

Introduction to Online Analysis

Cammille Carinan David Hammer Data Analysis Group

European XFEL Users' Meeting 2024 26 January 2024

European XFEL

Online analysis is data analysis that is performed in near-realtime, to guide an experiment.



tcp://127.0.0.1:22000 (localhos)

There are currently four ways to do online analysis at the facility.

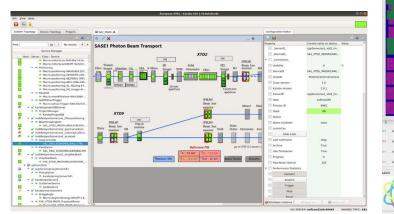
Karabo devices

EXtra-foam

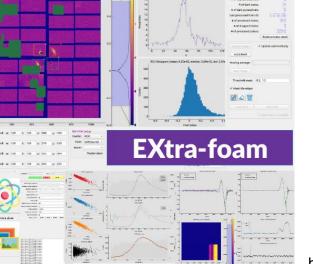
EXtra-metro

ss Detector(ViewGroup) module ax = 1 pulse_ax = -3

Custom tools



https://rtd.xfel.eu/docs/karabo/en/latest/index.html



https://extra-foam.readthedocs.io

Regions of interest
roi n: Parameter = get geometry_from_roi(ROIS['n']) Rate / Hz 9.9 roi 0: Parameter = get geometry from roi(ROIS['0'])
roi p: Parameter = get geometry from roi(ROIS['p']) 2.98k 2.98k 2.96k 2.98k 2.98k 2.98k 9.9 9.9 9.9 9.9 9.9 SA3_XTD10_MONO/MDL/PHOTO... SCS_DET_DSSC1M-1/DET/15CH0... SCS_ILH_LAS/DOOCS/BAM_1932... def __init__{self, prefix=' per(Detector, self). __init__(prefix=prefix) self.prefix = prefix self.flat_field = compute_flat_field_correction(2.96k 9.9 self.rois, PARAMETERS['flat field']) Wiew Matrix(names'(nrefix)data', hiddensTrue) lef data(self, images: f'karabo#(DETECTOR_SOURCE)@(DETECTOR_PATH)') Handle subscriptions automatically Profile if MODULE_SHAPE != images.shape[-2:] with flatfield/saturated/0 - default#-view#with fl... images = images.swapaxes(-3, -1) if images.ndim == 4 and images.shape[self.module_ax] == 1: images = np.squeeze(images, axis=self.module ax) dtype = np.float32 if use_dark: images[images == 0] = 256 dtype = np.uint16
images = images.astype(dtype) if drop intradark: if self.pulse_ax < 8: slice_ = [slice(None)] * images.ndim
slice_ self.pulse_ax] = slice(None, None, 2)

https://rtd.xfel.eu/docs/metropc/en/latest/index.html

self.pulse_ax += images.ndim



Further questions and requirements for your upcoming beamtime? Let us help you! Contact us at da@xfel.eu Or visit https://www.xfel.eu/data_analysis for more information.

tcp://127.0.0.1:22000 (localhost)

delay bam

data

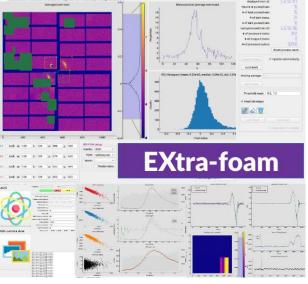
✓ Handle subscriptions automatically

Karabo devices

10 + + + × × SASE1 Photon Beam Tr ServerD, SA3_XTD2_IMGSR/CAM/ ON . . Devicet SAL XTD2 MOSR/CAN ClassID etonicScienceCamer 1.0 2.9.1 23.1 erveryba3_xtd2_im... exficon209 8962 ON Archive Connect Acquire Trigger Stop Recor

https://rtd.xfel.eu/docs/karabo/en/latest/index.html

EXtra-foam



https://extra-foam.readthedocs.io



Counts Rate / Hz 2.98k 9.9

Profiles

9.9 9.9 9.9 9.9 9.9

2.96k 9.9 2.96k 9.9

w#with flatfield/saturated/0 - default#-view#with fi... _ 🛛

#-view#with_flatfield\sat_percent[waveform] - E...

def __init__(self, prefix=''): super(Detector, self),__init__(prefix=prefix) self.fratf_field = compute_flat_field_correction(self.rdat_field = compute_flat_field'))

squeeze if single module if images.ndim == 4 and images.shape[self.module_ax] == 1: images = np.squeeze(images, axis=self.module_ax)

fix 256 value becoming spuriously 0 instead dtype = np.float32 if use_dark: images[images == 0] = 256 dtype = np.uin16 images.atype(dtype)

