KMD series are created using the expanded pool of Formulas of High Confidence. The new KMD series are then utilized to again search all remaining Refinable Formulas.

This is iterative algorithm is conceptually explained on **Slide 109**. The Formulas of High Confidence (panel A) are used to build KMD series (panel B). Then, the Refinable Formulas (panel C) are checked to see if any of them fall within these KMD series (panel D). Formulas that align (24 red dots son panel D) are then moved to the pool of Formulas of High Confidence (panel E). Then, new KMD series (panel F) are built using the new pool of Formulas of High Confidence (panel E). It can be seen that there are 3 new series (panel F) once new Formulas of High Confidence are added. These new KMD series (panel F) are then used to search the remaining Refinable Formulas (24 formulas from panel C are now removed from panel G). It can be seen that 5 new formulas are identified (panel I) and added to Formulas of High Confidence (panel J). Again, new KMD series (panel K) are built, and an additional 4 new series are identified once new Formulas of High Confidence are added. These new KMD series (panel K) are then used to search the remaining Refinable Formulas (an additional five formulas are missing in panel L, in comparison to panel G). It can be seen that 3 new formulas are identified to be Formulas of High Confidence (panel M) and are added to the new pool of Formulas of High Confidence (panel N). Again, new KMD series (panel O) are built, but in this case, no additional series are identified. This cycle of creating KMD series using Formulas of High Confidence, searching the Refinable Formulas for formulas of matching KMD values, and moving Refinable Formulas into the pool of Formulas of High Confidence is repeated multiple times until the iterative algorithm reaches a termination point.

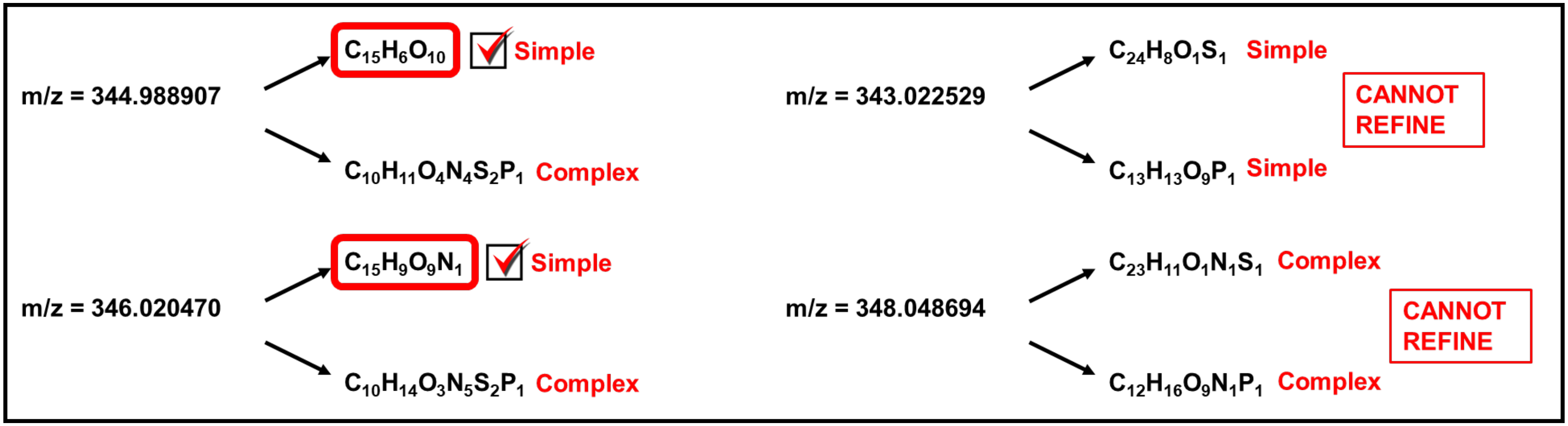
The termination point can occur in two different cases. In the first case (**Slide 109**), the algorithm will terminate if a new pool of Formulas of High Confidence (panel N) does not produce any new KMD series (panel O) relative to previous set of KMD series (panel K) using the previous pool of Formulas of High Confidence (panel J). In the second case (**Slide 110**), a new KMD series is produced (panel O), but the remaining Refinable Formulas (an additional three formulas are missing between panel L and panel P) do not give any formulas aligning within the new KMD series (panel Q).

It must be noted that the code checks if the formulas in each KMD series (i.e., formulas with the same mass defect within the user-defined precision in the **FTMS_ConfigurationAssignment**, line 50, variable *Precision*, **Slide 91**) are of the same elemental composition (CHO, CHON, CHOS, etc.). For example, CHO and CHON formulas cannot sit within the same KMD series but may coincidentally have the same KMD value for a certain building block. Each KMD series must be constructed of formulas of the same type (e.g., CHO only, CHON only). This additional step enhances the validity of the formulas refined using the KMD Filter and strengthens our confidence in the formulas selected by it.

If the user desires to perform the KMD search only once, they can disable the unlimited iterations feature in the in the **FTMS_ConfigurationAssignment** (line 103, variable *Filter_KMD_Unlimited_Iterations*, **Slide 92**). The pool of Formulas of High Confidence at the end of this filter is comprised of the formulas selected using Isotope and Uniqueness and some fraction of the Refinable Formulas that was selected using the KMD filter. This final pool of Formulas of High Confidence is exported in a newly created sheet “Refined w KMD” in the “_Refinement_Processing.xlsx” file (here, “Sample 1_Refinement_Processing.xlsx”) (**Slide 111**).

At the next stage of the formula refinement code (labeled as the “Middle Stage”), the code will compare the new pool of Formulas of High Confidence (found in the “Refined w KMD” sheet) with the remaining formulas in the Refinable Formulas pool. For each peak (i.e., each m/z value) in the Formulas of High Confidence, the code will identify formulas that are assigned to the same m/z in the Refined Formulas pool. As these formulas are ambiguous formulas that could not be placed into the Formulas of High Confidence pool using the Isotope, Unique, or KMD filters, they are assignments that are very unlikely to be the real molecule that was ionized and other formulas exist for that m/z value that are much more probable (of higher confidence). Thus, these ambiguous formulas are rejected and exported to a newly created sheet “Rejected at Middle Stage” (**Slide 111**). Any remaining Refinable Formulas that were not rejected are then concatenated with the Formulas of High Confidence, and this new matrix of data is exported in a newly created sheet “Middle Stage” (**Slide 111**). This sheet contains molecular formulas that are still possible assignments, and as such, this sheet will still contain many ambiguous formulas.

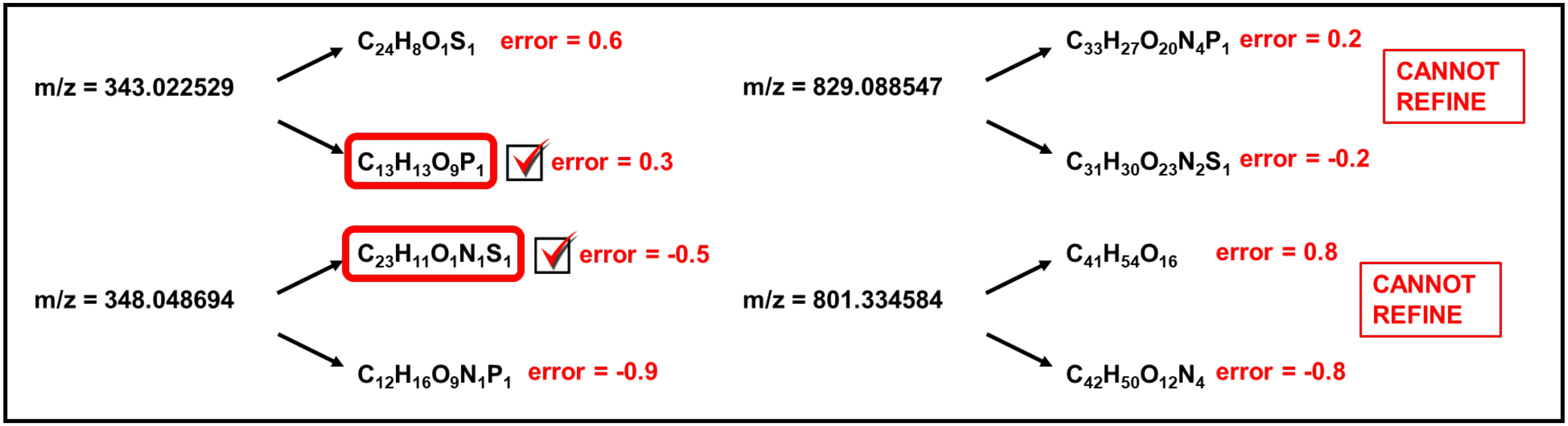
The next stage of the code (*%% Stage 5: Formula refinement using Compositional Filter*) refines the formula list based on the elemental composition. Formulas of CHO, CHON, CHOS, CHOP composition are labeled as “simple” versus other formulas containing more than one non-CHO element that are labeled “complex” (CHONS, CHONP, CHONSP, etc.). The code evaluates any remaining ambiguous assignments and selects the simple formula and rejects the complex one, as molecular formulas of lower complexity are more biogeochemically feasible (Koch et al., 2007). This is conceptually explained in the figure below.



In the two examples shown on the left side of the figure above, the code can clearly identify that the two peaks (m/z = 344.988907 and m/z = 346.020470) are assigned ambiguously. The code selects the formulas that are of simple composition (CHO, CHON, CHOS, CHOP). Complex formulas are rejected. This filter is unable to refine ambiguous formulas that are both simple or both complex, as shown on the right side of the figure. Further refinement is necessary for these cases. The Compositional Filter is enabled or disabled in the in the **FTMS_ConfigurationAssignment** (line 106, variable *Filter_Composition*) using *‘true’* or *‘false’* (**Slide 92**). If enabled, the code will export the refined formula list into a newly created sheet “Refined w Compositional Filter” in the “_Refinement_Processing.xlsx” file (**Slide 112**). Rejected formulas are also exported in a new sheet (“Rejected w Compositional Filter”) (**Slide 112**).

The next stage of the code (*%% Stage 6: Formula refinement using Error Filter)* refines the formula list based on the assignment error (difference between the m/z value (electron-corrected) and the exact mass of the assigned molecular formula). Formulas with an assignment error closer to 0 are more probable assignments than formulas with higher assignment errors. Formulas with the same error (absolute values!) cannot be further refined. This is conceptually explained in the figure below.

In the two examples shown on the left side of the figure below, the code can clearly identify that the two peaks (m/z = 343.022529 and m/z = 348.048694) are assigned ambiguously. These same ambiguous assignments could not be refined using the compositional filter earlier (see previous figure). The code can successfully choose a molecular assignment of lower error for these two peaks. On the right side of the figure below, two peaks are shown with ambiguous assignments that cannot be refined further (they have the same error and same complex type of elemental composition). The Error Filter is enabled or disabled in the **FTMS_ConfigurationAssignment** (line 109, variable *Filter_Error*) using *'true'* or *'false'* (**Slide 92**). If enabled, the code will export the refined formula list into a newly created sheet "Refined w Error Filter" in the "_Refinement_Processing.xlsx" file (**Slide 113**). Rejected formulas are also exported in a new sheet ("Rejected w Error Filter") (**Slide 113**).


m/z = 343.022529
$C_{24}H_8O_1S_1$ error = 0.6
$C_{13}H_{13}O_9P_1$ ☑ error = 0.3
m/z = 348.048694
$C_{23}H_{11}O_1N_1S_1$ ☑ error = -0.5
$C_{12}H_{16}O_9N_1P_1$ error = -0.9
m/z = 829.088547
$C_{33}H_{27}O_{20}N_4P_1$ error = 0.2
$C_{31}H_{30}O_{23}N_2S_1$ error = -0.2
CANNOT REFINE
m/z = 801.334584
$C_{41}H_{54}O_{16}$ error = 0.8
$C_{42}H_{50}O_{12}N_4$ error = -0.8
CANNOT REFINE

The next stage of the code (*%% Stage 7: Final Refinement*) is for finalizing the refinement process, to ensure that only one formula exists for each m/z value. The code first identifies any remaining ambiguous assignments and rejects them. For example, the two formulas per each of the peaks at m/z 829.088547 and 801.334584 shown on the right side of the figure above will be rejected from the list and these two peaks will be left unassigned. Choosing one or the other of these molecular formulas manually by the user would require manual inspection, but this will also introduce user-induced bias into the dataset. Should users decide to manually pick what they believe to be the "correct" formula is their own prerogative.

Once any remaining ambiguous assignments are rejected, the code evaluates if there are two or more peaks assigned with the same molecular formula. Such cases occur rarely but several reasons explained below can lead to their existence. For any incidence of multiple peaks assigned with the same exact formula, the code evaluates the differences in their m/z values, peak magnitudes, and assignment errors to determine how to proceed:

→ If there are two peaks (i.e., two m/z values with different magnitudes) with the same molecular formula assigned:

- (Common) If the two peaks have a larger difference in their m/z values (e.g., higher than 10), then two different ion species are detected per molecule. Commonly, either a sodiated ($[M+Na]^+$) or potassiated ($[M+K]^+$) ion is detected in addition to the hydrogenated ion ($[M+H]^+$). In more uncommon cases, both the sodiated ($[M+Na]^+$) or potassiated ($[M+K]^+$) ion species are detected and the hydrogenated ion ($[M+H]^+$) is not detected. Remember that if a peak is detected as two different ion species (e.g., detected as both $[M+H]^+$ and $[M+Na]^+$), the two ions will have two different m/z values (e.g., differing by ~23) but their corresponding ExactMass will be the same as the ExactMass corresponds to the neutral molecule. If a peak was detected as more than one species, the code will select the peak with the lower m/z value. Therefore, if both $[M+H]^+$ and $[M+Na]^+$ peaks are detected, the code will select the $[M+H]^+$ peak.
- (Rare) If the two peaks have similar m/z (within the user defined error, usually 1 ppm), their peak magnitudes differ by more than 20%, and the larger (higher magnitude) peak has the smaller error, this is a case when the smaller peak is a shoulder peak to the larger peak. While shoulder peaks are generally identified and manually removed during peak picking prior to the internal calibration of mass spectrum, shoulder peaks can be missed by the user. The code will retain the more abundant peak (higher magnitude, smaller error) and will reject the less abundant shoulder peak (smaller magnitude, higher error).
- (Moderately Rare) If the two peaks have similar m/z (within the user defined error, usually 1 ppm) and their magnitudes differ by less than 20%, this is a case of a split peak, which may be due to two poorly resolved ions. Peak splitting is also possible due to space-charge effects (Ledford et al., 1984), but if that was the case, most peaks in the spectrum (or most peaks at high m/z) would be split, and this would have been identified during the manual mass spectral inspection prior to internal calibration. In this situation, the code will select the peak with the smaller assignment error and reject the peak with the higher assignment error (peak magnitude is not considered).
- (Very Rare) If the two peaks have similar m/z (within the user defined error, usually 1 ppm), peak magnitudes differ by more than 20%, and the larger (higher magnitude) peak has the higher error, this is typically a case when there is a highly ionizable compound (a “spike”) that sticks up above the typical Gaussian-like mass spectrum of NOM. Often these peaks require an individual mass calibration or need to be assigned with a lower mass accuracy (up to 2 ppm instead of 1 ppm). In this situation, the code rejects both peaks and displays a warning to the user in the command window notifying that “Peaks with ExactMass = XXXXXXX need inspection and manual assignment!”. These peaks are then moved to the list of rejected peaks and if they are important, they may be manually assigned by the user and manually moved to the final list of formulas. We generally do not use these formulas further as we do not have high confidence in them.

→ If there are more than two peaks with the same molecular formula assigned:

- (Extremely Rare) In this situation, the three or more peaks with the same ExactMass are rejected and the code displays a warning to the user in the command window notifying that “ExactMass = XXXXXXX is assigned to more than two peaks – inspect spectrum!”. This may occur in positive ionization mode if $[M+H]^+$, $[M+Na]^+$, and $[M+K]^+$ were detected. The user will have to manually check out these abnormalities and manually extract the formula corresponding to the $[M+H]^+$ peak. This could also occur if electronic noise produces numerous peaks within a very small m/z window, in which case all formulas should be deleted (and those peaks should also be deleted from the original peak list). This type of occurrence is usually only confirmed when visually inspecting the mass spectra and seeing a disturbance of the baseline that looks abnormal. Generally, these abnormal frequency noise regions also occur in the blank and are thus typically removed at that stage of peak list refinement (during blank-subtraction).

After the final formula list is trimmed of any ambiguous formula assignments and of peaks assigned with the same formulas, the code exports the final formula assignment list into a newly created sheet "Final Refinement" in the "_Refinement_Processing.xlsx" file (**Slide 114**). Any rejected formulas are exported in a "Rejected w Final Refinement" sheet. The code also identifies which peaks of the refined peak list (that was used for formula assignment) were left unassigned and exports them into a newly created sheet "Not Assigned" in the "_Refinement_Processing.xlsx" file (**Slide 111**). Lastly, the code exports the final refined formula list into a separate Excel file named "_Final.xlsx" (here, "Sample 1_Final.xlsx") (**Slide 102**) that will be used as an example for the rest of the codes of TEnvR.

The next and last stage of the code (*%% Stage 8: Quality Control*) creates a quality control figure for the whole data processing pipeline. The figure is visualized by MATLAB in a maximized Figure window (**Slide 103**) and is also exported as a portable network graphic (PNG) picture file (named "FTMS Processing_Sample 1.png") (**Slide 102**). It is worth noting that if this figure is generated on a laptop, it may appear quite crowded and look slightly different from that shown in **Slide 103**, where the figure was generated on a larger monitor. Users are welcome to modify the code and adjust the figure parameters for different screen sizes and resolutions. The furthest left panel of the figure shows the van Krevelen (vK) diagram (Van Krevelen, 1950; Kim et al., 2003a) of the sample using the formulas after Final Refinement. This is a scatterplot of H/C vs O/C of the sample, and this particular vK diagram has its data points color-coded according to S/N (per the color bar to the right). This plot, along with the S/N histogram in the top middle panel can help assess the quality of the mass spectrum. The title of the S/N histogram will also list the center position, as well as the standard error of the fitted curve (μ ± σ). Abnormalities in these plots may indicate that the S/N threshold for peak picking must be increased. They can also be indicative of samples that were analyzed with poor instrumental conditions. The second histogram is of the assignment error. The title will list the center position, as well as the standard error of the fitted curve (μ ± σ). It is critical that this distribution is centered at ~zero. A large shift in the center position (more than 0.2 ppm) would indicate a significant misassignment of formulas, as seen in the example by Tolić et al. (2017). A histogram as such would indicate that there is an extra element that needs to be incorporated (e.g., a halogen), or the elemental criteria ($^{12}C_{5\text{-}\infty}$, $^{1}H_{5\text{-}100}$, $^{16}O_{1\text{-}30}$, $^{14}N_{0\text{-}5}$, $^{32}S_{0\text{-}4}$, $^{31}P_{0\text{-}2}$) or constraints (lines 77-85 in the **FTMS_ConfigurationAssignment**, **Slide 92**) need modification. The panel on the top right is assessing the comparability between assigned and estimated C-numbers. The goodness-of-fit (R^2) of the linear fit (dashed line shown in black), as well as the line equation, are given in the title, and a one-to-one line is also plotted (in red) for easier slope comparison. Generally, an R^2 value closer to 1 is targeted, though we are unable to provide recommendations for a specific R^2 threshold because, as mentioned earlier, the use of the estimated C-number from the $^{13}C/^{12}C$ ratio can be problematic when ^{13}C isotopologues of low S/N are used (Mitchell and Smith, 1995; Stenson et al., 2003; Koch et al., 2007).

The final and most critical part of this figure is the table in the lower right corner. It has all quality control data from the peak refinement stage (calculated by **FTMS_RefinementPeaks**). Additionally, the metrics for Unassignable and Assigned peaks are also listed. Their percentages are based on the number of peaks and the total magnitude of the Refined Peaks. Ergo, in the example with Sample 1, the number of Refined Peaks (6626) equals the number of Unassignable Peaks (152) + the number of Assignable Peaks (6474). The number-

and magnitude-based percentages of the Assigned Peaks must be above 80% (here, 97.71% and 99.02%, respectively). This validates the formula assignment and confirms that the assigned formulas represent the molecular composition of the analyzed sample. We have found that for less problematic NOM samples (e.g., riverine NOM), these percentages are often above 90% (number-based) and 95% (magnitude-based).

The final list of molecular formulas (in the "_Final.xlsx" file) can be further refined depending on the particular study. Refinement criteria vary per project and sometimes can be very conservative. For example, Khatami et al. (2019) only used formulas that had a corresponding ^{13}C isotopologue peak. Smith et al. (2016) used only formulas of CHO composition. Wozniak et al. (2020) only used formulas containing C, H, O, N and Cl (CHO, CHON, CHOCl, CHONCl) in their study. Wagner et al. (2019) used a less conservative approach but still performed further refinement by discarding the following formulas from their dataset: NSP (formulas containing N, S, and P), N_2S, N_3S, N_4S, N_2P, N_3P, N_4P, and NS_2. These examples highlight that there is no current consensus about formula refinement, and this is expected due to the known extreme molecular diversity of NOM. Further details in formula assignment and refinement can be found in previous publications on these topics (Koch and Dittmar, 2006; Koch et al., 2007; Sleighter and Hatcher, 2007; Reemtsma, 2009; Sleighter and Hatcher, 2011; Ohno and Ohno, 2013; Herzsprung et al., 2014; Herzsprung et al., 2016; Tolić et al., 2017; Qing-Long et al., 2020; Lu et al., 2021). In this tutorial, we cannot provide recommendations for which approach is best under all circumstances, and it is up to users of TEnvR to decide which formula refinement capabilities of TEnvR they should use and if any further manual refinements are needed. Future versions of this code will include additional filters for formula refinement.

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

- **FTMS_Process**

So far, the explained capabilities of TEnvR for processing FT-ICR-MS data have depended on the use of three separate codes (**FTMS_RefinementPeaks**, **FTMS_FormulaAssignment**, and **FTMS_RefinementFormulas**). This stepwise approach is very helpful for assigning problematic NOM samples (rainwater, aerosol, ice cores, marine oil snow, biochar leachates, etc.). However, this approach causes a discontinuity in the data processing routine and automation is challenging. For less problematic NOM samples, such as terrestrial (riverine, lake, swamp) and marine (estuarine, coastal, marine, deep ocean) NOM samples, as well as various soil extracts and their humic/fulvic fractions, the data processing does not require stepwise processing and detailed monitoring by the user. Automating these routines can allow for high efficiency when working with larger datasets. This will be exemplified later in this tutorial.

The **FTMS_Process** code loads the calibrated lists of a sample, a blank, and the **FTMS_ConfigurationAssignment** file. Prior to using this code, the user must define the elemental criteria (e.g., $^{12}C_{5-\infty}$, $^{1}H_{5-100}$, $^{16}O_{1-30}$, $^{14}N_{0-5}$, $^{32}S_{0-2}$, $^{31}P_{0-1}$), ionization mode, accuracy and precision, and other relevant formula assignment/refinement parameters inside the configuration code (lines 42-109, **Slides 91-92**). Then, the **FTMS_Process** code can be used as shown below. Here, we will employ it on Sample 2:

Slide 115: *FTMS_Process('Sample 2.txt','Blank PPL.txt',FTMS_ConfigurationAssignment)*

This code has the **FTMS_RefinementPeaks**, **FTMS_FormulaAssignment**, and **FTMS_RefinementFormulas** codes internally connected. Therefore, once the **FTMS_Process** code is executed using the command above, the sample and blank list are automatically processed using **FTMS_RefinementPeaks** as previously described. Then, the refined peak list (exported as "_Refinement.txt") is automatically re-loaded in MATLAB by the **FTMS_FormulaAssignment** code and formulas are assigned. The produced formula assignments are then directly refined as discussed previously using **FTMS_RefinementFormulas**. After the **FTMS_Process** code is employed, it will produce all files that were collectively produced for Sample 1 and described earlier (**Slide 115**), and a quality control figure is also visualized by MATLAB in a maximized Figure window (**Slide 116**).

In summary, TEnvR includes two data processing pipelines for ultrahigh resolution mass spectrometry data: a manual pipeline involving the use of **FTMS_RefinementPeaks**, **FTMS_RefinementFormulas**, and **FTMS_RefinementFormulas**; and an automated pipeline using **FTMS_Process**.

