Processing data using LUMA: LCMS-Based Untargeted Metabolomics Assistant

Jonathan Mosley and Marina Evich

##### Version 1.1

#### I. Introduction

The LCMS-Based Untargeted Metabolomics Assistant (LUMA) package is an automated data processing tool, bridging the outputs of XCMS and CAMERA to a comprehensive data matrix in the R environment for discovery-based studies. The LUMA package performs feature-reduction by minimizing single features which do not pass quality control which negatively impact downstream statistical analyses. LUMA contains functions that allow for rapid, automated workflows to perform these steps with minimal user input. Furthermore, to expedite manual data curation for potentially conflicting isotope and ion adduct annotations, data visualization is consolidated to a single graphic per metabolite group. This graphic contains all EIC plots and psSpectra from CAMERA and new correlation matrices and dendrograms for all features attributed to a single metabolite. Final processed metabolite data, containing normalized intensities and user-defined meta-data, can be exported to worksheets which are directly formatted to a number of analytical tools including MetaboAnalyst/Mummichog, while retaining traceability in the R environment.

The following tutorial is a guide to testing the basic functionality of LUMA using an LC-MS fish liver dataset using RStudio. Provided in this tutorial are 2 R scripts: install-LUMA-depend.R; test\_workflow.R and 1 binary package for lcmsfishdata\_0.1.0.zip, an accompanying R package containing LC-MS metabolomics data from fish livers (O:\Public\jmosl01).

#### II. Installation guide

1. Install the R package LUMA and dependencies by sourcing the provided script install-LUMA-depend.R in the R environment. Note, the package dbplyr must be updated to at least version 1.3.0. Also, there is a known bug when the package Formula is also loaded; unload and restart the R session if the Formula package is installed.
2. Install the R package lcmsfishdata from Install Packages, install from: Package Archive File (.zip) using the provided lcmsfishdata\_0.1.0.zip.

#### III. Directory organization

There will be three directories necessary for testing the LUMA R package: the working directory, the input file directory and the data directory.

1. Create the working directory on your local machine and paste the test\_workflow.R script into the working directory. Create an R project in this local directory by opening RStudio and create new project from your local working directory.
2. The location of the input directory will depend on your specific workstation and where lcmsfishdata was installed. This location can be printed to the R console by using the following code in R:

* input.dir <- system.file("extdata","Annotated\_library.csv", package = "lcmsfishdata", mustWork = T)  
  x <- unlist(gregexpr("/",input.dir))  
  keep <- substr(input.dir, 1, x[length(x)] - 1)  
  keep

1. The data directory containing the data files is located on the L drive. Do not delete or modify these files: O:\Public\jmosl01\LUMA Data

#### IV. Testing modules

LUMA is organized into modules, each of which has a specific function. To test each of these modules, open the provided R script file test\_workflow.R in the working directory and run one line of script at a time. After each module, record the output printed in the console, including any warnings or errors if they occur.

1. Remove all objects from the environment by running the first line of the script:

* rm(list = ls(all.names = TRUE))

1. Load the relevant libraries:

* library(LUMA)

1. Retrieve the location of the input file directory, described in III.B, for example, C:/Program Files/R/R-3.6.1/library/lcmsfishdata/extdata.
2. Initialize the workflow by running the first module. This module will open a series of dialog boxes for the user to select input files to process the blanks and sample data in both positive and negative ionization mode. Note, that the order of the dialog boxes may change.