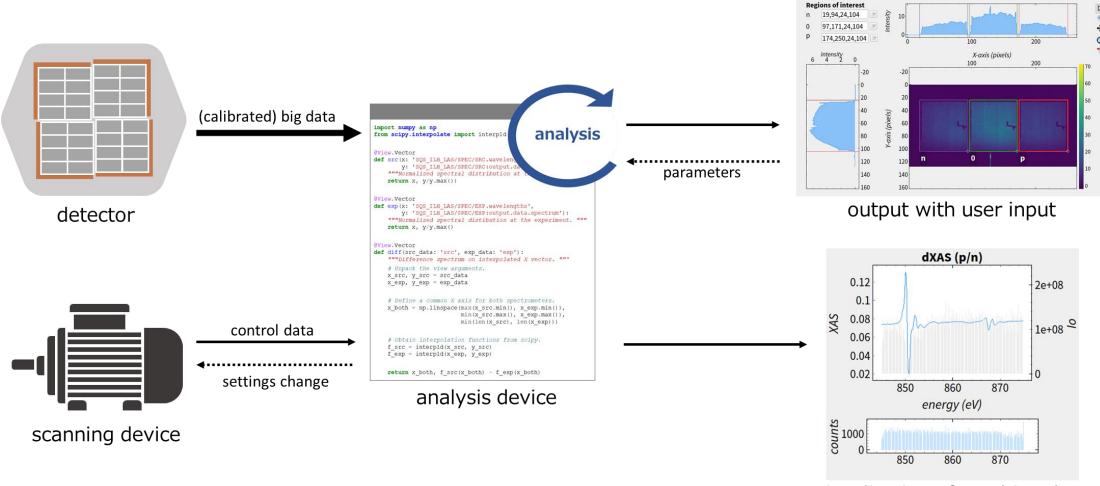
drap_intra_darks if drap_intradark: if self.pulse_mx < 0: self.pulse_mx += images.ndim stice_= = [slice(None)] * images.ndim stice_intradictions] * images.ndim</pre>

https://rtd.xfel.eu/docs/metropc/en/latest/index.html



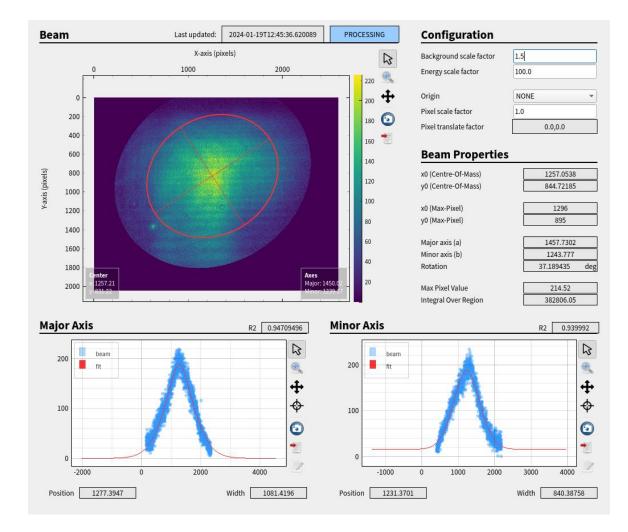
Custom tools

Karabo: the distributed control system is composed of small pluggable units (devices) that represent various components

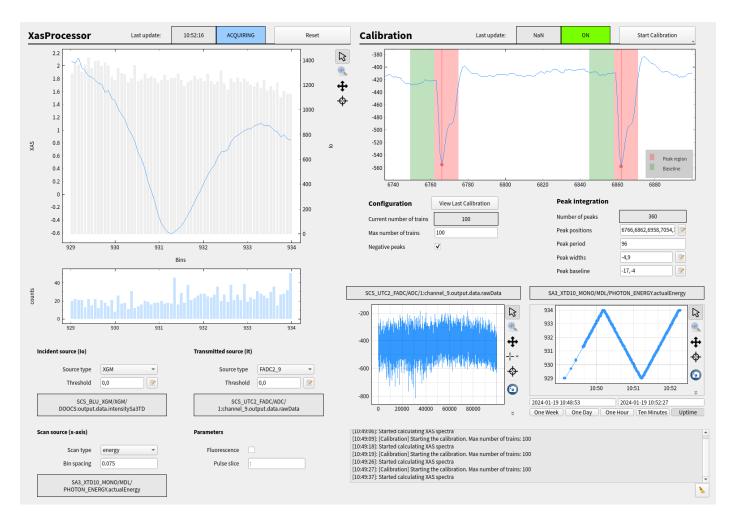


visualization of resulting data

BeamProcessor device: the quality of a laser beam is characterized by calculating its properties such as its centroid and widths



