## **Nutil to Usable (N2U)**

02.29.2024

***Purpose***

This program takes either the output from the QUINT workflow or a specific file format and allows you to create a heatmap that can be used to display the data or to QC the data for further analysis. It provides an easy way to create a value for regions of a chosen variable or calculate a percentage using multiple chosen variables. It outputs the heatmap itself and a CSV file which is compatible with a Shiny app created by Zach Madaj. The two applications are compiled onto a singular webpage at: **https://github.com/MXHend/Mouse-Brain-Heatmap-Website.**

***Preparation Using Nutil Files from the Quint Workflow***

First, create a folder which will contain all your data for this project and name it whatever you would like. Within this folder, create two folders labeled ‘left’ and ‘right’. If you only have one side of data, just create a singular folder labeled ‘data’.

After completing the QUINT workflow, you should have a folder (or more likely, multiple folders) called “Reports” which contains another folder with an ending label of RefAtlasRegions. In this folder will be at least two files with one of them having the same name as the folder. Ignore that file and copy the rest into the appropriate folder within the project folder you created in the beginning. Do this for every report folder you have until all the files are located in those two (or one) folders in your project folder.

You may now open the website.

***Preparation Using Outside Data***

First, create an excel spreadsheet file with the following columns:

* The first column should be labeled ‘**region**’ and should contain Alan Brain Atlas region abbreviations such as ACA for anterior cingulate area. These can be as specific or as broad as you would like
* The second column should be labeled ‘**side**’ and should contain either ‘left’ or ‘right’ based on what side the data is from.
* The rest of the columns will have names that you decide upon. Each of these columns will contain your numeric data. The column names should be **unique** from one another and should be enough for you to remember what type of data is contained in them.

Save this file as a **CSV**.

You may now open the website.

***Using N2U for the first time:***

1. Before running anything you should select the tab corresponding to what type of input you have
   1. If you have the **Nutil files**, select the **‘Nutil Data’ tab**
   2. If you have the **CSV** you created, select the **‘Non Nutil Data' tab**
2. **If you have Nutil Files:**
   1. Begin by hitting the browse button labeled for the **‘left side’** and select all files in the ‘left’ folder
      1. If you have a ‘data’ folder, select all files in that folder instead
   2. Hit the ‘open’ button to start loading the data
   3. While that is loading you can hit the browse button labeled for the **‘right side’** and select the files in the ‘right’ folder and hit the ‘open’ button to begin loading that data as well
   4. Wait until there is a message on the right side of the app that says the last side you loaded is done
3. **If you have the CSV:**
   1. Hit the browse button, find your file, select it, and then hit open
4. Double check that the above step has appeared and then click the button labeled ‘**click this when all files are done loading**’
5. Wait until the message on the right side changes and mentions the ‘blank annotation’ and the ‘checkpoint’ are done
6. Press the button labeled ‘**Download Checkpoint 1**’ and save it within your project folder – This will be used the next time you load the data
7. Press the button labeled ‘**Download blank annotation file**’ and save it within your project folder
8. Open your file explorer and navigate to your project folder
9. Open the **annotation** file you just downloaded and **fill** **out** all columns labeled with a NA (if you don’t have a column, then you may leave it NA)
   1. Mouse – Mouse name
   2. Sex – Mouse sex
   3. Treatment – Mouse treatment type
   4. MPI – Months post injection
   5. Genotype – Mouse genotype
   6. Marker – What you are comparing in the mice such as pSyn vs NeuN
   7. Batch – This is what you can use if you would like Batch correction in the MBH app
   8. Include – Use this column when doing QC to remove mice from the data easily
      1. Y = include
      2. N = do not include
10. **Save** this file as a CSV file with a different name so that you know you have completed it
11. **Reload** the webpage or stop and restart the RStudio app