Please note that all codes below are going to use the **FTMS_ConfigurationToolbox** configuration file (**Slide 93**).

- **FTMS_Metrics**

Once molecular formulas are refined and the formula list is finalized, a variety of metrics can be computed that can serve as bulk descriptors for the molecular composition. These are various averages of metrics derived from the molecular formulas. For example, five of these metrics were already discussed above (H/C, O/C, DBE, DBE/C, and AI_{MOD}). Metrics can be calculated in two modes: number-based and magnitude-weighted. The number-based averages are solely based on the metric for each molecular formula. For example, for the H/C average ($\overline{H/C}$), it is calculated by taking the arithmetic mean of all formulas (total number of formulas = n) in the formula list:

$$\overline{H/C} = \frac{\sum_1^n H/C_i}{n}$$

For Sample 1 (**Slide 117**):

$$\overline{H/C} = \frac{\sum_1^n H/C_i}{n} = \frac{1.00 + 0.89 + 0.83 + 1.80 + 2.00 + \cdots}{6465} = 0.98$$

The magnitude-weighted averages take into account the spectral magnitude of each peak. They are calculated by first scaling ("weighing") the metric of interest to the peak magnitude. This is done by multiplying the metric with the relative magnitude of the formula. Here, relative magnitude of peaks is calculated by dividing each peak magnitude by the total spectral magnitude (i.e., the sum of all peaks). This magnitude normalization approach is known as total sum scaling (TSS) and is the approach employed by the codes in TEnvR. Other normalization approaches do exist and some may be more appropriate for certain datasets (Thompson et al., 2021). The limitation of TEnvR to strictly employ TSS normalization will be overcome in future versions of it. The calculation of the magnitude-weighted H/C average ($\overline{H/C_w}$) is shown below:

$$\overline{H/C_w} = \sum_{1}^{n} \left[H/C_i \times \frac{Magnitude_i}{\sum_1^n Magnitude} \right]_i$$

For Sample 1 (**Slide 118**):

$$\overline{H/C_w} = \left[1.00 \times \frac{3678250}{2.35 \times 10^{11}}\right] + \left[0.89 \times \frac{3750509}{2.35 \times 10^{11}}\right] + \left[0.83 \times \frac{5339945}{2.35 \times 10^{11}}\right] + \left[1.80 \times \frac{4030800}{2.35 \times 10^{11}}\right] + \left[2.00 \times \frac{5368680}{2.35 \times 10^{11}}\right] + \cdots = 0.93$$

The user must carefully select which type of average to use in their study. Number-based averages are useful when one wants to see how all molecules[12] in a sample contribute to the particular metric. This number-based metric, however, can be biased by formulas of low magnitude having vastly different values for the particular metric. Thus, magnitude-weighted averages may be used that will not be severely impacted by formulas of low magnitude. The resultant value can be more representative of the average metric (e.g., H/C) of the whole sample as peaks of minor magnitudes (and likely low S/N) will barely impact this metric. However, whenever spectral magnitudes are used, it must be considered that they are a product of ionization and not quantity (D'Andrilli et al., 2020), as ESI-FT-ICR-MS is not quantitative. Thus, magnitude-weighted metrics can be particularly biased to represent a particular group of highly ionizable compounds, and other compounds of poor ionizability will be underrepresented. It is particularly critical for cases when magnitude-weighted averages are to be used that all spectra are acquired with proper tuning and are reproducible (Hawkes et al., 2020). If spectra are acquired on different days, they must be comparable and reproducible in order for their spectral magnitudes, and therefore their magnitude-weighed metrics, are to be compared later.

The **FTMS_Metrics** code computes number-based averages, number-based standard deviations, as well as magnitude-weighted averages[13] of 31 parameters derived from molecular formulas:

→ Average number of elements: C, H, O, N, S, P, E

→ Elemental ratios: O/C, H/C, N/C, Cl/C, H/N, O/N, H/S, H/P, O/S, O/P, N/S, P/S, H/E, O/E, N/E

→ Average mass of the whole molecule (ExactMass)

→ Double-bond equivalencies (DBE), C-normalized DBE (DBE/C), H-normalized DBE (DBE/H), O-normalized DBE (DBE/O), and O-corrected DBE (DBE-O). DBE/C, DBE/H, and DBE/O are generally using in classifying molecular formulas as "carboxyl-rich alicyclic molecules" or CRAM (Hertkorn et al., 2006). The DBE-O measure has been found useful in classifying molecules based on their photo-reactivity in photochemical studies of NOM (Gonsior et al., 2009). DBE/C is a measure of aromaticity (Hockaday et al., 2006), as described above.

→ Modified Aromaticity Index (AI_{MOD}) – a more robust measure of aromaticity. It is based on the probability of having a high density of π-bonds in a molecule (Koch and Dittmar, 2006, 2016).

[12] All molecules that have been observed in the analytical window of the ESI-FT-ICR-MS.

[13] A standard deviation cannot be calculated for the metric when it is magnitude-weighted.

→ Number of Condensed Rings (Ring) – an estimation of the number of rings of condensed aromatic molecules. This metric is derived using a list of 193 polycyclic aromatic hydrocarbons (PAHs) with known structure, number of condensed rings, and molecular formula. The compounds in the list are of a varied number of rings (2-21), molecular weight (117-1167 g/mol), and nitrogen content (0-3). This list of PAHs is provided in the “FTMS_PAHs.xlsx” file located in the Supplementary Files folder of TEnvR. It was identified that the number of rings (known from structure) and DBE have a strong relationship (R^2=0.9805) that can be used to develop a calibration curve (**Slide 119**). This calibration can be used to estimate the number of rings for formulas in unknown samples. It must be noted that this calculation is valid only for compounds of highly condensed character (PAHs and ionizable condensed aromatic compounds). Thus, this calculation is done only for compounds with AI_{MOD} ≥ 0.67 (Koch and Dittmar, 2006, 2016) and C-number ≥ 15 (Osterholz et al., 2016).

$$\mathrm{Ring} = \frac{\mathrm{DBE} - 3.1374}{2.436}$$

→ Nominal oxidation state of carbon (NOSC) – the average oxidation state of all carbons in a molecule. This parameter can be a proxy for polarity, solubility, biogeochemical reactivity, and bioavailability of substances (Kroll et al., 2011; LaRowe and Van Cappellen, 2011). For example, it has been used to study the alteration of NOM by polyvalent cations (Riedel et al., 2012). The presence of halogens is critical in this computation. The user must indicate whether a halogen was additionally assigned in the **FTMS_ConfigurationToolbox** file (line 49, variable *Heteroelement_Halogen*, **Slide 93**).

$$\text{If } \mathit{Heteroelement\_Halogen} = \text{'}\mathit{true}\text{'}:\ \mathrm{NOSC} = 4 - \frac{4C + H - 3N - 2O - 2S + 5P - Cl}{C}$$

$$\text{If } \mathit{Heteroelement\_Halogen} = \text{'}\mathit{false}\text{'}:\ \mathrm{NOSC} = 4 - \frac{4C + H - 3N - 2O - 2S + 5P}{C}$$

It must be noted that heteroelement-specific averages and standard deviations (N, S, P, E, N/C, Cl/C, H/N, O/N, H/S, H/P, O/S, O/P, N/S, P/S, H/E, O/E, N/E) are calculated for formulas containing at least one heteroelement. For example, N/C is calculated as the average of N/C ratios for all formulas containing at least 1 nitrogen (CHON, CHONS, CHONP, etc.), excluding formulas without nitrogen (CHO, CHOS, etc.). See the example described below for phosphorus.

The presence of a heteroelement is also critical for the computations of DBE and AI_{MOD}. These metrics are not computed by the **FTMS_Metrics** code but are loaded from the “_Final.xlsx”. Thus, the presence of heteroelement is accounted for in these computations as described earlier for the **FTMS_RefinementFormulas**.

To use the **FTMS_Metrics** code, keep the TestData_FTICRMS folder as your Current Folder:

Slide 120: *FTMS_Metrics('Sample 1_Final.xlsx')*

A new Excel file is created labeled "_Metrics.xlsx" (here, "Sample 1_Final_Metrics.xlsx") that will contain the formula list that was used in the code (everything from "Sample 1_Final.xlsx" is copied) with all metrics listed to the right (**Slides 121** and **122**). Here, because the maximum value of P atoms allowed is 1, any formulas containing P will only have 1 phosphorus atom. As such, the average number of P in the formulas including P is 1 (and the standard deviation is 0).

We recognize that other metrics exist (Merder et al., 2020) and the **FTMS_Metrics** code is not fully inclusive. Future versions of the code will aim to expand the list of metrics. Users are welcome to share feedback with the corresponding author if they have a request for any additional metrics to be coded for future versions.

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

- **FTMS_CompoundClass**

Molecular formulas, depending on their composition, can be categorized into different molecular categories referred to as "compound classes". These groups are exclusive (i.e., compound classes do not overlap, see **Slide 123**) and therefore this is a crude approach to classify observed molecular formulas based on assumed structure and biopolymeric source. This approach of separating the vK space into succinct regions has been incredibly useful in studying NOM biogeochemistry (Kim et al., 2003a; Koch et al., 2005; Kujawinski and Behn, 2006; Hockaday et al., 2009; Schmidt et al., 2009; Ohno et al., 2010; Santl-Temkiv et al., 2013; Ohno et al., 2014; Sleighter et al., 2014b; Ikeya et al., 2015; Sleighter et al., 2015; Ohno et al., 2016; Antony et al., 2017; He et al., 2018; Ohno et al., 2018; He et al., 2019). However, the compound class assignments to molecules are not unambiguous, which is an issue that has been discussed in detail by Sleighter and Hatcher (2008). Fundamentally, the reason for this is the high number of structural isomers that are possible for molecular formulas (Wieland et al., 1996; Hertkorn et al., 2007). Additionally, the stoichiometric boundaries used to define the different compound classes are highly variable among studies. We have provided a summary of several different compound class categorizations found in the literature in the "FTMS_CompoundClasses.pdf" document (**Slide 124**) found in the Supplementary Files folder of TEnvR. In a nutshell, the compound class categorization approach can be incredibly useful for a particular dataset and project, but it may also be detrimental to the study and create artificial trends. Thus, we advise that the employment of compound classes is done carefully and possibly in conjunction with a validation using another analytical technique. For example, findings using compound classes have been correlated with observations by fragmentation mass spectrometry (Stenson et al., 2003), NMR (Kim et al., 2003b; Kim et al., 2004; Hertkorn et al., 2006; Hertkorn et al., 2013), or UV-VIS and fluorescence spectroscopies (Stubbins et al., 2014; Goranov et al., in review), but in other cases, lack of such correlations has been also observed (Wagner et al., 2017; Wozniak et al., 2020).

Based on the different categorizations of compound classes in the literature (summarized in "FTMS_CompoundClasses.pdf", **Slide 124**), we have developed a categorization that is inclusive of most, as shown in the table below. Users are welcome to further edit the code and adjust the stoichiometric boundaries per their needs. These are also visualized on **Slide 123**.

Compound Class	Stoichiometric Boundaries
E-containing	E > 0
Condensed Aromatic Compounds (ConAC)	$AI_{MOD} \geq 0.67$ & $C \geq 15$
Small Condensed Aromatics (SCA)	$AI_{MOD} \geq 0.67$ & $C < 15$
Lignin	$AI_{MOD} < 0.67$ & $H/C < 1.5$ & $O/C < 0.67$ & $O/C > 0.1$
Tannin	$H/C < 1.5$ & $O/C \geq 0.67$ & $AI_{MOD} < 0.67$
Sulfonic Acid (SA)	$H/C \geq 2$ & $O/C < 0.67$ & $S > 0$
Amino Sugars (AS)	$H/C \geq 1.5$ & $O/C \geq 0.55$ & $N > 0$
Sugar	$H/C \geq 1.5$ & $O/C \geq 0.67$
Protein	$H/C \geq 1.5$ & $O/C < 0.55$ & $N > 0$
Fatty Acid (FA)	$H/C \geq 2$ & $O/C < 0.67$ & $N = 0$
Lipid (includes saturated aliphatics)	$H/C < 2$ & $H/C \geq 1.5$ & $O/C < 0.67$ & $N = 0$
Unsaturates	$H/C < 1.5$ & $AI_{MOD} > 0$ & $AI_{MOD} < 0.67$ & $O/C < 0.1$
Extra	Formulas that do not fit in any of the boundaries above

To use the **FTMS_CompoundClass** code, keep the TestData_FTICRMS folder as your Current Folder:

Slide 125: *FTMS_CompoundClass('Sample 1_Final.xlsx')*

The code first produces two figures. The first figure has three panels (**Slide 126**). The panel on the top left shows the van Krevelen (vK) diagram (Van Krevelen, 1950; Kim et al., 2003a) of the sample (a scatterplot of H/C vs O/C). The bottom left plot shows the H/C vs. Molecular Weight plot of the sample, which is also incredibly useful in interpreting ultrahigh resolution mass spectrometry data (Gonsior et al., 2018; Powers et al., 2019; Valle et al., 2020). The right plot shows a 3D vK diagram with relative magnitude (TSS-normalized (Thompson et al., 2021) and in $\log_{10}$-units), allowing for visualization of the spectral distribution of the ionized molecules. The figure panels of MATLAB are interactive and the user is able to zoom in/out, copy a sub-panel, change the view angle of a sub-plot, and click on an individual point to see its x, y, and z values. Please see the following website for instructions of the interactive capabilities of MATLAB: https://www.mathworks.com/help/matlab/creating_plots/interactively-explore-plotted-data.html. We show an example of how to rotate a 3D plot (**Slide 127**). The second figure shows vK diagrams of formulas categorized in two different ways (**Slide 128**). On the left panel, formulas are plotted with different markers and colors depending on their formula type (CHO, CHON, CHOS, CHOP, etc.). On the right panel, formulas are plotted with different markers and colors depending on their compound class as described in the table above (ConAC, SCA, Lignin, Tannin, etc.). These figures are also saved in your Current Folder as “vK_all_Sample 1_Final.png” and “vK_groups_Sample 1_Final.png” (**Slide 125**). The code also exports a new Excel file named labeled “_CompoundClass.xlsx” (here, “Sample 1_Final_CompoundClass.xlsx”) (**Slide 125**).

The first sheet (named "Sheet1") of the Excel file contains the formula list that was used in the code (here, "Sample 1_Final.xlsx") (**Slide 129**). Two more columns are added: Relative Magnitude (column T, labeled RelMagn) and Compound Class category (column U, labeled Class). To the right, there are numerous statistics calculated: 1) number of formulas and total spectral magnitude; 2) statistics for different formula types (CHO, CHON, CHOS, CHOP, etc.) including number of formulas and number- and magnitude-based percentages; and 3) statistics for different compound classes (ConAC, SCA, Lignin, Tannin, etc.) including number of formulas and number- and magnitude-based percentages.

This new file (here, "Sample 1_Final_CompoundClass.xlsx") also has a new sheet for each different formula type (CHO, CHON, CHOS, CHOP, etc.) and compound class (ConAC, SCA, Lignin, Tannin, etc.). The **FTMS_CompoundClass** also has the **FTMS_Metrics** code incorporated, and the latter is employed on all sheets. Using the **FTMS_CompoundClass** code on is a convenient way for researchers to quickly extract formulas of interest (for example, CHO as shown on **Slide 130**) and individually plot them or look into their metrics.

The **FTMS_CompoundClass** code provides two incredibly useful figures and a wealth of information about the molecular formulas that were loaded. This code is our first immediate approach to evaluate molecular formulas following data processing, as this code will immediately allow for data visualization and may allow for observation of immediate trends.

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

- **FTMS_Automation**

While the codes listed above can be individually applied on samples, this is inconvenient for datasets containing a large number of samples. Thus, these codes are incorporated into an automation routine. The **FTMS_Automation** script is capable of automatically refining peaks, assigning formulas, refining formulas (using **FTMS_Process**), as well as performing **FTMS_Metrics** and **FTMS_CompoundClass** on large datasets containing numerous samples. This code is controlled by the user in a similar way as the **EEM_Process_Generic** and **EEM_Process_Aqualog** are executed – section by section.

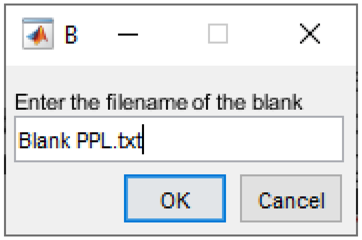
Prior to using the **FTMS_Automation** code, please delete all files that were created from the examples above (**Slide 131**). When **FTMS_Automation** is applied, only the raw sample data (samples + blank file) must be located in the Current Folder (i.e., Current Folder should be as shown on **Slide 89**).

Prior to running **FTMS_Automation**, it is important to note that the time required to process numerous samples can vary significantly with computer power. It is crucial to set the power settings on your computer or laptop so that the computer and screen never go to sleep and that the screen saver is disabled, to ensure continuous processing. If the computer goes to sleep, MATLAB will stop running (but will resume once woken).

Slide 132: *edit FTMS_Automation*

The **FTMS_Automation** code consists of several different sections for automating different FTMS codes. The first section *(%% Section 1: Automation for FTMS_Process*) is for automated peak refinement, formula assignment, and formula refinement using **FTMS_Process** (**Slide 132**). Click on the section and then click *Run Section*. A dialog window will pop up to request the name of the blank file; please input the full filename of the blank (Blank PPL.txt), including the extension, with no apostrophes as shown to the right.

Blank PPL.txt



Enter the filename of the blank

Blank PPL.txt

The code will then perform **FTMS_Process** on all 20 samples of the provided dataset and use the specified blank file. Once the code is finished, the current folder will be populated with the 6 files from **FTMS_Process** for each sample (therefore, 120 new files will be created) (**Slide 133**). The code outputs messages when each sub-code of **FTMS_Process** is employed (peak refinement using **FTMS_RefinementPeaks**, formula assignment, formula refinement using **FTMS_RefinementFormulas**) and also lists the time it took for each sub-code to run. Once the automated processing is done, the code will output a completion message in the Command Window: *Automated processing with FTMS_Process – Complete!* Again, the time required to run this step will vary by computer, and it is important to make sure that your computer and screen do not go to sleep while processing codes.

While we do not show an example of this in the tutorial slides, users can perform the same automation routine to process the example data obtained in positive-mode ESI. To do this, set TEnvR\TestData_FTICRMS\PositiveMode as your Current Folder (**Slide 90**). In **FTMS_ConfigurationAssignment**, change the *Ionization_Mode* variable (line 54) to *'positive'*, and allow for assignment of sodium and potassium (change *K_max* and *Na_max* variables on line 68 to *1*). Employ Section 1 of the FTMS_Automation code as described above and shown on **Slide 132**. The blank for this dataset is labeled “Blank_pos.txt”.

Once all samples are assigned, we can assess them using the previously described **FTMS_CompoundClass** and **FTMS_Metrics** codes in an automated fashion.

The next part of the **FTMS_Automation** code is to apply **FTMS_CompoundClass** on all samples. Scroll down to the next section *(%% Section 2 Automation for data exploration using FTMS_CompoundClass*). Click on the section and then click *Run Section* (**Slide 134**). The code will only load Excel files ending with “_Final.xlsx”; thus the other files of this folder will not obstruct the execution of the code. Once the automated processing of the 20 samples is finished, 40 figures (2 figures from **Slides 126** and **128** x 20 samples) will be produced (**Slide 135**) that can be then easily opened (externally via Explorer) and clicked through to observe the molecular composition of the 20 samples. An Excel file (“_CompoundClass.xlsx”) described previously will be output for each sample (**Slide 134**). Finally, the code will output an Excel file named “FTMS CompoundClass Master Report.xlsx” (**Slide 134**) that will contain a summary of all Compound

Class statistics for each sample (**Slide 136**). For this step of the tutorial, expect this code section to take anywhere from 20-60 min to complete.

The next part of the **FTMS_Automation** code is to apply **FTMS_Metrics** on all samples. Scroll down to the next section *(%% Section 3: Automation for data exploration using FTMS_Metrics*). Click on the section and then click *Run Section* (**Slide 137**). The code will only load Excel files ending with "_Final.xlsx"; thus the other files of this folder will not obstruct the execution of the code. Once the automated processing of the 20 samples is finished, an Excel file named "FTMS Compound Class Master Report.xlsx" will be output (**Slide 137**) that will contain a summary of all metrics for each sample (**Slide 138**). Sheet "Sheet1" contains number-based averages, sheet "STDEV" contains the number-based standard deviations, and sheet "Metrics Weighed" contains magnitude-weighted averages.

The last section of the code (*%% Section 4: Automation for generic function (FTMS_Function)*) is designed to provide a building block for automation for any other FTMS function that is to be employed on numerous processed ("_Final.xlsx") samples. This code piece can be used to employ **FTMS_Metrics**, **FTMS_CompoundClass**, **FTMS_Figures** (code described below), **FTMS_Peptides** (code described below), etc. We will not show an example of this section. If users want to run this section, they can follow the instructions on **Slide 139**.

The **FTMS_Automation** code utilizes external functions for alphanumeric sorting of filenames (*natsortfiles* and *natsort*) developed by Stephen Cobeldick (https://www.mathworks.com/matlabcentral/fileexchange/47434-natural-order-filename-sort). These codes and their license are found in TEnvR\Supplementary files\Internal codes.

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

- **FTMS_Figures**

To further explore the molecular composition of a studied sample, FT-ICR-MS or Orbitrap molecular formulas can be visualized in various ways using **FTMS_Figures**. To use it, keep the TestData_FTICRMS folder as your Current Folder:

Slide 140: *FTMS_Figures('Sample 1_Final.xlsx')*

The code produces four figures with numerous panels containing many different types of figures:

→ The first figure (**Slide 141**) contains 24 histograms of nearly all parameters that were used to calculate averages by **FTMS_Metrics**. This figure is saved in the Current Folder as "Fig_Histograms_filename.png" (**Slide 140**).

→ The second figure (**Slide 142**) contains 8 different 3D vK diagrams. The third dimension is based on magnitude, molecular weight, N/C, S/C, P/C, E/C, DBE, and NOSC. Such plots have been found very useful in interpreting the molecular composition of NOM (Wu et al., 2004; Sleighter and Hatcher, 2007). The figure panels of MATLAB are interactive and the user is able to zoom in/out, copy a sub-panel, change the view angle of a sub-plot, and click on an individual point to see its x, y, and z values. Please see the following website for instructions of the interactive capabilities of MATLAB:

https://www.mathworks.com/help/matlab/creating_plots/interactively-explore-plotted-data.html. This figure is saved in the Current Folder as “Fig_3DvKs_filename.png” (**Slide 140**).

→ The third figure (**Slide 143**) contains 8 plots evaluating different KMD homologous series (CH_2, H_2, COO, CH_2O, O_2, H_2O, NH_3, and HCl). KMD series analysis is a well-accepted approach for evaluating molecular composition (Kendrick, 1963; Hughey et al., 2001; Wu et al., 2004; Dittmar and Koch, 2006; Ikeya et al., 2015, 2020). This figure is saved in the Current Folder as “Fig_KMDplots_filename.png” (**Slide 140**).

→ The fourth and last figure (**Slide 144**) contains 10 plots of various parameters (H/C, O/C, N/C, S/C, P/C, NOSC, DBE, DBE/C, AI_{MOD}, C-normalized Ring) relative to the number of carbon atoms. The formulas are color-coded based on relative magnitude (TSS-normalized (Thompson et al., 2021) and in log_{10}-units). Evaluating molecular parameters relative to number of carbons is another useful approach to evaluate the molecular composition of NOM (Hsu et al., 2011; Cho et al., 2017). This figure is saved in the Current Folder as “Fig_Cnumberplots_filename.png" (**Slide 140**).

We recognize that the **FTMS_Figures** code is not fully inclusive to all possible figures that can be plotted using ultrahigh resolution mass spectrometry data. Furthermore, the produced figures are not of publication-grade quality. The sole purpose of the **FTMS_Figures** code, as well as to any of the other figure codes of TEnvR, is to provide a more detailed exploration of the sample that is being evaluated and to give the researcher ideas of how to further plot their data. Here, the **FTMS_Figures** code can be used to provide building blocks (e.g., a histogram plot) for creating codes that produce publication-grade figures. We have used such customized codes to produce high quality figures in our recent publications (Goranov et al., 2020; Goranov et al., in review).