XasProcessor device: the binned X-ray absorption spectrum of a digitizer is calculated by integrating its peaks and normalizing with an XGM



EXtra-foam Karabo devices **EXtra-metro Custom tools** a of head exchanging lass Detector(ViewGroup): module_ax = 1 pulse_ax = -3 tcp://127.0.0.1:22000 (localhost) # Regions of interest rol.m: Parameter = get geometry_from_rol(ROIS['n']) rol_9: Parameter = get_geometry_from_rol(ROIS['p']) rol_p: Parameter = get_geometry_from_rol(ROIS['p']) **10** Rate / Hz 9.9 Counts 2.98k + + + × × 9.9 9.9 9.9 9.9 9.9 SASE1 Photon Beam Tr ServerD, def __init__(selt, prefix=''): super(Detector, self).__init__(prefix=prefix) self.prefix = prefix self.rdi_field = compute flat_field_correction(self.rois, PARAMETERS['flat_field']) SA3_XTD2_IMGSR/CAM/ ON delay bam . . Devicet SAL XTD2 MOSR/CAN ClassID etonicScienceCamer 2.96k 2.96k 9.9 9.9 Thestoldmark 0.5. 1.0 2.9.1 - Mask tile edges data @View.Matrix{name='{prefix}data', hidden=True) def data(self, images: f'karabo#(DETECTOR_SOURCE)@(DETECTOR_PATH)'): 23-1 ierver;ha1_xtd2_im. exfican209 8962 ON 000 ✓ Handle subscriptions automatically Profiles if MODULE_SHAPE != images.shape[-2:]: w#with flatfield/saturated/0 - default#-view#with fi... _ 🛛 images = images.swapaxes(-3, -1) if images.ndim == 4 and images.shape[self.module ax] == 1: images = np.squeeze(images, axis=self.module_ax) CONFOR SELLO **EXtra-foam** FOM I Look at 1 # fix 256 value becoming spuriously θ instead dtype = np.float32 if use_dark: images[images == θ] = 256 Archive #-view#with_flatfield\sat_percent[waveform] - E... dtype = np.uint16 images = images.astype(dtype) Connect Acquire Trigger Stop Recor if drop intradark: drop_intradark: if self.pulse_ax < 0: self.pulse_ax += images.ndim slice_ = [slice(None)] * images.ndim slice_[self.pulse_ax] = slice(None, None, 2) https://rtd.xfel.eu/docs/karabo/en/latest/index.html 10 10 10 10 W

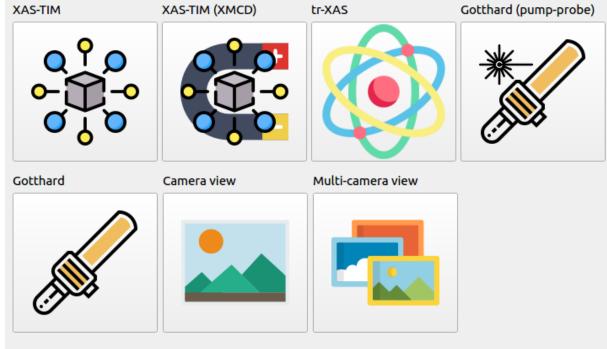
https://rtd.xfel.eu/docs/metropc/en/latest/index.html

https://extra-foam.readthedocs.io

EXtra-foam: an in-house developed framework that processes and visualizes data from experiments that uses 1D and 2D detectors

- **Fast Online Analysis Monitor**
- Graphical program for online analysis
- Supports features such as:
 - Detector preview from user-specified geometry (e.g. CrystFEL geometry file)
 - ROI analysis
 - Azimuthal integration
 - Correlating figures-of-merit with other data