***Using N2U when you have saved checkpoint 1 and completed filling out the annotation file:***

1. Begin by clicking the Browse button labeled by the phrase **‘Input checkpoint 1 here’**, select the “checkpoint1.rds” you saved earlier and click ‘open’
2. Check the right side of the screen for confirmation that the checkpoint is done loading and then click the browse button to **input in the annotation file** you finished creating
3. Once that has completed (check the right side for text saying has finished), **download Checkpoint 2** somewhere you have easy access to
   1. This will be for if you would like to come back to the app again later or if the app manages to fail at any later step.
   2. This checkpoint will explicitly go into the one labeled CSV as the other one is for older users who made checkpoints before the swap to CSVs.
4. Once that is complete you may notice that the box labeled for selecting your ‘**x axis variables**’ now has options to choose from. Please note the following when you select these variables:
   1. These variables will make up the **rows** of the heatmap with the labels for them showing on the **left side** of the heatmap
   2. These variables will also be displayed individually in the form of a color key on the **outside** **left** of the heatmap
   3. They will be combined and if they don’t make sense, such as combining ‘daughter’ (region) with ‘mouse’, they will mess up your heatmap or error the program
   4. **Common Examples**:
      1. ‘Mouse’
      2. ‘Marker’ and ‘Mouse’
      3. ‘Marker’, ‘Mouse’, ‘Genotype’, ‘Sex’, ‘Treatment’, ’Batch’, and ‘MPI’
5. Next, select the **‘y axis variables’** and note the following when selecting them:
   1. These variables will be the heatmaps **columns** and will be labeled on the **bottom**.
   2. You will choose the color key display for this axis separately
   3. They will be combined, so double check they make sense avoiding a similar issue such as described for the x axis
   4. **Common Examples**:
      1. ‘Daughter’
      2. ‘Side’ and ‘Daughter
      3. ‘Side’ and ‘Parent’
      4. ‘Side’ and ‘Specials’
         1. ‘Specials’ is a special variable containing the following ABA regions: CTXsp, fiber tracts, HPF, HY, Isocortex, OLF, PAL, STR, TH, MB, P, MY, and CB
6. Select the **‘display’ y axis variables** next noting the following:
   1. These variables will be displayed in the form of a color key on the **outside** **top** of the heatmap
   2. These must make sense for this axis such not using regions and mice together
   3. Please note that the more elements in a variable the less distinguishable the color key is going to be (i.e. don’t use ‘daughter’ and use ‘parent’ instead). This is a good place to use ‘specials’
   4. **Common Examples:**
      1. ‘parent’
      2. ‘side’ and ‘parent
      3. ‘side’, ‘parent’, and ‘specials’
7. When you have selected variables for all the above options, click the button labeled **‘Get Options’**.
   1. This button will use the data you just gave it to give you more specific options for your heatmap
8. Next, look at all the options that are selected in the box labeled with **‘variables to see’** and confirm you want to see all of the above variables on the heatmap
   1. You will only ever touch this in an instance where you want the calculations to be the same, but the display to only show part of it
   2. To not include something in the calculations, remove it using the annotation file
9. **If you are in the Nutil Data tab:**
   1. **This step will determine the values of the heatmap, so pay close attention:** From the dropdown list labeled **‘select the nutil variable’**, select the variable you will be using as your value (i.e. Object Count)
      1. Most of these variables are self-explanatory, but there is a bunch of options at the top of the list containing ‘load’ which are calculations run in the app. Scroll to the bottom of this document for an in depth explanation on these and how to use them. You may need to add an additional column to your checkpoint 2 to be able to use them.
10. **If you would like the data to be multiplied by 100** the check the corresponding box
11. **If you would like the data to be in** **log 10 scale**: then check the corresponding box
12. **If you would like your heatmap to be trimmed:** check the corresponding box (This does not affect the calculations)
    1. Atrimmed heatmapremoves any column with more than 50% of the data missing
13. If you would like **to remove regions** from your heatmap, select any amount of them from the drop down labeled **‘Select any regions to remove’**
14. **If you would like row and column names**: select from the corresponding drop-down boxes and select what you would like to see
    1. x and y are the combined column names that are created when you first select the x axis variables and y axis variables and will most likely contain the most information
15. The **color scale** is **not** something you will need to mess with your **first time** loading the heatmap
16. Now that you are done selecting the base variables for your heatmap, click the button labeled **‘Create heatmap’** and wait for a preview to appear in the right side
    1. Once this is done, you will probably note that the colors look wonky. This is due to the color scale I mentioned before to skip over.
17. **Now that the heatmap is done**, you will note some text at the top of the right side which says ‘min’ and ‘max’. Please enter these values into the textboxes labeled **‘Min Value’** and **‘Max Value’** for the **color scale**
18. At this point, you can look at the options under the label **‘Custom Colors**’ and determine if you would like to modify the colors at all.
19. The first option is to invert any Viridis color scheme used.
20. If you would like a different Viridis color scheme, select the checkbox and then select a new color scheme
21. If you would like a scale of 2 or 3 custom colors, select the corresponding boxes and adjust the colors.
22. If you accidentally select more than one it will go back to the default.
23. When you are done, click the **‘create heatmap’** button again and wait for the preview to appear
24. **To save the heatmap**: you have two options you can choose from or download both
    1. Click the button labeled **‘Download full Heatmap’** and this will download a heatmap with all information you selected above. Make sure to download it into a directory you can easily access to check if you like your chosen options
    2. Click the button labeled **‘Download Just the Plot’** and this will download a heatmap without any of the legends for easy editing. Make sure to download it into a directory you can easily access to check if you like your chosen options
25. Look at your heatmap and determine **if you need to change anything**. If you do, start from step 3 of this section and check over each step one at a time
    1. **Don’t forget** **to press the button in the middle** while changing options before attempting to recreate the heatmap again.

***Using N2U when you have saved checkpoint 2:***

1. Begin by clicking the Browse button labeled by the phrase **‘Input checkpoint 2 here’** and select “checkpoint2.csv” you saved earlier and click ‘open’
2. Check the right side of the screen for confirmation that the checkpoint is done loading and then **continue from step 4 above**.