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

- **FTMS_Figure_MD**

Another useful approach for assessing the molecular composition is based on mass defects. To use this code, keep the TestData_FTICRMS folder as your Current Folder:

Slide 145: *FTMS_Figures_MD ('Sample 1_Final.xlsx')*

The **FTMS_Figures_MD** code plots the molecular formulas on a vK diagram as well as an H/C vs. Molecular Weight plot (**Slide 145**). Formulas are coded based on their mass defect: 0-0.1 in pink, 0.1-0.2 in **blue**, 0.2-0.3 in **red**, 0.3-0.4 in **green**, 0.4-0.7 in **cyan**, and 0.95-1 in **black**. The black lines indicate modified aromaticity index cutoffs (Koch and Dittmar, 2006, 2016).

- **FTMS_Compare**

Once each individual formula list is evaluated using **FTMS_CompoundClass** and **FTMS_Figures**, more advanced data analysis approaches can be undertaken. A critical technique for evaluating two or more samples altogether is the Presence/Absence approach described in detail by Sleighter et al. (2012). This technique compares two formula lists and identifies common formulas (formulas that are “Present” in both samples being evaluated). Formulas that are unique to only one of the samples are also identified (formulas that are “Absent” from both samples being evaluated). Formula comparisons are done based on the computed molecular weight (“ExactMass”) of each formula. Molecular weights are rounded to the user-defined precision in the **FTMS_ConfigurationToolbox**, line 45, variable “*Precision*” (**Slide 93**). For the instruments we have experience with, we use a precision of 5 decimals. The Presence/Absence approach is a powerful technique for evaluating changes across environmental gradients (e.g., salinity transects, depth profiles) or during environmental processing (e.g., photochemical/microbial degradation) and has been used extensively in environmental research (Stubbins et al., 2010; Ward et al., 2014; Ward and Cory, 2016; Powers et al., 2019; Yuan et al., 2019; Goranov et al., 2020; Patel et al., 2021; Goranov et al., in review).

However, it must be noted that the Presence/Absence approach is prone to biases due to the common use of ESI sources in FT-ICR-MS work. Formulas that are absent in one sample may have vanished due to an actual change that occurred to the sample (e.g., photochemical degradation), but they also may have vanished artificially due to an ionization bias (Patriarca et al., 2020). This is often observed in photochemical studies: compound classes that are present in a control sample but are no longer present in a photo-irradiated sample may have vanished because of their photo-degradation. However, they may still be present but be outcompeted due to preferential ionization of photochemically produced molecules. And vice versa, compound classes that are absent in a control sample but become present in a photo-irradiated sample may be truly photo-produced, but they also may have been suppressed in the control sample and become more ionizable in the photo-irradiated sample, as seen previously by Goranov et al. (2020). Thus, the employment of the Presence/Absence approach must be employed with clear hypotheses and expectations to avoid incorrect interpretations (D’Andrilli et al., 2020). Furthermore, we recommend that the use of Presence/Absence analysis of FT-ICR-MS data is paired with supplementary analyses that will validate the mass spectrometric findings and eliminate the possibility of an ESI bias. Previously, we have validated findings from the Presence/Absence approach using absorbance, fluorescence, NMR, and compound-specific measurements (Goranov et al., 2020; Wozniak et al., 2020; Goranov et al., in review).

To use the **FTMS_Compare** code, keep the TestData_FTICRMS folder as your Current Folder:

Slide 146: *FTMS_Compare('Sample 1_Final.xlsx', 'Sample 2_Final.xlsx')*

The code produces three figures that pop up as maximized windows. On the first figure (**Slide 147**), the two whole samples are plotted and stacked on top of each other using vK (top left) and H/C vs molecular weight (bottom left) plots. Unique formulas to each sample are also plotted on vK (top right) and H/C vs molecular weight (bottom right) plots. Useful statistics (number of total, unique, and common formulas) are shown in the legends of each figure panel.

The second figure (**Slide 148**) plots only the common formulas on 3D vK diagrams colored according to the spectral magnitudes (TSS-normalized (Thompson et al., 2021) and in $\log_{10}$-units) of the first and second sample (here, “Sample 1_Final.xlsx” and “Sample 2_Final.xlsx”) on the left and middle panels, respectively. The color bars are normalized to the same range to enable an easier comparison between the two samples. Percent change is calculated using the relative magnitudes of each formula (i) from the two samples as shown below:

$$\%\mathrm{Change_i} = \frac{\mathrm{Rel. Magnitude\ (Sample\ 2)_i} - \mathrm{Rel. Magnitude\ (Sample\ 1)_i}}{\mathrm{Rel. Magnitude\ (Sample\ 1)_i}} \times 100$$

The third panel on the right has the common formulas color-coded using the calculated % Change. This plot should be used cautiously and interpreted very carefully. Peak magnitudes in ESI mass spectrometry mainly correspond to molecular ionizability rather than quantity. Thus, an increase or decrease of the magnitude of a group of formulas may not necessarily correspond to formulas increasing or decreasing in quantities. We generally use this plot to validate trends from plots of unique formulas (top right of the first figure, **Slide 147**). If a group of formulas vanished from Sample 1 and another group of formulas became present in Sample 2, we expect to see a negative and positive change, respectively, in the same regions amongst the common formulas (**Slide 148**). Please note that the figure panels are interactive and any of the 3D plots can be rotated in the three dimensions by the user (as shown on **Slide 127**).

The third figure (**Slide 149**) shows a Venn diagram on the left panel with statistics for the three pools of formulas – unique for sample 1, common for both samples, and unique for sample 2. The diagram lists number of formulas, as well as number- and magnitude-based percentages. The number-based percentages (num%) are based on the total number of formulas in the dataset, which are the formulas of the three pools (unique for sample 1 + common for both samples + unique for sample 2). Therefore, the total number of formulas (8099) is equal to the number of formulas of both samples (6465 + 6252), but here the common formulas among the two samples are duplicated and therefore 4618 must be subtracted, giving the total number of formulas has 6465 + 6252 - 4618 = 8099. The 8099 total formulas correspond to 1847 unique formulas to Sample 1, 4618 common formulas to both samples, and 1634 unique formulas to Sample 2. Magnitude-based percentages are on a per-sample basis. For example, the number of unique formulas in Sample 1 (1847) relative to the formulas in the three pools (1847+4618+1634) is 23%. The unique formulas in Sample 1 correspond to 14% of its spectral magnitude. The number of common formulas in Sample 1 (4618) relative to the formulas in the three pools (1847+4618+1634) is 57%. The common formulas in Sample 1 correspond to 86% of its spectral magnitude.

Venn diagram statistics are incredibly useful in interpreting trends across environmental gradients (Sleighter and Hatcher, 2008). The figure on the right panel is a scatterplot that evaluates the relative magnitude of common formulas in the two samples. The goodness-of-fit (R^2) of the linear fit (as a black dotted line) is listed in the title, and a one-to-one line is also plotted (in red) for easier slope comparison. Generally, a higher R^2 value closer to 1 indicates little change to the spectral distribution of common formulas. If two different samples are being compared, the R^2 can be used to deduce the similarity in magnitude distribution of their common formulas (assuming both samples were of equal ionizability). If the same sample being analyzed twice is being compared (as either an experimental or instrumental duplicate), this plot can serve as an evaluation of reproducibility (Sleighter et al., 2012).

The code also produces an Excel file (named "Comparison_filename1_filename2.xlsx", here "Comparison_Sample 1_Final_Sample 2_Final.xlsx") where the three different pools of formulas (unique for sample 1, common for both samples, unique for sample 2) are separated in different sheets (Unique1, Unique2, Common1[13], Common2[14]). An example is shown for the unique formulas for sample 1 on **Slide 150**. A new column with Relative Magnitude is added for each formula list (column T, label RelMagn). The two samples used in the code (here, "Sample 1_Final.xlsx" and "Sample 2_Final.xlsx") have their formula lists copied in this file also (in sheets "Data1" and "Data2", respectively). The default Excel sheet "Sheet1" contains the numeric statistics from the analysis (statistics found on the Venn diagram as well as slope parameters from the linear fit, **Slide 151**). The **FTMS_Compare** code also has the **FTMS_Metrics** code incorporated and the latter is employed on all sheets, providing a convenient way to evaluate the bulk averages of the molecular formulas that are unique or common to the two samples (**Slide 150**).

In addition to the Excel file, the three figures are exported as .png files. All exported files are labeled with the two sample names to create a paper trail (**Slide 150**). We recognize that the employment of TEnvR on computers with screens of different sizes and resolutions may change the appearance of these figures. If that is the case, we recommend users to use the code provided behind the figures and use it as a backbone for figures suitable for their systems. In future versions of TEnvR, we will advance these codes to be more universally applicable.

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

- **FTMS_Compare3**

This code is designed for comparing three samples using the presence/absence approach described above. This code is an extension of **FTMS_Compare**. Formula comparisons are done based on the computed molecular weight ("ExactMass") of each formula. Molecular weights are rounded to the user-defined precision in the **FTMS_ConfigurationToolbox**, line 45, variable: *Precision* (**Slide 93**). For the instruments we have experience with, we use a precision of 5 decimals. This code also produces publication-grade figures that we have used previously (Goranov et al., 2020).

To use the **FTMS_Compare3** code, keep the TestData_FTICRMS folder as your Current Folder:

Slide 152: *FTMS_Compare3('Sample 1_Final.xlsx', 'Sample 2_Final.xlsx','Sample 3_Final.xlsx')*

The code first compares the three formula lists and identifies common formulas that are present in all three samples. Common formulas are plotted for each sample on the first figure that pops up as a maximized window (**Slide 153**), which is color-coded based on the spectral magnitude for each sample (TSS-normalized (Thompson et al., 2021) and in $\log_{10}$-units). The black circle serves as a visual reference for easier comparisons. Formulas that are unique to one or two samples are separated as "Unique" and plotted on a second maximized figure (**Slide 154**). The modified aromaticity index (AI_{MOD}) range above 0.67 is boxed with black lines and formulas are color-coded based

[14] While molecular formulas in Common1 and Common2 are identical, their relative magnitudes, signal-to-noise (S/N), and estimated C-numbers (Estim #C) differ between the two samples.

on their carbon content. Figures are exported in Tagged Image Format (.tif) format (publication-grade) (named “Compare3_Common_filename1_filename2_filename3.tif” and “Compare3_Unique_C_filename1_filename2_filename3.tif”).

The code also produces an Excel file (“Comparison3_filename1_filename2_filename3.xlsx”, here “Comparison3_Sample 1_Final_Sample 2_Final_Sample 3_Final.xlsx”), where the three unique and three common pools of formulas are separated in different sheets (Unique1, Unique2, Unique3, Common1[14], Common2[15], Common3[14]). An example is shown for the unique formulas for sample 1 on **Slide 155**. A new column with Relative Magnitude is added for each formula list (column T, label RelMagn). The three samples used in the code (here, “Sample 1_Final.xlsx”, “Sample 2_Final.xlsx”, and “Sample 3_Final.xlsx”) have their formula lists copied in this file also (in sheets “Data1”, “Data2”, and “Data3”, respectively). The **FTMS_Compare3** code also has the **FTMS_Metrics** code incorporated and the latter is employed on all sheets providing a convenient way to evaluate the bulk averages of the molecular formulas that are unique or common to the three samples (**Slide 155**).

The **FTMS_Compare3** code has a capability to change the color-coding of the Unique formulas. Formulas are by default color-coded based on their carbon content (note that the file name for the TIF file generated above includes _C in the file name, **Slide 152**), but this can be changed to their N/C ratio. To edit the way samples are color coded, open the code by typing *edit FTMS_Compare3* in your Command Window (**Slide 156**). Change the *Z_scale* variable (line 13) from *'carbon'* to *'nitrogen'* to enable color-coding based on N/C ratio. Click *Save*. Run the code.

Slide 156: *FTMS_Compare3('Sample 1_Final.xlsx', 'Sample 2_Final.xlsx','Sample 3_Final.xlsx')*

The first figure (“Compare3_Common_filename1_filename2_filename3.tif”) and the Excel file (“Comparison3_ filename1_filename2_filename3.xlsx”) will be overwritten, but a new figure will be saved (named “Compare3_Unique_N_filename1_filename2_filename3.tif”), using N in the file name rather than C. The new figure is shown on **Slide 157**. We recognize that the employment of TEnvR on computers with screens of different sizes and resolutions may change the appearance of these figures. If that is the case, we recommend users to take the code provided behind the figures and use it as a backbone for figures suitable for their systems. In future versions of TEnvR, we will advance these codes to be more universally applicable.

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

- **FTMS_Compare4**

This code is designed for comparing four samples using the presence/absence approach described above. This code is an extension of **FTMS_Compare** and **FTMS_Compare3**. Formula comparisons are done based on the computed molecular weight (“ExactMass”) of each formula. Molecular weights are rounded to the user-defined precision in the **FTMS_ConfigurationToolbox**, line 45, variable:

[15] While molecular formulas in Common1, Common2 and Common3 are identical, their magnitude, signal-to-noise (S/N) and estimated C-numbers (Estim #C) differ between the three samples.

Precision (**Slide 93**). For the instruments we have experience with, we use a precision of 5 decimals. This code also produces publication-grade figures that we have used previously (Goranov et al., 2020).

To use the **FTMS_Compare4** code, keep the TestData_FTICRMS folder as your Current Folder:

Slide 158: *FTMS_Compare4('Sample 1_Final.xlsx', 'Sample 2_Final.xlsx','Sample 3_Final.xlsx', 'Sample 4_Final.xlsx')*

The code first compares the four formula lists and identifies common formulas that are present in all four samples. Common formulas are plotted for each sample on the first figure that pops up as a maximized window (**Slide 159**). Formulas are color-coded based on the spectral magnitude for each sample (TSS-normalized (Thompson et al., 2021) and in $\log_{10}$-units). The black circle serves as a visual reference for easier comparisons. Formulas that are unique to one, two, or three samples are separated as “Unique” and plotted on a second maximized figure (**Slide 160**). The modified aromaticity index (AI_{MOD}) range above 0.67 is boxed with black lines and formulas are color-color coded based on their carbon content. The Unique formulas are also plotted with no color-coding on a third maximized figure (**Slide 161**). Figures are exported in Tagged Image Format (.tif) format (publication-grade) (named “Compare4_Common_filename1_filename2_filename3_filename4.tif”, “Compare4_Unique_C_filename1_filename2_filename3_filename4.tif”, and “Compare4_UniqueSimple_filename1_filename2_filename3_filename4.tif”, respectively)

The code also produces an Excel file (“Comparison4_filename1_filename2_filename3_filename4.xlsx”, here “Comparison4_Sample 1_Final_Sample 2_Final_Sample 3_Final_Sample 4_Final.xlsx”) where the four unique and four common pools of formulas are separated in different sheets (Unique1, Unique2, Unique3, Unique4, Common1[15], Common2[16], Common3[15], Common4[15]). An example is shown for the unique formulas for sample 1 on **Slide 162**. A new column with Relative Magnitude is added for each formula list (column T, label RelMagn). The four samples used in the code (here, “Sample 1_Final.xlsx”, “Sample 2_Final.xlsx”, “Sample 3_Final.xlsx”, and “Sample 4_Final.xlsx”) have their formula lists copied in this file as well (in sheets “Data1”, “Data2”, “Data3”, and “Data4”, respectively). The **FTMS_Compare4** code also has the **FTMS_Metrics** code incorporated and the latter is employed on all sheets providing a convenient way to evaluate the bulk averages of the molecular formulas that are unique or common to the four samples (**Slide 162**).

As described above for **FTMS_Compare3**, the **FTMS_Compare4** code has a capability to change the color-coding of the Unique formulas. Formulas are by default color-coded based on their carbon content, but this can be changed to their N/C ratio. To edit the way samples are color coded, open the code by typing *edit FTMS_Compare4* in your Command Window (**Slide 163**). Change the *Z_scale* variable (line 14) from *‘carbon’* to *‘nitrogen’* to enable color-coding based on N/C ratio. Click *Save*. Run the code.

Slide 163: *FTMS_Compare4('Sample 1_Final.xlsx', 'Sample 2_Final.xlsx','Sample 3_Final.xlsx', 'Sample 4_Final.xlsx')*

[16] While molecular formulas in Common1, Common2, Common 3 and Common 4 are identical, their magnitude, signal-to-noise (S/N) and estimated C-numbers (Estim #C) differ between the four samples.

The first ("Compare4_Common_filename1_filename2_filename3_filename4.tif") and third ("Compare4_UniqueSimple_filename1_filename2_filename3_filename4.tif") figures, as well as the Excel file ("Comparison4_filename1_filename2_filename3_filename4.xlsx") will be overwritten. The produced figure (**Slide 164**) with the nitrogen color-coding is saved as "Compare4_Unique_N_filename1_filename2_filename3_filename4.tif". Again, the color-coding is indicated in the figure filename ("_C" or "_N"). We recognize that the employment of TEnvR on computers with screens of different sizes and resolutions may change the appearance of these figures. If that is the case, we recommend users to use the code provided behind the figures and use it as a backbone for figures suitable for their systems. In future versions of TEnvR, we will advance these codes to be more universally applicable.

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

- **FTMS_Peptides**

Molecular formulas can represent peptides and such species are commonly observed in various matrices. They are particularly easy to detect in positive ionization mode but can also be found in negative. The **FTMS_CompoundsClass** code described earlier classifies formulas as proteinaceous if they are within certain H/C and O/C ranges and if they have at least one nitrogen present. The **FTMS_Peptides** code can further classify formulas by determining if any of the molecular formulas in a given list are oligopeptides. Oligopeptides are defined as a series of linearly bound amino acids. To use the **FTMS_Peptides**, keep TestData_FTICRMS as your current folder:

Slide 165: *FTMS_Peptides('Sample 2_Final.xlsx',2,5)*

The second (*2*) and third (*5*) arguments are the minimum and maximum length of oligopeptide sequences (i.e., number of amino acids in a linear peptide). Fundamentally, the code takes this range (here, 2-5) and develops a list of possible combinations of amino acid monomers to result in dimers to pentamers. This is done using the *combn* function of the Toolbox for Dimensionality Reduction developed by J. van der Geest and Laurens van der Maaten. More information can be found on the following website: https://github.com/UMD-ISL/Matlab-Toolbox-for-Dimensionality-Reduction). The number oligopeptides, i.e., number of possible combinations of amino acids (denoted as C'), is combinatorically calculated based on the number of amino acids monomers (denoted as n) and the requested oligopeptide length (denoted as k):

$$C'_k(n) = \binom{n+k-1}{k} = \frac{(n+k-1)!}{k!\,(n+k-1-k)!} = \frac{(n+k-1)!}{k!\,(n-1)!}$$

For example, the number of possible oligopeptide sequences containing 5 amino acids, which can be any of the 20 possible amino acid monomers, is 42504.

$$C_5'(20) = \binom{20+5-1}{5} = \binom{24}{5} = \frac{24!}{5!\,(24-5)!} = \frac{24\cdot 23\cdot 22\cdot 21\cdot 20\cdot 19!}{5\cdot 4\cdot 3\cdot 2\cdot 1\cdot 19!} = 42504$$

The **FTMS_Peptides** code determines these combinations for oligopeptides of different lengths. Thus, the total number of combinations for oligopeptides of lengths "min" to "max" is calculated as:

$$Total\ Combinations\ (min - max) = \sum_{i=min}^{max} C_i'(20)$$

For the range given as an example here (2-5), the number of possible oligopeptide sequences containing 2-5 amino acids is 53109:

$$Total\ Combinations\ (2-5) = \sum_{i=2}^{5} C_i'(20) = C_2'(20) + C_3'(20) + C_4'(20) + C_5'(20) = \binom{21}{2} + \binom{22}{3} + \binom{23}{4} + \binom{24}{5}$$
$$= \frac{21!}{2!\,(21-2)!} + \frac{22!}{3!\,(22-3)!} + \frac{23!}{4!\,(23-4)!} + \frac{24!}{5!\,(24-5)!} = 210 + 1540 + 8855 + 42504 = 53109$$

This range can be further expanded, however, at a certain point the number of possible combinations becomes so big that the processing power of the computer will not be enough. In our experience, using a range larger than 2-5 or 2-6 on regular computers is not possible. If the user attempts to use a higher range (e.g., 2-7 or higher), and the computer is not powerful enough, an error message will be displayed and MATLAB and/or computer may freeze. The computer has frozen if the clock stops – you will likely need to do a hard reboot on the computer. We tested this code on a computer with 8 GB installed RAM and the largest range we could use was 2-5. On a separate computer with 64 GB installed RAM, the largest range we could use was 2-6. We could not employ larger ranges (e.g., 2-7) on the computers we tested TEnvR on. We have previously used this code on a supercomputer, and we have successfully used a range of 2-10 (Goranov et al., 2022).

The code then calculates the ExactMass of each of these oligopeptides and tries to match each formula's ExactMass to any of the ExactMasses of the possible oligopeptides. The matching is done by finding a mass difference that satisfies the user-defined mass accuracy in the **FTMS_ConfigurationToolbox** file (line 44, variable *MassAccuracy*, **Slide 93**). The code produces a new file named "_Peptides.xlsx" (here, "Sample 2_Peptides.xlsx") containing the original formula list (here, found in "Sample 2_Final.xlsx") with three additional columns (T, U, V) (**Slide 166**). If no oligopeptide sequences were matched to the ExactMass of a molecule, column T (labeled "Options") will be populated with a zero, column U (labeled "Sequence") will be empty, and column V (labeled "Error (ppm)") will be populated with 1000. If the ExactMass of a formula is indeed matched with a possible oligopeptide combination, the combination is going to be listed in column U (labeled "Sequence"). The number of matches, which may be 1 or more, will be listed in column T (labeled

“Options”). The mass difference (in ppm) will be listed in column V (labeled “Error (ppm)”) and its absolute value is going to be below the user-defined mass accuracy in the **FTMS_ConfigurationToolbox** file (line 44, variable *MassAccuracy*, **Slide 93**).

It must be noted that this code is unable to identify other proteinaceous species, such as higher order peptides or oligopeptides with more complex structures and motifs (e.g., sulfide bridges). A higher degree of data processing of proteinaceous species can be achieved using other software/databases (Välikangas et al., 2017; Schwämmle et al., 2021; Singh, 2021) and available code packages in MATLAB (e.g., Bioinformatics Toolbox).

In the Supplementary Files folder of TEnvR, we provide a file (“FTMS_AminoAcids.xlsx”) containing a list of all possible amino acids containing their abbreviations and exact masses.

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

- **FTMS_KMD**

Another common approach for evaluating molecular data is using Kendrick Mass Defect (KMD) series analysis (Kendrick, 1963; Hughey et al., 2001; Wu et al., 2004; Dittmar and Koch, 2006; Ikeya et al., 2015, 2020), as described above. To see an application of KMD analysis with a detailed explanation, see Figure 4 and corresponding discussion of Goranov et al. (in review). KMD plots are incorporated in the **FTMS_Figures** code, but on those plots data could only be crudely evaluated. Here with **FTMS_KMD**, a more detailed assessment can be performed. This code is designed to evaluate formulas that align on KMD series defined by the user. Here, an example will be shown with the carboxyl (COO) series, i.e., formulas differing in their number of COOH groups.

To use the **FTMS_KMD**, keep TestData_FTICRMS as your current folder:

Slide 167: *FTMS_KMD('Sample 1_Final.xlsx','COO')*

The code uses the exact mass to calculate the Kendrick Mass (KM), Kendrick Nominal Mass (KNM), and Kendrick Mass Defect (KMD) for the user-defined series (here, *'COO'*). The table below lists all possible series that are incorporated in the code and what chemical processes they generally correspond to. Users are welcome to modify the KMD codes and incorporate additional KMD series that fit their needs.

