Spike Analysis of Spike2 Files

Introduction

This project consists of two folders - **spike2_to_spykingcircus** and **spike_processing**. The Programs in **spike2_to_spykingcircus** are written with the purpose of transforming raw Spike2 (.smr) files into a file that Spyking Circus can read. The spike_processing folder consists of programs that can transform the Spyking Circus output into Excel files.

In depth information on the intermediate HDF5 files can be found in the accompanying guide "HDF5 Guide"!

This guide goes in more detail later but to start with, **spike2_to_spykingcircus** has two programs needed to be run inside it -

1. **spike2_to_hdf5** - This program accepts raw Spike2 recordings and transforms them into an HDF5 file that has data from all the channels recorded (e.g U1, LFP1 etc). If the user chooses to, they can input more data like dates and experiment conductors. Select atleast 2 files!
2. **channel_extract** - This program extracts certain data (channels) of the users specifications from spike2_to_hdf5. This is useful for extracting only certain channels like the Unit Recordings Channels for Spyking Circus.

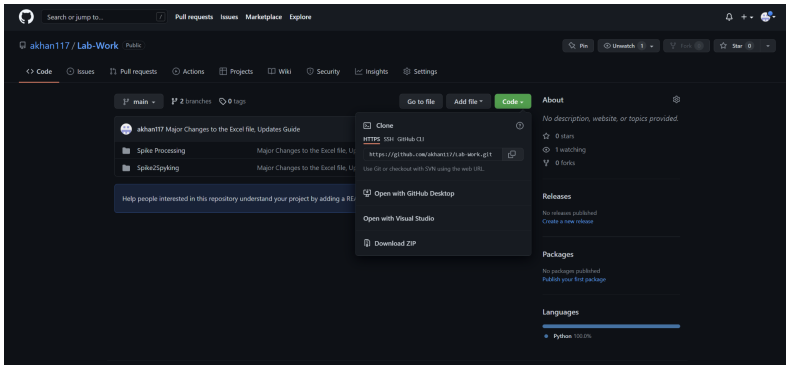
Similarly, **spike_processing** has two additional Programs in it, which come after running Spyking Circus.

1. **event_spike_merger** - The program creates the penultimate files needed for the complete Excel File. It combines Spike Data obtained from Spyking Circus with the Event Data (Usually the timings of the Odors).
2. **activity_ extractor** - This program, based on the users chosen parameters, uses the files obtained from event_spike_merger to create our readable excel file, detailing information on the spikes.

This program was written in a Conda environment using PyCharm, so I'd suggest getting those, but first here's the github link for the Project, which is now Public.

[akhan117/Lab-Work (github.com)]

You may download the project from there as a zip file.



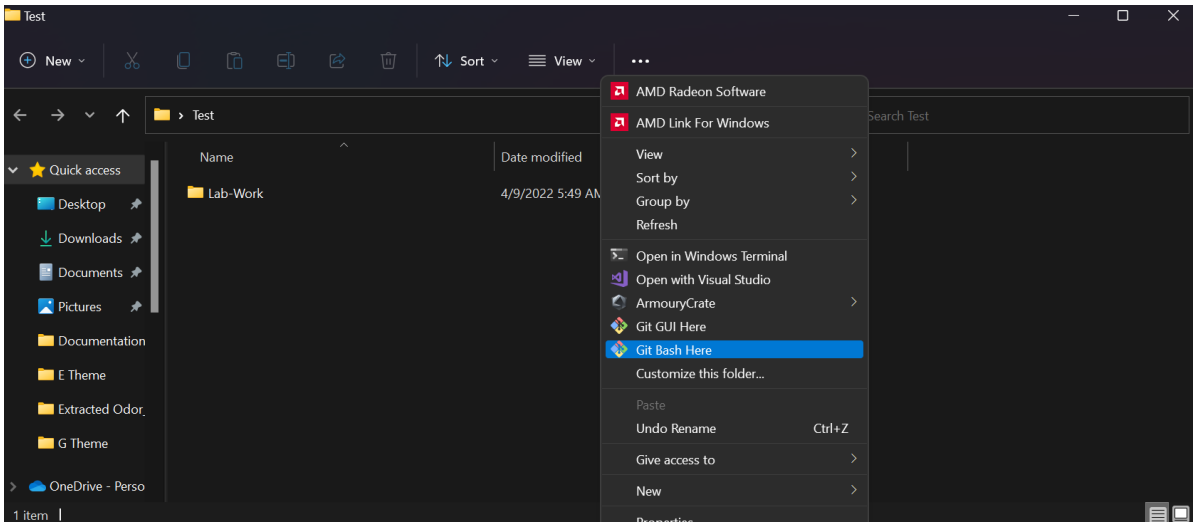
Alternatively, you may download and install Git, a command window for downloading and updating repositories.

