FERMI Data Analysis

Documentation

Simon Schreck

January 2018

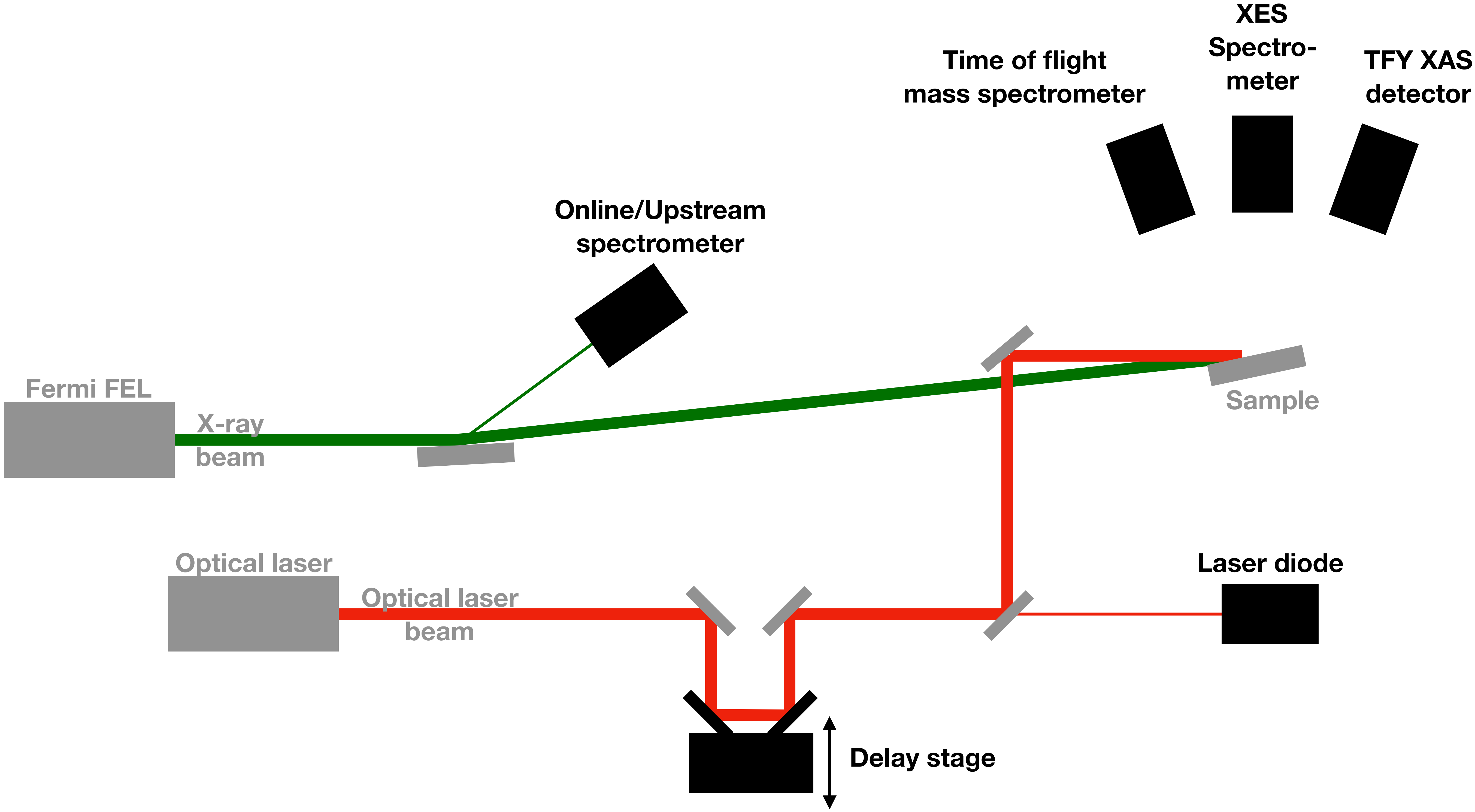
Stockholm University
Department of Physics, Chemical Physics Division

Content

- Experiment Overview
- Data acquisition and data structure
- Data sources
- Data reduction
  - Data collection
  - To come… Data Binning

Experiment Overview

Each black box in the diagram represents a data source that is described in more detail on the following pages.
All data sources are read out shot-to-shot.
<u>Note: Not for all runs all data sources are saved!</u>

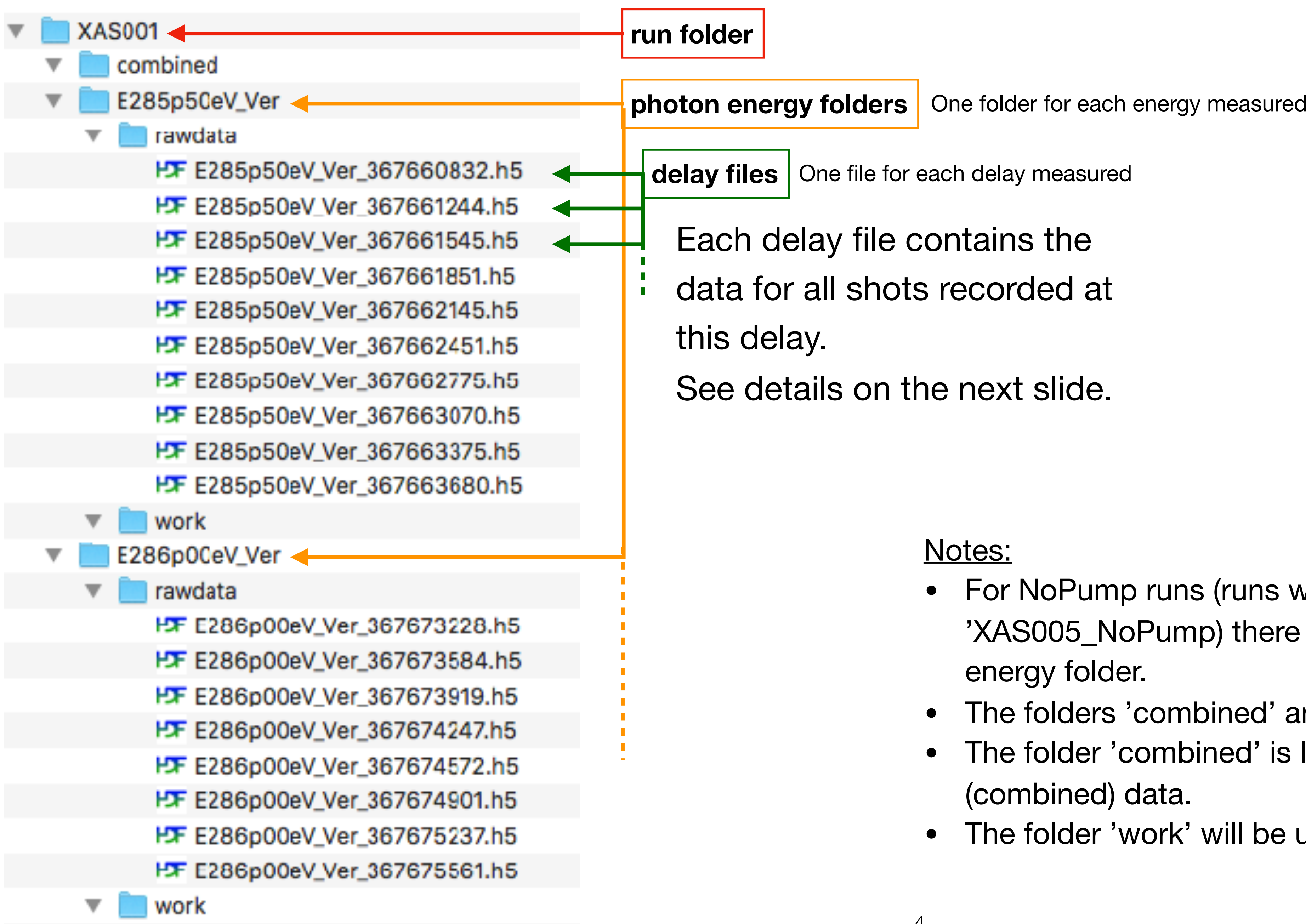


Data Acquisition and Folder Structure

For XAS Runs

For one XAS run (e.g. 'XAS001') we recorded several FEL photon energies and for each FEL photon energy several delays. For each delay we recorded a number of shots - typically a few 100.

This resulted in the following folder structure for the raw data:



Each delay file contains the data for all shots recorded at this delay.

See details on the next slide.