Function argument	Corresponding functional group/building block	Series factor (S)	Application[17]
'CH2'	Methyl	$S = \frac{14.0000000}{14.0156500}$	Addition/removal of CH_2 ≡ Addition/Removal of CH_2 building blocks; Molecular synthesis/degradation, Reduction or elongation of alkyl chains
'H2'	Dihydrogen	$S = \frac{2.0000000}{2.0156500}$	Addition/removal of H_2 ≡ Hydrogenation/dehydrogenation, Saturation/unsaturation of π-bonds (usually C=C)
'O'	Oxygen	$S = \frac{16.0000000}{15.9949146}$	Addition/removal of O ≡ Reduction/Oxidation Formation/removal of hydroxyl groups, S=O, N=O, etc.
'CO'	Keto	$S = \frac{28.0000000}{27.9949146}$	Addition/removal of CO ≡ Reduction/Oxidation, Formation/Removal of keto groups (aldehydes and ketones) as well as carboxylic acid derivatives
'COO'	Carboxyl	$S = \frac{44.0000000}{43.9898292}$	Addition/removal of COO ≡ Reduction/Oxidation, Formation/Removal of carboxyl groups
'CH2O'	Methoxy	$S = \frac{30.0000000}{30.0105646}$	Addition/removal of CH_2O ≡ Methoxylation/Demethoxylation, Formation/Removal of methoxy groups
'H2O'	Water	$S = \frac{18.0000000}{18.0105646}$	Addition/removal of H_2O ≡ Hydration/Condensation, Addition/Removal of water from π-bonds (usually C=C)
'NH3'	Ammonia	$S = \frac{17.0000000}{17.0265490}$	Addition/removal of NH_3 ≡ Amination/Deamination, Addition/Removal of NH_3 from π-bonds (usually C=C)
'HE'	Hydrogen chloride/bromide, etc.	$S\ (\mathrm{E} = {}^{35}Cl) = \frac{36.0000000}{35.9766777}$	Addition/removal of HE. In cases when E = halogen ≡ Halogenation/Dehalogenation, Addition/Removal of HE from π-bonds (usually C=C)

Once the user defines the series using the second argument of the function (here, 'COO'), the code determines the corresponding series factor (S) and calculates KM, KNM, and KMD as shown below.

$$\mathrm{KM} = \text{Molecular Weight } \times \mathrm{S}$$

$$\mathrm{KNM} = \text{integer of KM}$$

$$\mathrm{KMD} = \mathrm{KM} - \mathrm{KNM}$$

[17] The symbol ≡ means “is equivalent to”.

The KMD values are rounded to the number of decimals defined in the **FTMS_ConfigurationToolbox** (line 45, variable: *Precision*, **Slide 93**). For the instruments we have experience with, we use a precision of 5 decimals. Formulas with the same KMD value are found by the code. The first formula on a series (i.e., the one with the lowest KNM) is assumed to be the starting point for the series. In areas of metal complexation research, such formulas are referred to as "apo" formulas. Formulas that are metal-free are referred to as being in their "apo form", while formulas complexed with metals are characterized as being in their "metal form" (Boiteau et al., 2016). This terminology has been used in the **FTMS_KMD** code. Formulas that are not first on the KMD series are labeled as Sequential. Formulas that have unique KMD values cannot establish KMD series (a minimum of 2 points are required for a KMD series in this code). Such formulas are labeled as Unique.

The code produces two publication-grade figures (**Slide 167**). The first one is a scatter plot of KMD vs KM (saved as "KMD_COO_filename1.png") and is known as a KMD plot. Unique, Apo, and Sequential formulas are color-coded and statistics (number of formulas + number-based percentages) are shown in the legend. The user can use the interactive capabilities of the figure panel and zoom into individual KMD series (example shown on **Slide 127**). The code produces a second figure, a vK diagram, with the same groups of formulas and the same color-code (**Slide 167**). The vK diagram figure is saved as "KMDvK_COO_filename.png". The two figures are exported in .png format and the sample name and KMD series used (here, COO) are retained in the file name to create a paper trail.

- **FTMS_KMD_Ox**

This code has been specifically designed to evaluate oxidation. We have recently employed this approach to study the oxidative transformation of molecules during microbial incubations (Goranov et al., in review). This code is an extension of FTMS_KMD and involves simultaneous KMD analyses using O, CO, and COO series. Apo formulas are assumed to be substrates, while formulas with increased numbers of O, CO, and/or COO groups are considered to be products of oxidation reactions (such as those from hydroxyl radicals, see Waggoner et al. (2015); Waggoner et al. (2017), Waggoner and Hatcher (2017)). The mathematics behind the **FTMS_KMD** and **FTMS_KMD_Ox** codes are identical with the following difference: Some formulas may be categorized as oxidation products using more than one type of KMD series (i.e., one formula may be categorized as an oxidation product using two or three of the O, CO, and COO series). This creates redundant formulas categorized as "Oxidation Products" and the code simply removes these formula duplicates.

Formulas part of the same series are identified by their identical KMD values as described above. The KMD values are rounded to the number of decimals defined in the **FTMS_ConfigurationToolbox** (line 45, variable: *Precision*, **Slide 93**). For the instruments we have experience with, we use precision of 5 decimals.

To use the **FTMS_KMD_Ox**, keep TestData_FTICRMS as your current folder:

Slide 168: *FTMS_KMD_Ox('Sample 1_Final.xlsx')*

The code creates a vK diagram (**Slide 168**) showing the three different categories: formulas that are unique ("Not on KMD series"), apo formulas ("Substrates"), and sequential formulas on the oxidation series ("Oxygenation products"). Formulas are color-coded and statistics (number of formulas + number-based percentages) are shown in the legend. The user can use the interactive capabilities of the figure panel and zoom into individual KMD series (example shown on **Slide 127**). This figure is of publication-grade and is saved as "KMDvK_Oxidation_filename.png". Please note that KMD plots are not exported by this code, because it is not possible to create a composite KMD plot of the three different series (O, CO, and KMD) used in the oxidation assessment.

The use of this approach involves several important caveats that are to be considered. It is assumed that the first formula (i.e., the apo formula) of any of the O, CO, or COO series is fundamentally a control compound that was then oxidized. It is also assumed that the oxidation is not complete, i.e., both the starting compounds (the apo formulas) and the sequential oxidation products were detected by the ESI-FT-ICR-MS analysis. It is known that the varying degrees of oxygenation can affect molecular ionizability (Patriarca et al., 2020), and therefore the true control molecules may not be detected and thus, oxidation products may be misinterpreted as control compounds.

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

- **FTMS_KMD2**

This code has been designed to compare two samples using KMD analysis. This has been incredibly useful in studying environmental processes (Liu et al., 2011; Waggoner and Hatcher, 2017; Waggoner et al., 2017; Khatami et al., 2019; Goranov et al., in review). Fundamentally, **the FTMS_KMD2** code involves the same mathematics as the **FTMS_Compare** code for performing Presence/Absence analysis among two formula lists. It then incorporates the same mathematics as in the **FTMS_KMD** code to perform KMD series analysis.

To use the **FTMS_KMD2**, keep TestData_FTICRMS as your current folder. Here, an example will be shown with carboxyl (COO) series.

Slide 169: *FTMS_KMD2('Sample 1_Final.xlsx','Sample 2_Final.xlsx','COO')*

The code first compares two formula lists and identifies common and unique formulas. Formula comparisons are done based on the computed molecular weight ("ExactMass") of each formula. Molecular weights are rounded to the user-defined precision in the **FTMS_ConfigurationToolbox**, line 45, variable: *Precision* (**Slide 93**). For the instruments we have experience with, we use a precision of 5 decimals. The code then computes the KM, KNM, and KMD for all common and unique formulas using the user-defined series (here, *'COO'*). This is described in detail above for the FTMS_KMD code. KMD values are also rounded to the number of decimals defined in the **FTMS_ConfigurationToolbox**. The three different formula groups (unique for sample 1, unique for sample 2, and common to the two samples) are plotted on a KMD plot. Formulas are color-coded and Venn diagram statistics (number of formulas + number-based percentages) are shown in the legend. The user can use the interactive capabilities of the figure panel and zoom into individual KMD series (example shown on **Slide 127**). This figure is of publication-grade and is saved as "Compare_KMD_COO_filename1_filename2.png".

To evaluate the figure in more detail, type *set(gcf,'WindowState','maximize')* in the Command Window, which will maximize the figure to your screen size (**Slide 170**).

- **FTMS_AlignmentFormulas**

While FT-ICR-MS provides an extreme resolution to the sample composition, the produced data is particularly multivariate, and identifying trends among many formula lists each in the form of Excel files or another format is sometimes inefficient. The **FTMS_AlignmentFormulas** code is designed to load multiple spectra and align the peak lists into a matrix that displays which formula is present among the samples of the dataset. The alignment algorithm is described in detail and provided by Mantini et al. (2007). The resultant alignment matrix is useful in tracing how individual molecular formulas change in magnitude across a dataset and for performing multivariate statistical analyses as shown later.

To use the **FTMS_AlignmentFormulas** keep TestData_FTICRMS as your current folder. This code will load all Excel files in the folder that end with "_Final.xlsx". Thus, any other files except sample files ending with "_Final.xlsx" must be deleted. If you are following this tutorial, three interfering files ending with "_Final.xlsx" have been created by the comparison codes **FTMS_Compare**, **FTMS_Compare3**, and **FTMS_Compare4**. Please delete the following files: "Comparison_Sample 1_Final_Sample 2_Final.xlsx", "Comparison3_Sample 1_Final_Sample 2_Final_Sample 3_Final.xlsx", "Comparison4_Sample 1_Final_Sample 2_Final_Sample 3_Final_Sample 4_Final.xlsx" (**Slide 171**).

Slide 172: *FTMS_AlignmentFormulas*

The code produces an Excel sheet named "FTMS Alignment Matrix.xlsx" containing 5 sheets:

→ Sheet1 (**Slide 173**) – Contains the exact masses of formulas of the whole dataset. Absolute magnitudes corresponding to the formulas' peaks are shown for each sample where applicable. If a formula is not present in a certain sample, the corresponding magnitude value is set to zero. The matrix in this sheet is known as the "alignment matrix".

→ Formulas (**Slide 174**) – This sheet contains all information from "Sheet1" as well as molecular formula information on the right of the alignment matrix.

→ Trimmed (**Slide 175**) – This sheet contains a trimmed formula list. Formulas can be trimmed based on two criteria: Mass range and commonality. The mass range trimming algorithm removes any formulas outside of a user-defined range. The range is defined using the *MZ_low* and MZ_high variables on lines 54 and 55, respectively, of the **FTMS_ConfigurationToolbox** code (**Slide 93**). It is often beneficial to trim the data to a specific m/z range (e.g., 300-800), especially if there are significant tuning differences among samples and detection of ions below m/z *MZ_low* and *MZ_high* is less reproducible. For the FT-ICR-MS spectrometers we have experience with, we usually trim the data to m/z 300-800, especially if we plan on performing multivariate statistics. To disable filtering feature, simply put numbers outside of the range you will be working with. For example, with the provided dataset, using *MZ_low = 0* and *MZ_high = 2000* will be sufficient to disable this trimming feature. However, for this tutorial, we will trim the data to m/z 300-800. Alternatively, the commonality trimming algorithm removes any formulas that are not common to a certain number of samples. Sometimes, we require that formulas are present in more than one sample, and therefore, formulas unique to only one sample of the whole dataset are trimmed. The user can control this feature by the *MinSamples* variable on line 56 of the **FTMS_ConfigurationToolbox** code. This variable defines what is the minimum number of samples where a formula must be found in order to be included in the trimmed dataset. As an example here, we require that a formula is present in a minimum of 1 to be included in the trimmed dataset, and we sometimes change this threshold to 2 or 3.

→ Normalized (with magnitude, **Slide 176**) – The first option for normalizing data is based on relative magnitude, and this sheet contains all information from the Trimmed sheet, however, peak magnitudes here are normalized to the total spectral magnitude for each sample (TSS-normalized (Thompson et al., 2021)). We recommend using total spectral magnitude normalization when all samples were acquired using similar tuning and the data is of high reproducibility. We achieve this by validating our tuning using in-lab surrogate standards or IHSS standards (e.g., SRFA) following the recommendations by Hawkes et al. (2020).

→ Normalized (with presence-absence, **Slide 177**) – Instead of normalizing the data to total spectral magnitude (i.e., TSS-normalized) (Thompson et al., 2021), the second option is to normalize the data based on presence-absence. With the presence-absence normalization approach, magnitude values higher than zero are transformed to the value of 1 ("presence") and magnitude values equal to zero remain with the value of 0 ("absence"). This type of data transformation normalizes out differences in magnitudes among different peaks and accounts for variable ionizability of different molecular classes in the different samples of the dataset. All values are then further normalized to the number of formulas in each sample. This type of data transformation accounts for differences in instrumental sensitivity and detectability of different molecular classes (Kujawinski et al., 2009). To enable the presence-absence normalization, change the variable *PresenceAbsence* on line 57 of the **FTMS_ConfigurationToolbox** from *false* to *true*. We generally do not employ this normalization approach and use TSS-normalization, because using the spectral magnitude data is beneficial in revealing which molecular classes are most significant in driving the differences among samples, especially when we employ multivariate statistics (Sleighter et al., 2010). However, the presence-absence normalization is certainly useful for challenging datasets of unequally ionizing samples (accounting for variability in detectability/ionizability); for datasets comprised of samples analyzed on multiple instruments (accounting for variability in tuning); or for datasets comprised of samples analyzed on the same instrument across a large time frame (accounting for variability in tuning/instrumental drifts).

→ Normalized Common (**Slide178**) – This sheet contains formulas that are only found in all samples within the dataset. The data is normalized using the approach used to produce the data in the "Normalized" sheet. In the example of **Slide 178**, we have used TSS-normalization (Thompson et al., 2021).

Please do not delete the file "FTMS Alignment Matrix.xlsx" and do not change the sheet names "Normalized" and "Normalized Common" – these matrices will be directly used in several multivariate statistical routines (e.g., principal component analysis) as shown later in this tutorial.

The **FTMS_AlignmentFormulas** code utilizes external functions for alphanumeric sorting of filenames (*natsortfiles* and *natsort*) developed by Stephen Cobeldick (https://www.mathworks.com/matlabcentral/fileexchange/47434-natural-order-filename-sort). These codes and their license are found in TEnvR\Supplementary files\Internal codes.

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

Section 7. Nuclear Magnetic Resonance (NMR) Spectroscopy

- **NMR_Automation**

NMR spectroscopy is another tool that is highly utilized by various environmental research groups; however, we have not developed a significant number of scripts for processing NMR data. This is because NMR spectrometers typically come associated with software capable of a variety of data treatment and processing steps. Thus, we have found it easier to use the instrument software for data treatment (calibration, phasing, baseline-correction, integration, etc.) and data mining (integration, multiplet analysis, etc.). While automation of these processing steps is certainly possible, we have found that often, data treatment needs to be manually tuned and inspected for each sample. If users would like to explore possible computational routines for processing NMR data, there are various toolbox software packages that are already available (Günther et al., 2000; Ludwig and Günther, 2011). The **NMR_Automation** code is designed to organize one-dimensional NMR data into alignment matrix. The matrix format allows for preparation of publication-grade figures as well as multivariate statistical analysis.

In our experience, we have primarily used the TopSpin software associated with Bruker spectrometers to process and evaluate one- and two-dimensional data NMR data. We had found it necessary to create codes for processing 1D NMR data when it is necessary to export 1D NMR spectra and prepare publication-grade figures or evaluate the data using statistical analysis in MATLAB. The **NMR_Automation** code includes several routines allowing for automated reformatting of 1D NMR data into an Excel file as well as its pre-treatment prior to multivariate statistical analysis. To show an example of this code, we have provided 7 spectra acquired in COSMIC (ODU) on a 400 MHz (9.4 Tesla) Bruker BioSpin AVANCE II spectrometer fitted with a 4 mm solid-state HCN magic angle spinning (MAS) probe. Analysis was done using the ^{13}C MultiPulse Cross-Polarization MAS technique by Johnson and Schmidt-Rohr (2014), and these spectra have been previously published and interpreted by Bostick et al. (2018) and Wozniak et al. (2020). These spectra are located in TEnvR\TestData_NMR (**Slide 179**).

Slide 180: *NMR_Automation*

The code creates several new files and exports an Excel spreadsheet (named “NMR Alignment Matrix.xlsx”) described below. Open the code by typing *edit NMR_Automation* in the Command Window (**Slide 181**). The **NMR_Automation** code involves several stages:

→ *%% Stage 1: Reformat files from .txt files from TopSpin into 2D arrays.* 1D NMR spectra are exported by the Bruker TopSpin software as .txt files. Older versions of the software export 1D NMR data in a different format[18] than more recent versions[19]. The data format from a recent TopSpin version is shown on **Slide 182**, and the data format from an older TopSpin version is shown on **Slide 183**. To inform the code of the type of data format you will be using, change the variable *topspin_new* on line 47 (**Slide 181**) to *true* for files exported by recent versions and *false* for files exported from older versions. Please keep the .txt files as they are and do not delete any of the text in them – the code will extract the needed numerical information and ignore

[18] Older TopSpin versions export 1D spectra using the convbin2asc command.

[19] Newer TopSpin versions export 1D spectra using the totxt command.

the text data. We recognize that this code is currently exclusive to files acquired on Bruker spectrometers and future versions of TEnvR will have the option to import files from Varian spectrometers as well. The reformatting portion of the **NMR_Automation** code will export the reformatted spectra into text files labeled as "_ref.txt" (**Slide 180**), with chemical shift values in column 1 and intensity values in column 2 (**Slide 184**).

→ *%% Stage 2: Interpolation.* Even though the user defines the number of data points (the size of the free induction decay) prior to analysis, the resultant spectral files after post-processing often have a slight difference in the number of data points. For example, the seven files provided here in TestData_NMR have a slight variation in the number of data points: Sample 1 (1048131), Sample 2 (1048536), Sample 3 (1048206), Sample 4 (1048207), Sample 5 (1047675), Sample 6 (1048207), and Sample 7 (1047193). Prior to multivariate statistical analysis, all spectra must be of the same number of data points. Thus, spectra are interpolated using the *interp1* function to produce spectra of equal size (to learn more about this function, type *help interp1* in the Command Window). Interpolation can be disabled by setting the *Interpolation* variable on line 50 to *false* (**Slide 181**). Currently, interpolation is enabled using *true*. Interpolated spectra are exported as "_interp.txt" files (**Slide 180**).

→ *%% Stage 3: Denoising.* Spectra of NOM can often be noisy due to the very low concentration of NOM. Spectral noise becomes an issue when NMR data is to be used quantitatively and when it is evaluated using multivariate statistical analyses, such as PCA (Halouska and Powers, 2006). In our experience, we have successfully acquired spectra on samples have a carbon content of 1 mg/L and below (Whitty et al., 2019) but at the expense of very high number of scans (> 20,000). In cases when we were unable to use such a high number of scans, we have unfortunately obtained noisy spectra that needed to be further denoised. This can be done using post-acquisition processing in TopSpin. Commonly, we apply zero filling that increases the data size and generally improves the spectral quality. Exponential multiplication or other window functions can be also used to enhance the signal (Goranov et al., 2020; Goranov et al., in review) but that is at the expense of spectral resolution. Zero filling and exponential multiplication, if performed correctly with optimized parameters, are useful approaches that have shown to not affect multivariate statistical analyses such as PCA (Halouska and Powers, 2006). If none of these approaches work for denoising the spectrum, the **NMR_Automation** code contains a denoising algorithm proposed by Halouska and Powers (2006). Reformatted spectra are loaded, and the code will identify noise resonances in their baseline. The standard deviation of these resonances is taken, and it is multiplied by a factor of 5. Resonances in the whole spectrum below this "5 x standard deviation" threshold are set to zero and later eliminated. In our experience, we have rarely had to use this approach as we have managed to denoise spectra using the other approaches (zero filling, exponential multiplication) described above. However, this denoising approach has been shown beneficial prior to multivariate statistical analyses such as PCA (Halouska and Powers, 2006) and users are welcome to test it on their own data. The denoising algorithm can be enabled by setting the Denoise variable on line 53 to *true* (**Slide 181**). Currently, denoising is disabled using *false*. Denoised spectra are exported as "_denoised.txt" files (**Slide 180**).

→ *%% Stage 4: Binning.* It is common for NMR spectra to contain a very high number data points. A typical approach to reduce the spectral size is binning: data is grouped together by averaging it into a number of bins defined by the user using the binsize variable on line 57 (**Slide 181**). The main purpose of binning is to average out small deviations in the signal that may interfere with subsequent statistical analyses (Halouska and Powers, 2006). A secondary benefit of binning is that files become easier to handle and can be plotted/evaluated in Excel. In our experience, Excel is generally unable to work with the large unbinned files, and the software often crashes. We commonly use binning as a pre-processing procedure prior multivariate statistical analysis such as PCA (Sleighter et al., 2015; Wozniak et al., 2015; Tadini et al., 2021). Commonly, we bin liquid-state proton NMR data every 5-20 data points and solid-state carbon NMR data every 100-200 points. The *binsize* depends on the size of the free-induction decay and post-processing procedures such as zero filling. Thus, the *binsize* parameter needs to be tuned for each different dataset. Users are welcome to alter this parameter using the binsize variable on line 29 (**Slide 181**) and see how it alters the provided example spectra. If performed correctly with an optimized *binsize* parameter, binning is beneficial for improving multivariate statistical analysis results (Halouska and Powers, 2006) without reduction of spectral resolution (Sleighter et al., 2015). Binning can be disabled by setting the *Binning* variable on line 56 to *false* (**Slide 181**). Currently, binning is enabled using *true*. Binned spectra are exported as "_bin.txt" files (**Slide 180**) with the bin size retained into the filename prior bin (e.g., "Sample 1_ref_interp_200bin.txt").

→ *%% Stage 5: Alignment & Normalization.* Once the above pre-treatment procedures are performed, all NMR spectra are loaded and aligned into a matrix. Each NMR spectrum is then normalized to total spectral intensity to create a matrix of normalized data. The normalized matrix is further treated to set any negative intensity or NaN values to zero if such had not been removed using the denoising algorithm. Chemical shift values that have the same intensity value for all samples are removed from the dataset to avoid creating covariance values with missing values if multivariate statistical analyses such as PCA are employed.

→ *%% Stage 6: Export.* Unnormalized data and Normalized data are exported into an Excel file named "NMR Alignment Matrix.xlsx". The unnormalized data is placed in "Sheet1" (**Slide 185**). The Normalized data is written on a new sheet named "Normalized" (**Slide 186**). Data from this file can be further used for manual plotting in Excel or multivariate statistics as described below. Please do not delete the "NMR Alignment Matrix.xlsx" file and do not rename the sheet name "Normalized", as it will be directly used in several multivariate statistical routines as shown later in this tutorial.

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

Section 8. Multivariate Statistics

Multivariate data from UV-VIS, fluorescence, NMR, or FT-ICR-MS analyses is often difficult to interpret and explore on its own. Thus, multivariate statistical routines are usually employed to explore such datasets. Here, we present four routines for statistical analysis of multivariate data: hierarchical cluster analysis (HCA), principal component analysis (PCA), a code for developing correlation matrices with Pearson, Kendall, or Spearman correlations, and a code for Spearman correlation among FT-ICR-MS formulas and external parameters. For all statistical routines, a confidence level (α) of 95% is used corresponding to p-value threshold of significance of 0.05.

- **Stats_HCA**

Hierarchical cluster analysis (HCA) is a relatively simple but powerful approach for identifying samples that are similar or dissimilar to each other in a dataset. HCA results are displayed as a hierarchical tree also known as a dendrogram. Samples similar to each other fall on the same tree branch (are grouped together in a cluster), while samples dissimilar to each other are on different branches (are separated away from each other on different clusters). HCA can be used for evaluating datasets acquired using various analytical techniques (Xue et al., 2011; Gonsior et al., 2018), and it is a common approach for exploring multivariate data such as FT-ICR-MS formula assignments (Kujawinski et al., 2009; Hur et al., 2010; Sleighter et al., 2010; Liu et al., 2011; Dvorski et al., 2016; Xiao et al., 2020). Often HCA is employed in tandem with PCA, as they provide supplementary information and HCA can be used to validate PCA results. HCA is superior to PCA because it considers 100% of the variance of the dataset, while the first few components of PCA rarely explain over 70% of the dataset variance. However, a significant downside of HCA is that it loses variable information, and HCA is unable to explain which parameters of the dataset make the samples to be similar/different to each other (i.e., “drive” the sample clustering).