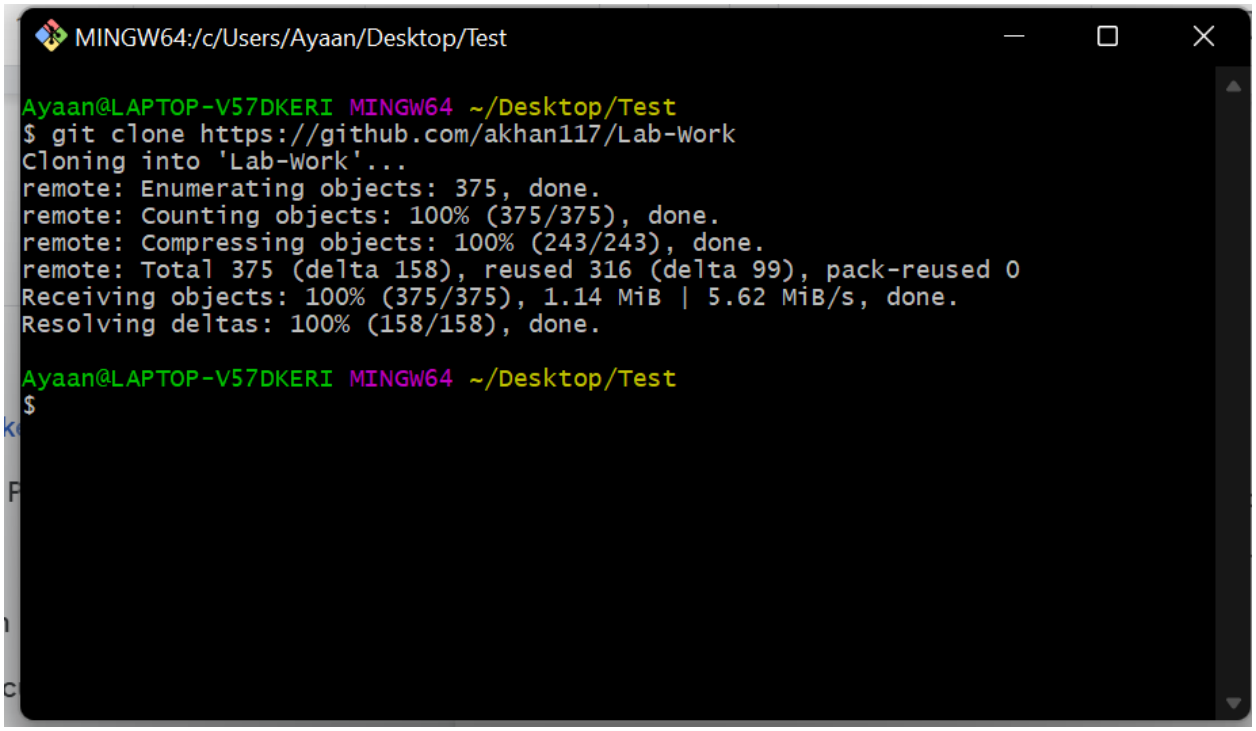
[Git for Windows]

[Git - Downloading Package (git-scm.com)] (For Mac)

Upon installing and running Git, you may obtain the repository by running the following command, just be sure to run the command in a folder you want to repository to be downloaded to -

git clone https://github.com/akhan117/Lab-Work





Installing Conda and PyCharm

If you're familiar with Python, Conda and have an IDE (Integrated Development Environment) - like for example Pycharm or Microsoft Visual Studio with its Python extension, feel free to skip this section.

First - What exactly is Conda? Well, Conda is simply a way to manage Python packages and update them- packages being Python programs that developers have written for others to use. We'll be using it to obtain a few packages required to run this project.

Secondly - What is Pycharm? In the simplest sense, Pycharm is a place to write and execute Python Code, also referred to as an IDE (Integrated Development Environment). A majority of the work will be done here.