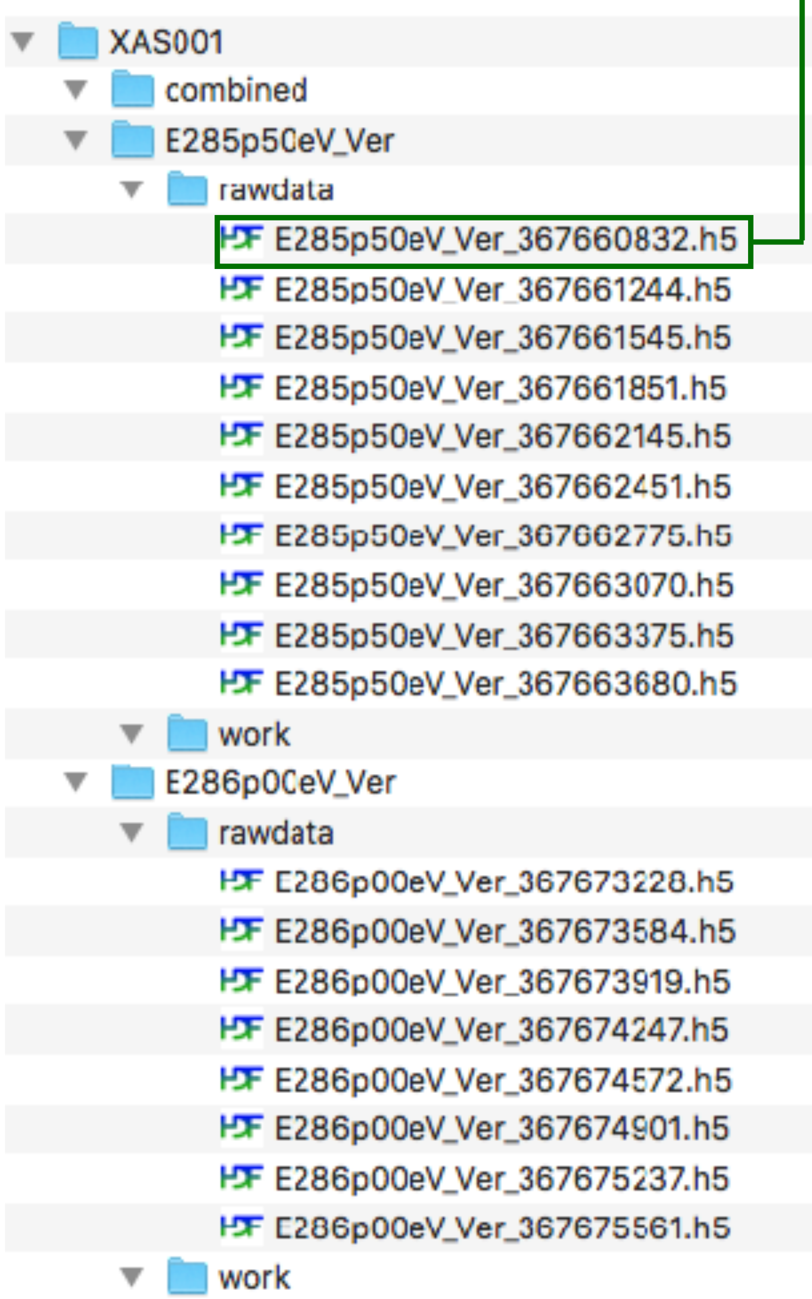
Notes:

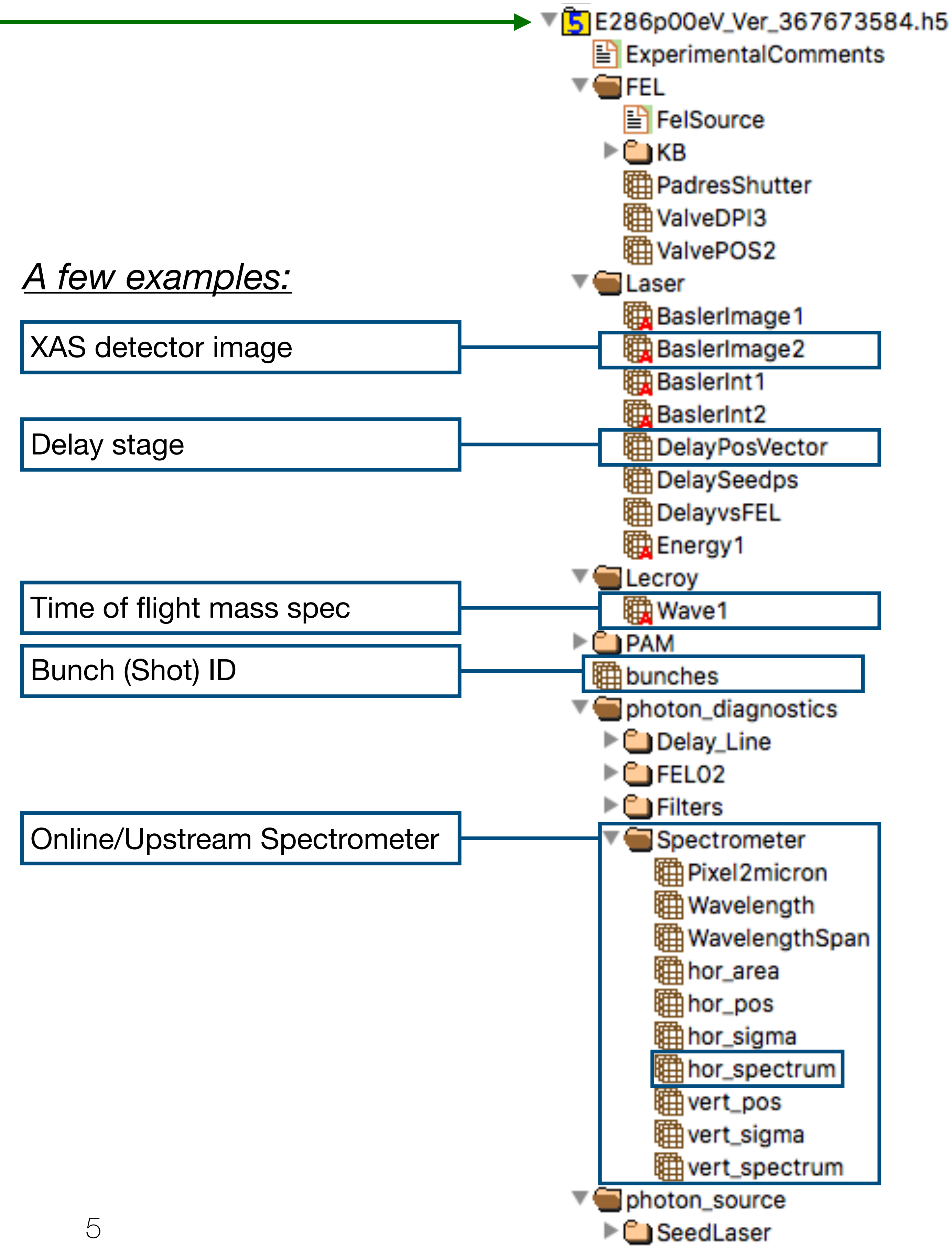
- For NoPump runs (runs without the optical laser, e.g. 'XAS005_NoPump) there is only one delay file per photon energy folder.
- The folders 'combined' and 'work' are empty.
- The folder 'combined' is later used to save the reduced (combined) data.
- The folder 'work' will be unused.

Delay File Structure (HDF5)

Raw data file

The delay files are in HDF5 format. Each delay file contains many variables of different format. Only a few of them are of interested for us. A few examples on the right.





Data Sources

Online/Upstream Spectrometer

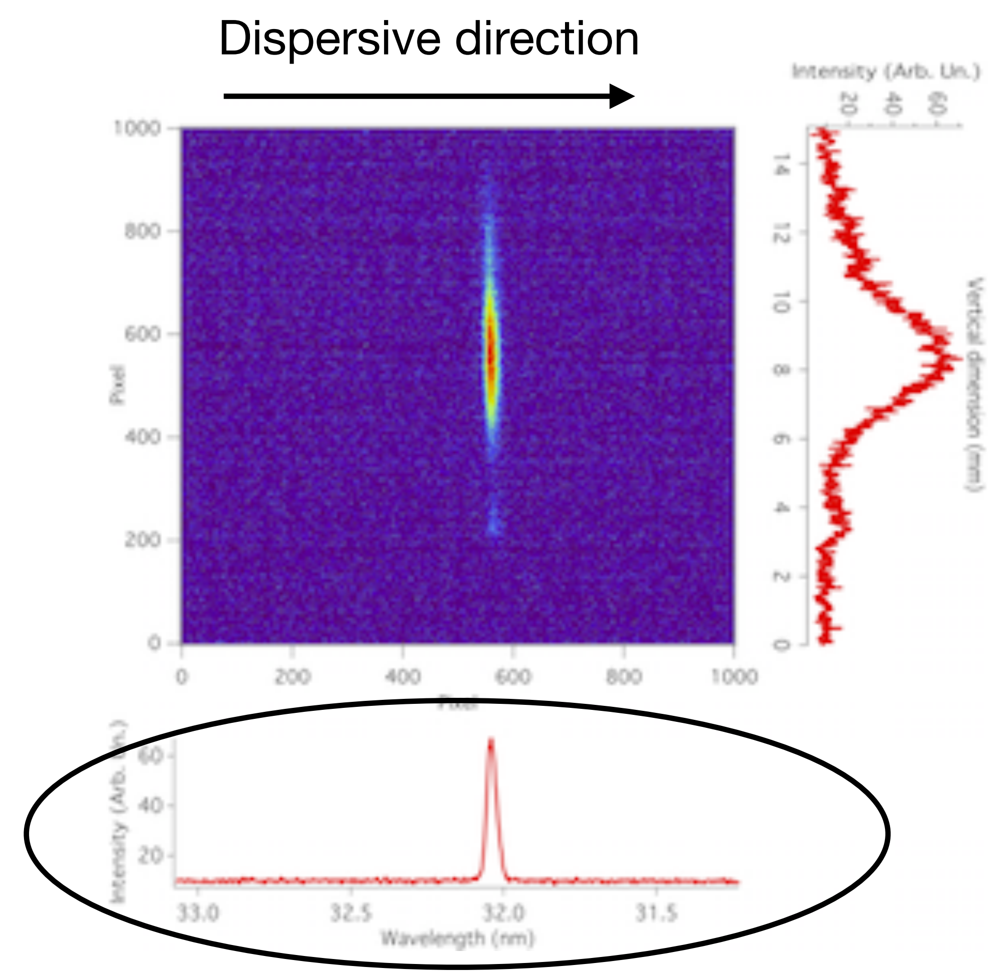
The **online/upstream spectrometer** is a CCD camera that sees a small fraction of the incident x-ray beam, which is diffracted (energy dispersed) onto the camera.