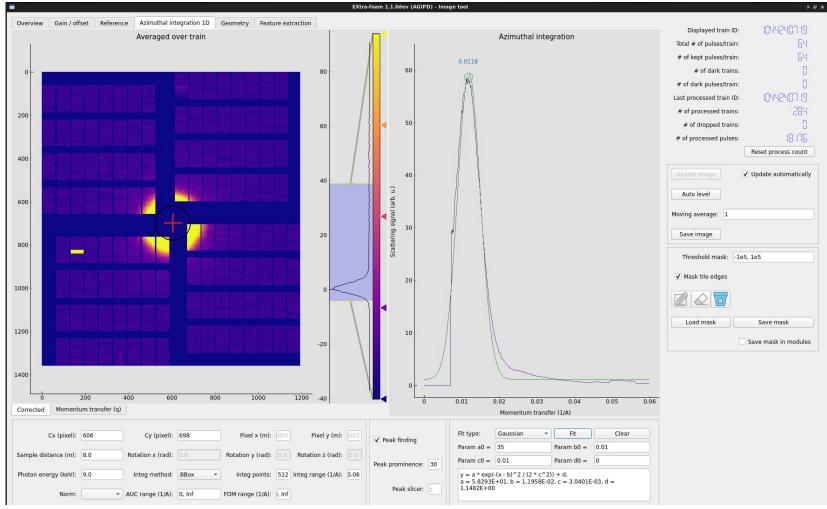
Masking



TOPIC: SCS

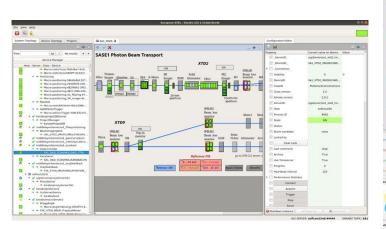
Special suites are small applications dedicated to specific online analysis.

EXtra-foam: analysis procedures for experiments such as azimuthal integration from a scattering experiment using AGIPD are provided



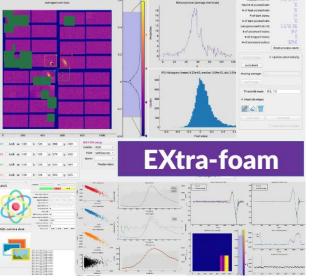
European XFEL

Karabo devices



https://rtd.xfel.eu/docs/karabo/en/latest/index.html

EXtra-foam



lass Detector(ViewGroup): module_ax = 1
pulse_ax = -3 tcp://127.0.0.1:22000 (localhost) # Regions of interest
rol_m: Parameter = get_geometry_from_rol(ROIS('n'))
rol_0: Parameter = get_geometry_from_rol(ROIS('n'))
rol_p: Parameter = get_geometry_from_rol(ROIS('p')) Rate / Hz 9.9 Counts 2.98k 9.9 9.9 9.9 9.9 9.9 def __init__(self, prefix=''): super(Detector, self).__init__(prefix=prefix) self.prefix = prefix self.fit_field = compute_flat_field_correction(self.rois, PARAMETERS['flat_field']) delay bam 2.96k 2.96k 9.9 9.9 data @View.Matrix{name='{prefix}data', hidden=True)
def data(self, images: f'karabo#(DETECTOR_SOURCE)@(DETECTOR_PATH)'): ✓ Handle subscriptions automatically Profiles if MODULE_SHAPE != images.shape[-2:]: w#with flatfield/saturated/0 - default#-view#with fi... _ 🛛 if images.ndim == 4 and images.shape[self.module ax] == 1: images = np.squeeze(images, axis=self.module_ax) # fix 256 value becoming spuriously θ instead dtype = np.float32 if use_dark: images[images == θ] = 256 -view#with_flatfield\sat_percent[waveform] - E... dtype = np.uint16
images = images.astype(dtype) if drop intradark: drop_intradark: if self.pulse_ax < 0: self.pulse_ax += images.ndim slice_ = [slice(None)] * images.ndim slice_[self.pulse_ax] = slice(None, None, 2)

https://rtd.xfel.eu/docs/metropc/en/latest/index.html

images = images.swapaxes(-3, -1)

EXtra-metro



Custom tools

https://extra-foam.readthedocs.io

- Express analysis as small connected steps
- Each step is a simple Python function

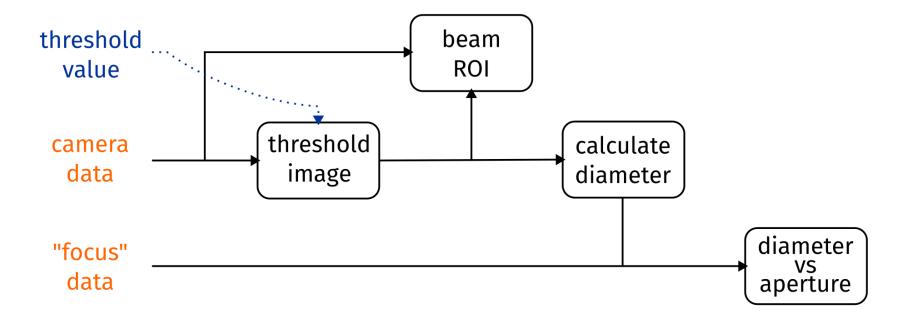
European XFEL

- Code can be changed at runtime
- Allows exploratory data analysis with live data

