Guide to Prepare Data from Network Formation Assay Micro-electrode array Experiments for Dose-Response and Hit Calling with the ToxCast Pipeline

# Next steps

(remember ultimate goal is to keep pre=processing a bit at EPA, then send up to IUF! And they may supersede with their scripts!)

* Briefly (10 min) check for existing list of things to update for the MEA NFA (so that I’m not rediscovering and recording issues)
* Keep blazing through this guide, the notes template, and the run\_me code. Want to end with something that is “to the end”, even if I’ve earmarked a few holes.
  + I’m not sure if I should try to update the run\_me template at this point, if I should even make that a thing.. yeah, when I send to IUF, I think that would be good.
* Make little updates as you go. Whenever you encounter something big that may need to be updated, note that
* Review recent updates to the HCI data pre-processing, and any to integrate to the MEA to the list of to-do
* Anything else to add to General/Tickets for future updates to scripts?
* Then go back and triage -> what absolutely needs to be done? What’s medium, what’s lower priority? Probs get Tim’s insight as well.
* Could the scripts/guide possibly be ready enough for Kelly to drive the bus for the Hitting a nerve project?

# Notes

Turquoise highlights = things that are not directly translatable to IUF

Yellow = need to go over this

# Setting up the MEA NFA pre-processing folder for the first time

* Go to your Documents folder (on your C drive)
* Open Git Bash
* Clone the MEA NFA pre-processing folder
  + ‘git clone <https://ccte-bitbucket.epa.gov/scm/mea/pre-process_mea_nfa_for_tcpl.git>’
* Initialize an R project in the newly created folder “pre-process\_mea\_nfa\_for\_tcpl”

Note: all file path references going forward will be relative to your folder “pre-process\_mea\_nfa\_for\_tcpl”.

# Setting up the data to be processed

Decide on a name for your dataset. Usually I use “name of compounds” followed by the year the experiments were started, e.g. “PFAS2018.” Don’t use any spaces.

Create a copy of the “Template” folder in this directory and rename it as your dataset name.

# Data Preparation

Open the OneNote Notebook: L:\Lab\NHEERL\_MEA\Carpenter\_Amy\pre-process\_mea\_nfa\_for\_tcpl\Pre-processing\_MEA\_NFA\_for\_TCPL

Make a copy of the *Note Template* tab, and name it as your dataset name.

Follow the instructions to fill out the form.

\*\*Go over the notes template

# Running the scripts

## 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. “openxlsx” – includes functions for reading .xlsx Excel files
   2. “data.table” – for robust data manipulation
   3. “gtools” – includes useful functions such as ‘asc’ for getting ascii character code
   4. “devtools” – includes function needed to install packages from GitHub
   5. “pracma” – used in mutual information scripts
   6. “compiler” – used in mutual information scripts
2. Install the package “rhdf5” for reading, writing, and opening h5 files. 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.

For additional information on Bioconductor and the hdf5 package, see <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 information: <https://github.com/dianaransomhall/meadq>, <https://github.com/sje30/sjemea>

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

## The “run\_me” script

In the folder *nfa-spike-list-to-mc0-r-scripts\R*, create a copy of the script *run\_me\_template.R* and save to the same folder as the rest of the scripts. Rename the copy of the scrip to *run\_me\_datasetname.R*.

In the “run\_me” script, fill out the “USER INPUT” section:

* Set the dataset\_title to the dataset name you created.
* Set pause\_between\_steps to TRUE, or FALSE if you have don’t want to pause between steps.
* Set save\_notes\_graphs to FALSE to view all output and graphs in the console. After you have ran through the steps, set to TRUE and re-source the run\_me to save a log of the notes and plots.
* Set default\_ControlTreatmentName. This is usually “DMSO”.
* 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()).
* 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.
* Set spidmap\_file to the file path of the Excel file containing the Sample ID’s and stock concentrations of the compounds in the current data set. Be sure to change all backslashes “\” to forward slashes “/”.
* Set spid\_sheet. To the sheet in the spidmap\_file that you want to use. Can be a number or the name of the sheet.
* Set scripts.dir. Use the default setting. This is the folder containing the scripts that will be referenced.
* Set root\_output\_dir. This is where the output will be created. Use the default setting.