The mathematics behind HCA are complex and are described in detail elsewhere (Murtagh and Contreras, 2012, 2017; Vijaya and Bateja, 2017). Briefly, independent variables are first quantitatively compared among each other. A measure of similarity among variables is computed using a distance matrix calculation (using the MATLAB function *pdist*). We typically use Euclidean distances, but other types of distance metrics (SEuclidean, City Block, Jaccard, etc.) can also be used as described later. Then, to determine the similarity and dissimilarity among samples, samples are compared to each other using the computed pair-wise distances among their variables. This is done using a linkage criterion that determines how the samples are compared and linked together on branches in a hierarchical tree (using the MATLAB function *linkage*). We typically use the unweighted pair group method with arithmetic mean (UPGMA) as a linkage criterion, but other types of linkages (n, City Block, Jaccard, etc.) can also be used as described later.

HCA is a two-way technique, i.e., it requires an input of one dimension of independent variables and a second dimension of dependent variables that are associated with the independent variables for each sample. For UV-VIS data, wavelength values are the independent variables and their associated absorbance values (for each different sample) are the dependent variables (**Slide 187**). In the case when FT-ICR-MS formulas are used, the molecular weights of the formulas (their ExactMass-es) are the independent variables and their associated spectral magnitude values are the dependent variables (**Slide 187**). For NMR data, chemical shift values are the independent variables and their associated intensity values (for each different sample) are the dependent variables (**Slide 187**). Data of higher dimensionality (e.g., 3D fluorescence EEM spectra) cannot be directly evaluated using HCA or PCA. The two dimensions of independent variables in EEMs (excitation and emission wavelengths) or 2D NMR (two chemical shift dimensions) must be folded into one single dimension of independent variables (**Slide 187**). This has been explained and shown earlier for EEM data using **EEM_Fold** (**Slides 80 – 83**), but can be analogously performed on 2D NMR data (Hedenström et al., 2009). A code for folding 2D NMR data will be developed and provided in a future version of TEnvR.

HCA requires the import of a matrix that has the independent variables (molecular weight, chemical shift, wavelengths, etc.) for all samples aligned. The **UVVIS_Automation**, **EEM_Fold**, **FTMS_AlignmentFormulas**, and **NMR_Automation** codes produce such alignment matrices in their outputted files (saved in a sheet named “Normalized”). It is critical that data is normalized to account for concentration (in UV-VIS, EEM, or NMR data) or ionization (in ESI-FT-ICR-MS data) differences among samples. It is also required that no independent variable has the same dependent variable for all samples. For example with UV-VIS data, if all samples have absorbance at 600 nm of 0 AU, then the wavelength of 600 must be removed from a dataset. All codes (**UVVIS_Automation**, **EEM_Fold**, **FTMS_AlignmentFormulas**, and **NMR_Automation**) include an algorithm that removes such variables.

The **Stats_HCA** code takes a matrix of normalized data and outputs a dendrogram figure. The **Stats_HCA** code is versatile and can be applied on five different types of data as explained below. Please note that the larger the number of samples and number of variables, the longer it will take for the code to run.

→ **Stats_HCA** on UV-VIS spectra

To employ HCA on UV-VIS spectra, the different absorbance wavelengths are used as independent variables and the normalized[20] absorbance values are uses as dependent variables. To run this code, set TestData_UVVIS as your Current Folder. You must have the file “UVVIS_MasterReport.xlsx” in there.

Slide 188: *Stats_HCA('UVVIS_MasterReport.xlsx','UVVIS')*

The code outputs a dendrogram figure showing the differences and similarities among samples. The figure is saved as a .png file, and the filename of the matrix and type of data are retained into the figure file name (here, figure is named “HCA_UVVIS_MasterReport_UVVIS.png”). It can be seen that Samples 6 and 17 are very similar to each other. Samples 1 and 19 are also very similar to each other, but of less similarity that samples 6 and 17 (there is a difference in the distance, i.e., the length of the branch). From this figure, it appears that Sample 16 is most different from the whole dataset. Please note that these conclusions are only based on UV-VIS analysis, having an analytical window only covering light-absorbing molecules. Thus, this must be considered when interpreting similarities and differences among samples.

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

[20] Normalization is done based on total spectral intensity. Only absorbance values up to 450 nm are used (Helms et al., 2008).

→ **Stats_HCA** on EEM spectra

To employ HCA on EEM spectra, the EEM spectra must have been folded from 3D into 2D format using the **EEM_Fold** code. The new excitation-emission wavelength variables (e.g., EX300EM400, EX300EM401, …) are used as independent variables and the normalized[21] intensity values are used as dependent variables. To run the **Stats_HCA** code, set TEnvR\TestData_EEM\TestData_Generic_Processed as your Current Folder. You must have the file "EEM_AlignmentMatrix.xlsx" in there.

Slide 189: *Stats_HCA('EEM_AlignmentMatrix.xlsx','EEM')*

The produced dendrogram is saved as "HCA_EEM_AlignmentMatrix_EEM.png".

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

→ **Stats_HCA** on FT-ICR-MS formulas

To employ HCA on FT-ICR-MS molecular data, the exact masses of molecular formulas are used as independent variables and the normalized[22] magnitude values are uses as dependent variables. To run this code, set TEnvR\TestData_FTICRMS as your Current Folder. You must have the file "FTMS Alignment Matrix.xlsx" in there.

Slide 190: *Stats_HCA('FTMS Alignment Matrix.xlsx','FTICRMS')*

The produced dendrogram is saved as "HCA_FTMS Alignment Matrix_FTICRMS.png".

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

→ **Stats_HCA** on NMR spectra

To employ HCA on NMR spectra, the different chemical shifts are used as independent variables and the normalized[21] intensity values are used as dependent variables. To run this code, set TEnvR\TestData_NMR as your Current Folder. You must have the file "NMR Alignment Matrix.xlsx" in there.

Slide 191: *Stats_HCA('NMR Alignment Matrix.xlsx','NMR')*

The produced dendrogram is saved as "HCA_NMR Alignment Matrix_NMR.png".

[21] Normalization is done based on total spectral intensity.

[22] Normalization is done based on total spectral magnitude (TSS, Thompson et al., 2021).

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

→ **Stats_HCA** on mixed variables

HCA can be employed on datasets of lower dimensionality, such as sets of diverse variables from different analytical techniques. For example, a dataset of variables can be created by combining dissolved organic carbon, pH, salinity, select UV-VIS/EEM/FTMS metrics, NMR integrals, etc. Here, we will show an example of how to use the **Stats_HCA** code on mixed variables using the list of UV-VIS metrics that was generated earlier by the **UVVIS_Automation** code.

First, open the "UVVIS_MasterReport.xlsx" file located in TEnvR\TestData_UVVIS. Right-click on the sheet "Metrics" and select "Move or Copy…". In the popped up window, change "To Book" to "(new book)" and select "Create a copy" (**Slide 192**). The whole sheet will be opened into a new Excel file. Evaluate the data and make sure it is valid. Here, the matrix contains dilution factors in column B (not real measurements) and E_4:E_6 measurements with negative values (invalid). Thus, these variables need to be removed – delete columns B and K as shown on **Slide 193**. Save the file into your TEnvR\TestData_FTICRMS folder by clicking Ctrl + S on your keyboard or File → Save As. Save the file as "ExternalVariables.xlsx" (**Slide 193**).

Please note that the format of the matrix here is different. While in the previous files ("UVVIS_MasterReport.xlsx", "EEM_AlignmentMatrix.xlsx", "FTMS Alignment Matrix.xlsx", "NMR Alignment Matrix.xlsx"), sample names were as column labels and independent variables were as row labels. Here (in "ExternalVariables.xlsx") the data is transposed: sample names are as row labels and independent variables are as column labels. This is a common source of error in using the **Stats_HCA** code, and we urge users to be careful with their data format. The different data format is accounted for in the second argument of the function (i.e., *'UVVIS'*, *'EEM'*, *'NMR'*, *'FTICRMS'*, *'Variables'*).

Then, set TEnvR\TestData_FTICRMS as your Current Folder.

Slide 194: *Stats_HCA('ExternalVariables.xlsx','Variables')*

The produced dendrogram is saved as "HCA_ExternalVariables_Variables.png".

The **Stats_HCA** code can be modified to alter the type of distance metric and type of linkage criterion.

Slide 195: *edit Stats_HCA*

To modify the type of distance metric used, edit the value of the *Type_Distance* variable (line 9, **Slide 195**) to any of the following options: 'euclidean', *'squaredeuclidean'*, *'seuclidean'*, *'cityblock'*, *'minkowski'*, *'chebychev'*, *'mahalanobis'*, *'cosine'*, *'correlation'*, *'spearman'*, *'hamming'*, *'jaccard'*, or a customized function. A description of these types of distance can be found by typing *help pdist* in the Command Window or going to the *pdist* function listing on MathWorks (https://www.mathworks.com/help/stats/pdist.html?s_tid=srchtitle). We generally use Euclidian distances ('euclidean') in order to make the HCA results comparable with PCA analysis results.

To modify the type of linkage criterion used, change the value of the *Type_Tree* variable (line 10, **Slide 195**) to any of the following options: 'single', *'complete'*, *'average'*, *'weighted'*, *'centroid'*, *'median'* or *'ward'*. A description of these types of distance can be found by typing *help linkage* in the Command Window or going to the *linkage* function listing on MathWorks (https://www.mathworks.com/help/stats/linkage.html?s_tid=doc_ta). We generally use the unweighted pair group method with arithmetic mean (UPGMA) linkage criterion corresponding to value *'average'* for the *Type_Tree* variable (line 10, **Slide 195**). UPGMA is also known as the unweighted average distance method.

Please note that it is up to the researcher to determine what type of distance metric and linkage criterion are most appropriate for the HCA evaluation of their data. Other types of clustering, such as K-means and C-means Clustering, can also be useful in exploring multivariate data such as formula lists from FT-ICR-MS or spectral data (Cuss and Guéguen, 2016; Zhang et al., 2020), and researchers are welcome to explore such and other statistical approaches that will better suit their datasets.

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

- **Stats_PCA**

Principal component analysis (PCA) is another common technique for exploratory statistics employed in numerous areas of research (Gewers et al., 2021). Different from HCA, PCA evaluates inputted variables (known as observations) and determines which independent variables are most important for determining similarities or differences among samples. Thus, PCA is often viewed as a dimensionality reduction technique, as it also informs which variables are unimportant in explaining the variance of the dataset (i.e., which dimensions of the data are unnecessary). Commonly, PCA is employed on multivariate data such as formula lists from FT-ICR-MS (Hur et al., 2010; Sleighter et al., 2010; Sleighter et al., 2014a; Sleighter et al., 2014b; Gurganus et al., 2015; Sleighter et al., 2015; Antony et al., 2018), as well as EEM (Boehme et al., 2004) or NMR spectra (Hedenström et al., 2009; Wozniak et al., 2013; Wozniak et al., 2015; Willoughby et al., 2016) or datasets of mixed variables (Xue et al., 2011; Wozniak et al., 2020).

The mathematics behind PCA are complex and are described in detail elsewhere (Jolliffe and Cadima, 2016). Briefly, data must have been normalized on a per-sample basis. The codes of TEnvR do this by normalizing individual intensity values to total spectral intensity (UV-VIS, EEMs, NMR). For FT-ICR-MS data, peak magnitudes are normalized to total spectral magnitude (TSS-normalized) (Thompson et al., 2021) or using presence-absence normalization. Thus, the data loaded in the **Stats_PCA** code has already been normalized on a per-sample basis to account for differences in concentration (UV-VIS, EEMs, NMR) or ionizability (ESI-FT-ICR-MS) to make samples comparable to each other. Once data is loaded in the **Stats_PCA** code, data for each variable must also be normalized to make variables comparable to each other. The average is taken for values of each independent variable. The average is then subtracted from all values. This is known as mean-centering, i.e., the mean for each variable is shifted to zero. The normalization and mean-centering are critical to ensure that the resultant principal components capture only the variance of the dataset and are not influenced by the absolute quantity of the variables or differences among sample concentration/ionizability. Without these mathematical transformations, the first principal component may erroneously correspond to the direction of data mean instead of the direction of maximum variance. Mean-centered data is then used to compute a covariance matrix. Covariance is the measure of how much two variables vary together. Each variable of the dataset is correlated with all other variables, and the resultant covariance data is output into a matrix, hence, covariance matrix. Covariances values closer to ±1 indicate strong correlations, while values closer to zero indicate weak correlations. The covariance data is deconvoluted using one of the following algorithms: single-value decomposition (MATLAB function *svd*), eigenvalue decomposition (MATLAB function *eig*), or an alternating least squares low-rank matrix factorization (MATLAB function *alsmf*). In our experience, we have found that the overall PCA results from these three different algorithms do not differ significantly, even though the three different algorithms have their own benefits and downsides. For example, single-value decomposition is a more stable algorithm than the eigenvalue decomposition, but both are sensitive to missing values and errors in the covariance matrix. The alternating least squares low-rank matrix factorization approach, on the other hand, is insensitive to missing data and can fill in missing values using interpolation routines (Severson et al., 2017). We have exclusively used single-value decomposition routines in our previous research, because it is a faster and more stable routine, and because we pre-treat our alignment matrices to remove variables with missing values (either manually in Excel or automatically in MATLAB as described above). Thus, we have not found the need to use alternating least squares low-rank matrix factorization algorithms previously, and the **Stats_PCA** code is developed to employ single-value decomposition using the *svd* function. Researchers are welcome to further modify this script for incorporating different algorithms. Researchers are also welcome to also explore the MATLAB function *pca*, which is highly tunable and has versatile capabilities (to learn more about this function, type *help pca* in the Command Window). PCA exclusively uses Euclidian distances in its computations, and variables are evaluated using linear correlations. In cases when Euclidean distances and linear correlations are not appropriate, another technique must be used. Paliy and Shankar (2016) provide an excellent guide for choosing alternative approaches for exploratory statistics. Non-metric multidimensional scaling (NMDS) is a similar technique to PCA but is more versatile, and the type of distance metric can be altered (Kellerman et al., 2014; Kellerman et al., 2015). A code for NMDS will be developed for a future version of TEnvR.

In parallel with HCA, PCA is also a two-way technique, i.e., it requires an import of a matrix that has one dimension of aligned independent variables (molecular weight, chemical shift, wavelengths, etc.) and one dimension of sample-dependent variables (**Slide 187**). The **UVVIS_Automation**, **EEM_Fold**, **FTMS_AlignmentFormulas**, and **NMR_Automation** codes produce such alignment matrices in their outputted files (saved in a sheet named “Normalized”). It is critical that data is normalized to account for large concentration (in UV-VIS, EEM, or NMR data) or large ionization (in FT-ICR-MS data) differences among samples. It is also required that no independent variable has the same dependent variable for all samples. For example with UV-VIS data, if all samples have absorbance at 600 of 0 AU, then the wavelength of 600 must be removed from a dataset. All codes (**UVVIS_Automation**, **EEM_Fold**, **FTMS_AlignmentFormulas**, and **NMR_Automation**) include an algorithm that removes such variables.

The **Stats_PCA** code takes a matrix of normalized data and deconvolutes it using single-value decomposition. To learn more about this approach, type *help svd* in the Command Window. The **Stats_PCA** code is versatile and can be applied on different types of data as explained below. Please note that the larger the number of samples and number of variables, the longer it will take for the code to run.

→ **Stats_PCA** on UV-VIS spectra

To employ PCA on UV-VIS spectra, the different absorbance wavelengths are used as independent variables and the normalized[23] absorbance values are uses as dependent variables. To run this code, set TestData_UVVIS as your Current Folder. You must have the file "UVVIS_MasterReport.xlsx" in there. The code will search for a sheet named "Normalized" and will load the data from there.

Slide 196: *Stats_PCA('UVVIS_MasterReport.xlsx','UVVIS')*

The code produces an Excel file named "PCA_Results (UVVIS).xlsx" that contains four sheets: "Sheet1" containing the original data matrix to create a paper trail (**Slide 197**), "Scores" (**Slide 198**), "Loadings" (**Slide 199**), and "Eigenvalues %" (**Slide 200**). The number of principal components (PCs) is equal to the number of inputted independent variables. There is a score value for each sample and for each component. The percentage value in the principal component labels is the corresponding eigenvalue. Plotting the first principal component versus the second principal component (**Slide 198**) generally shows which samples are similar to each other and which ones are different. This is highly dependent on the type of data that was input. For example, samples that are similar based on UV-VIS data may be different based on NMR data, due to the two different analytical windows of the techniques.

The loadings are used to determine which variables cause the samples to differ or be similar among each other (**Slide 199**). The eigenvalues describe how much variance of the dataset each component explains (**Slide 200**). The eigenvalues are generally used to determine which components describe the data best. A common way to visualize eigenvalues is using a Scree plot (**Slide 200**). While there are quantitative approaches that can be used to determine which components are statistically significant (Jackson, 1993), some approaches such as the Kaiser-Guttman rule have been shown to be problematic (Preacher and MacCallum, 2003). Furthermore, to our knowledge, it is unknown if any quantitative approach has been tested and is acceptable for biogeochemical research and at present there are no guidelines for multivariate statistical analyses of FT-ICR-MS data. Thus, we generally choose components manually and we typically consider components with more than 10% explained variance to be important. In our experience, we have found that in an ideal case with NOM samples, more that 50% of the summed variance is explained by the components we select. Commonly, only PC1 and PC2 contribute majorly to the explained variance of the dataset. Rarely, we find that PC3 is also important (Gurganus et al., 2015; Willoughby et al., 2016). Other components usually show redundant and/or unimportant trends in the datasets we have previously worked with.

→ **Stats_PCA** on EEM spectra

To employ PCA on EEM spectra, the EEM spectra must have been folded from 3D into 2D format using the **EEM_Fold** code. The new excitation-emission wavelength variables (e.g., EX300EM400, EX300EM401, …) are used as independent variables and the normalized[24] intensity values are used as dependent variables. To run the **Stats_PCA** code, set TEnvR\TestData_EEM\TestData_Generic_Processed

[23] Normalization is done based on total spectral intensity. Only absorbance values up to 450 nm are used (Helms et al., 2008).

[24] Normalization is done based on total spectral intensity.

as your Current Folder. You must have the file "EEM_AlignmentMatrix.xlsx" in there. The code will search for a sheet named "Normalized" and will load the data from there.

Slide 201: *Stats_PCA('EEM_AlignmentMatrix.xlsx','EEM')*

The produced Excel file is named "PCA_Results (EEM).xlsx". The loadings are of the same format as of the matrix that was inputted – in a 2D folded format. Here, the **EEM_Unfold** code is used to convert the folded loadings into the traditional 3D EEM format.

Slide 202: *EEM_Unfold('PCA_Results (EEM).xlsx',1)*

The first argument of the function is the filename of the PCA report (*'PCA_Results (EEM).xlsx'*), and the second argument is the principal component number (*1*) that is to be unfolded. The code exports a new file in a .csv format named "PCA_Results (EEM)_comp1_unfold.csv" containing the data in a typical 3D EEM format (**Slide 203**). This data is then visualized using **EEM_Visualize**:

Slide 204: *EEM_Visualize('PCA_Results (EEM)_comp1_unfold.csv')*

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

→ **Stats_PCA** on FT-ICR-MS formulas

To employ PCA on FT-ICR-MS molecular data, the exact masses of molecular formulas are used as independent variables and the normalized[25] magnitude values are uses as dependent variables. To run this code, set TEnvR\TestData_FTICRMS as your Current Folder. You must have the "FTICRMS_AlignmentMatrix.xlsx" file in there. The code will search for a sheet named "Normalized" and will load the data from there.

Slide 205: *Stats_PCA('FTMS Alignment Matrix.xlsx','FTICRMS')*

The code produces an Excel file named "PCA_Results (FTICRMS).xlsx" with four sheets as described previously. When FT-ICR-MS formulas are used, an additional sheet "pvalues" is also produced (**Slide 206**). The code has a capability of computing p-values for each independent variable per each principal component. P-values are calculated by performing Pearson correlation among sample scores (for each component) and the mean-centered dependent variables (Yamamoto et al., 2014). These p-values are useful in determining which formulas are significant in driving the changes among principal components. We generally use a p-value threshold of 0.05 corresponding to confidence level (α) of 95%, but researchers are welcome to employ a more strict or liberal threshold.

[25] Normalization is done based on total spectral magnitude (TSS, Thompson et al., 2021).

→ **Stats_PCA** on NMR spectra

To employ PCA on NMR spectra, the different chemical shifts are used as independent variables and the normalized[26] intensity values are uses as dependent variables. To run this code, set TEnvR\TestData_NMR as your Current Folder. You must have the file “NMR Alignment Matrix.xlsx” in there. The code will search for a sheet named “Normalized” and will load the data from there.

Slide 207: *Stats_PCA('NMR Alignment Matrix.xlsx','NMR')*

The code produces an Excel file named “PCA_Results (NMR).xlsx” with four sheets as described previously. Please note that here only seven samples were used for the example of the code. In our experience, a minimum of 8 samples and 10 variables is generally required to obtain a PCA model, however, model robustness may be of question with such limited matrix size.

→ **Stats_PCA** on mixed variables

PCA can be employed on datasets of lower dimensionality, such as sets of diverse variables from different analytical techniques. For example, a dataset of variables can be created by combining dissolved organic carbon, pH, salinity, select UV-VIS/EEM/FTMS metrics, NMR integrals, etc. An example is a recent study by Wozniak et al. (2020) where FTMS metrics, CompoundClass statistics, NMR integrals, and benzenepolycarboxylic acids[27] molecular marker measurements were used in a PCA investigation to study the heterogeneity of pyrogenic dissolved organic matter. Here, we will show an example of how to use the **Stats_PCA** code on mixed variables using the list of UV-VIS metrics that was generated earlier by the **UVVIS_Automation** code.

Here, we will use the “ExternalVariables.xlsx” matrix that is located in the TEnvR\TestData_FTICRMS folder (**Slide 208**) that was generated as described above in the HCA section. Please note that the format of the matrix here is different. While in the previous files (“UVVIS_MasterReport.xlsx”, “EEM_AlignmentMatrix.xlsx”, “FTMS Alignment Matrix.xlsx”, “NMR Alignment Matrix.xlsx”) sample names were as column labels and independent variables were as row labels, here (in “ExternalVariables.xlsx”) the data is transposed: sample names are as row labels and independent variables are as column labels. This is a common source of error in using the **Stats_PCA** code and we urge users to be careful with their data format. The different data format is accounted for in the second argument of the function (i.e., *'UVVIS'*, *'EEM'*, *'NMR'*, *'FTICRMS'*, *'Variables'*).

Slide 208: *Stats_PCA('ExternalVariables','Variables')*

The code produces an Excel file named “PCA_Results (Variables).xlsx” with sheets “Sheet1”, “Scores”, “Loadings”, “Eigenvalues %”, and “pvalues”.

[26] Normalization is done based on total spectral intensity.

[27] The benzenepolycarboxylic acid (BPCA) molecular marker method is a technique for quantifying condensed aromatic compounds (ConAC) in environmental matrices (Glaser et al., 1998; Chang et al., 2018).

Please note that if **Stats_PCA** is employed on a matrix with more than 20 variables, the code will only export parameters (scores, loadings, eigenvalues, and p-values when applicable) for the first 20 principal components and the other components (PC21, PC22, …) will not be exported. Users are welcome to modify the code to export a larger range of components if that is desired.

Statistical evaluation of multivariate data using PCA analysis is often considered modeling. This is because the dataset (data matrix) being evaluated often needs refinement for removal of outlier samples and removal/addition of variables. Excluding sample outliers in order to refine the PCA model can reveal more distinctive trends in comparison to using the whole dataset (Wozniak et al., 2013). Analogously, the set of variables can be also modified to present trends among samples in a clearer way. For example, in a dataset of molecular data from global rivers, Wagner et al. (2015b) did not use all molecular formulas, but only formulas corresponding to condensed aromatic molecules. They extracted the data for this particular compound class and used those molecular formulas as variables in the PCA modeling, revealing trends among molecular data and watershed characteristics.