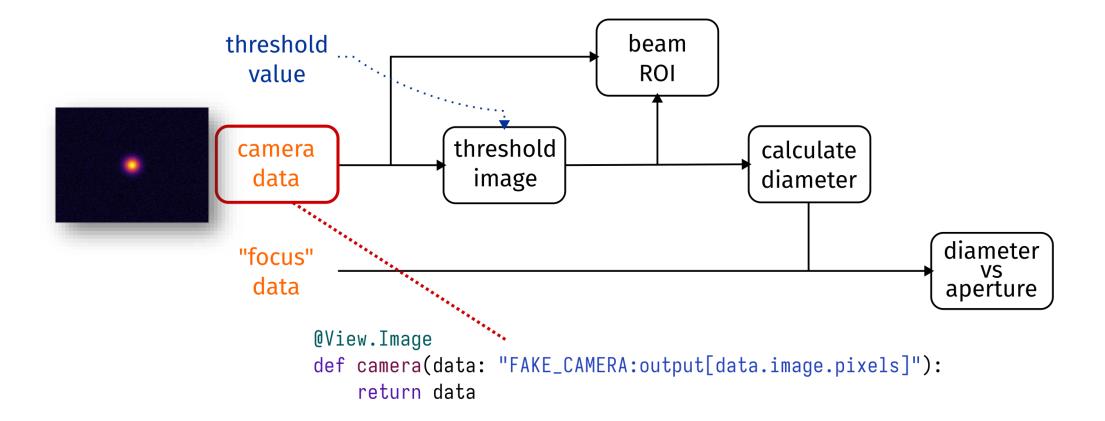
- The framework handles:
- Getting data from the control system
- Executing your analysis
- Visualizing (intermediate) results