Source the *run\_me* script line by line:

* Under the section “run the main steps”, source the script *source\_steps.R.* This script will automatically run through each step. If pause\_between\_steps is set to TRUE, you will be prompted to enter y/n before continuing each step. The script will also check if a step has been run before. If so, you will be able to select if you want to continue with the existing files, remake all of them (i.e., overwrite), append to the existing files, or quit. You can quit and re-source this line as many times as needed.
  + Selecting files
    - Select all file types needed for the analysis (\_spike\_list.csv, \_MaestroExperimentLog\_Ontogeny.csv, and Calculations/Summary xlsx files containing the cytotoxicity data)
    - When you have selected all files, hit “Cancel” – then the selected files will be saved in a text file.
* Under the section “prepare spidmap”, you will read in the spidmap\_file. (Be sure to close the spidmap file in Excel before reading the file in R). You will need to standardize the names of the treatment, stock\_conc, and spid columns in the spidmap. In the line that says “setnames”,
  + Update trt\_col to the column name in the spidmap that corresponds to the chemical names. The chemical names should match the names in the “treatment” column of the AUC and cytotoxicity data.
  + Update the conc\_col to the column in the spidmap that lists the stock concentration of the chemicals (this will be used to confirm the concentration-correction where the stock concentration is not exactly 20mM).
  + Update the spid\_col to the column in the spidmap that lists the EPA Sample ID corresponding to each compound. This column is . The sample IDs (or SPIDs) usually begin with a prefix such as “EPA”,“EX”, “TP” or “TX” followed by a 6-8 digit code.
  + If you need multiple spidmaps, you can read them in separately and then combine them with rbind.
  + The “expected\_stock\_conc” is the target concentration. This is the expected concentration that the dilutions were based on. This is usually 20mM. Sometimes, it is 30mM. Sometimes, for individual compounds in a dataset, the lab technician sees that the actual concentration is not 20, and so adjust the source\_conc’s accordingly… more to explain
    - Show examples, esp where expected is 10 and actual is 10.1?
* Run tcpl\_MEA\_dev\_AUC
  + If you get an error stating that some treatments don’t have a corresponding spid in the spidmap, you may need to rename any compounds that were misspelled in the auc/cytotox data. Uncomment the section under “rename any compounds” and update as needed.
  + This script will also check the concentration corrections for each compound. Follow the prompts to update any concentrations that look off. It assumes that the expected aliquot concentration for each compound is 20. If that is not the case, you can change this default by adding the argument expected\_target\_conc = 30 (for example)
  + Other things to check:
    - If it appears that the conc’s were partially conc-corrected (e.g. corrected in cyto data but not AUC dat) -> need to standardize the conc’s before you can continue
      * Suggest how user could do that??
    - If it appears that the conc’s were conc-corrected incorrectly (show example -> spidmap\_guess\_conc’s does not agree with actual conc’s, but actual conc’s are not 0.1,0.3,etc) -> Then need to standardized the concs or something before can correct them
* Run the final data checks
  + Read through the output and confirm that there are the expected number of cultures, plates, etc., no missing data, etc.
  + Take a look at the output plots. There isn’t really anything specific to look for in the plots… just check that nothing looks waay off, many missing values, etc. Compare control wells to treated values in each plot and see if it looks reasonable.
  + Feel free to do any other checks that you want!

In the end, you should have a file in the *output* folder called “datasetname\_longfile.csv”

Once you have successfully made it through all of the steps, set save\_notes\_graphs to TRUE and pause\_between\_steps to FALSE. Then source the entire “run\_me” script again. A “run\_log” text file and a “summary\_plots” folder will be created as documentation.

Conc-correction function:

* If a compound in your dataset is tested at different concentrations than those listed under “expected\_target\_concs”, this compound will be flagged as a compound whose concentrations should be corrected. However, if the stock concentration is 20, the concentration-correction will not affect these values.
  + Not a great long-term solution, because what if the stock conc is not 20!

Dataset checks

* Make sure you review the results, make sure the values look reasonable, no glaring unexpected holes!

# Final checks (!)

(per recommendation from Kelly)

Might be good to add a final step after pre-processing, before the data is sent to TCPL, to:

1. Have the lab look at the data, particularly to make sure that the chemicals that were expected to be present are all there (nothing missing, nothing added, probably expected number of replicates, conc range, etc).
2. Have a biologist look at the data, see if things look as expected