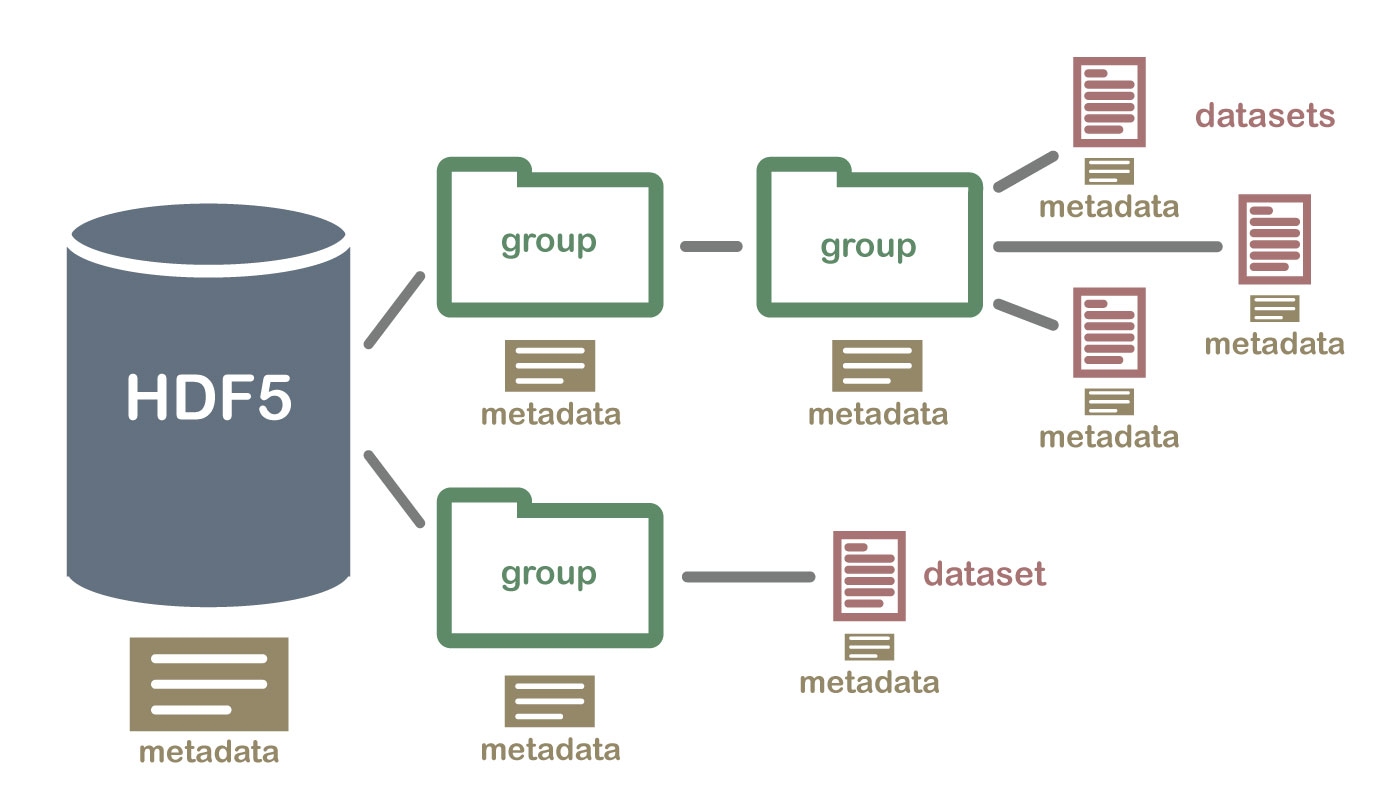
Working with HDF5 Files in Python

HDF5 files do not have a particularly complicated file structure and they support lossless compression. In addition, Spyking-Circus outputs HDF5 files as well.

**The General Structure of an HDF5 file.**

HDF5 files use a “file directory” like structure. The typical HDF5 file would consist of groups and datasets.

* Groups – These are like folders in a file directory – they can contain other folders, or the datasets.
* Datasets – These correspond to the actual data in an HDF5 file.



Picture Credit - [Hierarchical Data Formats - What is HDF5? | NSF NEON | Open Data to Understand our Ecosystems (neonscience.org)](https://www.neonscience.org/resources/learning-hub/tutorials/about-hdf5)

**H5PY**

**I’d suggest using official documentation over this section. If that’s unclear, this may be helpful.**

H5PY is the package python uses to interact with HDF5 files. It’s very similar to Pythons own file I/O.

You begin by installing and importing h5py. Opening an HDF5 file is simple.

with h5py.File(read\_from2, 'r+') as f:

To get a particular dataset is very simple as well – although you need to know what group it would be in.

Consider the example where the dataset **data** is in the group **group**. To access the dataset, you need only do the following –

i = “group/data”

temp = f.get(i)[:]

temp now contains whatever was stored in the dataset named **data** in the group **group** in the file referred to as **read\_from2**.

**The Spyking-Circus Files**

Upon running Spyking-Circus, it will generate several HDF5 files. The HDF5 file we’re interested in will end in result-merged.hdf5. An example of this would be the file named -

**U1, U2, U3 combined - ParameterTest\_OE8\_091119\_Preinfusion and ParameterTest\_OE8\_091119\_Postinfusion.result-merged.hdf5**

**File Structure:**

**I’d suggest reading spyking-circus’s own documentation of this file. It’s under the Fitting heading.**

[Generated Files — SpyKING CIRCUS 1.0.1 documentation (spyking-circus.readthedocs.io)](https://spyking-circus.readthedocs.io/en/latest/advanced/files.html)

But to summarize, we are interested in the **spiketimes** group, which contains datasets under the name **temp\_i**, where the **i** is a number signaling a particular neuron.

For example, in a situation where 3 neurons were detected, there would be 3 datasets under the **spiketimes** group, named **temp\_0, temp\_1, temp\_2**. Each of these datasets is a regular list of times where spikes were detected.

**The File obtained from EventSpikeMerger**

The python file EventSpikeMerger creates an HDF5 file as well. However, in this case due to the relative simplicity, there are no groups, there are only datasets.

The first three datasets are known as **Odor\_Names, Odor\_Onsets and Odor\_Offsets.**

* Odor\_Names is a python list of the order in which the odors were used.
* Odor\_Onsets is a python list of the times at which the odor was released into the box.
* Odor\_Offsets is a python list of times at which the odor was removed from the box.

These 3 Lists will have the same number of elements as each other, and put together, they will show you the name of the odor introduced, the time it was introduced at, and when it was removed.

Additionally, there are more datasets as well, lifted from the previous file. These are the **spiketimes/temp\_i** datasets, but they are renamed to **neuron\_i** in this HDF5 File.

As previously described, each of these are from a neuron detected in spyking circus, so the number of these datasets depends on the number of neurons detected.

Each of these is a list of times at which spikes were detected, so this HDF5 files consists of odor information and spike timings, if I were to summarize.