If you're not familiar with installing conda, I'd suggest using this guide –

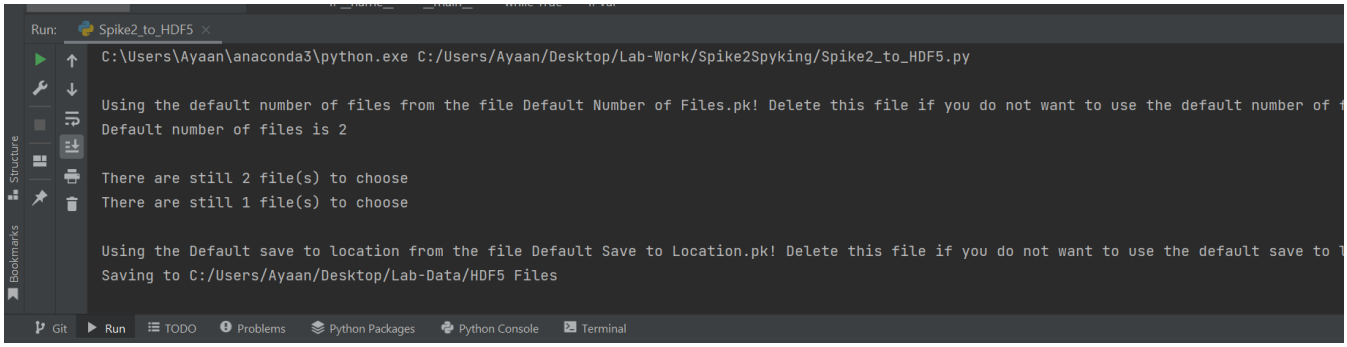
> Downloading conda — conda 4.10.3.post40+c1579681 documentation

After that, install PyCharm and open the project. you'll probably want to use the Conda installation you just downloaded for it, so I suggest following this guide –

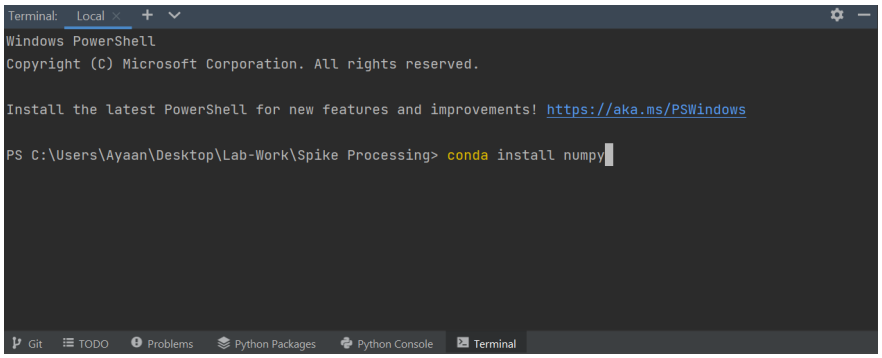
> Configure a Conda virtual environment | PyCharm (jetbrains.com)

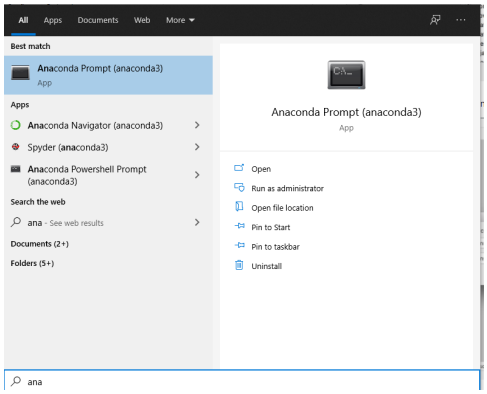
Required Packages

There are some packages required to be installed to run this program. Try running the following lines of code in the PyCharm terminal. If it's giving you trouble and you're not familiar with getting conda to run, try using the anaconda terminal to run these commands.



In this above image of Pycharm, the Terminal option is in the **"Bottom Middle"**





Pictured above is how to access the Anaconda terminal.

You'll need to install numpy, h5py, neo version 0.6.1 and of course, spyking-circus. Run these commands.

conda install numpy

conda install h5py

conda install –c conda-forge python-neo=0.6.1

conda install –c conda-forge –c spyking-circus spyking-circus

I don't suggest copy pasting these commands, since word seems to do something to the hyphens – you may have to manually type them out. There may be other packages needed as well, but python should let you know.

Running the Program

Now, we've downloaded our program from github, opened it in PyCharm, set Conda as the environment for it, and installed the packages we need to run it, if the steps above have been followed and work. So how do we actually run the program?