We use only the projection onto the dispersive direction, which is called ‚hor_spectrum‘ and saved under:

/photon_diagnostics/Spectrometer/hor_spectrum

From ‚hor_spectrum‘ we extract:

- Incident intensity (I0) = area under peak
- Incident photon energy = peak position
- Bandwidth of incident photon energy = peak width



hor_spectrum

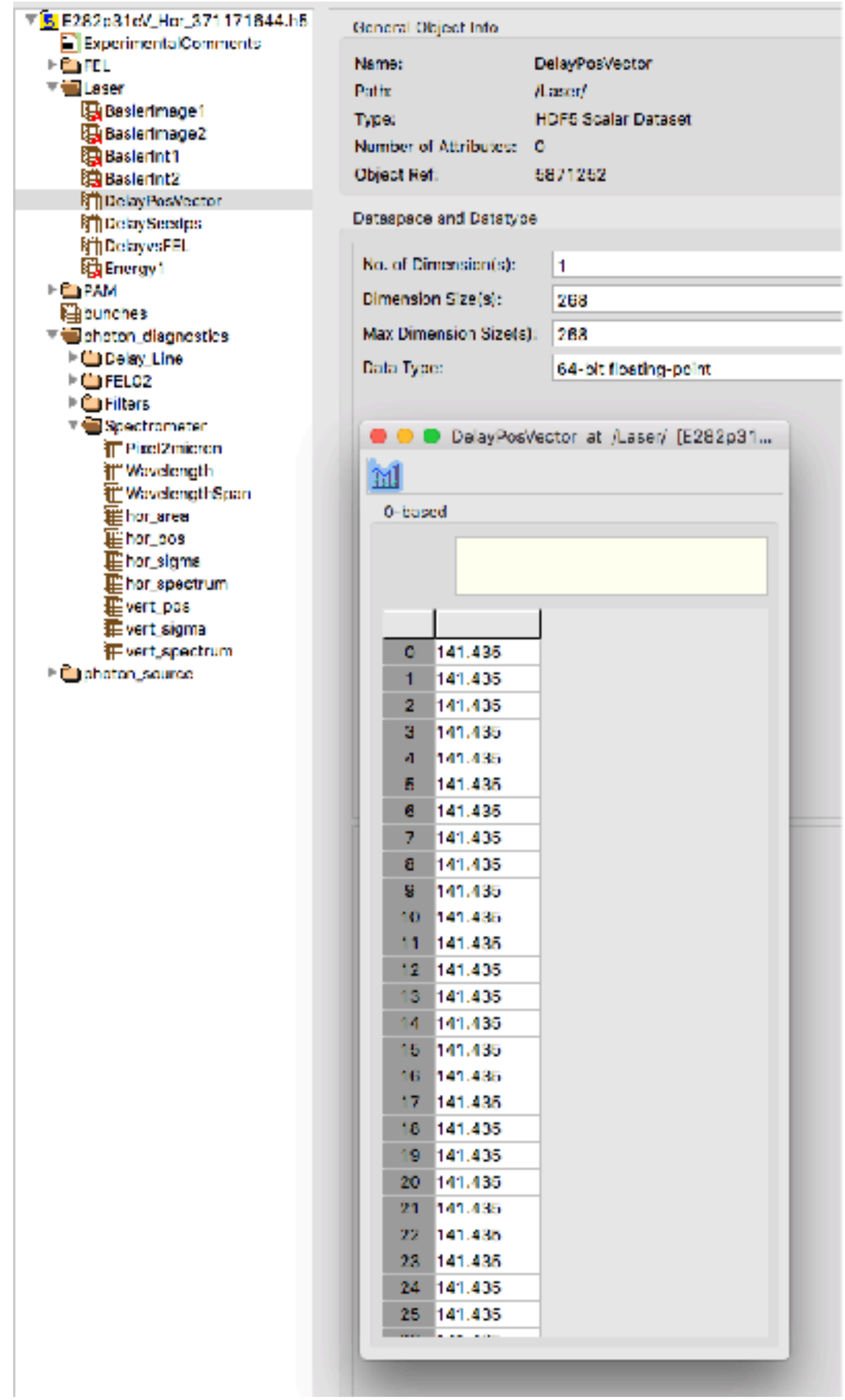
Delay stage

The **delay stage** is a linear drive that moves two mirrors which ultimately changes the path length of the optical laser and with this the delay between X-ray and optical laser pulse.

The delay stage position in mm is saved under:

/Laser/DelayPosVector

With the delay stage position and the time-zero position the delay can be calculated for each shot in femtoseconds.



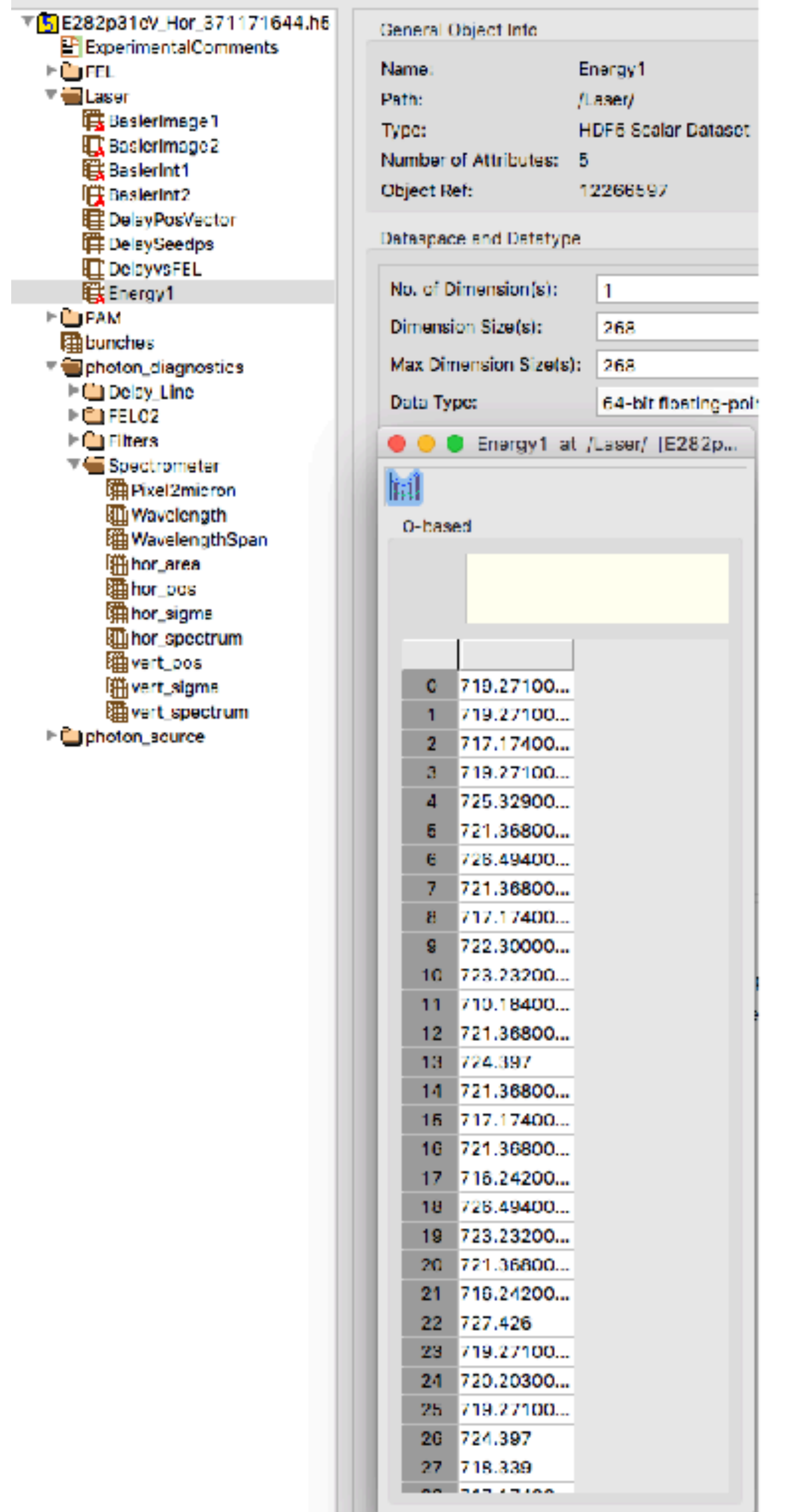
Laser diode

The **laser diode** is a photodiode that sees a fraction of the incident optical laser beam and with this supplies an intensity that is proportional to the laser intensity on the sample.