It must be also noted that while there are various statistical tools that can be employed on multivariate data (Paliy and Shankar, 2016), there is limited information about appropriateness for the different analytical techniques described in this tutorial. For FT-ICR-MS formula lists particularly, data has been explored using PCA (Sleighter et al., 2010; Sleighter et al., 2014a; Sleighter et al., 2014b), Principal Coordinate Analysis (PCoA, (Hawkes et al., 2020)), NMDS (Kellerman et al., 2014; Kellerman et al., 2015), and 2D correlation (Abdulla et al., 2013; Sleighter et al., 2014b). While comparisons of different statistical techniques exist (Gauch Jr. and Whittaker, 1972), limited information about the applicability of each technique is available for environmental data and especially FT-ICR-MS molecular data.

- **Stats_CorrMatrix**

This code is specifically used for generating a correlation matrix, an approach where each variable is correlated with each variable. Three types of correlations are possible: Pearson, Kendall, and Spearman. The user must determine which type is most appropriate for their data. We provide some generic guidelines below.

Pearson correlation is appropriate for continuous or interval variables. An interval variable is one that has a meaningful difference (i.e., interval) among each two data points. The Pearson correlation also requires variables that are normally distributed (i.e., having a Gaussian “bell-shaped” distribution). Such distribution is also referred to as parametric distribution, and thus, the Pearson correlation is often classified as a parametric statistical test. A significant limitation of the Pearson correlation is that it is sensitive to outliers and does not perform well with small sample sets. A major difference between Pearson correlations and Kendall/Spearman correlations is that Pearson assesses how linearly related are the variables being tested (i.e., changing at a constant rate = slope), while Kendall/Spearman correlations assess monotonic relationships among the variables being tested. Monotonic relationships measure the likelihood of two variables moving in the same direction, but not necessarily at a constant rate. Thus, a linear relationship is not a requirement for Kendall and Spearman correlations, and thus, they may assess data of any shape (exponential, polynomial, etc.).

Kendall correlation is often used when the data does not fit the requirements for Pearson correlations: Kendall correlations can be used on non-continuous data, non-interval data, non-normally distributed (non-parametric) data, and non-linearly related data.
Spearman correlation is similar to the Kendall correlation in terms of requirements: it can be used on non-continuous data, non-normally distributed (non-parametric) data, and non-linearly related data. Kendall is sometimes preferred over Spearman, because Kendall is simpler to compute and has less associated errors. In statistical terms, Kendall has smaller gross error sensitivity (i.e., is more robust) as well as smaller asymptotic variance (i.e., is more efficient) (Croux and Dehon, 2010). However, a major benefit of the Spearman correlation is that its correlation coefficient (Spearman's Rho, ρ) has a similar meaning to the coefficient of determination R^2 (i.e., explained variance). By contrast, the Kendall's correlation coefficient (Kendall's Tau, τ) is more of a test statistic for nonlinear correlations. In the environmental sciences, it appears that Pearson and Spearman are most commonly used types of correlation.

To show the application of this code, we will use the "ExternalVariables.xlsx" matrix located in the TEnvR\TestData_FTICRMS folder. This matrix was generated as described above in the HCA section. Please set your Current Folder to be TEnvR\TestData_FTICRMS.

Slide 209: *Stats_CorrMatrix('ExternalVariables.xlsx','Pearson')*

The code will produce a file named "CorrMatrix_Pearson.xlsx" containing the original data in "Sheet1" to promote a paper-trail (**Slide 210**), the Pearson correlation coefficients in the second sheet (named "Coeffs", **Slide 211**), the squared Pearson correlation coefficients in the second sheet (named "Coeffs2", **Slide 212**), and the p-values for each correlation in the fourth sheet (named "pvalues", **Slide 213**).

By changing the second function argument from *'Pearson'* to *'Kendall'* or *'Spearman'*, you can change the type of correlation analysis you want to perform on the data. The produced Excel files are named "CorrMatrix_Kendall.xlsx" and "CorrMatrix_Spearman.xlsx", respectively.

Slide 214: *Stats_CorrMatrix('ExternalVariables.xlsx','Kendall')*
Slide 215: *Stats_CorrMatrix('ExternalVariables.xlsx','Spearman')*

Before proceeding, type *close all*, *clear,* and then *clc* in the Command Window.

- **FTMS_SpearmanCorrelation**

Spearman correlations are incredibly useful in mathematically coupling FT-ICR-MS and other (external) parameters such as UV-VIS (Berg et al., 2019) or fluorescence/PARAFAC metrics (Stubbins et al., 2014). It correlates the normalized[28] magnitude values of molecular formulas common among all samples with values of the external parameters. The correlation is done for each formula and for each external parameter.

[28] Normalization is done based on total spectral magnitude (TSS, Thompson et al., 2021).

The **FTMS_SpearmanCorelation** code requires two different inputs: aligned and magnitude-normalized molecular formulas as well as external parameters. The formulas are loaded from the “FTMS Alignment Matrix.xlsx” file. The code will search for a sheet named “Normalized Common” and will load the data from there. The code also requires a sheet of external parameters. Here, we will use the “ExternalVariables.xlsx” file with UV-VIS metrics and correlate them with the common formulas among the 20 example FT-ICR-MS samples. To run this code, set TestData_FTICRMS as your Current Folder. Please note that the code will not run properly if any of the variables are labeled with symbols such as / ? * [ ].

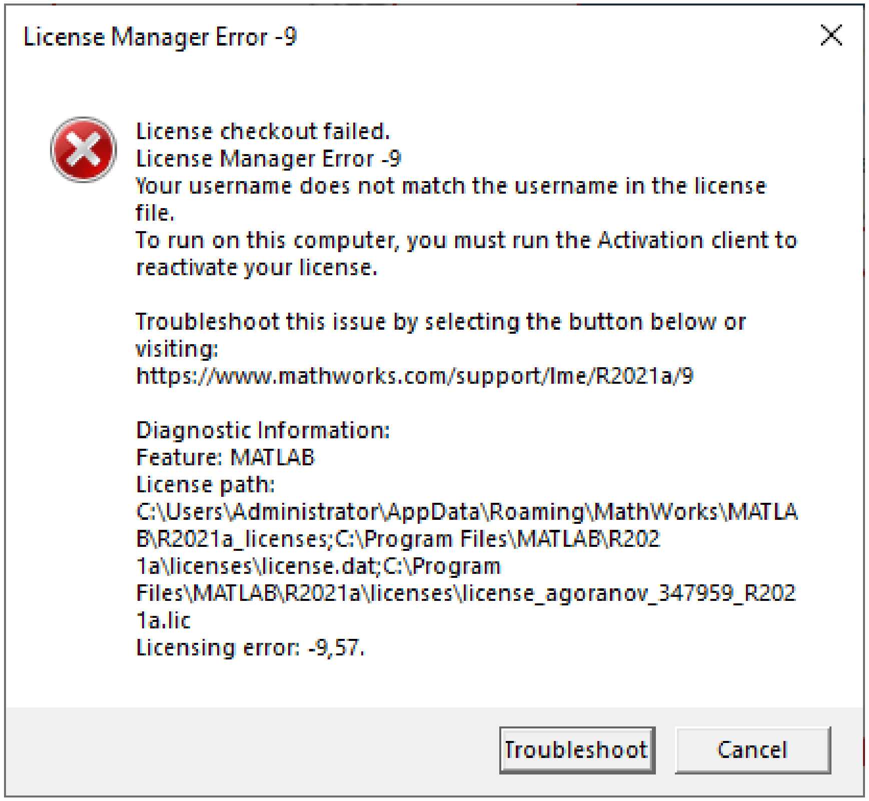
Slide 216: *FTMS_SpearmanCorrelation('FTMS Alignment Matrix.xlsx','ExternalVariables.xlsx')*

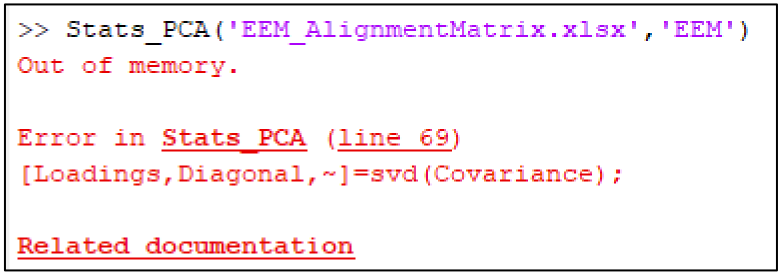
The code will output a van Krevelen diagram for each metric (here, 15 metrics ≡ 15 figures), which are saved as “vK_Spearman_MetricName.png” (e.g., “vK_Spearman_Abs-230 (dec).png”, “vK_Spearman_Slope(275-295).png”, etc.). Formulas significantly positively correlated ($p < 0.05$) with the metrics are colored in red, negatively correlated ($p < 0.05$) in blue, and formulas that are not significantly correlated are colored in gray ($p \geq 0.05$). Corresponding Venn diagram statistics (number of formulas + number-based percentages) are shown in the legend. It should be noted that the legend can be moved around on the plot, simply by clicking on it and then moving it to the desired location.

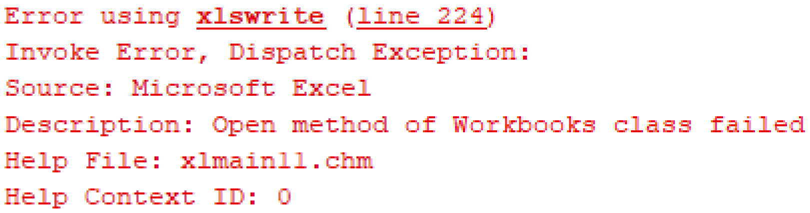
The code also outputs an Excel file (“FTMS_SpearmanResults.xlsx”) containing all original molecular formula data (from “FTMS Alignment Matrix.xlsx”) in “Sheet1” (**Slide 217**), all original external variables data (from “ExternalVariables.xlsx”) in sheet “ExternalVar” (**Slide 218**), and the molecular formula correlations for each external variable in other sheets named with the external variable name (example shown for Abs-230 (dec) on **Slide 219**). Molecular formula information is listed along with Spearman’s rank correlation coefficient ρ (Greek letter rho) (labeled r-value in column N), the t-statistic (labeled as t-value in column O), and the p-value in column P.

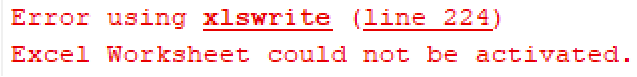
Section 9. Troubleshooting (Common Errors)

In this section, we describe the most common errors we have experienced while using TEnvR. If users experience an error that is not listed below, we first recommend checking for formatting issues. The codes of TEnvR require specific format of the files that are to be loaded. We provide examples in this tutorial and describe the formats in detail above. For persisting issues, users are welcome to communicate with the corresponding authors via email, ResearchGate, or the website of TEnvR.

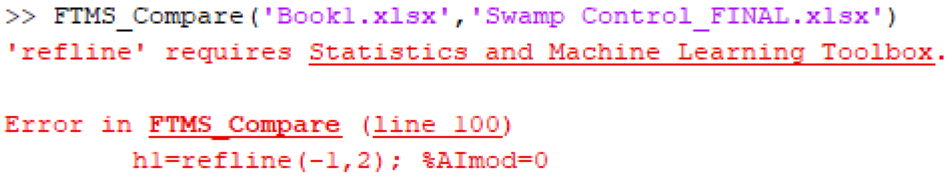
Error Message	Solution
Error (when attempting to open MATLAB): *License Manager Error -9*  License Manager Error -9 License checkout failed. License Manager Error -9 Your username does not match the username in the license file. To run on this computer, you must run the Activation client to reactivate your license. Troubleshoot this issue by selecting the button below or visiting: https://www.mathworks.com/support/lme/R2021a/9 Diagnostic Information: Feature: MATLAB License path: C:\Users\Administrator\AppData\Roaming\MathWorks\MATLAB\R2021a_licenses;C:\Program Files\MATLAB\R2021a\licenses\license.dat;C:\Program Files\MATLAB\R2021a\licenses\license_agoranov_347959_R2021a.lic Licensing error: -9,57. 	There is an issue with the MATLAB license. Reactivation of the MATLAB license with the activation client generally fixes this issue. Activation agent is found in C:\Program Files\MATLAB\R2021a\bin\win64\activatematlab.exe Please follow the instructions on the following link: https://www.mathworks.com/matlabcentral/answers/99067-why-do-i-receive-license-manager-error-9

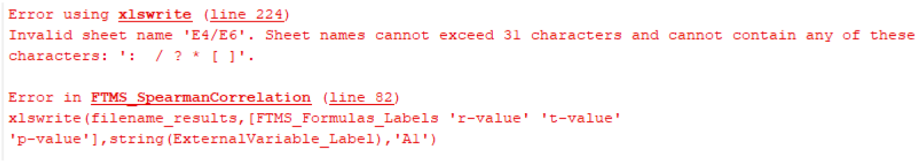
Error: *Out of memory.*

`>> Stats_PCA('EEM_AlignmentMatrix.xlsx','EEM')`
`Out of memory.`
 	
`Error in Stats_PCA (line 69)`
`[Loadings,Diagonal,~]=svd(Covariance);`
 	
`Related documentation`
	The computer cannot handle the size of the dataset. Please run the code on a more powerful computer or a supercomputer.

Error: *Invoke Error, Dispatch Exception*

`Error using xlswrite (line 224)`
`Invoke Error, Dispatch Exception:`
`Source: Microsoft Excel`
`Description: Open method of Workbooks class failed`
`Help File: xlmain11.chm`
`Help Context ID: 0`
	Type *clear all* in the Command Window.
	
	If error persists, restart MATLAB.
	
	If error persists, close MATLAB. Open task manager (Ctrl+Shift+Esc), go to the tab Details and close any processes related to Excel.

Error: *Excel Worksheet could not be activated.*

`Error using xlswrite (line 224)`
`Excel Worksheet could not be activated.`
	Type *clear all* in the Command Window.
	
	Open task manager (Ctrl+Shift+Esc), go to the tab Details and close any processes related to Excel.
	
	If error persists, restart MATLAB. If error persists, restart your computer.

Error: *PNG library failed: Could not open file.*	Filename is too long. Reduce filename.

Error: *'refline' requires Statistics and Machine Learning Toolbox.*

`>> FTMS_Compare('Book1.xlsx','Swamp Control_FINAL.xlsx')`
`'refline' requires Statistics and Machine Learning Toolbox.`
 	
`Error in FTMS_Compare (line 100)`
`        hl=refline(-1,2); %AImod=0`
	One of the functions requires an additional toolbox. Please click on the required toolbox (the red underlined toolbox statement in the command window is clickable). You will be redirected to MathWorks where you will be asked to install the missing toolbox. See **Slide 2** for required toolboxes for TEnvR).

Error: *Invalid sheet name 'E4/E6'. Sheet names cannot exceed 31 characters and cannot contain any of these characters: ': / ? * [ ] '*  Error using **xlswrite** (line 224) Invalid sheet name 'E4/E6'. Sheet names cannot exceed 31 characters and cannot contain any of these characters: ': / ? * [ ]'. Error in **FTMS_SpearmanCorrelation** (line 82) xlswrite(filename_results,[FTMS_Formulas_Labels 'r-value' 't-value' 'p-value'],string(ExternalVariable_Label),'A1')	You used an illegal character (here, a slash /). Remove the character from the label.

References

Abdulla, H.A.N., Sleighter, R.L. and Hatcher, P.G. (2013) Two dimensional correlation analysis of Fourier transform ion cyclotron resonance mass spectra of dissolved organic matter: A new graphical analysis of trends. Analytical Chemistry 85, 3895-3902, https://doi.org/10.1021/ac303221j.

Antony, R., Willoughby, A.S., Grannas, A.M., Catanzano, V., Sleighter, R.L., Thamban, M., Hatcher, P.G. and Nair, S. (2017) Molecular insights on dissolved organic matter transformation by supraglacial microbial communities. Environmental Science & Technology 51, 4328-4337, https://doi.org/10.1021/acs.est.6b05780.

Antony, R., Willoughby, A.S., Grannas, A.M., Catanzano, V., Sleighter, R.L., Thamban, M. and Hatcher, P.G. (2018) Photo-biochemical transformation of dissolved organic matter on the surface of the coastal East Antarctic ice sheet. Biogeochemistry 141, 229-247, https://doi.org/10.1007/s10533-018-0516-0.

Bae, E., Yeo, I.J., Jeong, B., Shin, Y., Shin, K.-H. and Kim, S. (2011) Study of double bond equivalents and the numbers of carbon and oxygen atom distribution of dissolved organic matter with negative-mode FT-ICR MS. Analytical Chemistry, 4193-4199, https://doi.org/10.1021/ac200464q.

Baker, A. (2001) Fluorescence Excitation−Emission Matrix Characterization of Some Sewage-Impacted Rivers. Environmental Science & Technology 35, 948-953, https://doi.org/10.1021/es000177t.

Berg, S.M., Whiting, Q.T., Herrli, J.A., Winkels, R., Wammer, K.H. and Remucal, C.K. (2019) The Role of Dissolved Organic Matter Composition in Determining Photochemical Reactivity at the Molecular Level. Environmental Science & Technology 53, 11725-11734, https://doi.org/10.1021/acs.est.9b03007.

Bianca, M.R., Baluha, D.R., Gonsior, M., Schmitt-Kopplin, P., Del Vecchio, R. and Blough, N.V. (2020) Contribution of ketone/aldehyde-containing compounds to the composition and optical properties of Suwannee River fulvic acid revealed by ultrahigh resolution mass spectrometry and deuterium labeling. Analytical and Bioanalytical Chemistry 412, 1441-1451, https://doi.org/10.1007/s00216-019-02377-x.

Boehme, J., Coble, P., Conmy, R. and Stovall-Leonard, A. (2004) Examining CDOM fluorescence variability using principal component analysis: seasonal and regional modeling of three-dimensional fluorescence in the Gulf of Mexico. Marine Chemistry 89, 3-14, https://doi.org/10.1016/j.marchem.2004.03.019.

Boiteau, R.M., Till, C.P., Ruacho, A., Bundy, R.M., Hawco, N.J., McKenna, A.M., Barbeau, K.A., Bruland, K.W., Saito, M.A. and Repeta, D.J. (2016) Structural Characterization of Natural Nickel and Copper Binding Ligands along the US GEOTRACES Eastern Pacific Zonal Transect. Frontiers in Marine Science 3, https://doi.org/10.3389/fmars.2016.00243.

Bostick, K.W., Zimmerman, A.R., Wozniak, A.S., Mitra, S. and Hatcher, P.G. (2018) Production and composition of pyrogenic dissolved organic matter from a logical series of laboratory-generated chars. Frontiers in Earth Science 6, 1-14, https://doi.org/10.3389/feart.2018.00043.

Bro, R. (1997) PARAFAC. Tutorial and applications. Chemometrics and Intelligent Laboratory Systems 38, 149-171, https://doi.org/10.1016/s0169-7439(97)00032-4.

Brown, T.L. and Rice, J.A. (2000) Effect of Experimental Parameters on the ESI FT-ICR Mass Spectrum of Fulvic Acid. Analytical Chemistry 72, 384-390, https://doi.org/10.1021/ac9902087.

Cabaniss, S.E. and Shuman, M.S. (1987) Synchronous fluorescence spectra of natural waters: tracing sources of dissolved organic matter. Marine Chemistry 21, 37-50, https://doi.org/10.1016/0304-4203(87)90028-4.

Chang, Z., Tian, L., Li, F., Zhou, Y., Wu, M., Steinberg, C.E.W., Dong, X., Pan, B. and Xing, B. (2018) Benzene polycarboxylic acid - A useful marker for condensed organic matter, but not for only pyrogenic black carbon. Science of The Total Environment 626, 660-667, https://doi.org/10.1016/j.scitotenv.2018.01.145.

Chen, H., Stubbins, A. and Hatcher, P.G. (2011) A mini-electrodialysis system for desalting small volume saline samples for Fourier transform ion cyclotron resonance mass spectrometry. Limnology and Oceanography: Methods 9, 582-592, https://doi.org/10.4319/lom.2011.9.582.

Chen, H., Stubbins, A., Perdue, E.M., Green, N.W., Helms, J.R., Mopper, K. and Hatcher, P.G. (2014) Ultrahigh resolution mass spectrometric differentiation of dissolved organic matter isolated by coupled reverse osmosis-electrodialysis from various major oceanic water masses. Marine Chemistry 164, 48-59, https://doi.org/10.1016/j.marchem.2014.06.002.

Chen, H., Johnston, R.C., Mann, B.F., Chu, R.K., Tolic, N., Parks, J.M. and Gu, B. (2017) Identification of Mercury and Dissolved Organic Matter Complexes Using Ultrahigh Resolution Mass Spectrometry. Environmental Science & Technology Letters 4, 59-65, https://doi.org/10.1021/acs.estlett.6b00460.

Chen, Y., Senesi, N. and Schnitzer, M. (1977) Information Provided on Humic Substances by E4/E6 Ratios. Soil Science Society of America Journal 41, 352-358, https://doi.org/10.2136/sssaj1977.03615995004100020037x.

Cho, Y., Birdwell, J.E., Hur, M., Lee, J., Kim, B. and Kim, S. (2017) Extension of the Analytical Window for Characterizing Aromatic Compounds in Oils Using a Comprehensive Suite of High-Resolution Mass Spectrometry Techniques and Double Bond Equivalence versus Carbon Number Plot. Energy & Fuels 31, 7874-7883, https://doi.org/10.1021/acs.energyfuels.7b00962.

Coble, P.G., Schultz, C.A. and Mopper, K. (1993) Fluorescence contouring analysis of DOC intercalibration experiment samples: a comparison of techniques. Marine Chemistry 41, 173-178, https://doi.org/10.1016/0304-4203(93)90116-6.

Coble, P.G., Lead, J., Baker, A., Reynolds, D.M. and Spencer, R.G.M. (2014) Aquatic Organic Matter Fluorescence. Cambridge University Press, New York, NY.

Cooper, W.T., Llewelyn, J.M., Bennett, G.L., Stenson, A.C. and Salters, V.J.M. (2005) Organic phosphorus speciation in natural waters by mass spectrometry, in: Turner, B.L., Frossard, E., Baldwin, D.S. (Eds.), Organic phosphorus in the environment. CABI Publishing, Wallingford, UK, pp. 45-74.

Cory, R.M. and McKnight, D.M. (2005) Fluorescence Spectroscopy Reveals Ubiquitous Presence of Oxidized and Reduced Quinones in Dissolved Organic Matter. Environmental Science & Technology 39, 8142-8149, https://doi.org/10.1021/es0506962.

Cory, R.M., Miller, M.P., McKnight, D.M., Guerard, J.J. and Miller, P.L. (2010) Effect of instrument-specific response on the analysis of fulvic acid fluorescence spectra. Limnology and Oceanography: Methods 8, 67-78, https://doi.org/10.4319/lom.2010.8.0067.

Croux, C. and Dehon, C. (2010) Influence functions of the Spearman and Kendall correlation measures. Statistical Methods & Applications 19, 497-515, https://doi.org/10.1007/s10260-010-0142-z.

Cuss, C.W. and Guéguen, C. (2016) Analysis of dissolved organic matter fluorescence using self-organizing maps: mini-review and tutorial. Analytical Methods 8, 716-725, https://doi.org/10.1039/c5ay02549d.

D'Andrilli, J., Fischer, S.J. and Rosario-Ortiz, F.L. (2020) Advancing critical applications of high resolution mass spectrometry for DOM assessments: Re-engaging with mass spectral principles, limitations, and data analysis. Environmental Science & Technology 54, 11654-11656, https://doi.org/10.1021/acs.est.0c04557.

De Haan, H. and De Boer, T. (1987) Applicability of light absorbance and fluorescence as measures of concentration and molecular size of dissolved organic carbon in humic Lake Tjeukemeer. Water Research 21, 731-734, https://doi.org/10.1016/0043-1354(87)90086-8.

Dittmar, T. and Koch, B.P. (2006) Thermogenic organic matter dissolved in the abyssal ocean. Marine Chemistry 102, 208-217, https://doi.org/10.1016/j.marchem.2006.04.003.

Dittmar, T., Koch, B., Hertkorn, N. and Kattner, G. (2008) A simple and efficient method for the solid-phase extraction of dissolved organic matter (SPE-DOM) from seawater. Limnology and Oceanography: Methods 6, 230-235, https://doi.org/10.4319/lom.2008.6.230.

Dvorski, S.E., Gonsior, M., Hertkorn, N., Uhl, J., Muller, H., Griebler, C. and Schmitt-Kopplin, P. (2016) Geochemistry of dissolved organic matter in a spatially highly resolved groundwater petroleum hydrocarbon plume cross-section. Environmental Science & Technology 50, 5536-5546, https://doi.org/10.1021/acs.est.6b00849.

