**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 version: 3.6.1

**Preliminary notes**

* Input data:
  + Spike List files from MEA recordings, created by the AxIS Spike Detector
    - 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 single 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 of the relevant chemicals and the corresponding sample ID (The sample ID column may be labelled “EPA\_SAMPLE\_ID” or “SPID”)
* Target output:
  + An mc0 file that is ready to be processed in the ToxCast Pipeline
  + Along the way, you will also get:
    - csv files with all ontogeny values
    - csv file with all AUC values

**Step 0: Prepare the files**

Download the R scripts from the BitBucket repository. Go to this link: <https://ncct-bitbucket.epa.gov/projects/NSLTM/repos/nfa-spike-list-to-mc0-r-scripts/browse>

Underneath the header “Source” there should be 2 gray boxes that say “master” and “…”. Click on the “…”. Then select “Download” and save the zip file. Create a folder for the scripts in your project folder and unzip the scripts into that folder.

Locate the spike list files for the current data set. These are usually named, for example, *ON\_20151209\_MW1056-28\_05\_00(000)\_spike\_list.csv*. This long file should have 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 named *[date]\_MW[Plate#]\_MaestroExperimentLog\_Ontogeny.csv*

Locate the files containing the cytotoxicity data for all plates. You can either use the “Calculations” excel files located in each culture date folder (e.g. *[date]\_ON [Group] Calculations.xlsx*) or the “Summary” excel files located in each plate folder (e.g. *ON\_[date]\_MW[Plate#]\_Summary.xlsx*). The “Calculations” files contain data for 3 plates, and the “Summary” files contain data for one plate. The script **cytotox\_prep06.R** will extract the raw blank-corrected data from these files.

Check the lab notebook to confirm which culture dates and plates should be used. If you need to remove only some wells from a plate or culture date, you can do that in step 7.

Copy all of the scripts in the folder “L:\Lab\NHEERL\_MEA\NFA Spike List to mc0 R Scripts\BitBucket Connect” to your working directory.

Processing Option 1 (preferred):

Put all 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, e.g. “All Master Chem Lists”. Lastly, put all the files containing the cytotoxicity data in another folder. (Note: the script **h5\_conversion.R** may have accessibility issues if the spike lists files are on the OneDrive – Environmental Protection Agency (EPA) drive)

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.

**Install required R packages**

1. Use the command install.packages(“package name”) to install any of the following packages that you do not already have:
   1. “h5”
   2. “gtools”
   3. “xlsx”
   4. “reshape”
   5. “data.table”
   6. “devtools”
   7. “BioInstaller”
2. To install “rhdf5”, use 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. Use the command devtools::install\_github(“package name”)to install the following packages from GitHub:
   1. "sje30/sjemea"
   2. "dianaransomhall/meadq"

Additional info: <https://github.com/dianaransomhall/meadq>, <https://github.com/sje30/sjemea>

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

**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 named “*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.
   1. For all directory names in R, change all backward slashes to forward slashes
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 asking for the Master Chemical Lists
      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 6 for each plate.
   5. This script completes approximately 1 h5 file per minute

**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**

Overview: 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 named “*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 statement will appear multiple times:

\*\*\* No network spikes found

Warnings generated are normal

* 1. If following Option 2, repeat step 9 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 (e.g. *[name]\_all\_prepared\_data.csv*)
   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 prepared data files created in step 9 (named *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 is significantly more computationally intensive (approximately 1 hour per plate). 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
3. Source **MI\_script\_all.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
      2. If following Option 2, select the 4 h5 files for each DIV for the first plate
   2. If following Option 2, repeat step 14 for all plates. If all the output files are not in the same folder, put them in one master folder before the next step.
4. 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.
   2. Source **comb\_summary.R**
   3. A dialog box will open. Select all the MI files (*MI\_[date].csv*)
   4. The output file will be put in the same “*All\_MI*” folder as the selected files