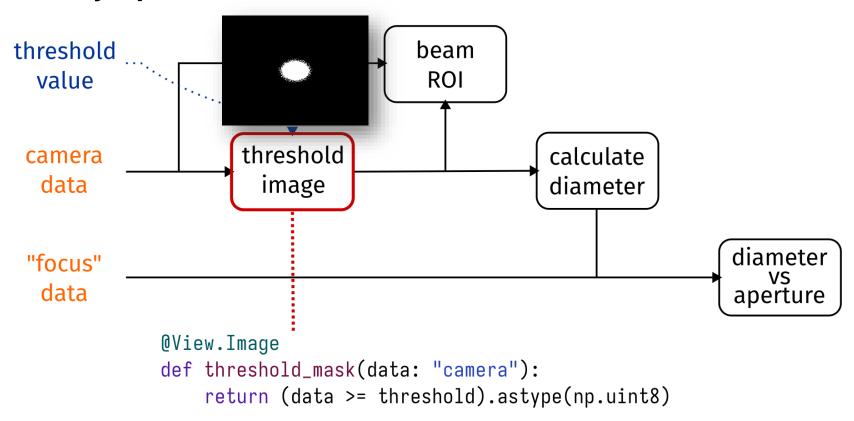
Toy example: analyze camera data



Simple view: access data from Karabo channel

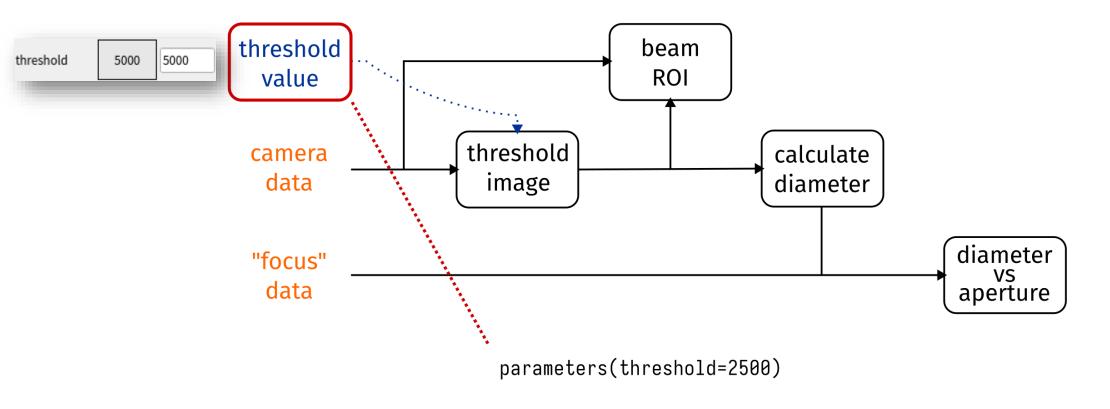


Use regular NumPy operations

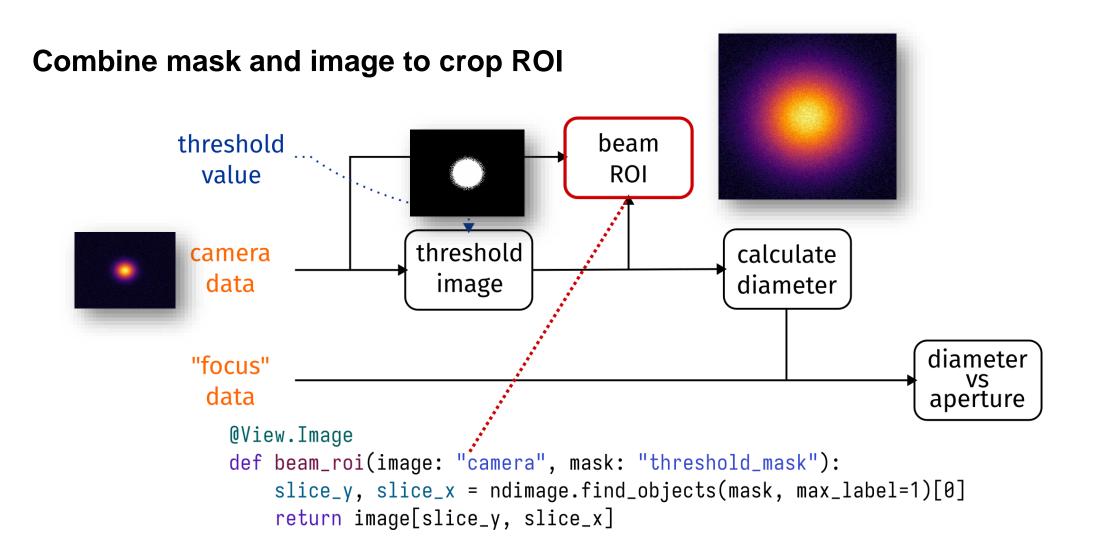


Note: threshold value not yet defined, don't want to hardcode in context file

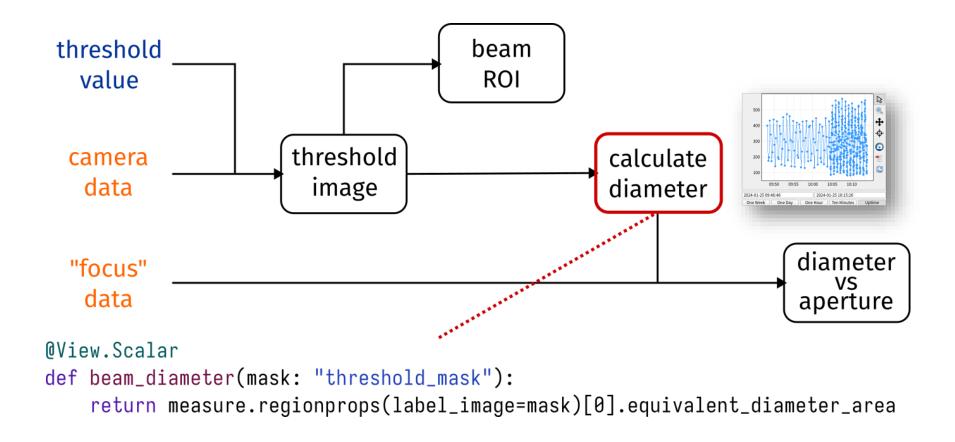
Specify parameters to tweak at runtime

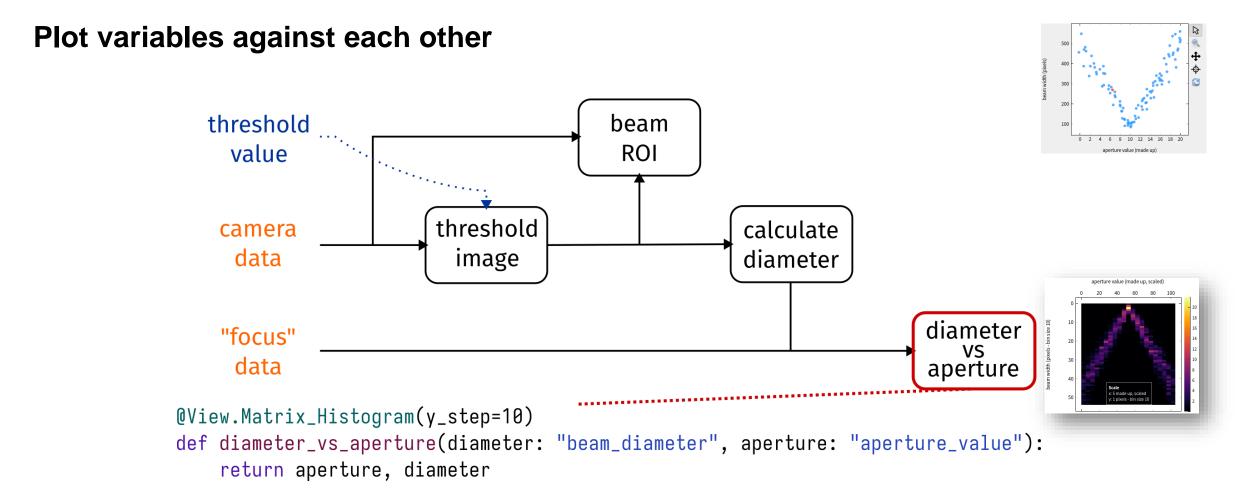


User-supplied value automatically used in previous view

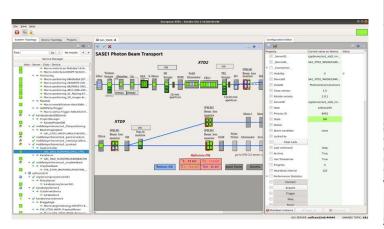


Use standard Python scientific libraries to compute metrics



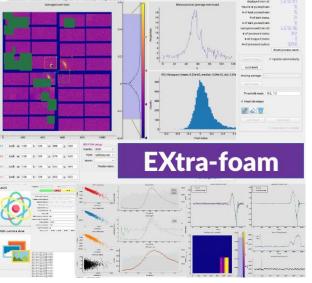


Karabo devices



https://rtd.xfel.eu/docs/karabo/en/latest/index.html

EXtra-foam



https://extra-foam.readthedocs.io

lass Detector(ViewGroup): module_ax = 1
pulse_ax = -3 tcp://127.0.0.1:22000 (localhost) Rate / Hz 9.9 Counts 2.98k 9.9 9.9 9.9 9.9 9.9 delay bam 2.96k 2.96k 9.9 9.9 data ✓ Handle subscriptions automatically Profiles w#with flatfield/saturated/0 - default#-view#with fl... –

EXtra-metro