Gauch Jr., H.G. and Whittaker, R.H. (1972) Comparison of Ordination Techniques. Ecology 53, 868-875, https://doi.org/10.2307/1934302.

Gewers, F.L., Ferreira, G.R., Arruda, H.F.D., Silva, F.N., Comin, C.H., Amancio, D.R. and Costa, L.D.F. (2021) Principal Component Analysis: A Natural Approach to Data Exploration. ACM Computing Surveys 54, 1-34, https://doi.org/10.1145/3447755.

Glaser, B., Haumaier, L., Guggenberger, G. and Zech, W. (1998) Black carbon in soils: The use of benzenecarboxylic acids as specific markers. Organic Geochemistry 29, 811-819, https://doi.org/10.1016/S0146-6380(98)00194-6.

Gonsior, M., Peake, B.M., Cooper, W.T., Podgorski, D., D'Andrilli, J. and Cooper, W.J. (2009) Photochemically induced changes in dissolved organic matter identified by ultrahigh resolution Fourier transform ion cyclotron resonance mass spectrometry. Environmental Science & Technology 43, 698-703, https://doi.org/10.1021/es8022804.

Gonsior, M., Hertkorn, N., Hinman, N., Dvorski, S.E., Harir, M., Cooper, W.J. and Schmitt-Kopplin, P. (2018) Yellowstone hot springs are organic chemodiversity hot spots. Scientific Reports 8, 1-13, https://doi.org/10.1038/s41598-018-32593-x.

Goranov, A.I., Wozniak, A.S., Bostick, K.W., Zimmerman, A.R., Mitra, S. and Hatcher, P.G. (2020) Photochemistry after fire: Structural transformations of pyrogenic dissolved organic matter elucidated by advanced analytical techniques. Geochimica et Cosmochimica Acta 290, 271-292, https://doi.org/10.1016/j.gca.2020.08.030.

Goranov, A.I., Wozniak, A.S., Bostick, K.W., Zimmerman, A.R., Mitra, S. and Hatcher, P.G. (2022) Microbial labilization and diversification of pyrogenic dissolved organic matter. Biogeosciences 19, 1491-1514, https://doi.org/10.5194/bg-19-1491-2022.

Goranov, A.I., Wozniak, A.S., Bostick, K.W., Zimmerman, A.R., Mitra, S. and Hatcher, P.G. (in review) Labilization and diversification of pyrogenic dissolved organic matter by microbes. Biogeochemistry.

Green, N.W., Perdue, E.M., Aiken, G.R., Butler, K.D., Chen, H., Dittmar, T., Niggemann, J. and Stubbins, A. (2014) An intercomparison of three methods for the large-scale isolation of oceanic dissolved organic matter. Marine Chemistry 161, 14-19, https://doi.org/10.1016/j.marchem.2014.01.012.

Green, S.A. and Blough, N.V. (1994) Optical absorption and fluorescence properties of chromophoric dissolved organic matter in natural waters. Limnology and Oceanography 39, 1903-1916, https://doi.org/10.4319/lo.1994.39.8.1903.

Günther, U.L., Ludwig, C. and Rüterjans, H. (2000) NMRLAB—Advanced NMR Data Processing in Matlab. Journal of Magnetic Resonance 145, 201-208, https://doi.org/10.1006/jmre.2000.2071.

Gurganus, S.C., Wozniak, A.S. and Hatcher, P.G. (2015) Molecular characteristics of the water soluble organic matter in size-fractionated aerosols collected over the North Atlantic Ocean. Marine Chemistry 170, 37-48, https://doi.org/10.1016/j.marchem.2015.01.007.

Halouska, S. and Powers, R. (2006) Negative impact of noise on the principal component analysis of NMR data. Journal of Magnetic Resonance 178, 88-95, https://doi.org/10.1016/j.jmr.2005.08.016.

Harris, D.C. (2015) Quantitative chemical analysis. W. H. Freeman.

Hawkes, J.A., D'Andrilli, J., Agar, J.N., Barrow, M.P., Berg, S.M., Catalán, N., Chen, H., Chu, R.K., Cole, R.B., Dittmar, T., Gavard, R., Gleixner, G., Hatcher, P.G., He, C., Hess, N.J., Hutchins, R.H.S., Ijaz, A., Jones, H.E., Kew, W., Khaksari, M., Palacio Lozano, D.C., Lv, J., Mazzoleni, L.R., Noriega-Ortega, B.E., Osterholz, H., Radoman, N., Remucal, C.K., Schmitt, N.D., Schum, S.K., Shi, Q., Simon, C., Singer, G., Sleighter, R.L., Stubbins, A., Thomas, M.J., Tolic, N., Zhang, S., Zito, P. and Podgorski, D.C. (2020) An international laboratory comparison of dissolved organic matter composition by high resolution mass spectrometry: Are we getting the same answer? Limnology and Oceanography: Methods 18, 235-258, https://doi.org/10.1002/lom3.10364.

He, Z., Guo, M., Sleighter, R.L., Zhang, H., Chanel, F. and Hatcher, P.G. (2018) Characterization of defatted cottonseed meal-derived pyrolysis bio-oil by ultrahigh resolution electrospray ionization Fourier transform ion cyclotron resonance mass spectrometry. Journal of Analytical and Applied Pyrolysis 136, 96-106, https://doi.org/10.1016/j.jaap.2018.10.018.

He, Z., Sleighter, R.L., Hatcher, P.G., Liu, S., Wu, F., Zou, H. and Olanya, O.M. (2019) Molecular level comparison of water extractives of maple and oak with negative and positive ion ESI FT-ICR mass spectrometry. Journal of Mass Spectrometry 54, 655-666, https://doi.org/10.1002/jms.4379.

Hedenström, M., Wiklund-Lindström, S., Öman, T., Lu, F., Gerber, L., Schatz, P., Sundberg, B. and Ralph, J. (2009) Identification of Lignin and Polysaccharide Modifications in Populus Wood by Chemometric Analysis of 2D NMR Spectra from Dissolved Cell Walls. Molecular Plant 2, 933-942, https://doi.org/10.1093/mp/ssp047.

Helms, J.R., Stubbins, A., Ritchie, J.D., Minor, E.C., Kieber, D.J. and Mopper, K. (2008) Absorption spectral slopes and slope ratios as indicators of molecular weight, source, and photobleaching of chromophoric dissolved organic matter. Limnology and Oceanography 53, 955-969, https://doi.org/10.4319/lo.2008.53.3.0955.

Hemmler, D., Gonsior, M., Powers, L.C., Marshall, J.W., Rychlik, M., Taylor, A.J. and Schmitt-Kopplin, P. (2019) Simulated sunlight selectively modifies Maillard reaction products in a wide array of chemical reactions. Chem. Eur. J. 25, 13208-13217, https://doi.org/10.1002/chem.201902804.

Hertkorn, N., Benner, R., Frommberger, M., Schmitt-Kopplin, P., Witt, M., Kaiser, K., Kettrup, A. and Hedges, J.I. (2006) Characterization of a major refractory component of marine dissolved organic matter. Geochimica et Cosmochimica Acta 70, 2990-3010, https://doi.org/10.1016/j.gca.2006.03.021.

Hertkorn, N., Ruecker, C., Meringer, M., Gugisch, R., Frommberger, M., Perdue, E., Witt, M. and Schmitt-Kopplin, P. (2007) High-precision frequency measurements: Indispensable tools at the core of the molecular-level analysis of complex systems. Analytical and Bioanalytical Chemistry 389, 1311-1327, https://doi.org/10.1007/s00216-007-1577-4.

Hertkorn, N., Harir, M., Koch, B.P., Michalke, B. and Schmitt-Kopplin, P. (2013) High-field NMR spectroscopy and FTICR mass spectrometry: Powerful discovery tools for the molecular level characterization of marine dissolved organic matter. Biogeosciences 10, 1583-1624, https://doi.org/10.5194/bg-10-1583-2013.

Herzsprung, P., Hertkorn, N., von Tümpling, W., Harir, M., Friese, K. and Schmitt-Kopplin, P. (2014) Understanding molecular formula assignment of Fourier transform ion cyclotron resonance mass spectrometry data of natural organic matter from a chemical point of view. Analytical and Bioanalytical Chemistry 406, 7977-7987, https://doi.org/10.1007/s00216-014-8249-y.

Herzsprung, P., Hertkorn, N., von Tumpling, W., Harir, M., Friese, K. and Schmitt-Kopplin, P. (2016) Molecular formula assignment for dissolved organic matter (DOM) using high-field FT-ICR-MS: chemical perspective and validation of sulphur-rich organic components (CHOS) in pit lake samples. Analytical and Bioanalytical Chemistry 408, 2461-2469, https://doi.org/10.1007/s00216-016-9341-2.

Hockaday, W.C., Grannas, A.M., Kim, S. and Hatcher, P.G. (2006) Direct molecular evidence for the degradation and mobility of black carbon in soils from ultrahigh-resolution mass spectral analysis of dissolved organic matter from a fire-impacted forest soil. Organic Geochemistry 37, 501-510, https://doi.org/10.1016/j.orggeochem.2005.11.003.

Hockaday, W.C., Purcell, J.M., Marshall, A.G., Baldock, J.A. and Hatcher, P.G. (2009) Electrospray and photoionization mass spectrometry for the characterization of organic matter in natural waters: A qualitative assessment. Limnology and Oceanography: Methods 7, 81-95, https://doi.org/10.4319/lom.2009.7.81.

Hsu, C.S., Lobodin, V.V., Rodgers, R.P., McKenna, A.M. and Marshall, A.G. (2011) Compositional Boundaries for Fossil Hydrocarbons. Energy & Fuels 25, 2174-2178, https://doi.org/10.1021/ef2004392.

Hu, C., Muller-Karger, F.E. and Zepp, R.G. (2002) Absorbance, absorption coefficient, and apparent quantum yield: A comment on common ambiguity in the use of these optical concepts. Limnology and Oceanography 47, 1261-1267, https://doi.org/10.4319/lo.2002.47.4.1261.

Hughey, C.A., Hendrickson, C.L., Rodgers, R.P., Marshall, A.G. and Qian, K. (2001) Kendrick mass defect spectrum: a compact visual analysis for ultrahigh-resolution broadband mass spectra. Analytical Chemistry 73, 4676-4681, https://doi.org/10.1021/ac010560w.

Hur, M., Yeo, I., Park, E., Kim, Y.H., Yoo, J., Kim, E., No, M.-h., Koh, J. and Kim, S. (2010) Combination of Statistical Methods and Fourier Transform Ion Cyclotron Resonance Mass Spectrometry for More Comprehensive, Molecular-Level Interpretations of Petroleum Samples. Analytical Chemistry 82, 211-218, https://doi.org/10.1021/ac901748c.

Ikeya, K., Sleighter, R.L., Hatcher, P.G. and Watanabe, A. (2015) Characterization of the chemical composition of soil humic acids using Fourier transform ion cyclotron resonance mass spectrometry. Geochimica et Cosmochimica Acta 153, 169-182, https://doi.org/10.1016/j.gca.2015.01.002.

Ikeya, K., Sleighter, R.L., Hatcher, P.G. and Watanabe, A. (2020) Chemical compositional analysis of soil fulvic acids using Fourier transform ion cyclotron resonance mass spectrometry. Rapid Communications in Mass Spectrometry 34, 1-11, https://doi.org/10.1002/rcm.8801.

Jackson, D.A. (1993) Stopping Rules in Principal Components Analysis: A Comparison of Heuristical and Statistical Approaches. Ecology 74, 2204-2214, https://doi.org/10.2307/1939574.

Johnson, R.L. and Schmidt-Rohr, K. (2014) Quantitative solid-state ^{13}C NMR with signal enhancement by multiple cross polarization. Journal of Magnetic Resonance 239, 44-49, https://doi.org/10.1016/j.jmr.2013.11.009.

Jolliffe, I. and Cadima, J. (2016) Principal component analysis: A review and recent developments. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 374, 1-16, https://doi.org/10.1098/rsta.2015.0202.

Kalbitz, K., Geyer, W. and Geyer, S. (1999) Spectroscopic properties of dissolved humic substances — a reflection of land use history in a fen area. Biogeochemistry 47, 219-238, https://doi.org/10.1007/BF00994924.

Kalbitz, K., Geyer, S. and Geyer, W. (2000) A comparative characterization of dissolved organic matter by means of original aqueous samples and isolated humic substances. Chemosphere 40, 1305-1312, https://doi.org/10.1016/S0045-6535(99)00238-6.

Kellerman, A.M., Dittmar, T., Kothawala, D.N. and Tranvik, L.J. (2014) Chemodiversity of dissolved organic matter in lakes driven by climate and hydrology. Nature Communications 5, 1-8, https://doi.org/10.1038/ncomms4804.

Kellerman, A.M., Kothawala, D.N., Dittmar, T. and Tranvik, L.J. (2015) Persistence of dissolved organic matter in lakes related to its molecular characteristics. Nature Geoscience 8, 454-457, https://doi.org/10.1038/ngeo2440.

Kendrick, E. (1963) A mass scale based on CH_2 = 14.0000 for high resolution mass spectrometry of organic compounds. Analytical Chemistry 35, 2146-2154, https://doi.org/10.1021/ac60206a048.

Khatami, S., Deng, Y., Tien, M. and Hatcher, P.G. (2019) Formation of water-soluble organic matter through fungal degradation of lignin. Organic Geochemistry 135, 64-70, https://doi.org/10.1016/j.orggeochem.2019.06.004.

Kim, S., Kramer, R.W. and Hatcher, P.G. (2003a) Graphical method for analysis of ultrahigh-resolution broadband mass spectra of natural organic matter, the van Krevelen diagram. Analytical Chemistry 75, 5336-5344, https://doi.org/10.1021/ac034415p.

Kim, S., Simpson, A.J., Kujawinski, E.B., Freitas, M.A. and Hatcher, P.G. (2003b) High resolution electrospray ionization mass spectrometry and 2D solution NMR for the analysis of DOM extracted by C_{18} solid phase disk. Organic Geochemistry 34, 1325-1335, https://doi.org/10.1016/s0146-6380(03)00101-3.

Kim, S., Kaplan, L.A., Benner, R. and Hatcher, P.G. (2004) Hydrogen-deficient molecules in natural riverine water samples—evidence for the existence of black carbon in DOM. Marine Chemistry 92, 225-234, https://doi.org/10.1016/j.marchem.2004.06.042.

Koch, B.P., Witt, M., Engbrodt, R., Dittmar, T. and Kattner, G. (2005) Molecular formulae of marine and terrigenous dissolved organic matter detected by electrospray ionization Fourier transform ion cyclotron resonance mass spectrometry. Geochimica et Cosmochimica Acta 69, 3299-3308, https://doi.org/10.1016/j.gca.2005.02.027.

Koch, B.P. and Dittmar, T. (2006) From mass to structure: An aromaticity index for high-resolution mass data of natural organic matter. Rapid Communications in Mass Spectrometry 20, 926-932, https://doi.org/10.1002/rcm.2386.

Koch, B.P., Dittmar, T., Witt, M. and Kattner, G. (2007) Fundamentals of molecular formula assignment to ultrahigh resolution mass data of natural organic matter. Analytical Chemistry 79, 1758-1763, https://doi.org/10.1021/ac061949s.

Koch, B.P. and Dittmar, T. (2016) From mass to structure: An aromaticity index for high-resolution mass data of natural organic matter (Erratum). Rapid Communications in Mass Spectrometry 30, 1, https://doi.org/10.1002/rcm.7433.

Konermann, L., Ahadi, E., Rodriguez, A.D. and Vahidi, S. (2013) Unraveling the mechanism of electrospray ionization. Analytical Chemistry 85, 2-9, https://doi.org/10.1021/ac302789c.

Korak, J.A., Dotson, A.D., Summers, R.S. and Rosario-Ortiz, F.L. (2014) Critical analysis of commonly used fluorescence metrics to characterize dissolved organic matter. Water Research 49, 327-338, https://doi.org/10.1016/j.watres.2013.11.025.

Kothawala, D.N., Murphy, K.R., Stedmon, C.A., Weyhenmeyer, G.A. and Tranvik, L.J. (2013) Inner filter correction of dissolved organic matter fluorescence. Limnology and Oceanography: Methods 11, 616-630, https://doi.org/10.4319/lom.2013.11.616.

Kroll, J.H., Donahue, N.M., Jimenez, J.L., Kessler, S.H., Canagaratna, M.R., Wilson, K.R., Altieri, K.E., Mazzoleni, L.R., Wozniak, A.S., Bluhm, H., Mysak, E.R., Smith, J.D., Kolb, C.E. and Worsnop, D.R. (2011) Carbon oxidation state as a metric for describing the chemistry of atmospheric organic aerosol. Nature Chem. 3, 133-139, https://doi.org/10.1038/nchem.948.

Kujawinski, E.B. and Behn, M.D. (2006) Automated analysis of electrospray ionization Fourier transform ion cyclotron resonance mass spectra of natural organic matter. Analytical Chemistry 78, 4363-4373, https://doi.org/10.1021/ac0600306.

Kujawinski, E.B., Longnecker, K., Blough, N.V., Del Vecchio, R., Finlay, L., Kitner, J.B. and Giovannoni, S.J. (2009) Identification of possible source markers in marine dissolved organic matter using ultrahigh resolution mass spectrometry. Geochimica et Cosmochimica Acta 73, 4384-4399, https://doi.org/10.1016/j.gca.2009.04.033.

Kurek, M.R., Harir, M., Shukle, J.T., Schroth, A.W., Schmitt-Kopplin, P. and Druschel, G.K. (2021) Seasonal transformations of dissolved organic matter and organic phosphorus in a polymictic basin: Implications for redox-driven eutrophication. Chemical Geology 573, 1-12, https://doi.org/10.1016/j.chemgeo.2021.120212.

LaRowe, D.E. and Van Cappellen, P. (2011) Degradation of natural organic matter: A thermodynamic analysis. Geochimica et Cosmochimica Acta 75, 2030-2042, https://doi.org/10.1016/j.gca.2011.01.020.

Lawaetz, A. and Stedmon, C. (2009) Fluorescence Intensity Calibration Using the Raman Scatter Peak of Water. Applied Spectroscopy 63, 936-940, https://doi.org/10.1366/000370209788964548.

Ledford, E.B., Rempel, D.L. and Gross, M.L. (1984) Space charge effects in Fourier transform mass spectrometry. II. Mass calibration. Analytical Chemistry 56, 2744-2748, https://doi.org/10.1021/ac00278a027.

Liu, Z., Sleighter, R.L., Zhong, J. and Hatcher, P.G. (2011) The chemical changes of DOM from black waters to coastal marine waters by HPLC combined with ultrahigh resolution mass spectrometry. Estuarine, Coastal and Shelf Science 92, 205-216, https://doi.org/10.1016/j.ecss.2010.12.030.

Louchouarn, P., Opsahl, S. and Benner, R. (2000) Isolation and Quantification of Dissolved Lignin from Natural Waters Using Solid-Phase Extraction and GC/MS. Analytical Chemistry 72, 2780-2787, https://doi.org/10.1021/ac9912552.

Lu, J.H. and Wu, L. (2001) Spectrophotometric Determination of Polyacrylamide in Waters Containing Dissolved Organic Matter. Journal of Agricultural and Food Chemistry 49, 4177-4182, https://doi.org/10.1021/jf010430o.

Lu, K., Li, X., Chen, H. and Liu, Z. (2021) Constraints on isomers of dissolved organic matter in aquatic environments: Insights from ion mobility mass spectrometry. Geochimica et Cosmochimica Acta, https://doi.org/10.1016/j.gca.2021.05.007.

Ludwig, C. and Günther, U.L. (2011) MetaboLab - advanced NMR data processing and analysis for metabolomics. BMC Bioinformatics 12, 1-6, https://doi.org/10.1186/1471-2105-12-366.

Maizel, A.C. and Remucal, C.K. (2017) The effect of advanced secondary municipal wastewater treatment on the molecular composition of dissolved organic matter. Water Research 122, 42-52, https://doi.org/10.1016/j.watres.2017.05.055.

Mantini, D., Petrucci, F., Pieragostino, D., Del Boccio, P., Di Nicola, M., Di Ilio, C., Federici, G., Sacchetta, P., Comani, S. and Urbani, A. (2007) LIMPIC: a computational method for the separation of protein MALDI-TOF-MS signals from noise. BMC Bioinformatics 8, 1-17, https://doi.org/10.1186/1471-2105-8-101.

Marshall, A.G., Hendrickson, C.L. and Jackson, G.S. (1998) Fourier transform ion cyclotron resonance mass spectrometry: A primer. Mass Spectrom Rev 17, 1-35, https://doi.org/10.1002/(SICI)1098-2787(1998)17:1%3C1::AID-MAS1%3E3.0.CO;2-K.

McKnight, D.M., Boyer, E.W., Westerhoff, P.K., Doran, P.T., Kulbe, T. and Andersen, D.T. (2001) Spectrofluorometric characterization of dissolved organic matter for indication of precursor organic material and aromaticity. Limnology and Oceanography 46, 38-48, https://doi.org/10.4319/lo.2001.46.1.0038.

McLafferty, F.W. and Turecek, F. (1993) Interpretation of mass spectra, 4th Edition ed. University Science Books.

Merder, J., Freund, J.A., Feudel, U., Hansen, C.T., Hawkes, J.A., Jacob, B., Klaproth, K., Niggemann, J., Noriega-Ortega, B.E., Osterholz, H., Rossel, P.E., Seidel, M., Singer, G., Stubbins, A., Waska, H. and Dittmar, T. (2020) ICBM-OCEAN: Processing ultrahigh-resolution mass spectrometry data of complex molecular mixtures. Analytical Chemistry 92, 6832-6838, https://doi.org/10.1021/acs.analchem.9b05659.

Mitchell, D.W. and Smith, R.D. (1995) Cyclotron motion of two Coulombically interacting ion clouds with implications to Fourier-transform ion cyclotron resonance mass spectrometry. Phys. Rev. E 52, 4366-4386, https://doi.org/10.1103/physreve.52.4366.

Mopper, K., Stubbins, A., Ritchie, J.D., Bialk, H.M. and Hatcher, P.G. (2007) Advanced instrumental approaches for characterization of marine dissolved organic matter: extraction techniques, mass spectrometry, and nuclear magnetic resonance spectroscopy. Chemical Reviews 107, 419-442, https://doi.org/10.1021/cr050359b.

Murphy, K.R., Butler, K.D., Spencer, R.G.M., Stedmon, C.A., Boehme, J.R. and Aiken, G.R. (2010) Measurement of dissolved organic matter fluorescence in aquatic environments: An interlaboratory comparison. Environmental Science & Technology 44, 9405-9412, https://doi.org/10.1021/es102362t.

Murphy, K.R. (2011) A note on determining the extent of the water Raman peak in fluorescence spectroscopy. Applied Spectroscopy 65, 233-236, https://doi.org/10.1366/10-06136.

Murphy, K.R., Stedmon, C.A., Graeber, D. and Bro, R. (2013) Fluorescence spectroscopy and multi-way techniques. PARAFAC. Analytical Methods 5, 6541-6882, https://doi.org/10.1039/c3ay41160e.

Murphy, K.R., Stedmon, C.A., Wenig, P. and Bro, R. (2014) OpenFluor– an online spectral library of auto-fluorescence by organic compounds in the environment. Analytical Methods 6, 658-661, https://doi.org/10.1039/C3AY41935E.

Murtagh, F. and Contreras, P. (2012) Algorithms for hierarchical clustering: an overview. WIREs Data Mining and Knowledge Discovery 2, 86-97, https://doi.org/10.1002/widm.53.

Murtagh, F. and Contreras, P. (2017) Algorithms for hierarchical clustering: an overview, II. WIREs Data Min. Knowl. 7, e1219, https://doi.org/10.1002/widm.1219.

Obeid, W.A. (2015) Investigation of the ootential for algaenan to produce hydrocarbon based fuels from algae by hydrous pyrolysis, Department of Chemistry and Biochemistry. Old Dominion University, pp. 1-223.

Ohno, T. (2002) Fluorescence Inner-Filtering Correction for Determining the Humification Index of Dissolved Organic Matter. Environmental Science & Technology 36, 742-746, https://doi.org/10.1021/es0155276.