**Step 4: Calculate the Area Under the Curve of the 16 ontogeny values and the mutual information**

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 over time in 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. It is suggested that you create an “*Intermediate Output*” folder for this file
2. Set the desired “filename” for the output file (e.g. *[name]\_AUC.csv*)
3. Set “parameter\_data” to the combined parameter .csv file, created in Step 2.11. (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 3.15
   1. Remember to use the folder location and file name
   2. Change all backward slashes to forward slashes
5. 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 script:

**cytotox\_prep06.R**

Overview: This script reads in the raw, blank-corrected cytotoxicity Alamar Blue and LDH date and arranges the data in the mc0 long file format. You can either input files with data for one plate (usually called *ON\_[date]\_MW[Plate#]\_Summary.xlsx*), 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.

1. In the USER INPUT section, set the desired “basepath” for the location of the output file and the desired name of the output file (“filename”)
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 one culture in each file, set sheetdata = “three”
3. If you are running the script for the first time or want to overwrite an existing file with the same filename, set newFile = TRUE. If you are appending to an existing file of the same name, set newFile = FALSE.
   1. If you have some “Summary” files and some “Calculations” files, you can run this script for the first set of files and then re-run with the new sheetdata value and newFile = FALSE.
4. Source **cytotox\_prep\_06.R**
   1. A dialog box will ask for the input cytotoxicity data
      1. If following Option 1, select all files that have the specified number of sheets in “sheetdata”
      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). You can 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 in the cytotoxicity data match the names in the MEA data. If you want to just use the chemical names in the cytotoxicity data files, just hit cancel.
   3. If the script cannot find the tab in the excel sheets corresponding to the Alamar Blue or LDH data, you will be prompted to enter the name of the tab in the console
   4. The output file will go in the specified folder.
5. Repeat from step 22 until you have one file with LDH and AB data for all plates and culture dates.

**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 rows are appended to the merged with the AUC data. This script also adds the label of wllt = “n” for all control wells and renames the treatment to the user-specified control treatment names (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 (“filename”)
3. Set the” AUCsourcefilename” to the input AUC file created in step 4.20
4. Set the “cytotox\_filename” to the cytotoxicity file created in step 5.25
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 in the string vector “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. Set “check\_unique\_apid” = TRUE. The script will check if a plate ID was used in multiple culture dates. If so, a letter suffix will be added to distinguish the 2 plate IDs. The script will also compare each plate ID to the plate ID’s in the previously pipelined data (located here L:\Lab\NHEERL\_MEA\tcpl\_nheerl\_mea\_dev\source\_files) and will rename the plate IDs if needed. Any plate ID name changes will be printed to the console.
7. Source **tcpl\_MEA\_dev\_AUC.R**
   1. The output file will be created in the specified folder

**Step 7: (optional) Clean up the data set as needed**

For each data set, you may need to create a custom script to make adjustments as needed. A few issues to consider:

* Are there any notes in the lab notebook indicating that some wells are not useable? If so, set wllq = 0 for these wells. The data for these wells will be removed in level 2 in the ToxCast Pipeline.
* Are there compounds that were tested on these plates that should not be included in the dataset? For example,
  + Was some of this data already pipelined for specific compounds on some plates (we don’t want duplicate data and artificially increase the number of replicates)
  + Are there any compounds for which you do not have Sample ID’s?

Remove the data rows corresponding to these compounds as needed. Note that the control treatment wells have already been renamed to “DMSO” (or other vehicle control name), so you will have to reference these data rows by rowi and coli, not just the treatment used in that plate row. The script **removeRows2.R** may serve as a useful starting point for implementing these changes.

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

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 IDs.

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 (“filename”).
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 the line spidmap = read.xlsx(spidmap\_filename) to read the appropriate file extension
5. Set the “sheetName” to 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. Make sure that there is only 1 column with that name in the sheet.
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 message, read the message, or type “chem” in the console to see the current chemical that is having issues. Check 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.