The laser diode reading in μJ is saved under:

/Laser/Energy1

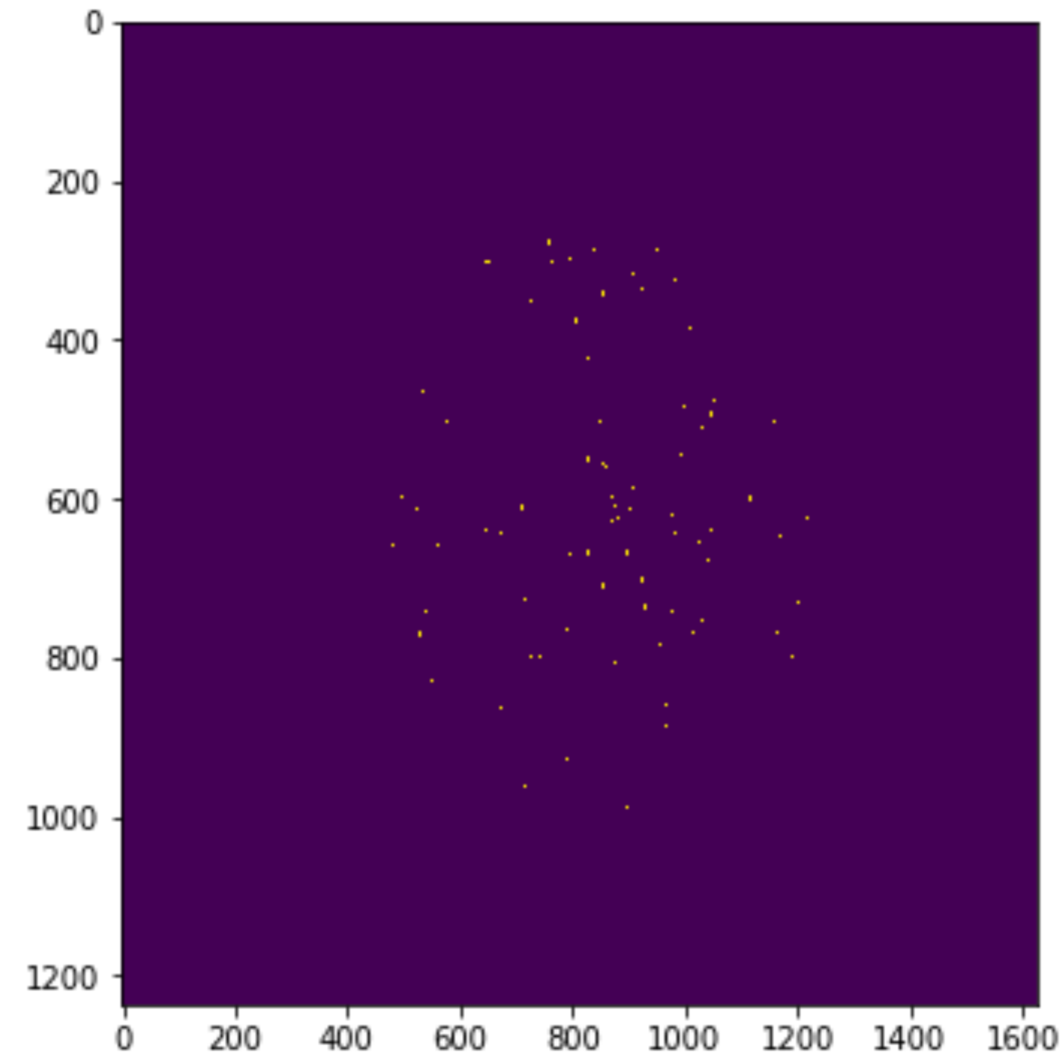
For now we don't use the laser diode in the analysis, but it is saved in the processes data in case we need it later.



TFY XAS detector

The **TFY** (total fluorescence yield) **XAS detector** is a CCD camera that looks at the phosphor screen behind our MCP x-ray detector. Each bright spot, which we called a blob, on the detector corresponds to one photon.

The integrated intensity on the camera (after proper background treatment) is promotional to the total fluoresce coming from the sample and with this prontoional to the absorption.



The camera image is saved under:

/Laser/BaslerImage2

For some runs also the integrated intensity is saved directly under:

/Laser/BaslerInt2

XES Spectrometer

The **XES Spectrometer** is a CCD camera in the same way as for the TFY XAS detector. However, since the x-rays are dispersed by a grating before hitting the XES CCD camera, one dimension of the camera is the dispersive direction.

Projection onto the dispersive direction under consideration of the curvature yields the x-ray emission spectrum.

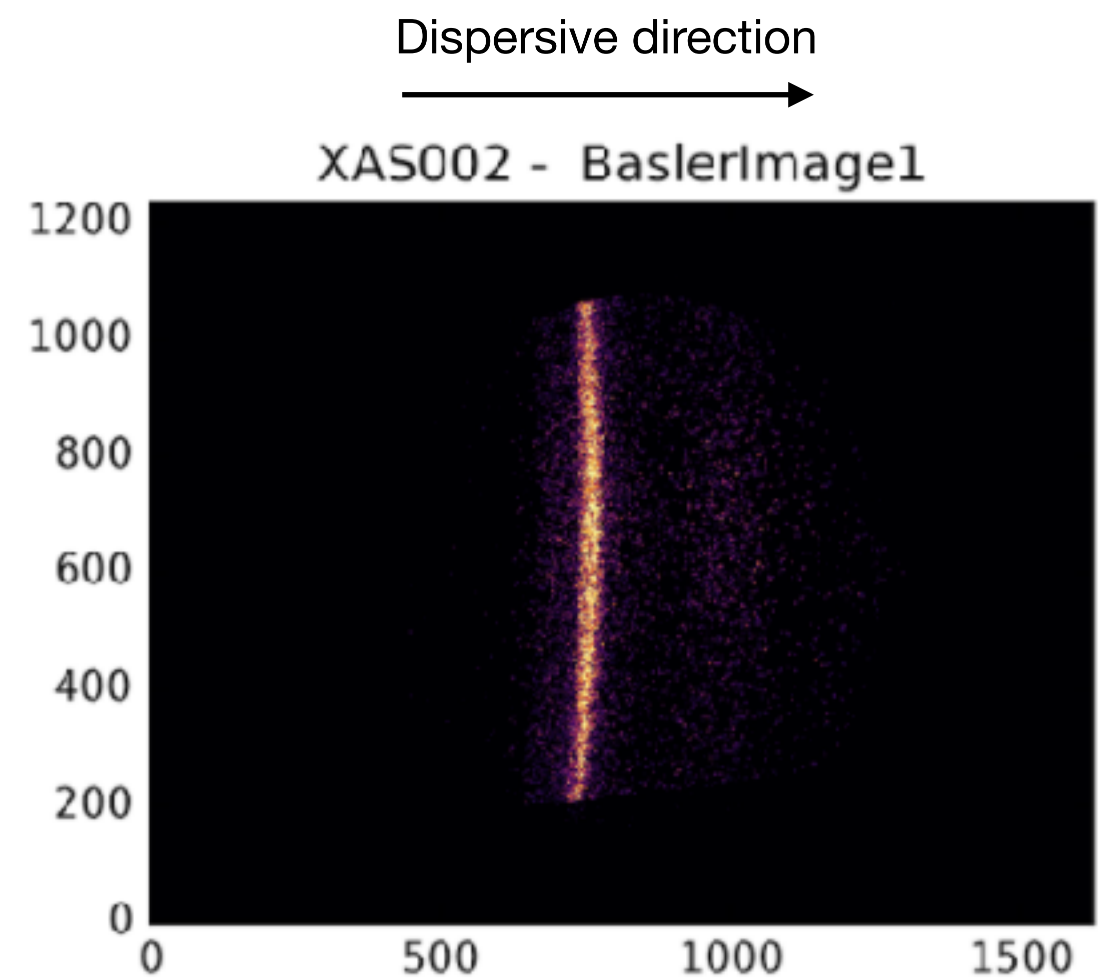
The camera image is saved under:

/Laser/BaslerImage1

For some runs also the integrated intensity is saved directly under:

/Laser/BaslerInt1

XES analysis is not yet included in the scripts! (As of Jan 2018)



The above image is the sum of many 100 shots. On a single shot there will be only a 1 to a few blobs!