Ohno, T., He, Z., Sleighter, R.L., Honeycutt, C.W. and Hatcher, P.G. (2010) Ultrahigh resolution mass spectrometry and indicator species analysis to identify marker components of soil- and plant biomass-derived organic matter fractions. Environmental Science & Technology 44, 8594-8600, https://doi.org/10.1021/es101089t.

Ohno, T. and Ohno, P.E. (2013) Influence of heteroatom pre-selection on the molecular formula assignment of soil organic matter components determined by ultrahigh resolution mass spectrometry. Analytical and Bioanalytical Chemistry 405, 3299-3306, https://doi.org/10.1007/s00216-013-6734-3.

Ohno, T., Parr, T.B., Gruselle, M.C.I., Fernandez, I.J., Sleighter, R.L. and Hatcher, P.G. (2014) Molecular Composition and Biodegradability of Soil Organic Matter: A Case Study Comparing Two New England Forest Types. Environmental Science & Technology 48, 7229-7236, https://doi.org/10.1021/es405570c.

Ohno, T., Sleighter, R.L. and Hatcher, P.G. (2016) Comparative study of organic matter chemical characterization using negative and positive mode electrospray ionization ultrahigh-resolution mass spectrometry. Analytical and Bioanalytical Chemistry 408, 2497-2504, https://doi.org/10.1007/s00216-016-9346-x.

Ohno, T., Sleighter, R.L. and Hatcher, P.G. (2018) Adsorptive fractionation of corn, wheat, and soybean crop residue derived water-extractable organic matter on iron (oxy)hydroxide. Geoderma 326, 156-163, https://doi.org/10.1016/j.geoderma.2018.04.006.

Osterholz, H., Kirchman, D.L., Niggemann, J. and Dittmar, T. (2016) Environmental drivers of dissolved organic matter molecular composition in the Delaware estuary. Frontiers in Earth Science 4, 1-14, https://doi.org/10.3389/feart.2016.00095.

Paliy, O. and Shankar, V. (2016) Application of multivariate statistical techniques in microbial ecology. Molecular Ecology 25, 1032-1057, https://doi.org/10.1111/mec.13536.

Parlanti, E., Wörz, K., Geoffroy, L. and Lamotte, M. (2000) Dissolved organic matter fluorescence spectroscopy as a tool to estimate biological activity in a coastal zone submitted to anthropogenic inputs. Organic Geochemistry 31, 1765-1781, https://doi.org/10.1016/S0146-6380(00)00124-8.

Patel, K.F., Tejnecký, V., Ohno, T., Bailey, V.L., Sleighter, R.L. and Hatcher, P.G. (2021) Reactive oxygen species alter chemical composition and adsorptive fractionation of soil-derived organic matter. Geoderma 384, 1-8, https://doi.org/10.1016/j.geoderma.2020.114805.

Patriarca, C., Balderrama, A., Može, M., Sjöberg, P.J.R., Bergquist, J., Tranvik, L.J. and Hawkes, J.A. (2020) Investigating the ionization of dissolved organic matter by electrospray ionization. Analytical Chemistry 92, 14210-14218, https://doi.org/10.1021/acs.analchem.0c03438.

Patriarca, C. and Hawkes, J.A. (2020) High Molecular Weight Spectral Interferences in Mass Spectra of Dissolved Organic Matter. Journal of the American Society for Mass Spectrometry 32, 394-397, https://doi.org/10.1021/jasms.0c00353.

Peacock, M., Evans, C.D., Fenner, N., Freeman, C., Gough, R., Jones, T.G. and Lebron, I. (2014) UV-visible absorbance spectroscopy as a proxy for peatland dissolved organic carbon (DOC) quantity and quality: considerations on wavelength and absorbance degradation. Environmental Science: Processes & Impacts 16, 1445-1461, https://doi.org/10.1039/C4EM00108G.

Perrette, Y., Delannoy, J.-J., Desmet, M., Lignier, V. and Destombes, J.-L. (2005) Speleothem organic matter content imaging. The use of a Fluorescence Index to characterise the maximum emission wavelength. Chemical Geology 214, 193-208, https://doi.org/10.1016/j.chemgeo.2004.09.002.

Peuravuori, J. and Pihlaja, K. (1997) Molecular size distribution and spectroscopic properties of aquatic humic substances. Analytica Chimica Acta 337, 133-149, https://doi.org/10.1016/S0003-2670(96)00412-6.

Powers, L.C., Hertkorn, N., McDonald, N., Schmitt-Kopplin, P., Del Vecchio, R., Blough, N.V. and Gonsior, M. (2019) *Sargassum sp.* act as a large regional surce of marine dissolved organic carbon and polyphenols. Global Biogeochemical Cycles 33, 1423-1439, https://doi.org/10.1029/2019GB006225.

Preacher, K.J. and MacCallum, R.C. (2003) Repairing Tom Swift's Electric Factor Analysis Machine. Understanding Statistics 2, 13-43, https://doi.org/10.1207/S15328031US0201_02.

Proctor, C.J., Baker, A., Barnes, W.L. and Gilmour, M.A. (2000) A thousand year speleothem proxy record of North Atlantic climate from Scotland. Climate Dynamics 16, 815-820, https://doi.org/10.1007/s003820000077.

Qing-Long, F., Manabu, F. and Thomas, R. (2020) Development and comparison of formula assignment algorithms for ultrahigh-resolution mass spectra of natural organic matter. Analytica Chimica Acta 1125, 247-257, https://doi.org/10.1016/j.aca.2020.05.048.

Reemtsma, T. (2009) Determination of molecular formulas of natural organic matter molecules by (ultra-) high-resolution mass spectrometry: Status and needs. Journal of Chromatography A 1216, 3687-3701, https://doi.org/10.1016/j.chroma.2009.02.033.

Riedel, T., Biester, H. and Dittmar, T. (2012) Molecular fractionation of dissolved organic matter with metal salts. Environmental Science & Technology 46, 4419-4426, https://doi.org/10.1021/es203901u.

Santl-Temkiv, T., Finster, K., Dittmar, T., Hansen, B.M., Thyrhaug, R., Nielsen, N.W. and Karlson, U.G. (2013) Hailstones: a window into the microbial and chemical inventory of a storm cloud. PLoS ONE 8, e53550, https://doi.org/10.1371/journal.pone.0053550.

Schmidt, F., Elvert, M., Koch, B.P., Witt, M. and Hinrichs, K.-U. (2009) Molecular characterization of dissolved organic matter in pore water of continental shelf sediments. Geochimica et Cosmochimica Acta 73, 3337-3358, https://doi.org/10.1016/j.gca.2009.03.008.

Schwämmle, V., Harrow, J. and Ienasescu, H. (2021) Proteomics Software in bio.tools: Coverage and Annotations. Journal of Proteome Research 20, 1821-1825, https://doi.org/10.1021/acs.jproteome.0c00978.

Senesi, N., Miano, T.M., Provenzano, M.R. and Brunetti, G. (1989) Spectroscopic and compositional comparative characterization of I.H.S.S. reference and standard fulvic and humic acids of various origin. Science of The Total Environment 81-82, 143-156, https://doi.org/10.1016/0048-9697(89)90120-4.

Severson, K.A., Molaro, M.C. and Braatz, R.D. (2017) Principal Component Analysis of Process Datasets with Missing Values. Processes 5, 1-18, https://doi.org/10.3390/pr5030038.

Singh, A. (2021) Proteomic analysis with MaxDIA. Nat. Methods 18, 988-988, https://doi.org/10.1038/s41592-021-01267-4.

Sleighter, R.L. and Hatcher, P.G. (2007) The application of electrospray ionization coupled to ultrahigh resolution mass spectrometry for the molecular characterization of natural organic matter. Journal of Mass Spectrometry 42, 559-574, https://doi.org/10.1002/jms.1221.

Sleighter, R.L. and Hatcher, P.G. (2008) Molecular characterization of dissolved organic matter (DOM) along a river to ocean transect of the lower Chesapeake Bay by ultrahigh resolution electrospray ionization Fourier transform ion cyclotron resonance mass spectrometry. Marine Chemistry 110, 140-152, https://doi.org/10.1016/j.marchem.2008.04.008.

Sleighter, R.L., McKee, G.A., Liu, Z. and Hatcher, P.G. (2008) Naturally present fatty acids as internal calibrants for Fourier transform mass spectra of dissolved organic matter. Limnology and Oceanography: Methods 6, 246-253, https://doi.org/10.4319/lom.2008.6.246.

Sleighter, R.L., McKee, G.A. and Hatcher, P.G. (2009) Direct Fourier transform mass spectral analysis of natural waters with low dissolved organic matter. Organic Geochemistry 40, 119-125, https://doi.org/10.1016/j.orggeochem.2008.09.012.

Sleighter, R.L., Liu, Z., Xue, J. and Hatcher, P.G. (2010) Multivariate statistical approaches for the characterization of dissolved organic matter analyzed by ultrahigh resolution mass spectrometry. Environmental Science & Technology 44, 7576-7582, https://doi.org/10.1021/es1002204.

Sleighter, R.L. and Hatcher, P.G. (2011) Fourier transform mass spectrometry for the molecular level characterization of natural organic matter: Instrument capabilities, applications, and limitations in: Nikolic, G. (Ed.), Fourier Transforms - Approach to Scientific Principles. InTech, pp. 295-320.

Sleighter, R.L., Chen, H., Wozniak, A.S., Willoughby, A.S., Caricasole, P. and Hatcher, P.G. (2012) Establishing a measure of reproducibility of ultrahigh-resolution mass spectra for complex mixtures of natural organic matter. Analytical Chemistry 84, 9184-9191, https://doi.org/10.1021/ac3018026.

Sleighter, R.L., Chin, Y.-P., Arnold, W.A., Hatcher, P.G., McCabe, A.J., McAdams, B.C. and Wallace, G.C. (2014a) Evidence of Incorporation of Abiotic S and N into Prairie Wetland Dissolved Organic Matter. Environmental Science & Technology Letters 1, 345-350, https://doi.org/10.1021/ez500229b.

Sleighter, R.L., Cory, R.M., Kaplan, L.A., Abdulla, H.A.N. and Hatcher, P.G. (2014b) A coupled geochemical and biogeochemical approach to characterize the bioreactivity of dissolved organic matter from a headwater stream. Journal of Geophysical Research: Biogeosciences 119, 1520-1537, https://doi.org/10.1002/2013JG002600.

Sleighter, R.L., Caricasole, P., Richards, K.M., Hanson, T. and Hatcher, P.G. (2015) Characterization of terrestrial dissolved organic matter fractionated by pH and polarity and their biological effects on plant growth. Chemical and Biological Technologies in Agriculture 2, 1-19, https://doi.org/10.1186/s40538-015-0036-2.

Smith, C.R., Hatcher, P.G., Kumar, S. and Lee, J.W. (2016) Investigation into the sources of biochar water-soluble organic compounds and their potential toxicity on aquatic microorganisms. ACS Sustainable Chemistry & Engineering 4, 2550-2558, https://doi.org/10.1021/acssuschemeng.5b01687.

Stenson, A.C., Landing, W.M., Marshall, A.G. and Cooper, W.T. (2002) Ionization and fragmentation of humic substances in electrospray ionization Fourier transform-ion cyclotron resonance mass spectrometry. Analytical Chemistry 74, 4397-4409, https://doi.org/10.1021/ac020019f.

Stenson, A.C., Marshall, A.G. and Cooper, W.T. (2003) Exact masses and chemical formulas of individual Suwannee River fulvic acids from ultrahigh resolution electrospray ionization Fourier transform ion cyclotron resonance mass spectra. Analytical Chemistry 75, 1275-1284, https://doi.org/10.1021/ac026106p.

Stubbins, A., Spencer, R.G.M., Chen, H., Hatcher, P.G., Mopper, K., Hernes, P.J., Mwamba, V.L., Mangangu, A.M., Wabakanghanzi, J.N. and Six, J. (2010) Illuminated darkness: Molecular signatures of Congo River dissolved organic matter and its photochemical alteration as revealed by ultrahigh precision mass spectrometry. Limnology and Oceanography 55, 1467-1477, https://doi.org/10.4319/lo.2010.55.4.1467.

Stubbins, A., Niggemann, J. and Dittmar, T. (2012) Photo-lability of deep ocean dissolved black carbon. Biogeosciences 9, 1661-1670, https://doi.org/10.5194/bg-9-1661-2012.

Stubbins, A., Lapierre, J.F., Berggren, M., Prairie, Y.T., Dittmar, T. and del Giorgio, P.A. (2014) What's in an EEM? Molecular signatures associated with dissolved organic fluorescence in boreal Canada. Environmental Science & Technology 48, 10598-10606, https://doi.org/10.1021/es502086e.

Summers, R.S., Cornel, P.K. and Roberts, P.V. (1987) Molecular size distribution and spectroscopic characterization of humic substances. Science of The Total Environment 62, 27-37, https://doi.org/10.1016/0048-9697(87)90478-5.

Tadini, A.M., Martin-Neto, L., Goranov, A.I., Milori, D.M.B.P., Bernardi, A.C.C., Oliveira, P.P.A., Pezzopane, J.R.M., Colnago, L.A. and Hatcher, P.G. (2021) Chemical characteristics of soil organic matter from integrated agricultural systems in southeastern Brazil. European Journal of Soil Science 73, 1-18, https://doi.org/10.1111/ejss.13136.

Thompson, A.M., Stratton, K.G., Bramer, L.M., Zavoshy, N.S. and McCue, L.A. (2021) Fourier transform ion cyclotron resonance mass spectrometry (FT-ICR-MS) peak intensity normalization for complex mixture analyses. Rapid Communications in Mass Spectrometry 35, e9068, https://doi.org/10.1002/rcm.9068.

Tolić, N., Liu, Y., Liyu, A., Shen, Y., Tfaily, M.M., Kujawinski, E.B., Longnecker, K., Kuo, L.-J., Robinson, E.W., Paša-Tolić, L. and Hess, N.J. (2017) Formularity: Software for Automated Formula Assignment of Natural and Other Organic Matter from Ultrahigh-Resolution Mass Spectra. Analytical Chemistry 89, 12659-12665, https://doi.org/10.1021/acs.analchem.7b03318.

Välikangas, T., Suomi, T. and Elo, L.L. (2017) A comprehensive evaluation of popular proteomics software workflows for label-free proteome quantification and imputation. Briefings in Bioinformatics 19, 1344-1355, https://doi.org/10.1093/bib/bbx054.

Valle, J., Harir, M., Gonsior, M., Enrich-Prast, A., Schmitt-Kopplin, P., Bastviken, D. and Hertkorn, N. (2020) Molecular differences between water column and sediment pore water SPE-DOM in ten Swedish boreal lakes. Water Research 170, 1-11, https://doi.org/10.1016/j.watres.2019.115320.

Van Krevelen, D.W. (1950) Graphical-statistical method for the study of structure and reaction processes of coal. Fuel Processing Technology 29, 269-228.

Vijaya , A.S. and Bateja, R. (2017) A Review on Hierarchical Clustering Algorithms. Journal of Engineering and Applied Sciences 12, 7501-7507, https://doi.org/10.36478/jeasci.2017.7501.7507.

Waggoner, D.C., Chen, H., Willoughby, A.S. and Hatcher, P.G. (2015) Formation of black carbon-like and alicyclic aliphatic compounds by hydroxyl radical initiated degradation of lignin. Organic Geochemistry 82, 69-76, https://doi.org/10.1016/j.orggeochem.2015.02.007.

Waggoner, D.C. and Hatcher, P.G. (2017) Hydroxyl radical alteration of HPLC fractionated lignin: Formation of new compounds from terrestrial organic matter. Organic Geochemistry 113, 315-325, https://doi.org/10.1016/j.orggeochem.2017.07.011.

Waggoner, D.C., Wozniak, A.S., Cory, R.M. and Hatcher, P.G. (2017) The role of reactive oxygen species in the degradation of lignin derived dissolved organic matter. Geochimica et Cosmochimica Acta 208, 171-184, https://doi.org/10.1016/j.gca.2017.03.036.

Wagner, S., Dittmar, T. and Jaffé, R. (2015a) Molecular characterization of dissolved black nitrogen via electrospray ionization Fourier transform ion cyclotron resonance mass spectrometry. Organic Geochemistry 79, 21-30, https://doi.org/10.1016/j.orggeochem.2014.12.002.

Wagner, S., Riedel, T., Niggemann, J., Vahatalo, A.V., Dittmar, T. and Jaffé, R. (2015b) Linking the molecular signature of heteroatomic dissolved organic matter to watershed characteristics in world rivers. Environmental Science & Technology 49, 13798-13806, https://doi.org/10.1021/acs.est.5b00525.

Wagner, S., Ding, Y. and Jaffé, R. (2017) A new perspective on the apparent solubility of dissolved black carbon. Frontiers in Earth Science 5, 1-16, https://doi.org/10.3389/feart.2017.00075.

Wagner, S., Fair, J.H., Matt, S., Hosen, J.D., Raymond, P., Saiers, J., Shanley, J.B., Dittmar, T. and Stubbins, A. (2019) Molecular Hysteresis: Hydrologically Driven Changes in Riverine Dissolved Organic Matter Chemistry During a Storm Event. Journal of Geophysical Research: Biogeosciences 124, 759-774, https://doi.org/10.1029/2018jg004817.

Ward, C.P., Sleighter, R.L., Hatcher, P.G. and Cory, R.M. (2014) Insights into the complete and partial photooxidation of black carbon in surface waters. Environmental Science: Processes & Impacts 16, 721-731, https://doi.org/10.1039/C3EM00597F.

Ward, C.P. and Cory, R.M. (2016) Complete and Partial Photo-oxidation of Dissolved Organic Matter Draining Permafrost Soils. Environmental Science & Technology 50, 3545-3553, https://doi.org/10.1021/acs.est.5b05354.

Weishaar, J.L., Aiken, G.R., Bergamaschi, B.A., Fram, M.S., Fujii, R. and Mopper, K. (2003) Evaluation of specific ultraviolet absorbance as an indicator of the chemical composition and reactivity of dissolved organic carbon. Environmental Science & Technology 37, 4702-4708, https://doi.org/10.1021/es030360x.

Whitty, S.D., Waggoner, D.C., Cory, R.M., Kaplan, L.A. and Hatcher, P.G. (2019) Direct noninvasive ^{1}H NMR analysis of stream water DOM: Insights into the effects of lyophilization compared with whole water. Magnetic Resonance in Chemistry 59, 540-553, https://doi.org/10.1002/mrc.4935.

Wieland, T., Kerber, A. and Laue, R. (1996) Principles of the Generation of Constitutional and Configurational Isomers. Journal of Chemical Information and Computer Sciences 36, 413-419, https://doi.org/10.1021/ci9502663.

Willoughby, A., Wozniak, A. and Hatcher, P. (2016) Detailed Source-Specific Molecular Composition of Ambient Aerosol Organic Matter Using Ultrahigh Resolution Mass Spectrometry and 1H NMR. Atmosphere 7, 1-24, https://doi.org/10.3390/atmos7060079.

Wilson, H.F. and Xenopoulos, M.A. (2008) Effects of agricultural land use on the composition of fluvial dissolved organic matter. Nature Geoscience 2, 37-41, https://doi.org/10.1038/ngeo391.

Wilson, M.A., Gillam, A.H. and Collin, P.J. (1983) Analysis of the structure of dissolved marine humic substances and their phytoplanktonic precursors by 1H and 13C nuclear magnetic resonance. Chemical Geology 40, 187-201, https://doi.org/10.1016/0009-2541(83)90029-3.

Wozniak, A.S., Shelley, R.U., Sleighter, R.L., Abdulla, H.A.N., Morton, P.L., Landing, W.M. and Hatcher, P.G. (2013) Relationships among aerosol water soluble organic matter, iron and aluminum in European, North African, and Marine air masses from the 2010 US GEOTRACES cruise. Marine Chemistry 154, 24-33, https://doi.org/10.1016/j.marchem.2013.04.011.

Wozniak, A.S., Shelley, R.U., McElhenie, S.D., Landing, W.M. and Hatcher, P.G. (2015) Aerosol water soluble organic matter characteristics over the North Atlantic Ocean: Implications for iron-binding ligands and iron solubility. Marine Chemistry 173, 162-172, https://doi.org/10.1016/j.marchem.2014.11.002.

Wozniak, A.S., Goranov, A.I., Mitra, S., Bostick, K.W., Zimmerman, A.R., Schlesinger, D.R., Myneni, S. and Hatcher, P.G. (2020) Molecular heterogeneity in pyrogenic dissolved organic matter from a thermal series of oak and grass chars. Organic Geochemistry 148, 1-18, https://doi.org/10.1016/j.orggeochem.2020.104065.

Wu, Z., Rodgers, R.P. and Marshall, A.G. (2004) Two-and three-dimensional van Krevelen diagrams: A graphical analysis complementary to the Kendrick mass plot for sorting elemental compositions of complex organic mixtures based on ultrahigh-resolution broadband Fourier transform ion cyclotron resonance mass measurements. Analytical Chemistry 76, 2511-2516, https://doi.org/10.1021/ac0355449.

Wünsch, U.J., Bro, R., Stedmon, C.A., Wenig, P. and Murphy, K.R. (2019) Emerging patterns in the global distribution of dissolved organic matter fluorescence. Analytical Methods 11, 888-893, https://doi.org/10.1039/C8AY02422G.

Xiao, Y., Carena, L., Näsi, M.-T. and Vähätalo, A.V. (2020) Superoxide-driven autocatalytic dark production of hydroxyl radicals in the presence of complexes of natural dissolved organic matter and iron. Water Research 177, 1-8, https://doi.org/10.1016/j.watres.2020.115782.

Xue, J., Lee, C., Wakeham, S.G. and Armstrong, R.A. (2011) Using principal components analysis (PCA) with cluster analysis to study the organic geochemistry of sinking particles in the ocean. Organic Geochemistry 42, 356-367, https://doi.org/10.1016/j.orggeochem.2011.01.012.

Yamamoto, H., Fujimori, T., Sato, H., Ishikawa, G., Kami, K. and Ohashi, Y. (2014) Statistical hypothesis testing of factor loading in principal component analysis and its application to metabolite set enrichment analysis. BMC Bioinformatics 15, 1-9, https://doi.org/10.1186/1471-2105-15-51.

Yuan, C., Sleighter, R.L., Weavers, L.K., Hatcher, P.G. and Chin, Y.P. (2019) Fast photomineralization of dissolved organic matter in acid mine drainage impacted waters. Environmental Science & Technology 53, 6273-6281, https://doi.org/10.1021/acs.est.9b00202.

Zhang, X., Han, J., Zhang, X., Shen, J., Chen, Z., Chu, W., Kang, J., Zhao, S. and Zhou, Y. (2020) Application of Fourier transform ion cyclotron resonance mass spectrometry to characterize natural organic matter. Chemosphere 260, 1-10, https://doi.org/10.1016/j.chemosphere.2020.127458.

Zherebker, A., Shirshin, E., Rubekina, A., Kharybin, O., Kononikhin, A., Kulikova, N.A., Zaitsev, K.V., Roznyatovsky, V.A., Grishin, Y.K., Perminova, I.V. and Nikolaev, E.N. (2020a) Optical Properties of Soil Dissolved Organic Matter Are Related to Acidic Functions of Its Components as Revealed by Fractionation, Selective Deuteromethylation, and Ultrahigh Resolution Mass Spectrometry. Environmental Science & Technology 54, 2667-2677, https://doi.org/10.1021/acs.est.9b05298.

Zherebker, A., Yakimov, B., Rubekina, A., Kharybin, O., Fedoros, E.I., Perminova, I.V., Shirshin, E. and Nikolaev, E.N. (2020b) Photoreactivity of humic-like polyphenol material under irradiation with different wavelengths explored by FTICR MS and deuteromethylation. European Journal of Mass Spectrometry 26, 1-9, https://doi.org/10.1177/1469066720917067.

Zsolnay, A., Baigar, E., Jimenez, M., Steinweg, B. and Saccomandi, F. (1999) Differentiating with fluorescence spectroscopy the sources of dissolved organic matter in soils subjected to drying. Chemosphere 38, 45-50, https://doi.org/10.1016/S0045-6535(98)00166-0.