**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 –

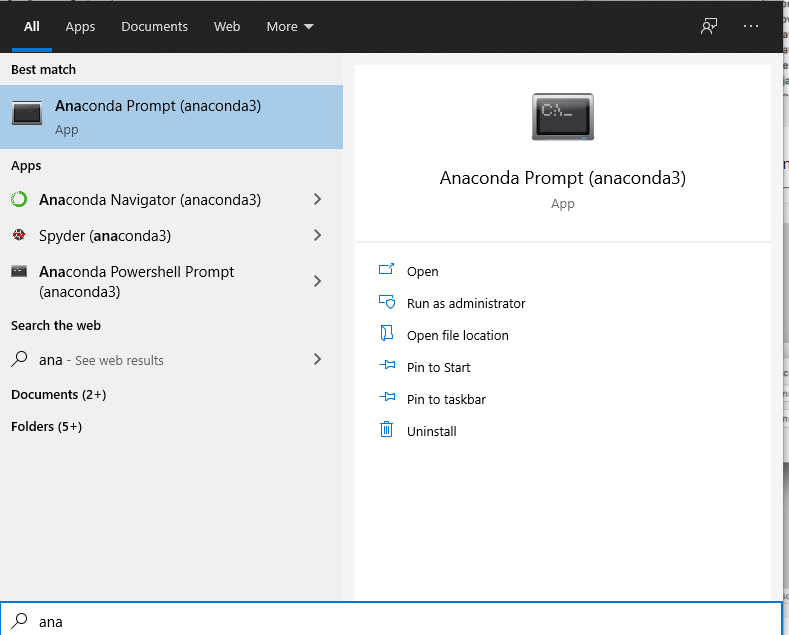
[Downloading conda — conda 4.10.3.post40+c1579681 documentation](https://docs.conda.io/projects/conda/en/latest/user-guide/install/download.html)

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**

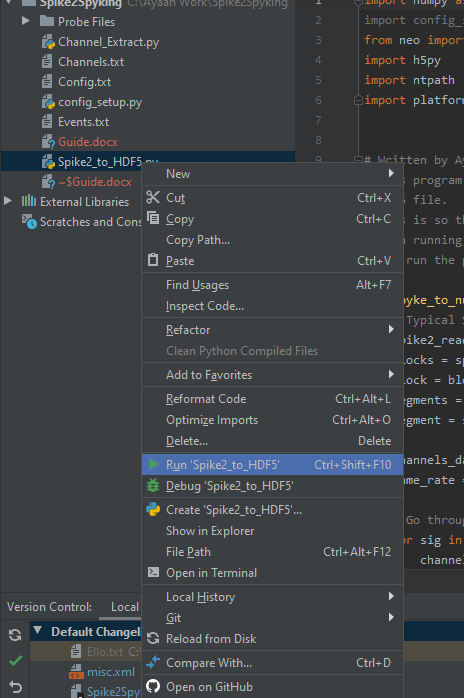
**conda install –c conda-forge python-neo=0.6.1**

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

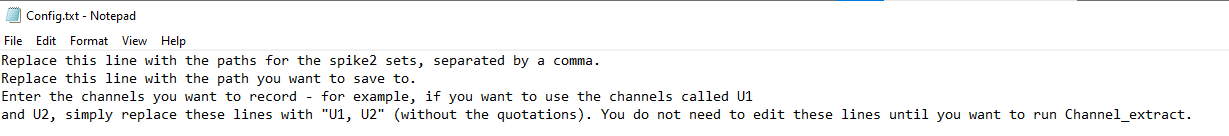
**Running the Program**

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

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

****

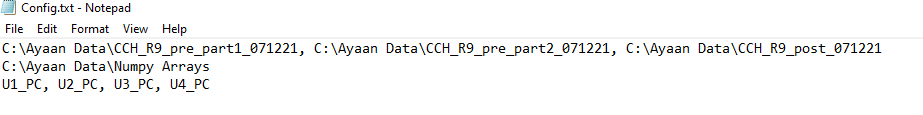
This will create a config file for you to edit -



You want to order the spike2 files in the order of recording – pre infusion first and post infusion later, and place these all on the first line.

The second line should be where you want to save the created file to.

Here’s an example of a config file that’s been updated –



As noted, you don’t need to edit the third line until later.

Now run Spike2\_to\_HDF5.py again and it should create the data set you requested at the location you specified. Note that it also creates a file called Channels.txt that you’ll need later as it contains the sampling rates.

1. **Channel\_Extract.py**

This program can run if you’ve edited the config file to include the channel names you want to extract. If not, just edit it as shown above. If you need the channel names, check Channel.txt. It will create a new, spyking-circus ready file for you in the save to location you specified.

1. **Running Spyking-Circus.**

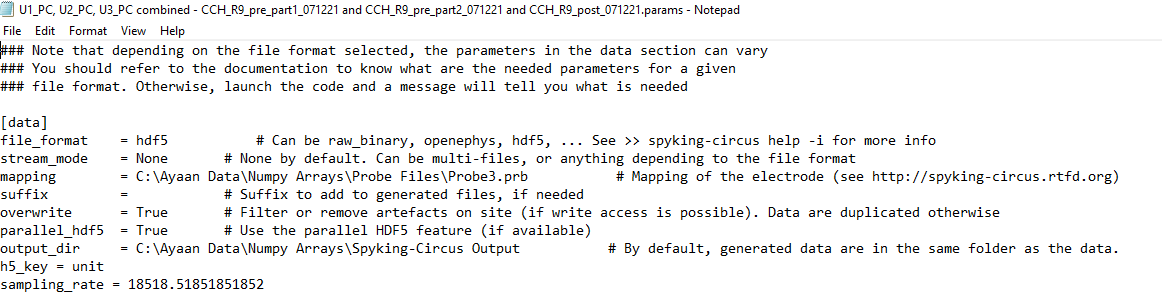
In the anaconda prompt or PyCharm terminal run the following command, where you replace the name path/mydata with the name and location of the saved file from Channel\_Extract.

**spyking-circus path/mydata.hdf5**

Similar to the config file, 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.

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

**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 %.**