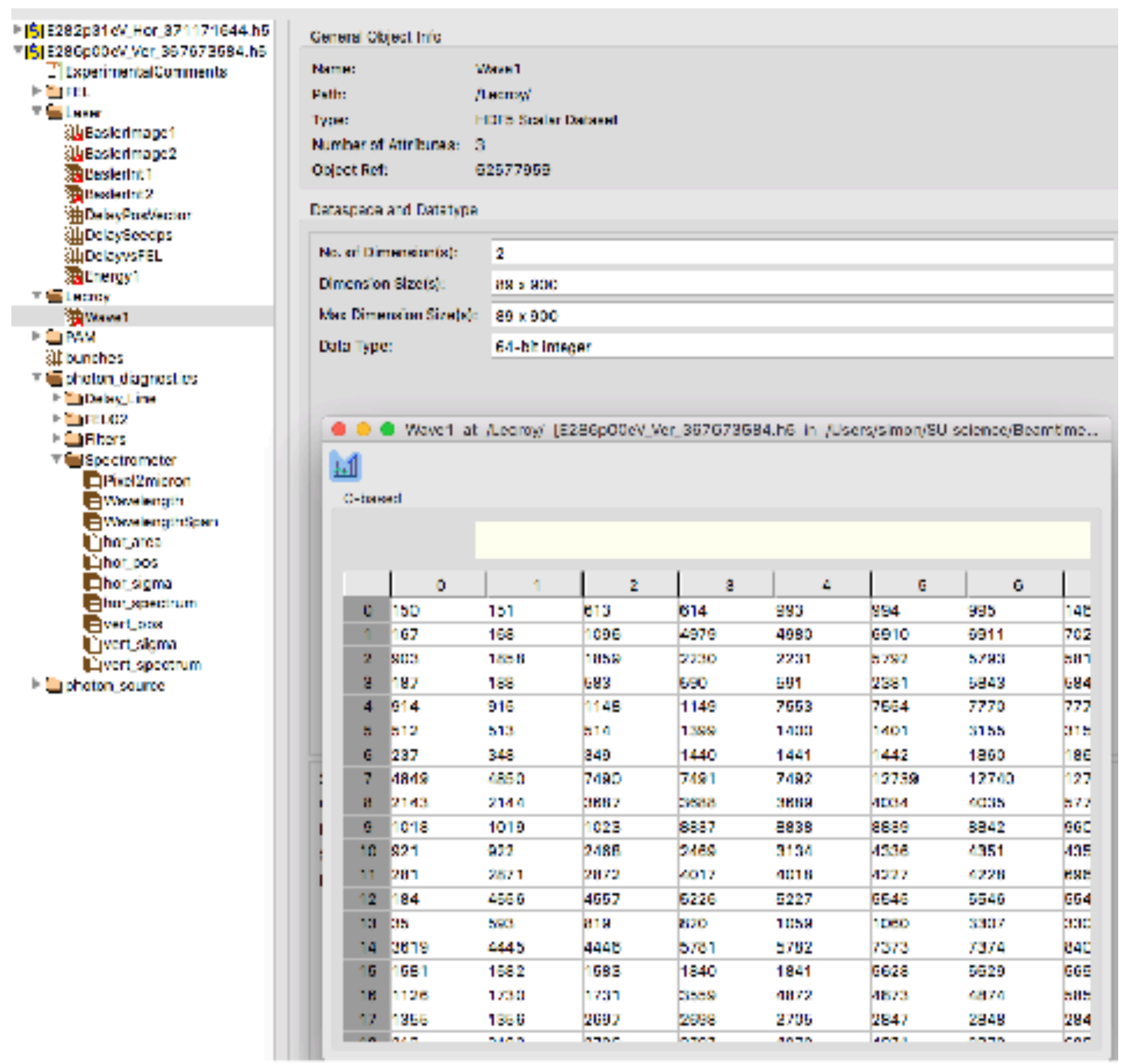
Time of Flight Mass Spectrometer

The **time of flight mass spectrometer** records molecules of a specific mass that desorbed from the surface as a function of time after the optical laser hit the sample.

In the data source the arrival time of each count is saved in units of 5e-9 sec.

The arrival times for each shot are saved under:

/Lecroy/Wave1



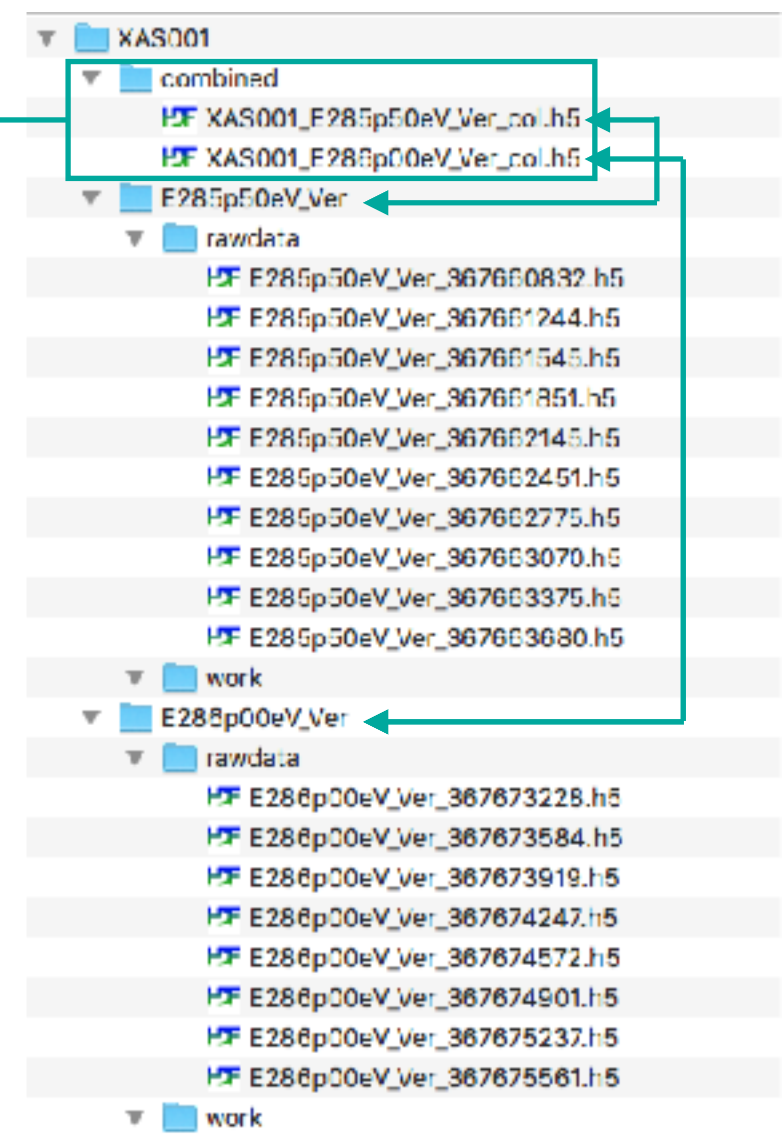
Data Reduction Structure

The reduction of the raw data is done in two steps:

Step 1: Data Collection (script: *16_data_collection.ipynb)*

- In this step all relevant data from the raw HDF5 files (delay files) are collected and saved in a new HDF5 file.
- For the more complex data sources (e.g. XAS detector or online/upstream spectrometer) the data are reduced to more compact formats. E.g. for the XAS detector the image is reduced to a single number representing the integrated intensity in the image.
- The data from all delay files within one photon energy folder are combined in a single HDF5 file with the name *run_photonenergy_polarization*_col.h5 e.g.: *XAS001_E285p50eV_Ver_col.h5*
  This combined file is saved in the ‚combined' folder of the run folder.

- In the combined files the data from each data source is saved for each shot.



Step 2: Data Binning

- In this step all combined files for a number of selected runs (or a single run) are loaded.
- Filters are applied on a shot-to-shot basis on all shots in the selected runs.
  E.g. Filtering out all shots with an incident intensity below a certain value.
- All *good shots* (the ones not filtered out) are then binned into a 2d-raster with dimensions photon energy X delay, which provides delay dependent X-ray absorption spectra.
- The binned data are saved in new HDF5 file called XAS_*RunNo1_RunNoLast_bin*.h5 (e.g.: *XAS_001_003_bin.h5)* inside the folder 'Binned' in the main data path (*data_path*).

Data Collection

(script: *16_data_collection.ipynb)*

The data collection script consists of three cells:

1. **Initialization**

   Initialize python by loading packages and defining some settings.

   This cell needs to be run only once after starting the juyter notebook.

2. **Define Parameters**

   Define which data to collect and parameters for e.g. the XAS detector camera.

   This cell needs to be run whenever you have changed a parameter defined here.

3. **Do the Data Collection**

   This cell performs the main data collection.

   In the top part of the cell you can define which run(s) (run, run type, beamtime) to collect the data from.

   Also you have to set correct main data path.

3. Do the Data Collection

(Cell 3 in script: *16_data_collection.ipynb)*

The main data collection consists of a three nested loops with the following structure:

1. **Loop over selected runs**

   - Collect data which are global for the entire run (harmonic, polarization)

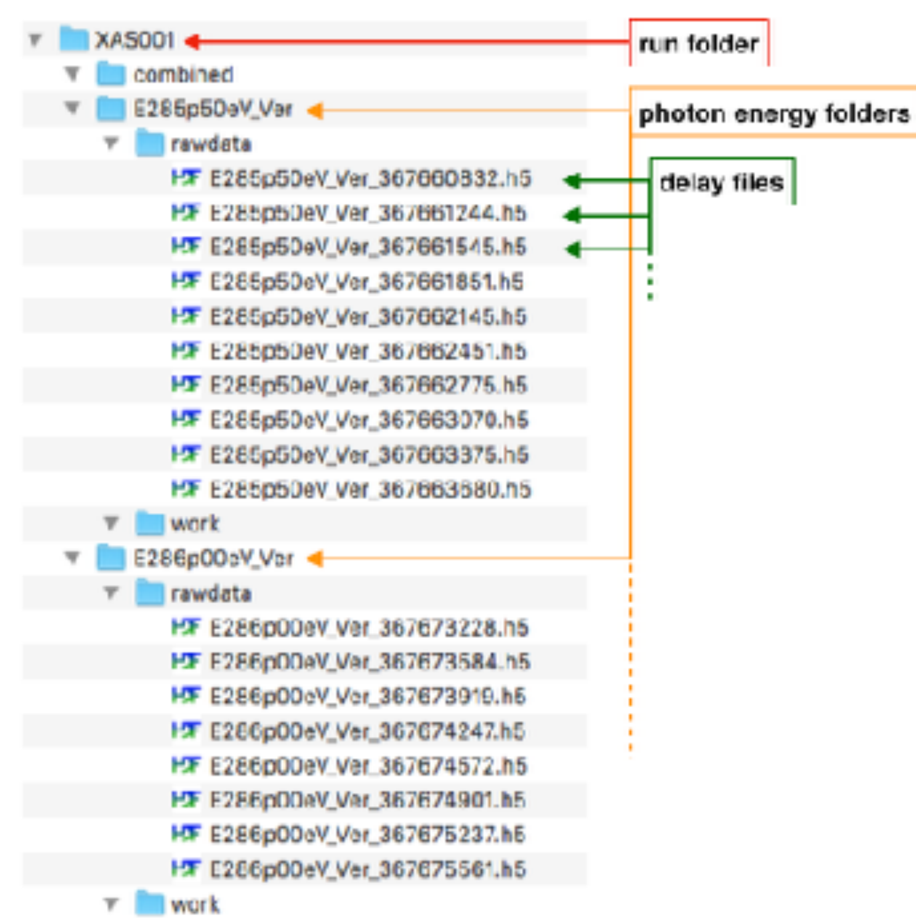
   1.1. **Loop over photon energy folders for this run**

