**Spike List to mc0**

**This document adapted from:** L:\Lab\NHEERL\_MEA\Comprehensive AUC Analysis\Comprehensive\_Analysis.docx, Last rev. June 20, 2019 by Brittany Lynch

**Software info:**

RStudio version: 1.2.1335

R console version: 3.6.1

**Preliminary notes**

* Input data:
  + Spike List files for each DIV, from each plate and culture date
    - e.g., ON\_[date]\_MW[Plate#]\_[DIV]\_00(000)\_spike\_list.csv
  + Master Chemical List for each plate
    - e.g. [date]\_MW[Plate#]\_MaestroExperimentLog\_Ontogeny.csv
  + Cytotoxicity data
    - e.g. "ON\_[date]\_MW[Plate#]\_\_Summary.xlsx” for 1 plate data, or
    - “[date]\_ON G8\_2 Calculations.xlsx” for files with data for 3 plates
  + Sample ID file
    - This file should contain a list the relevant chemicals and the corresponding sample ID (The sample ID column may be labelled “EPA\_SAMPLE\_ID” or “SPID”)
* Target output:
  + A file that is ready to process in the ToxCast pipline using tcplWriteLvl0(data, type = “mc”)
  + Along the way, you will also get:
    - Excel files with all ontogeny values
    - Excel file with all AUC values

**Downloading and install required R packages**

1. Download “meadq”, “rdfh”, “bioinstaller”, and “sjemea” packages into R
   1. Install the “devtools” package, copy and paste the following command:

install.packages("devtools")

* 1. To install “rhdf5”, copy and paste the following commands:

if(!requireNamespace("BiocManager", quietly = TRUE))

+ install.packages("BiocManager")

BiocManager::install("rhdf5")

If it asks to Update all/some/none packages, select all.

Additional info: <https://bioconductor.org/install> and <https://stackoverflow.com/questions/15974643/how-to-deal-with-hdf5-files-in-r>

* 1. To install “bioInstaller”, copy and paste the following command:

install.packages(“BioInstaller”)

* 1. To install “sjmea”, copy and paste the following command:

devtools::install\_github("sje30/sjemea")

* 1. To install “meadq”, copy and paste the following command:

devtools::install\_github("dianaransomhall/meadq")

Additional info: <https://github.com/dianaransomhall/meadq>

<https://cran.r-project.org/web/packages/githubinstall/vignettes/githubinstall.html>

* 1. To install “h5”, copy and paste the following command:

install.packages(“h5”)

* 1. To install “gtools”, copy and paste the following command:

install.packages(“gtools”)

* 1. To install “xlsx”, copy and paste the following command:

install.packages(“xlsx”)

* 1. To install “reshape”, copy and paste the following command:

install.packages(“reshape”)

* 1. To install “data.table”, copy and paste the following command:

install.packages(“data.table”)

**Step 0: Preparing the spike list files**

Locate the spike list files for the data set. These are usually named something like “ON\_20151209\_MW1056-28\_02\_00(000)\_spike\_list.csv.” This should be a long file with 3 columns that record the time, electrode, and amplitude of each spike. You will use only the 4 spike list files for DIV 5, 7, 9, 12 for each plate. Do not include data for DIV 2, Bicuculline-treated wells (a second recording on DIV 12), or files that go past 900 seconds (if a files says “corrected for time”, use that one).

Locate the Master Chemical Lists for each plate. These are usually located in a file called “csv files”, and are usually named “[date]\_MW[Plate#]\_MaestroExperimentLog\_Ontogeny.csv”

Also locate the files containing the cytotoxicity data. These files are usually located in the culture date folder (e.g. "[date]\_ON [Group] Calculations.xlsx") or the plate folder (e.g. “ON\_[date]\_MW[Plate#]\_Summary.xlsx”). These files will contain multiple tabs, including 2 tabs for LDH and AB data. You can use either files that contain data for one plate or 3 plates. In step 5, the script cytotox\_prep05\_rawValues.R will extract the raw blank-corrected data for each plate.

Check which culture dates should be used. If a culture was repeated because of poor data quality, do not use that data. If you need to keep only some data from a culture date, you can do the analysis for all data from that culture date. In step 6, you will be able to selectively remove unwanted data from that culture date.