DISCLAIMER : The programs will at times give you the option to name files. Outside of these, do not rename any of the other files generated by the programs - this will break its functionality!

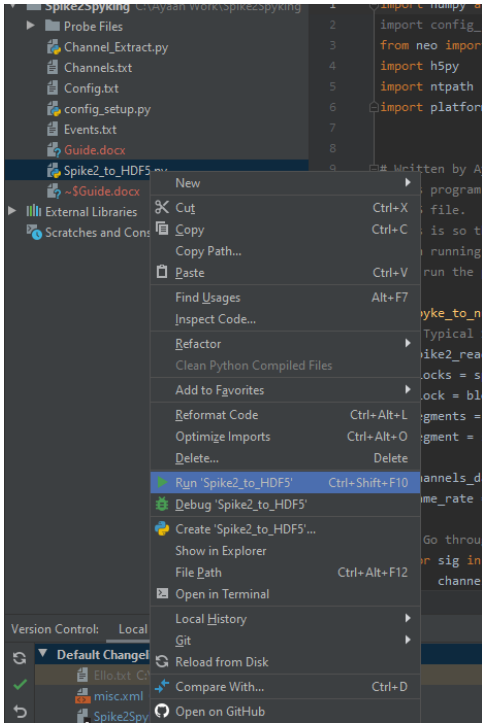
Spike2Spyking

In PyCharm, open Spike2Spyking.

a) spike2_to_hdf5.py

DISCLAIMER: Always use at least 2 files - this program was created under the assumption that you use both pre and post infusion data. If you do not do this, things may break in activity_extractor!

spike2_to_hdf5.py is the first program we need to run - this program accepts Spike2 recordings (.smr files) from a singular recording session, and transforms them into HDF5 files that contain all the data contained within each of the channels in the Spike2 recordings.

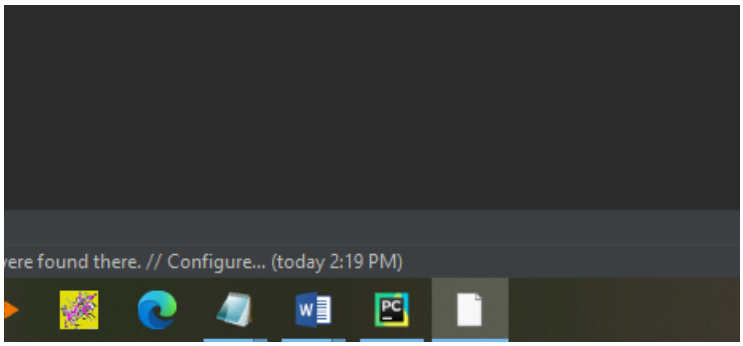


You'll be asked how many files you want to run the program with. In all likelihood, you'll want 2 files, 1 for pre infusion and 1 for post infusion. However, some sessions seem to have been recorded in 3 parts.

The program will then ask you if you want to save this as the default value – this means that whenever you run the program after this, it will assume an input of the value you entered, and you

won't have to enter this again. This makes sense for you to do if you are going to be working with sessions with the same amount of files (usually 2) for a while. The default value can be deleted if you need to change it – delete the file "Default Number of Files.pk" in the folder Default Values.

Next, the GUI for file selection should open up. It may open in your taskbar (in Windows, the bottom taskbar), so you may have to click on it. It should be represented by a plain white icon.



Now, you simply select the files from the session you want to work with. Simply selecting them one by one is what you need to do - do not highlight to select multiple - after you select one, another window will open that will let you select the next file. Do this until you've reached the number of files that you entered before this. The order matters as well - the order you've selected them is the order they will be processed in. **In other words, select the Spike2 recordings in the order they actually occurred.**

After you're done selecting your files, you need to select a folder in a similar GUI that you want to save your processed files to (however as mentioned, in this case you simply choose a folder to save to - not a file). Similar to the "Default number of files'', you can also have a "Default save to location", which can be useful if you plan on saving multiple files to the same folder, so you don't have to select your save location every time. It is also saved in "Default Values", so you can delete the default value if need be.

Following this, you will be asked if you want to enter miscellaneous data such as the rat names, the date of the experiment, etc. If you wish to do so enter "y" if not, then "n", and then answer the questions asked - if you wish to skip any, just press enter. This information will then be appended to the created HDF5 file.