   - Check that all data that should be collected do exist (checks this for the first delay file in the first photon energy folder)

   1.1.1. **Loop over delay files for this photon energy folder**

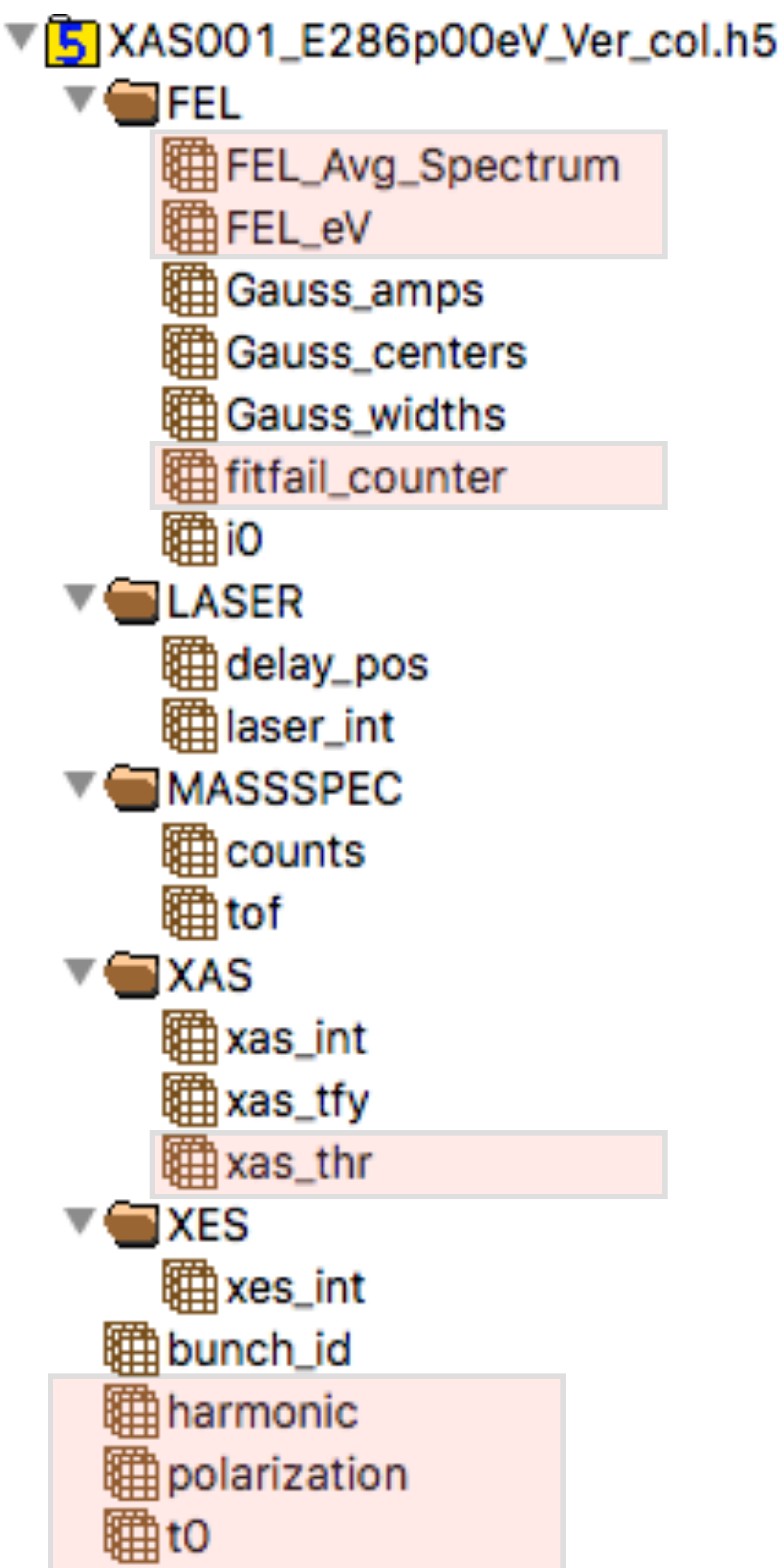
   - Collect the data for all delay files

   - Save the collected data for this photon energy folder in the combined file.



Collected Date in the Combined File

After running the Data Collection

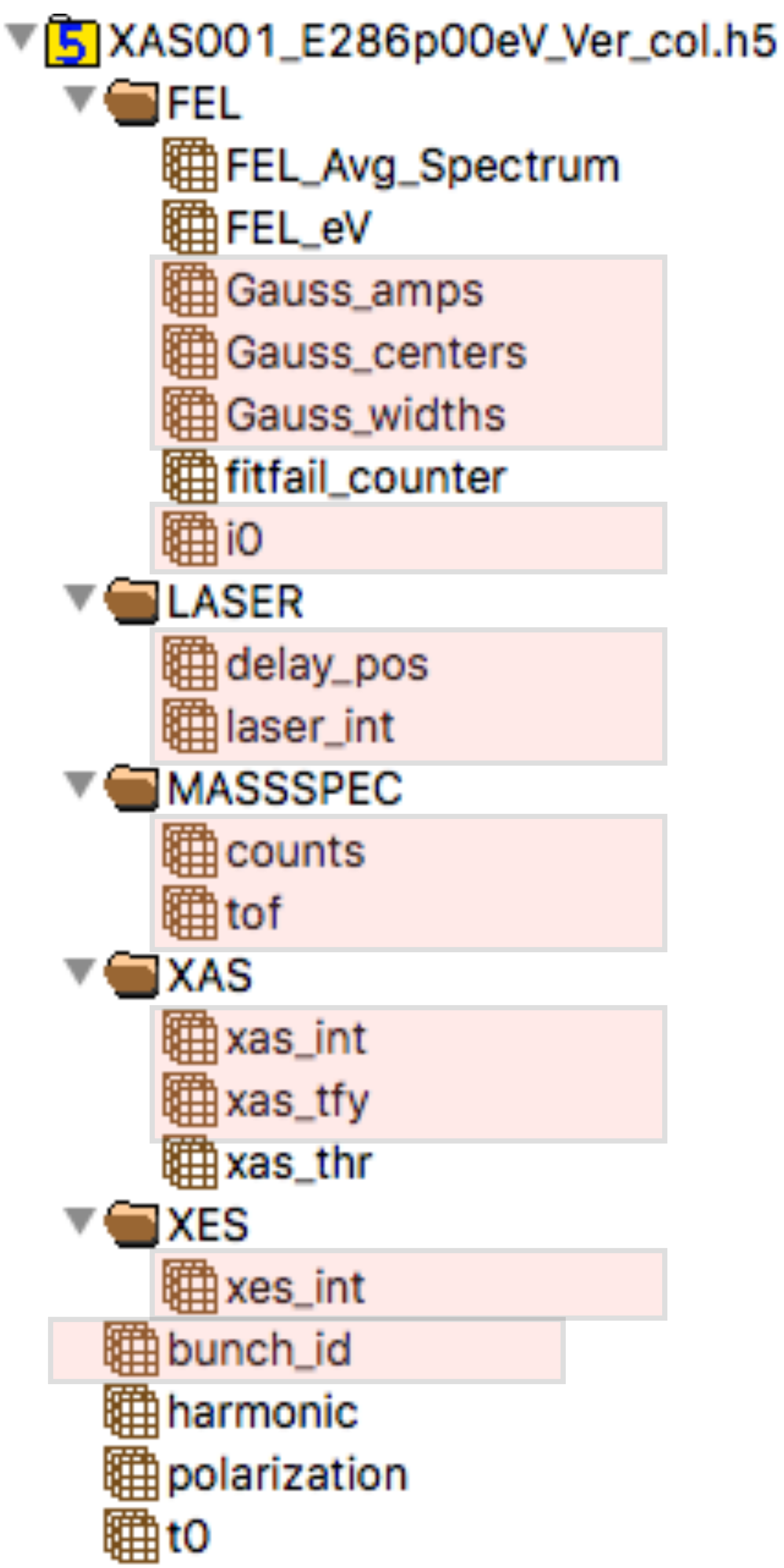


Global data (not shot-to-shot)

- **FEL_Avg_Spectrum** and **FEL_eV**
  each 1d array with length 1000
  Average spectrum of all shots form the online/upstream spectrometer (FEL_Avg_Spectrum is the intensity and FEL_eV is the photon energy)
- **fitfail_counter**
  integer number
  Number of shots for which the gaussian fit of the FEL spectrum failed. Typically 0…10
- **xas_thr**
  floating point number
  Intensity threshold for the XAS meteor images. This number is set in the data collection parameters.
- **harmonic**
  integer number
  FEL harmonic
- **Polarisation**
  integer number
  FEL polarization: 1 = linear vertical, 0 = linear horizontal
- **t0**
  floating point number
  Zime zero position on the delay stage in mm as it was set during data acquisition.
  NOTE: THIS IS NOT THE FINAL CALIBRATED TIME ZERO!