Processing Option 1 (preferred):

Put all of the spike list files that you want to analyze in one folder, e.g. “All Spike Lists”. Put all the Master Chemical Lists in another folder (anywhere), e.g. “All Master Chem List”. Lastly, put all of the cytotoxicity data containing files in another folder.

Processing Option 2 (if you don’t want to move all the spike list files):

Leave spike list files, master chemical lists, and cytotoxicity data files in their current folders. You will run h5\_conversion.R and every other script for each plate individually.

**Step 1: Create h5 Files from spike list files**

Open the following R scripts:

h5\_conversion.R, and spike\_list\_functions.R

Overview: This script will create one h5file from each spike list file. The output will go in a folder called “*h5Files*”

1. Source spike\_list\_functions.R. This script contains functions that h5\_conversion.R will use.
2. In h5\_conversion.R, in the USER INPUT section, set the “basepath” variable to where you want the h5Files folder to be created. Note: if the h5Files folder is created in the same location as an existing h5Files folder, the new h5Files folder will overwrite the existing h5Files folder.
3. Source h5\_conversion.R
   1. This script will open a dialog box that asks for all the spike list files (the dialog box might not pop up or create an icon on the taskbar. If so, try minimizing RStudio to see the dialog box)
   2. Select the spike list files that you want to use
      1. If following Option 1, select all of the spike list files
      2. If following Option 2, select the 4 spike list files for the first plate
   3. Another dialog box will open afterwards asking for the Master Chemical List
      1. If following Option 1, select all of the Master Chemical List files
      2. If following Option 2, select the Master Chemical List for the first plate
   4. If following Option 2, repeat step 4 for each plate.

**Step 2: Calculate 16 ontogeny values (all except MI) from the h5 files**

Open the following R scripts:

create\_ont\_csv.R, create\_burst\_ont \_Data.R, local.corr.all.ont.ae.filter.R, and comb\_summary.R

This script will calculate 16 parameter values from the h5files and each DIV. There will be one file for each plate. The output will be in a folder called “*prepared\_data*”

1. Source create\_burst\_ont \_Data.R and **local.corr.all.ont.ae.filter.R**. These files contains functions that create\_ont\_csv.R will use
2. In create\_ont\_csv.R, in the USER INPUT section, define the “basepath” variable for where you want the prepared data folder to be created
   1. Change all backward slashes to forward slashes
3. Source create\_ont\_csv.R
   1. This script will open a dialog box that asks for the h5files (the dialog box might not pop up or create an icon on the taskbar. If so, try minimizing RStudio to see the dialog box)
      1. If following Option 1, select all h5files for every plate
      2. If following Option 2, select the 4 h5 files for each DIV for the first plate
   2. The following message will appear multiple times:

\*\*\* No network spikes found

Warnings generated are normal

* 1. If following Option 2, repeat step 7a for all plates. If all the output files are not in the same folder, put them in one master folder before the next step.

1. In comb\_summary.R, in the USER INPUT section, set the desired name of the output csv file
   1. Change all backward slashes to forward slashes
2. Source comb\_summary.R
   1. This script will create a file that combines all the input data into one file. The file will be put in the same “*prepared\_data*” folder as the selected files
   2. A dialog box will open. Select all the prepared data files (these files are called “ont\_data\_summary\_ABEfilt\_[date]\_[plate].csv”)

**Step 3: Calculate the Normalized Mutual Information from the h5 files**

Open the following R scripts:

spikeLoadRountines.R, nmi2\_final.R, nmi\_wrapper.R, MI\_script\_all.R, and comb\_summary.R

Overview: These scripts will calculate the normalized mutual information parameter from the h5 files. This parameter is calculated separately because it takes a long time (approximately 1 hour per plate). It is best to run this script overnight. One file will be created for each culture date. The output will go in a folder called “*All\_MI*”.