The file created will be named after the spike2 files you used for it – there will just be an "and" between the two names. For e.g., data1.smr and data2.smr would create **"data1 and data2.hdf5".**

Additionally, the file will save the list of Odor events with the name and time. You'll be prompted to save data about the experiment as well, if you want.

A guide to the file can be found in the document "HDF5 Guide".

b) channel_extract.py

Channel Extract is the second program we need to run - this program extracts channels of our choosing from the file we created using the spike2_to_hdf55.py. In order for the user to actually know the names of the channels, a text file named "Channels.txt" will be created to read in the spike2_to_spykingcircus/temporary .text files folder.

Upon running this program, you'll select the file (The one we just created) you want to extract channels from using the GUI. Following this, you'll be asked to select which channels to extract – for example, if you want to extract the channels U1, U2 and U3, just input the following – U1, U2, U3. Similar to some variables in spike2_to_hdf5, you can save this as a default value under "Default Channels" in Default Values, which you can delete later if need be.

The file created will be named after the file you worked and the channels extracted. For e.g., if you just worked on "data1 and data2.hdf5" to extract the channels U1 and U2, the file will be named "**U1, U2 data1 and data2.hdf5".**

A guide to the file can be found in the document "HDF5 Guide".

c) Running Spyking-Circus.

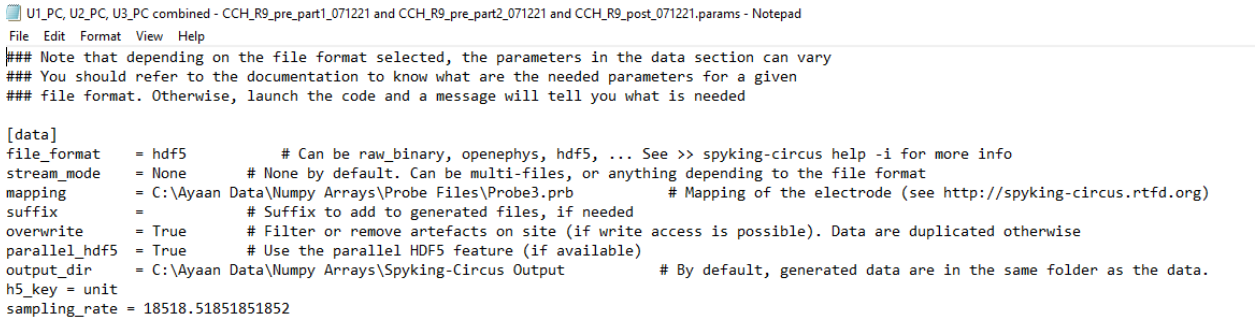
In the anaconda prompt or PyCharm terminal run the following command, where you replace the name path/mydata with the name and location of the saved file from Channel_Extract.

spyking-circus path/*dataname*

This will create a parameter file in the same folder, which you can read more in more detail about in the following link. If you are planning on understanding Spyking Circus and its various parameters, the link is important –

[Configuration File — SpyKING CIRCUS 1.0.1 documentation (spyking-circus.readthedocs.io)]

However, if you just want to run it, you'll need to change only the very first section as follows –



As shown above you'll need to edit some fields-

- The file_format field has to be set to hdf5, since this is the file format Channel Extract creates.
- The mapping will look for probe files with the extension .prb. Three probe files are included with the program in the Probe Files folder. If you're extracting 3 channels set the directory to Probe3.prb, if extracting 4 channels then use Probe4.prb. I'd suggest following up the link included in the params file to see what's going on with these if you're curious.
- Change output_dir if you want to save the output in another folder, which is suggested since Spyking Circus creates many folders and files.
- You have to insert the h5_key and sampling rate for hdf5 files.

- h5_key = unit can be used unedited. This is the name Channel Extract stores the data in inside the file it creates, you don't really need to worry about this.
- You can find the sampling rate in the text file called Channels, as alluded to earlier.

If by any chance, you get an error that ends in -

'configparser.InterpolationSyntaxError: '%' must be followed by '%' or '(', found: '% of max dtype) [0,1]'

You'll need to find the % symbol in the param file, and add another % to it, so that it's %% instead of %.

Now Spyking Circus should truly be able to run. It will save its output in the output_dir, as previously mentioned.

Run **spyking-circus path/*dataname***, (again) and then,

Run this command -

spyking-circus path/*dataname* -m converting -c 1

and then select s for "Some".