Regions of interest
rol_m: Parameter = get_geometry_from_rol(ROIS('n'))
rol_0: Parameter = get_geometry_from_rol(ROIS('n'))
rol_p: Parameter = get_geometry_from_rol(ROIS('p'))

def __init__(self, prefix=''): super(Detector, self).__init__(prefix=prefix) self.prefix = prefix self.fit_field = compute_flat_field_correction(self.rois, PARAMETERS['flat_field'])

@View.Matrix{name='{prefix}data', hidden=True)
def data(self, images: f'karabo#(DETECTOR_SOURCE)@(DETECTOR_PATH)'): if MODULE_SHAPE != images.shape[-2:]: images = images.swapaxes(-3, -1)

if images.ndim == 4 and images.shape[self.module ax] == 1: images = np.squeeze(images, axis=self.module ax)

fix 256 value becoming spuriously θ instead dtype = np.float32 if use_dark: images[images == θ] = 256 dtype = np.uint16
images = images.astype(dtype)

if drop intradark: drop_intradark: if self.pulse_ax < 0: self.pulse_ax += images.ndim slice_ = [slice(None)] * images.ndim slice_[self.pulse_ax] = slice(None, None, 2)

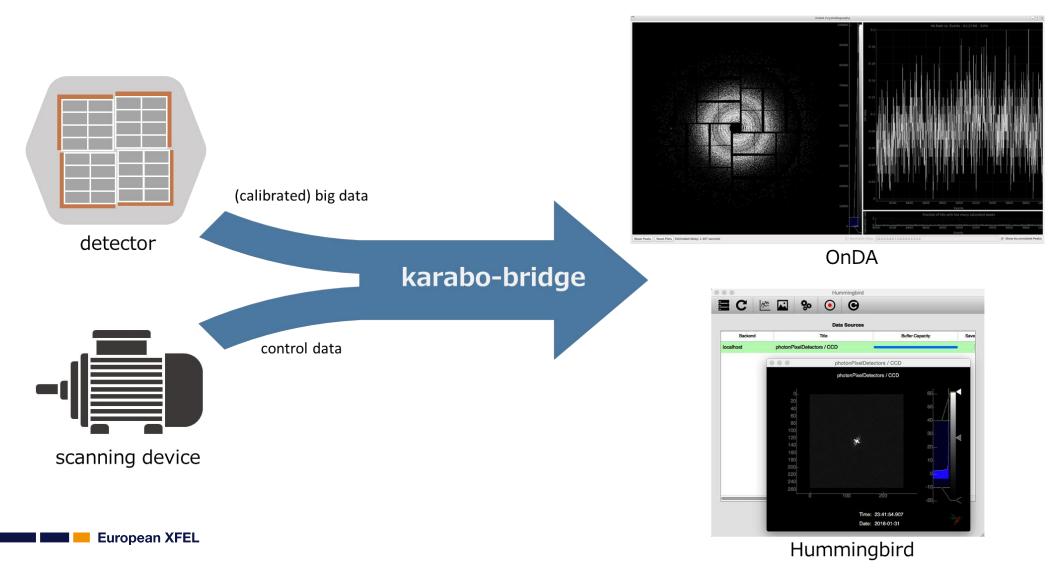
https://rtd.xfel.eu/docs/metropc/en/latest/index.html