1. Source spikeLoadRountines.R, nmi2\_final.R, and nmi\_wrapper.R
2. In MI\_script\_all.R, in the USER INPUT section, set the “basepath” variable to where you want the “*All\_MI*” folder to be created
   1. Change all backward slashes to forward slashes
   2. Source the MI\_script\_all.R
   3. This script will open a dialog box that asks for the h5files (the dialog box might not pop up or create an icon on the taskbar. If so, try minimizing RStudio to see the dialog box)
      1. If following Option 1, select all h5files
      2. If following Option 2, select the 4 h5 files for each DIV for the first plate
   4. If following Option 2, repeat step 11a for all plates. If all the output files are not in the same folder, put them in one master folder before the next step.
3. Combine the MI files into one master file, just as in the previous step
   1. In comb\_summary.R, in the USER INPUT section, set the desired name of the output csv file. Change all backward slashes to forward slashes
   2. Source comb\_summary.R
   3. The file will be put in the same “*All\_MI*” folder as the selected files
   4. A dialog box will open. Select all the MI files (“MI\_[date].csv”)

**Step 4: Calculate the Area Under the Curve from the 16 ontogeny values (from Step 2) and the mutual information (Step 3)**

Open the following R scripts:

burst\_parameter\_to\_AUC.R

Overview: This script merges the MI data with the other parameter values. Then, it calculates the trapezoidal area under the curve for each parameter and each well.

1. In burst\_parameter\_to\_AUC.R, in the USER INPUT section, set the “basepath” variable for where you want the output file to be created.
   1. The basepath should be in the main folder for all of the experiments (e.g. Organophosphates, if doing them in a batch all together)
2. Set the desired “filename” for the output
3. For “parameter\_data”, set it as the combined parameter .csv file, created in Step 2.9. (located in the “*prepared\_data*” folder)
   1. Remember to use the folder location and file name
   2. Change all backward slashes to forward slashes
4. For the “mi\_data”, set the file as the combined MI csv file, created in Step 3.12
   1. Remember to use the folder location and file name
   2. Change all backward slashes to forward slashes
5. If you want to normalize the data after calculating the AUC values, set normalize = TRUE. Otherwise, set normalize = FALSE
6. If you want to make the normalized response a positive, zero-centered response, set zero\_center\_positive\_response = TRUE. Otherwise, set it to FALSE
7. Source burst\_parameter\_to\_AUC.R. The output will go to the folder specified in “basepath”

**Step 5: Prepare the cytotoxicity data**

Open the following R scripts:

cytotox\_prep05\_rawValues.R

Overview: This script reads in the raw, blank-corrected cytotoxicity data for Alamar Blue and LDH and arranges the data in the mc0 file format. You can either input files with data for one plate (usually called "ON\_[date]\_MW[Plate#]\_Summary.xlsx”) found in the plate folder or for 3 plates for a given data (usually called "[date]\_ON [Group] Calculations.xlsx"). These files are usually found in the plate folders or culture date folders.folder.

1. In the USER INPUT section, set the basepath and the desired name of the output file (“outfile”)
   1. The basepath should but the main folder for all of the experiments (e.g. Organophosphates, if doing them in a batch all together) (same place as AUC file)
2. If your input files have LDH and Cell Titer Blue data for one plate in each file, set sheetdata = “one”. If your input files have data for 3 plates from a culture in each file, set sheetdata = “three”
3. Set ABname to the exact name of the excel sheet tab in your input files for the Alamar Blue data (sometimes called “CTB”, “Cell Titer Blue”, “AB”, etc.). Do the same for the LDHname
4. If you are creating a new output for the first time, set newFile = TRUE. If you are appending to an existing file of the same name, set newFile = FALSE.
   1. If the assays have different names across the different experiments, then create a new file for one set of experiments with the same name and then append to that file for those with a different name.
5. Source cytotox\_prep05\_rawValues.R
   1. A dialog box will ask for the input cytotoxicity data
      1. If following Option 1, select all files that have the specified sheetdata, ABname, and LDHname
      2. If following Option 2, select the desired file containing the cytotoxicity data
   2. Another dialog box will ask for the Master Chemical List(s). Select all of the master chemical lists corresponding to the plates represented in the selected input files. This is used to ensure that the chemical names are the same in the cytotoxicity data. If you want to use the chemical names in your input cytotoxicity data files, just hit cancel.
   3. The output file will go in the specified folder.
6. Repeat until you have one file for all plates and culture dates. Set newFile = FALSE, then repeat from step 21