The "-m converting" parameters allows us to export our results into the phy (spike sorting library) format, allowing the GUI to work. "-c 1" dictates the amount of cores Spyking Circus will use. I have found that running 1 core does not slow down the running too much, and using a single core is likely safest. You may run "-c N" if you wish to use more cores depending on your Laptop/ Computer.

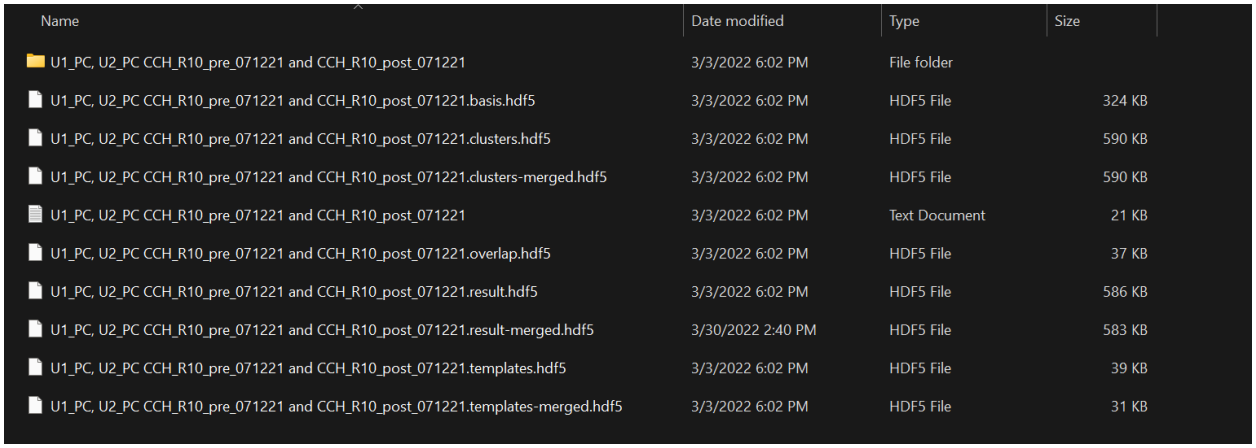
Selecting some is what Thoms Lab suggests, potentially due to size and speed issues.

Then to Launch the gui, run

circus-gui-python path/*dataname*

You may need to install some packages to run the gui, such as phy. This will likely depend on what you already have installed.

For context, pictured below is an example of a Spyking Circus created folder. Data, like the timing of the spikes, is located mainly in the file appended by result-merged, albeit in HDF5 format. -



Name	Date modified	Type	Size
U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221	3/3/2022 6:02 PM	File folder	
U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221.basis.hdf5	3/3/2022 6:02 PM	HDF5 File	324 KB
U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221.clusters.hdf5	3/3/2022 6:02 PM	HDF5 File	590 KB
U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221.clusters-merged.hdf5	3/3/2022 6:02 PM	HDF5 File	590 KB
U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221	3/3/2022 6:02 PM	Text Document	21 KB
U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221.overlap.hdf5	3/3/2022 6:02 PM	HDF5 File	37 KB
U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221.result.hdf5	3/3/2022 6:02 PM	HDF5 File	586 KB
U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221.result-merged.hdf5	3/30/2022 2:40 PM	HDF5 File	583 KB
U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221.templates.hdf5	3/3/2022 6:02 PM	HDF5 File	39 KB
U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221.templates-merged.hdf5	3/3/2022 6:02 PM	HDF5 File	31 KB

Spike Processing

This section of the project focuses on working with spikes you extract from the Spyking-Circus files. These programs are in a different folder – they are in the spike_processing folder instead of the spike2_to_spykingcircus folder.

Running the Program

a) event_spike_merger.py

This program accepts output from Spyking-Circus, extracts the spike data, and combines it with the odor data.

Upon running this program, it will ask you for the result-merged.hdf5 file obtained from running Spyking-circus. This file can be found with the rest of your Spyking-circus output.

The second input required will be where you want to save to – similar to previous save to variables, you can choose if you want to save the directory as the default place to save to for this program. This folder has its own default values folder as well.

The third input will be the sampling rate – use the same input for sampling rate that you did for spyking-circus. You may also save this as a default value.

The final input required will be the amount of time the rat was exposed to the odor.

Now you may have noticed that you never actually have to feed in the Event Data to the program - this is because spike2_to_hdf5 has prepared and stored this data in advance in the folder “Events and Sampling Rate Data”, using the name of the data. This is why the first disclaimer exists, informing you not to change any names you are not prompted to - finding the event data relies on the name of the file you’ve chosen from Spyking Circus!