Collected Date in the Combined File

After running the Data Collection



Shot-to-shot data

- **Gauss_amps**, **Gauss_centers** and **Gauss_width**
  each 1d array with length of number of shots
  Amplitude, central photon energy and photon energy width of the incident x-ray pulse retrieved from a gauss fit to the online spectrometer.
- **i0**
  1d array with length of number of shots
  Incident x-ray intensity calculated as the intensity in the online spectrometer over a region of interest.
- **delay_position** and **laser_int**
  each 1d array with length of number of shots
  Delay stage position in mm and laser diode reading in μJ
- **MASSSPEC/counts**
  1d array with length of number of shots
  Number of mass spec counts for each shot
- **MASSSPEC/tof**
  1d array with length = sum of all mass spec counts
  Time of flight of each mass spec count. If the first shot had e.g. three mass spec counts, the first three values in tof give the time of flight of these three counts.
- **xas_int** and **xas_tfy**
  each 1d array with length of number of shots
  Both give the integrated intensity on the XAS detector (difference is the thresholding). - **USE xas_tfy FOR NOW!**
- **xes_int**
  Integrated intensity on the XES detector
- **bunch_id**
  each 1d array with length of number of shots
  Unique number identifying each shot. Monotonically increasing from shot to shot.

Data Binning

(script: *26_data_binner.ipynb)*

The data binning script filters the collected data on a shot-to-shot basis and bins all good shots into incident photon energy and delay. It consists also of three cells:

1. Initialization

Initialize python by loading packages and defining some settings.
This cell needs to be run only once after starting the juyter notebook.

2. Define Parameters

Define which data to use as XAS intensity.
Define based on which data to filter the data.
Define binning parameters (bin edges and/or centers)
Define time zero

3. Do the Data Binnning

This cell performs the main data binning.
In the top part of the cell you define which run(s) (run, run type, beamtime) to perform the binning on.
Also you have to set correct main data path, which needs to be consistent with the path in the data collector.

A VERY simplified example of how the Data Binner works

(script: *26_data_binner.ipynb*)

Say we have a Run with the following 20 shots:

(Values are arbitrary and not in a typical order of magnitude)

Shot No	hv_in (gauss_center)	bandwidth (gauss_width)	delay	I_0	XAS intensity
1	285.0	0.5	-0.5	75	8
2	287.3	0.2	1.4	62	5
3	285.3	0.4	0.3	85	9
4	286.7	1.1	2.3	10	2
5	287.0	0.7	-0.9	65	5
6	286.2	0.65	0.45	74	6
7	286.0	1.5	1.8	6	0
8	288.9	0.1	2.7	73	8
9	287.9	0.3	0.2	82	7
10	288.5	0.35	1.9	63	7
11	289.0	0.45	-0.1	68	7
12	285.1	0.6	2.4	91	10
13	286.0	0.45	1.9	59	6
14	286.1	1.0	2.2	68	6
15	287.5	0.9	2.5	47	4
16	288.0	0.8	1.0	67	7
17	285.0	0.32	0.1	82	9
18	286.6	0.85	1.7	12	1
19	288.6	0.72	2.7	41	5
20	288.0	0.4	2.0	96	10

A VERY simplified example of how the Data Binner works

(script: *26_data_binner.ipynb*)

Say we have a Run with the following 20 shots:

(Values are arbitrary and not in a typical order of magnitude)

Shot No	hv_in (gauss_center)	bandwidth (gauss_width)	delay	I_0	XAS intensity
1	285.0	0.5	-0.5	75	8
2	287.3	0.2	1.4	62	5
3	285.3	0.4	0.3	85	9
4	286.7	1.1	2.3	10	2
5	287.0	0.7	-0.9	65	5
6	286.2	0.65	0.45	74	6
7	286.0	0.6	1.8	6	0
8	288.9	0.1	2.7	73	8
9	287.9	0.3	0.2	82	7
10	288.5	0.35	1.9	63	7
11	289.0	0.45	-0.1	68	7
12	285.1	0.6	2.4	91	10
13	286.0	0.45	1.9	59	6
14	286.1	0.5	2.2	18	6
15	287.5	0.9	2.5	47	4
16	288.0	0.8	1.0	67	7
17	285.0	0.32	0.1	82	9
18	286.6	0.85	1.7	12	1
19	288.6	0.72	2.7	41	5
20	288.0	0.4	2.0	96	10

Step 1: Filtering based on…

…bandwidth: min = 0, max = 0.7

…I0: min = 20, max = infinity

A VERY simplified example of how the Data Binner works

(script: *26_data_binner.ipynb*)

Say we have a Run with the following 20 shots:

(Values are arbitrary and not in a typical order of magnitude)

Shot No	hv_in (gauss_center)	bandwidth (gauss_width)	delay	I_0	XAS intensity
1	285.0	0.5	-0.5	75	8
2	287.3	0.2	1.4	62	5
3	285.3	0.4	0.3	85	9
4	286.7	1.1	2.3	10	2
5	287.0	0.7	-0.9	65	5
6	286.2	0.65	0.45	74	6
7	286.0	0.6	1.8	6	0
8	288.9	0.1	2.7	73	8
9	287.9	0.3	0.2	82	7
10	288.5	0.35	1.9	63	7
11	289.0	0.45	-0.1	68	7
12	285.1	0.6	2.4	91	10
13	286.0	0.45	1.9	59	6
14	286.1	0.5	2.2	18	6
15	287.5	0.9	2.5	47	4
16	288.0	0.8	1.0	67	7
17	285.0	0.32	0.1	82	9
18	286.6	0.85	1.7	12	1
19	288.6	0.72	2.7	41	5
20	288.0	0.4	2.0	96	10

Step 1: Filtering based on…

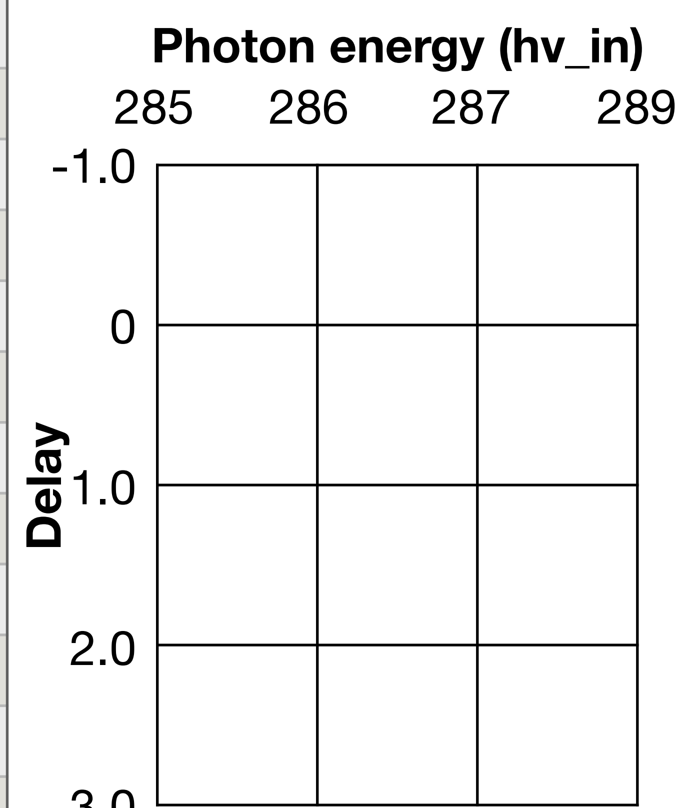
…bandwidth: min = 0, max = 0.7

…I0: min = 20, max = infinity

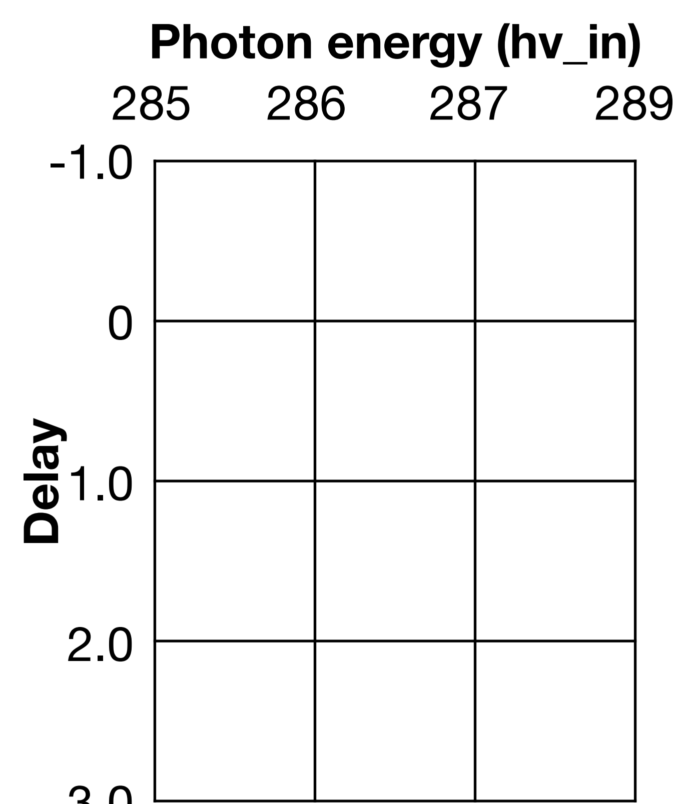
Step 2: Binning into …

…hv_in: Bin width = 1.0 eV

…delay: Bin edges = [-1, 0, 1, 2, 3]