**Step 6: Create the mc0 file**

Open the following R scripts:

tcpl\_MEA\_dev\_AUC.R

Overview: This script takes the AUC data and the cytotoxicity data as input. The AUC data is reformatted into an mc0 long file. Then, the cytotoxicity data is merged with the AUC (i.e., the rows are stacked together). This script also adds the label of wllt = “n” for all control wells and renames the treatment to the user-specified ControlTreatmentName (e.g., DMSO or water)

1. In the USER INPUT section, set the working directory to where you want the output file to go
2. Set the desired name for the output file “outfile”
3. Set the AUCsourcefilename to the input AUC file
4. Set the cytotox\_filename to the input cytotoxicity file created in step 5
5. Set the default\_ControlTreatmentName to the name of the most common vehicle control used in your experiment (usually DMSO)
   1. If there are some compounds that have a vehicle control other than the default, enter the compound names as a string vector in the variable different\_vehicleControlCompounds. If there are no other vehicle controls used, leave this variable as an empty list (c() ).
   2. Add the corresponding vehicle control names to the variable different\_vehicleControls. There should be a one to one correspondence between the control treatment names in this list and the different\_vehicleControlCompounds list.
6. Source tcpl\_MEA\_dev\_AUC.R
   1. The output file named “MEA\_DEV\_AUC\_mc0.csv” will be created in the specified folder
   2. The sample id column (“spid”) is currently set to the treatment name. This will be updated in the near future to show the actual sample id

**Step 7: (optional) Remove unneeded rows of data for specific plates and compounds**

Open the following R scripts:

removeRows2.R

Overview: Use this script if you need to selectively remove some rows of data from the output from step 6 for a given set of plates, for some compounds (for example, if some compounds from a different dataset were tested on extra rows for a set of plates).

This script will also create a text file that documents the changes you make.

1. In the USER INPUT section, set the desired working directory for the location of the output file
2. Set the desired name of the output file (“outfile”).
3. Set plates to the contain the names of the plates for which you want to remove some rows (see the ‘assay plate id (apid)’ column in the input file)
4. Set compoundName to the compound in those plates that you want to keep or remove. (Note the exact name of the compound in the ‘treatment’ column of the input file)
5. Set keepCompound
   1. If you want to remove all of the plate data except the rows with compoundName, set keepCompound = TRUE
   2. If you want to remove only the rows with compoundName for the plate data, set keepCompound = FALSE
6. This script will also create a log file, documenting the changes you make at this step. If you are doing this step for the first time, set newLogFile = TRUE. If you are making additional changes, set newLogFile = FALSE to append the current changes to an existing log file
7. Source removeRows2.R
   1. A dialog box will ask you for the input file. Select the mc0 file created in step 6 that you want to trim
   2. The output file will go in the specified folder
8. Repeat step 38 for any additional files you need to trim

**Step 8: Replace the “treatment” column with the sample ID’s (spid)**

Open the following R script:

spid\_mapping.R

Overview: This script is designed to replace the treatment column in the mc0 file created by tcpl\_MEA\_dev\_AUC.R with the corresponding sample ID.

1. In the USER INPUT section, set the basepath for the location of the output file
2. Set the desired name of the output file (“outfile”).
3. Set the mc0\_filename. This should be the output file from tcpl\_MEA\_dev\_AUC.R
4. Set the spidmap\_filename. This is the file that contains the sample ID’s.
   1. This file is expected to be .xlsx. If it is not, change spidmap = read.xlsx(spidmap\_filename) to read the appropriate file extension
5. Set the sheetName of the sheet in the spidmap\_filename that contains the sample IDs and chemical names
6. Set the spidCol as the name of the column in spidmap\_filename that contains the sample IDs
7. Set the mapMatchCol as the name of the column in spidmap\_filename that contains the chemical names in the same format as in the treatment column of the mc0 file
8. Set the mc0MatchCol as the name of the column in the mc0 file that contains the chemical names (this will always be “treatment”, unless you are using data from another source)
9. Source spid\_mapping.R.
   1. If you get an error, type “chem” in the console to see the current chemical that is having issues. See if there is a difference in how the chemical is named in the mc0 and spidmap files. If there is a difference, change the spelling, as demonstrated in the commented code around line 38.
   2. The output will go in the specified basepath location.