A guide to the file can be found in the document “HDF5 Guide”.

b) activity_extractor.py

This Program accepts the output from event_spike_merger.py (Any Number of inputs can be taken) and outputs it to Excel.

Upon running the program, you will be asked for input (through a file browning gui)- select how many ever output files from event_spike_merger.py, and when you’re done simply hit the exit red cross.

At this point you will be presented with 2 options -

1) Spontaneous Activity Before any Odors
2) Odor Responses

Depending on what you choose, the options will differ - let’s go through option 1, which can be selected by simply entering 1.

To start with you’ll be asked for the start point of the data - where you want to start from pre infusion and post infusion - for example if you pick 5, you start from 5 seconds after the beginning of the data pre and post infusion. Next, you will be asked for the amount of data you want in seconds.

Now you can select which folder you want to save to, and then enter the name of the file (extension not included). Of course like earlier, you can use the default save to location, and this one will be saved to Default Save to E.pk. You can then navigate to and open your excel file, to see something like this -

A17 fx

	A	B	C	D	E	F	G
1	File Name	Start Time	Duration	Neuron ID	Pre Infusi	Post Infusion Spikes	
2	U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221.result-merged.hdf5	0	5	neuron_1	7	32	
3	U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221.result-merged.hdf5	0	5	neuron_2	8	4	
4	U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221.result-merged.hdf5	0	5	neuron_3	12	11	
5	U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221.result-merged.hdf5	0	5	neuron_4	4	23	
6	U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221.result-merged.hdf5	0	5	neuron_5	6	46	
7	U1_PC, U2_PC Scop_R12_pre_072721 and Scop_R12_post_072721.result-merged.hdf5	0	5	neuron_1	335	3	
8	U1_PC, U2_PC Scop_R12_pre_072721 and Scop_R12_post_072721.result-merged.hdf5	0	5	neuron_2	123	39	
9							
10							
11							

As you can see, the above excel file has the data we just inputted, and outputs the Pre Infusion and Post infusion spikes for our chosen time periods for each Neuron.

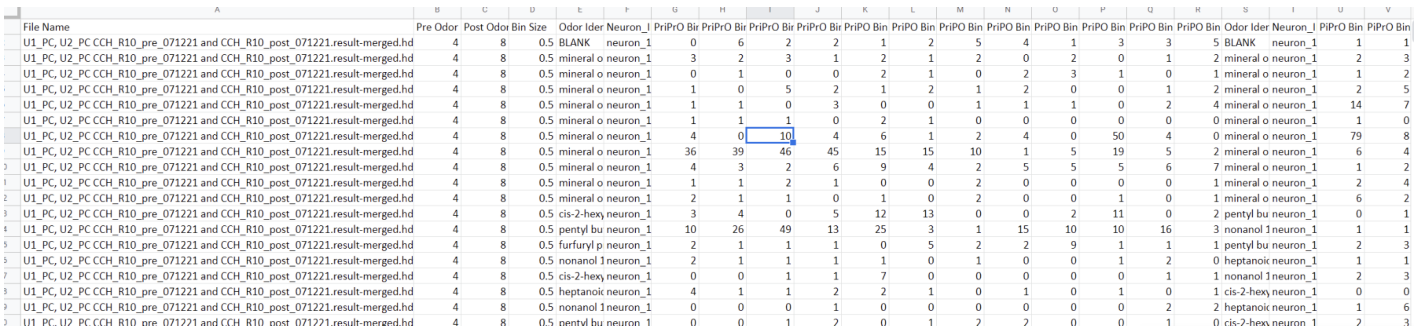
Now, we can move on to the more complicated option 2 - **Odor responses.** Simply enter 2 for this option.

You will first be prompted to enter the amount of time before Odors to analyze - so if you want 2 seconds from before each odor to analyze and contrast for example, you could choose 2 seconds. Next, you’ll be prompted for time after the Odors - basically time from after when the odor was introduced. For this example, I choose 4 seconds - so that’s **2 seconds from before every Odor and 4 seconds from after.**

The next option will be the bin size - so we can control the amount of bins we want pre and post odor. In this example, I choose 0.5 seconds. This means I have 4 bins Pre Odor, and 8 bins Post Odor and the program will inform you as such.

Now you can select which folder you want to save to, and then enter the name of the file (extension not included). Of course like earlier, you can use the default save to location, and this one will be saved to Default Save to E2.pk.