XAS Intensity



I0

A VERY simplified example of how the Data Binner works

(script: *26_data_binner.ipynb)*

Say we have a Run with the following 20 shots:

(Values are arbitrary and not in a typical order of magnitude)

Shot No	hv_in (gauss_center)	bandwidth (gauss_width)	delay	I_0	XAS intensity
1	285.0	0.5	-0.5	75	8
2	287.3	0.2	1.4	62	5
3	285.3	0.4	0.3	85	9
4	286.7	1.1	2.3	10	2
5	287.0	0.7	-0.9	65	5
6	286.2	0.65	0.45	74	6
7	286.0	0.6	1.8	6	0
8	288.9	0.1	2.7	73	8
9	287.9	0.3	0.2	82	7
10	288.5	0.35	1.9	63	7
11	289.0	0.45	-0.1	68	7
12	285.1	0.6	2.4	91	10
13	286.0	0.45	1.9	59	6
14	286.1	0.5	2.2	18	6
15	287.5	0.9	2.5	47	4
16	288.0	0.8	1.0	67	7
17	285.0	0.32	0.1	82	9
18	286.6	0.85	1.7	12	1
19	288.6	0.72	2.7	41	5
20	288.0	0.4	2.0	96	10

Step 1: Filtering based on…

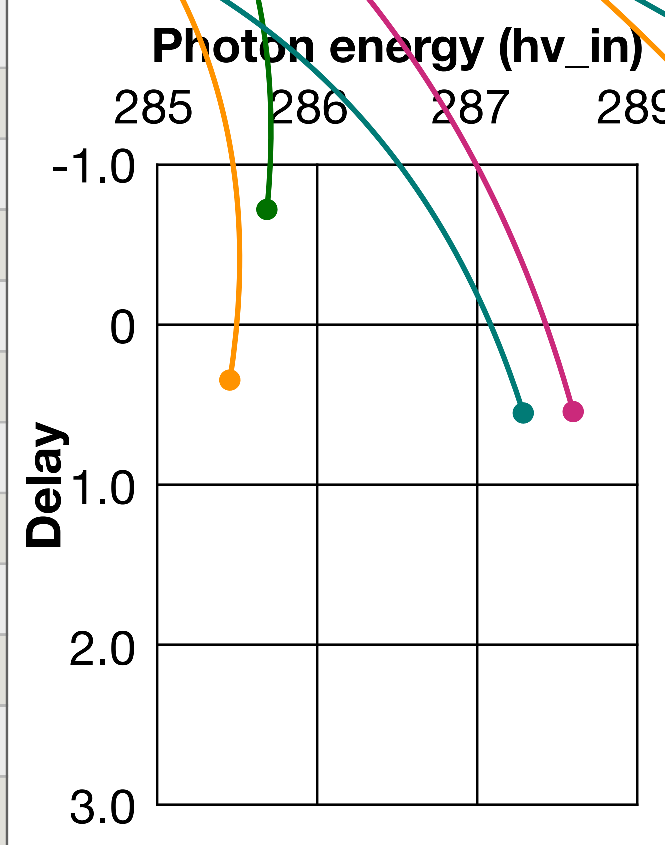
…bandwidth: min = 0, max = 0.7

…I0: min = 20, max = infinity

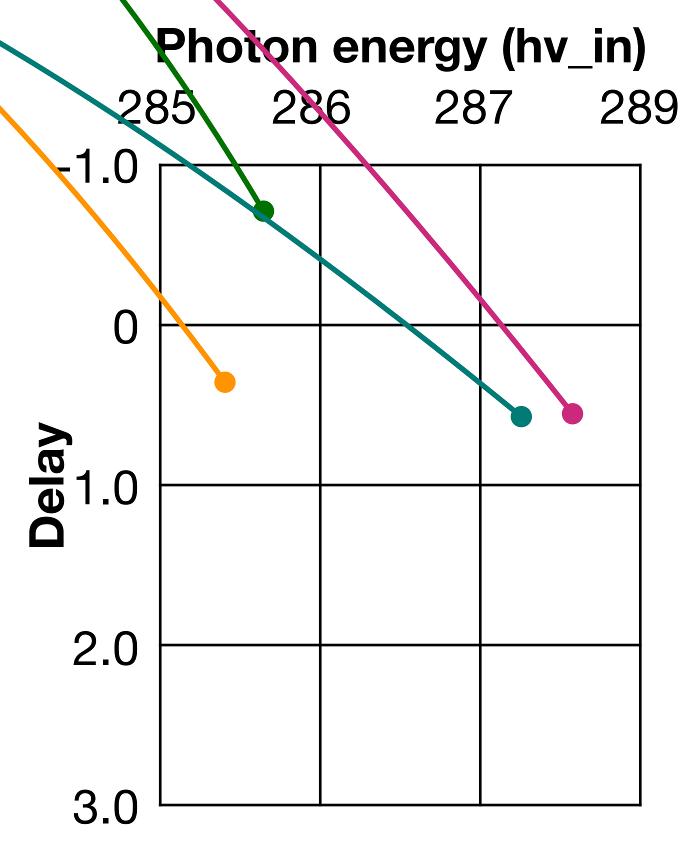
Step 2: Binning into …

…hv_in: Bin width = 1.0 eV

…delay: Bin edges = [-1, 0, 1, 2, 3]

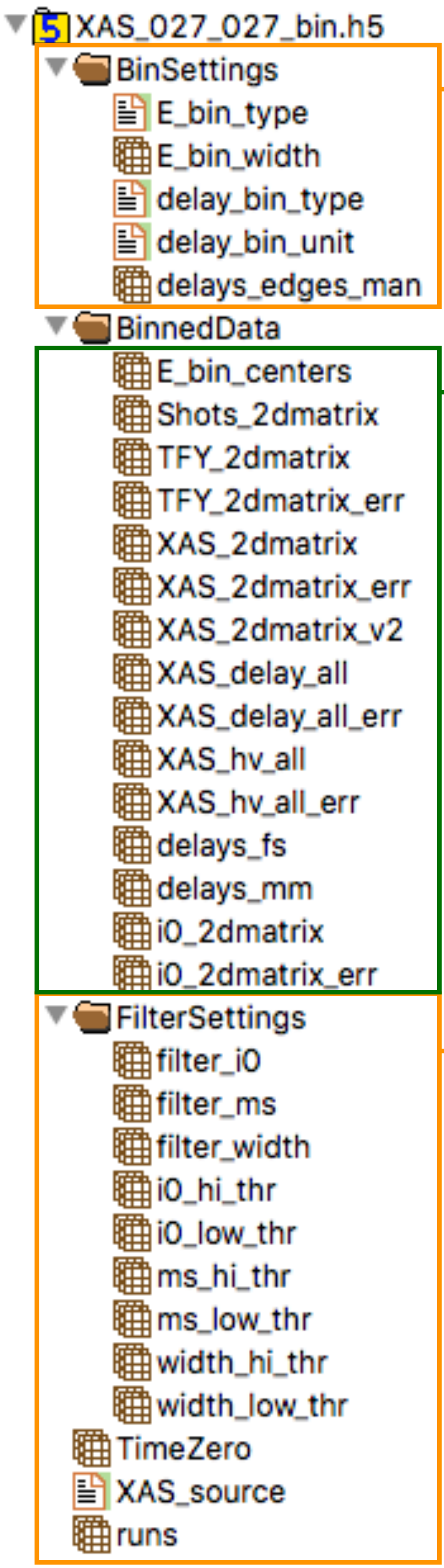


XAS Intensity



I0

Binned data file



Filter and bin setting defined in the parameters cell

Binned data:

- **E_bin_centers, delays_fs and delays_mm**
  1d arrays with photon energy axis and delay axis in fs and mm
- **i0_2dmatrix** and **i0_2dmatrix_err**
  2d arrays (1st dimension is delay, 2nd dimension is photon energy)
  Incident intensity per shot, _err contains the standard error for each bin
- **TFY_2dmatrix** and **TFY_2dmatrix_err**
  2d arrays (1st dimension is delay, 2nd dimension is photon energy)
  Intensity per shot from XAS source (not normalized by I0)
  _err contains the standard error for each bin
- **XAS_2dmatrix** and **XAS_2dmatrix_err**
  2d arrays (1st dimension is delay, 2nd dimension is photon energy)
  XAS intensity per shot calculated as sum(tfy_i / i0_i)
  _err contains the standard error for each bin
- **XAS_2dmatrix_v2**
  Same as XAS_2dmatrix, but calculated as sum(tfy_i) / sum (i0_i).
  For big enough data sets XAS_2dmatrix_v2 and XAS_2dmatrix are identical
- **XAS_hv_all** and **XAS_hv_all_err**
  1d array (same length as E_bin_centers)
  Projection of XAS_2dmatrix onto photon energy axis —> XAS spectrum summed over all delays
- **XAS_delay_all** and **XAS_delay_all_err**
  1d array (same length as delays_fs and delays_mm)
  Projection of XAS_2dmatrix onto delay axis —> XAS delay trace summed over all photon energies.