**Preparing Spike2 files to be used in Spyking-Circus**

This project, named Spyke2Spyking aims to convert files from the .smr format(Spike2) files to files suitable for use in Spyking-Circus. This program was written in a Conda environment using PyCharm, so I’d suggest getting those.

If you’re familiar with Python, Conda and have an IDE, feel free to skip this next section.

**Installing Conda and PyCharm**

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

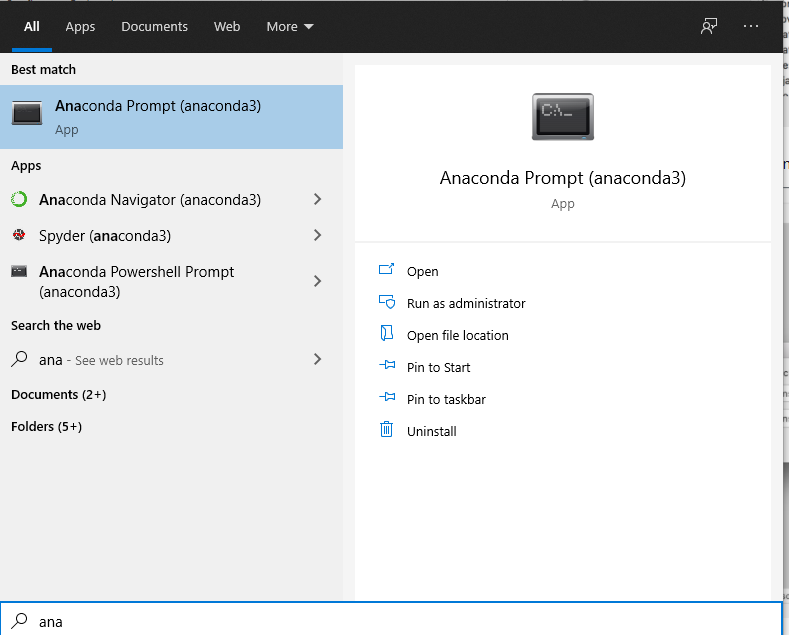
[Downloading conda — conda 4.10.3.post40+c1579681 documentation](https://docs.conda.io/projects/conda/en/latest/user-guide/install/download.html)

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)](https://www.jetbrains.com/help/pycharm/conda-support-creating-conda-virtual-environment.html?keymap=primary_windows#e417d6ec)

**Required Packages**

There are some packages are required to install this program. Try running the following lines of code in 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. You can also use the terminal provided in PyCharm.



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**

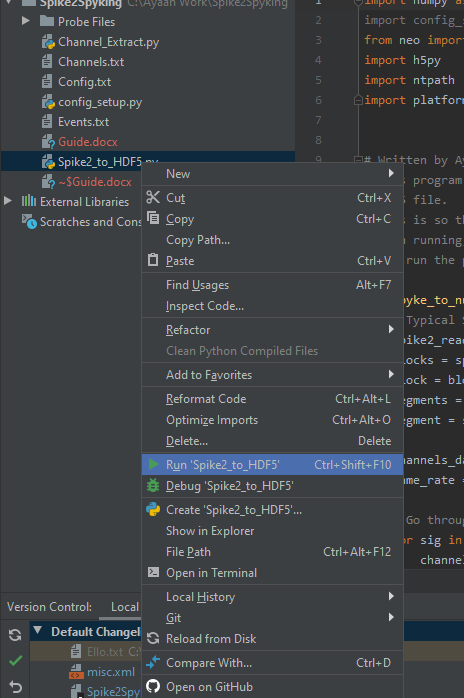
**conda install –c conda-forge –c spyking-circus spyking-circus**

**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**

1. **Spike2\_to\_HDF5.py**

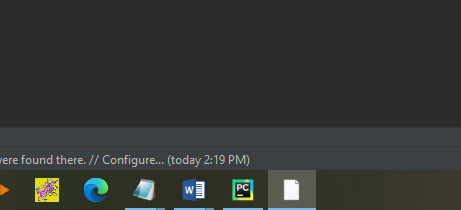
The first step is to run Spike2\_to\_HDF5.py.

****

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 whatever value you entered. This makes sense for you to do if you are going to be working with sessions with the same amount of files 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 task bar, 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. After you’re done selecting your files, you need to a folder you want to save your processed files to. 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 all 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.

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”.

1. **Channel\_Extract.py**

Upon running this program, you’ll select the file 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”.

1. **Running Spyking-Circus.**

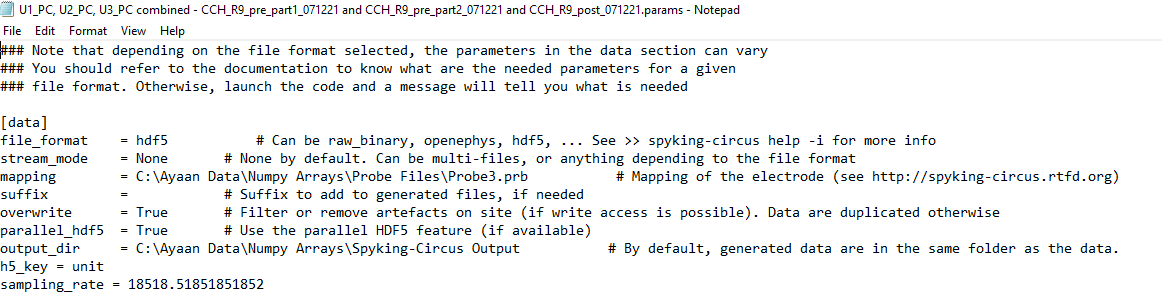
In the anaconda prompt or PyCharm terminal run the following command, where you replace the name path/mydata with the name and location of the saved file from Channel\_Extract.