* InitWorkflow()
  1. In the first dialog box, select the input directory, described in III.B.
  2. Select OK to use the LUMA recommended data directory.
  3. Select OK to process blanks first or cancel to process samples first.
  4. Enter 1 or Positive in the console to process positive ionization mode first. Alternatively, select 2 or Negative to process negative ionization mode.
  5. In the dialog box, select the data directory, described in III.C.
  6. In the dialog box, select the Sample Class file (Sample\_Class.txt). This file contains sample sex, class and n information as well as a flag for endogenous (control class).
  7. In the dialog box, select the Sample Data file (Sample\_Data.csv). This file contains metadata for each sample file to be used.
  8. In the dialog box, select the Annotated Library file (Annotated\_library.csv). This file contains annotation information such as name, formula, molecular weight and retention times for metabolites.
  9. In the dialog box, select the Search Parameters file (Search\_Parameters.txt). This file contains additional processing information including the void retention time, % CV cutoffs and additional flags, such as the option to keep or remove singletons, i.e., metabolites with only a single feature.
  10. Select OK to use an existing XCMS objects file.
  11. Select OK to use an existing CAMERA objects file.
  12. In the dialog box, select the adducts file (primary\_adducts\_pos.csv if selected positive in step IV.D.4. or primary\_adducts\_neg.csv if selected negative in IV.D.4.)
  13. In the dialog box, select the XCMS objects file (for example, XCMS\_objects\_Blanks\_Pos if selected processing blanks in positive mode first).
  14. Enter 2 in the console to select the 2nd sample as the XCMS center parameter. Hit enter to confirm all previous selections.
  15. In the dialog box, select the CAMERA objects file (for example, CAMERA\_objects\_Blanks\_Pos if selected processing blanks in positive mode first).

1. The next module will remove peaks that elute prior to the void volume for the LC system, set by the input file selected in IV.D.9.

* CullVoidVolume(from.table = "From CAMERA\_with Minfrac", to.table = "Trimmed by RT", method = "mz")

1. The following module will annotate the culled peak list from annotations in the Annotated Library file selected in IV.D.8.

* AnnotatePeaklist(from.table = "Trimmed by RT", to.table = "Annotated")

1. The following module, ParseCAMERA, will parse the annotations generated by CAMERA and remove redundant ion adduct annotations using predefined rules and will result in generating the first .csv output file used for further evaluation to eliminate conflicting annotations.
   * if running positive ionization mode data, use the ParseCAMERA module with the following arguments:
   * ParseCAMERA(from.table = "Annotated", to.table = "output\_parsed", CAMERA.obj = "anposGa")
   * if running negative ionization mode data, use the following:
   * ParseCAMERA(from.table = "Annotated", to.table = "output\_parsed", CAMERA.obj = "annegGa")
2. The following module, CombineFeatures, will sum the isotopic and adduct peaks and then combines all features into a single metabolite group.

* CombineFeatures(from.table = "output\_parsed", to.table = "Combined Isotopes and Adducts")

1. The next module, CullCV, will remove components with a coefficient of variation (CV) greater than the user-specified cutoff across the Pooled QC samples. Removing these variable components reduces overall instrument specific variation in the data. In this case, a CV cutoff of 30% is used. Run this module only when processing data samples (if selected “Cancel” in step IV.D.3.).

* CullCV(from.table = "Combined Isotopes and Adducts", to.table = "Trimmed by CV")

1. The CullMF module then removes components with a minimum fraction of less than 80% within a specific class. Removing these components reduces overall biological variation per class in the data, as these are not considered reliable biomarkers for a given class. Run this module only when processing data samples (if selected Cancel in step IV.D.3.).

* CullMF(from.table = "Trimmed by CV", to.table = "Trimmed by MinFrac")

1. Repeat steps D-J for the other samples until positive blanks, negative blanks, positive sample data and negative sample data have all been processed.
2. The CullBackground module then removes background components which have been identified form the positive and negative blank samples.

* CullBackground(from.tables = c("Trimmed by MinFrac","Combined Isotopes and Adducts"), to.tables c("Peaklist\_Pos\_Solvent Peaks Removed", "Peaklist\_Neg\_Solvent Peaks Removed", "Peaklist\_Pos\_Solvent Peaks Only", "Peaklist\_Neg\_Solvent Peaks Only"), method = "monoMass")

1. The next module, SimplyPeaklists, will combine the two peak lists from the positive sample data and negative sample data into a single list. This module will generate a Peaklist\_combined.csv file which contains samples and intensities for each metabolite.