We can now navigate to our file and view the result -



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
	File Name	Pre Odor	Post Odor	Bin Size	Odor Ider	Neuron_I	PriPrO Bir	PriPrO Bir	PriPrO Bir	PriPrO Bir	PriPO Bin	PriPO Bin	PriPO Bin	PriPO Bin	PriPO Bin	PriPO Bin	PriPO Bin	PriPO Bin	Odor Ider	Neuron_I	PiPrO Bin	PiPrO Bin
	U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221.result-merged.hd	4	8	0.5	BLANK	neuron_1	0	6	2	2	1	2	5	4	1	3	3	5	BLANK	neuron_1	1	1
	U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221.result-merged.hd	4	8	0.5	mineral o	neuron_1	3	2	3	1	2	1	2	0	2	0	1	2	mineral o	neuron_1	2	3
	U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221.result-merged.hd	4	8	0.5	mineral o	neuron_1	0	1	0	0	2	1	0	2	3	1	0	1	mineral o	neuron_1	1	2
	U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221.result-merged.hd	4	8	0.5	mineral o	neuron_1	1	0	5	2	1	2	1	2	0	0	1	2	mineral o	neuron_1	2	5
	U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221.result-merged.hd	4	8	0.5	mineral o	neuron_1	1	1	0	3	0	0	1	1	1	0	2	4	mineral o	neuron_1	14	7
	U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221.result-merged.hd	4	8	0.5	mineral o	neuron_1	1	1	1	0	2	1	0	0	0	0	0	0	mineral o	neuron_1	1	0
	U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221.result-merged.hd	4	8	0.5	mineral o	neuron_1	4	0	10	4	6	1	2	4	0	50	4	0	mineral o	neuron_1	79	8
	U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221.result-merged.hd	4	8	0.5	mineral o	neuron_1	36	39	46	45	15	15	10	1	5	19	5	2	mineral o	neuron_1	6	4
	U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221.result-merged.hd	4	8	0.5	mineral o	neuron_1	4	3	2	6	9	4	2	5	5	5	6	7	mineral o	neuron_1	1	2
	U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221.result-merged.hd	4	8	0.5	mineral o	neuron_1	1	1	2	1	0	0	2	0	0	0	0	1	mineral o	neuron_1	2	4
	U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221.result-merged.hd	4	8	0.5	mineral o	neuron_1	2	1	1	0	1	0	2	0	0	1	0	1	mineral o	neuron_1	6	2
	U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221.result-merged.hd	4	8	0.5	cis-2-hex	neuron_1	3	4	0	5	12	13	0	0	2	11	0	2	pentyl bu	neuron_1	0	1
	U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221.result-merged.hd	4	8	0.5	pentyl bu	neuron_1	10	26	49	13	25	3	1	15	10	10	16	3	nonanol 1	neuron_1	1	1
	U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221.result-merged.hd	4	8	0.5	furfuryl p	neuron_1	2	1	1	1	0	5	2	2	9	1	1	1	pentyl bu	neuron_1	2	3
	U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221.result-merged.hd	4	8	0.5	nonanol 1	neuron_1	2	1	1	1	1	0	1	0	0	1	2	0	heptanoic	neuron_1	1	1
	U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221.result-merged.hd	4	8	0.5	cis-2-hex	neuron_1	0	0	1	1	7	0	0	0	0	0	1	1	nonanol 1	neuron_1	2	3
	U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221.result-merged.hd	4	8	0.5	heptanoic	neuron_1	4	1	1	2	2	1	0	1	0	1	0	1	cis-2-hex	neuron_1	0	0
	U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221.result-merged.hd	4	8	0.5	nonanol 1	neuron_1	0	0	0	1	0	0	0	0	0	0	2	2	heptanoic	neuron_1	1	6
	U1_PC, U2_PC CCH_R10_pre_071221 and CCH_R10_post_071221.result-merged.hd	4	8	0.5	pentyl bu	neuron_1	0	0	1	2	0	1	2	2	0	0	1	0	cis-2-hex	neuron_1	2	3

We can see some new columns - most notably the ones such as “PrIPrO” or “PrIPO”. Here’s what these mean - Pr stands for Pre, P stands for Post, I stands for infusion, and O stands for Odor. PrIPO for example would be Pre Infusion, Post Odor. Since each of these would have several bins, short forms were used to make the names less cumbersome. In this example, each Bin represents a period of 0.5 seconds, and we can see the number of spikes in each bin.