**spyking-circus path/*dataname***

This will create a parameter file, which you can read more about here –

[Configuration File — SpyKING CIRCUS 1.0.1 documentation (spyking-circus.readthedocs.io)](https://spyking-circus.readthedocs.io/en/latest/code/config.html)

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 can just be set to hdf5.
* The mapping will look for probe files with the extension .prb. Two probe files are included with 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.
* Change output\_dir if you want to save the output in another folder.
* You have to insert the h5\_key and sampling rate for hdf5 files.
* h5\_key = unit can be used unedited.
* You can find the sampling rate in the text file called Channels, as alluded to earlier.

**If by 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 simply run Spyking-Circus the same way again and it will output its data to ouput\_dir.

Following that, run this command -

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

and select s for “Some”

To Launch the gui, run

**circus-gui-python *dataname***

**You may need to install some packages to run the gui.**

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 Spike2Spyking folder.

**Running the Program**

1. **EventSpikeMerger.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.

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

1. **ActivityExtractor.py**

This Program accepts the output from EventSpikeMerger (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 EvenSpikeMerger.py, and when you’re done simply hit the exit red cross.

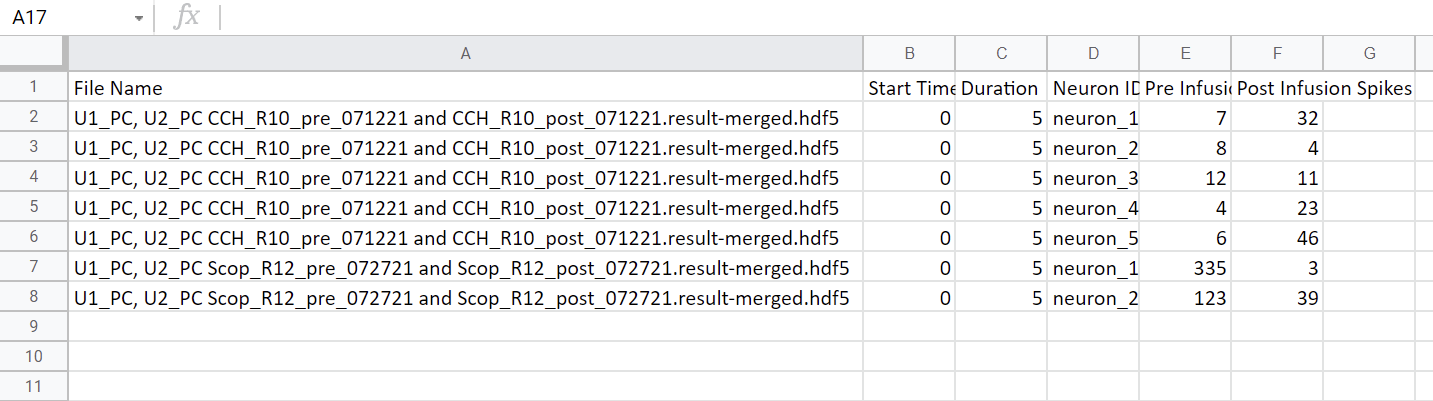
At this point you will be presented with 2 options -

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 -



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.

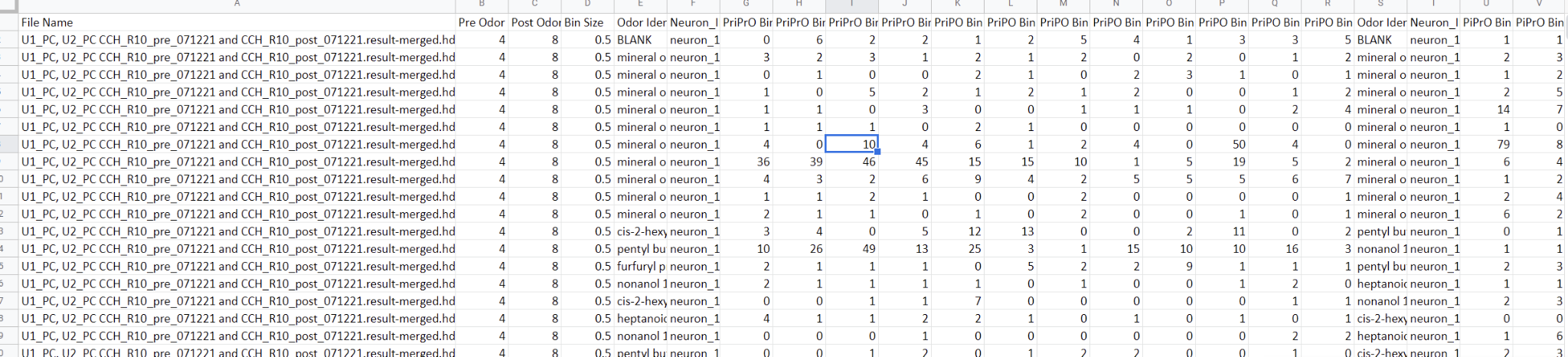
Now, we can move on to the more complicated option 2 - **Odor responses.** Simply enter 2 for this option.

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 -



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.