***For integration with the Mouse Brain Heatmap (MBH) app OR additional numeric and/or annotation information for the heatmap:***

1. If you think all of your data is being created the way you want it to and would like **to take it into the MBH app**, click the ‘**Download MBH Table’** button and save it wherever you can access it
2. If you would like **the color scheme to also be the same in the MBH app**, click the **‘Download Colors’** button
3. If you would like the values of the heatmap or any of the axis annotation information of the heatmap, click the buttons labeled **‘Download Data’, ‘Download AnnoX’, and ‘Download AnnoY’**
4. Finally, **if you would like access to the many variables created during this program running or the data frame that contains the raw data at any point**:
   1. Open the app in R Studio
   2. Run through all of the steps
   3. Once it has created a heatmap, wait a minute or two for everything to fully load
   4. Click back into R Studio without closing the app
   5. Find the console in the bottom left corner and click the button that looks like a stop sign ONCE and only once
   6. Wait very patiently for the app to close on its own
   7. If it gives you an error like below click ‘No’ and wait some more
      1. Graphical user interface, text, application, email

         Description automatically generated
   8. Eventually it will stop the app and variables will appear in the top right box labeled ‘environment’
   9. To see these variables, double click their names to open a viewer pane
   10. To save a variable as a csv file write the following code in the console:
       1. write.csv(variableName, “filename.csv”)
       2. **If you would like to designate the location of the file,** include the file path in front of the filename with **all back slashes replaced with forward slashes**
          1. “~\projects\file.csv” is **incorrect**
          2. “~/projects/file.csv” is **correct**

***Measures***

To elaborate the options available in the measurement variable step (step 9) of N2U, below are descriptions for each option:

* **Load**: This measure calculates the proportion of pathology occupying each region. Nutil outputs contain a column for object area (segmented pathology) and for region area. The two load measures in N2U (‘*ParentLoad’* and ‘*DaughterLoad’*) generate the load outputs by taking the object area/region area. This proportion can be converted to a percent by multiplying by 100.
  + **Daughter load:** Regions are split down to the daughter level (individual layers/subregions). Within the same mouse, some daughter regions get sampled across multiple figures. Daughter load averages object area and averages region area as they appear across the same mouse to generate a daughter load value that takes account for difference in region size (e.g CP appears bigger in fig 46 than it does in fig 67 or 75, by dividing the average of object area by average region area we get a more accurate load value that takes region size into account as it appears across the 3 figures).
    - This is the same as dividing the sum of object area and sum of region area as it appears across figures/section
  + **Parent Load:** Regions are split into the parent level (layers/subregions are compiled into single regions). Since Nutil always outputs data at the daughter level, N2U is responsible for averaging across subregions. Instead of taking the average load value (object/region area) of all daughter regions, when you select for ‘ParentLoad’, the app will divide the average object area over average region area as they appear across subregions. This will result in a parent region load value that takes the size of the parts into account (the app has been programmed this way to avoid weighting subregions with very little area (e.g. cortical layer 6b) equally with other larger subregions).
    - Subregions spanning across multiple figures/sections will be averaged in a similar way as in daughter load.
* **Inclusions/mm^2:** This measure calculates the density of positive cells in each parent or daughter region at the mm^2 level. The measure is derived from taking the Nutil columns *Object Count* and *Region Area (*μm*^2):* the app divides object counts (+ cells) by the region area converted to mm^2 (area in μm^2 \* 10^-6) for each region.
  + Be sure you turn off the setting for Object Splitting in Nutil, otherwise you’ll end up with NAs in the column for Object Count in the Nutil outputs and this measure will not run.
* **Neurite Load:** This measure calculates the proportion of pathology outside of cells in each region. You can run two separate segmentations in QuPath: one for all pixels above a threshold across the full image (total area) and another for pixels above a threshold within detected cells (inclusion area). By subtracting inclusion area from total area, we’ll have neurite area (extracellular area) that can be used to calculate neurite load. The steps to utilize N2U to run this calculation are listed below:
  + To start, run the Nutil files generated from the full or total area segmentation through N2U up to checkpoint 2, this will be our *total* checkpoint2
    - the checkpoint2 csv should have a column titled ‘Object Area’ that refers to whatever you had segmented which in this case is the *total* area
  + Run the Nutil files generated from the segmentation restricted within cells through N2U up to checkpoint2, this will be our *inclusion* checkpoint2
    - In this checkpoint2, object area refers the *inclusion* area
  + Copy the *total* checkpoint2, rename it to checkpoint2 *neurite*
  + Copy the object area column from the *inclusion* area checkpoint2 and paste it into the next free column in the checkpoint2 *neurite* (after all the total
  + Rename the newly pasted column as ‘inclusion area’
    - It is important that inclusion area is named that so that it can be found by N2U

When you select for ‘NeuriteLoad’, the app subtracts the *inclusion* area from *total* area (total area labelled ‘*object* area’ in checkpoint2 neurite) giving us a difference that represents neurite (or extracellular) area. These values are then divided by region area to generate neurite load. The same subregion and multiple sampling averaging as in parent/daughter load occurs here.