* file.copy(from = paste0(keep,"/EIC\_index\_pos.txt"), to = "./EIC\_index\_pos.txt", overwrite = TRUE)  
  file.copy(from = paste0(keep,"/EIC\_index\_neg.txt"), to = "./EIC\_index\_neg.txt", overwrite = TRUE)  
  SimplyPeaklists(from.tables = c("Peaklist\_Pos\_Solvent Peaks Removed", "Peaklist\_Neg\_Solvent Peaks Removed"), to.table = "Peaklist\_Combined", peak.db = "Peaklist\_db")

1. The NormalizePeaklists module will next normalize the peaklist using the default normalization, set to unit normalize.

* NormalizePeaklists(from.table = "Peaklist\_Combined", to.table = "Peaklist\_Normalized")

1. The FormatForMetaboAnalystR module will format the peaklist and generate files for an acceptable format to be used with MetaboAnalystR for further analysis.

* FormatForMetaboAnalystR(from.table = "Peaklist\_Normalized", to.csv = "Peaklist\_for\_MetaboAnalyst", data.type = "pktable", anal.type = "stat")

1. The final module, FinalWorkflow, will finalize the LUMA workflow, closing the databases and generating a LUMA log for traceability.

* FinalWorkflow(peak\_db = peak\_db, lib\_db = lib\_db)

1. After running the final module, save the R Workspace image. Run sessionInfo() and copy and paste the output in the console to a Word document saved in the working directory as “session\_info.docx”.

* sessionInfo()

#### V. Traceability in LUMA

After running the FinalWorkflow module, the LUMA log will be printed to the console. This log contains the names of all tables that exist in the SQLite databases that were saved during the LUMA workflow that was ran in this SOP. LUMA contains database querying functions that can pull these tables into the R environment. An example of each function is shown below. Check the R documentation for more details.

#### VI. Submitting results to MetaboAnalyst

After completing the final module, you will have generated all of the necessary files to submit this dataset to MetaboAnalyst for further analysis. For the purposes of this SOP, we only ask that you verify the successful upload of the dataset into MetaboAnalyst (<https://www.metaboanalyst.ca/>).

1. In the “Upload Data” section, set the following parameters:
2. In the “Upload Data” section, set the following parameters:
3. In the “Upload Data” section, set the following parameters:
4. In the “Upload Data” section, set the following parameters:
   1. For Data Type, choose "Concentrations".
   2. For Format, choose "Samples in rows (unpaired)".
   3. Use the "Choose File" buttom to upload the 'Peaklist\_for\_MetaboAnalyst.csv' file.
   4. Click “Submit”.
5. The next window is the ‘Data Integrity’ check. After reading about the data processing information, click “Skip” (this data has already had missing value imputation performed).
6. On the ‘Data Filtering’ window, choose “None (less than 5000 features)” and click ‘Proceed’ (this data has already been filtered)
7. On the ‘Normalization Overview’ window, choose the following parameters:
   1. For Data Type, choose "Concentrations".
   2. For Format, choose "Samples in rows (unpaired)".
   3. Use the "Choose File" buttom to upload the 'Peaklist\_for\_MetaboAnalyst.csv' file.
   4. Click “Submit”.
   5. Click “Submit”.
8. At this point, the dataset has been successfully uploaded into MetaboAnalyst if you see the “Select an analysis path to explore” page. Feel free to explore the various analysis options the MetaboAnalyst provides; however, these results are beyond the scope of this SOP.

#### VII. Submitting beta-testing results to developers

After completing the final module, you will have generated several output files including Microsoft .csv and .xlsx files in your working directory. Compress the entire working directory into a zip file. Rename the .zip extension to .piz to prevent Outlook from blocking the file, and send it to [evich.marina@epa.gov](mailto:evich.marina@epa.gov).