Custom tools

-view#with_flatfield\sat_percent[waveform] - E...

karabo-bridge: enables external applications to access data from the control system



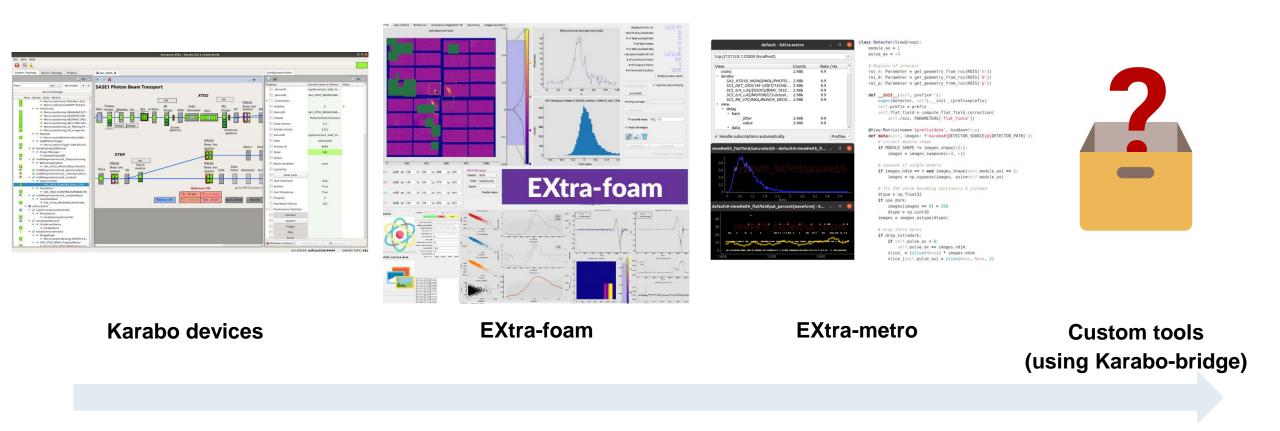
karabo-bridge: the data can be retrieved from a bridge server by using the inhouse developed bridge client

Client libraries:

https://github.com/European-XFEL/karabo-bridge-py https://github.com/European-XFEL/karabo-bridge-cpp

```
>>> from karabo_bridge import Client
>>> krb_client = Client('tcp://server-host-name:12345')
>>> data, metadata = krb_client.next()
>>> data.keys()
dict_keys(['source1', 'source2', 'source3'])
>>> data['source1'].keys()
dict keys(['param1', 'param2'])
>>> metadata['source1']
{'source1': {'source': 'source1',
  'timestamp': 1528476983.744877,
  'timestamp.frac': '744877000000000000',
  'timestamp.sec': '1528476983',
  'timestamp.tid': 10000000073}}
```

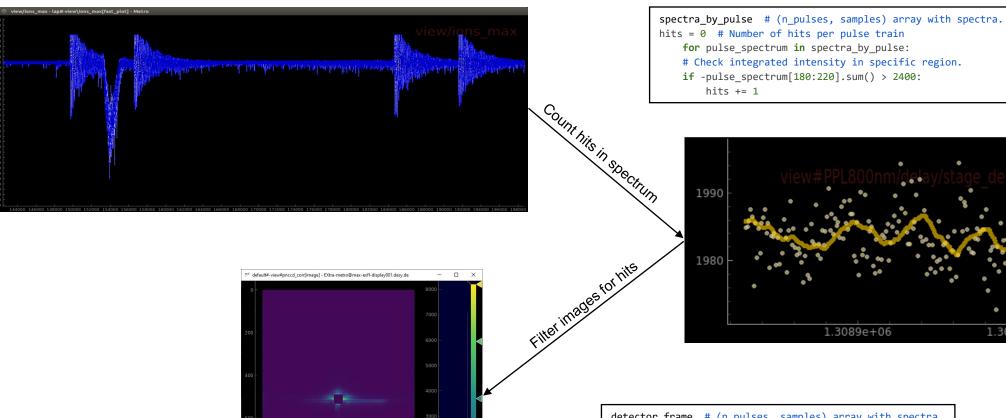
Summary: different levels of flexibility

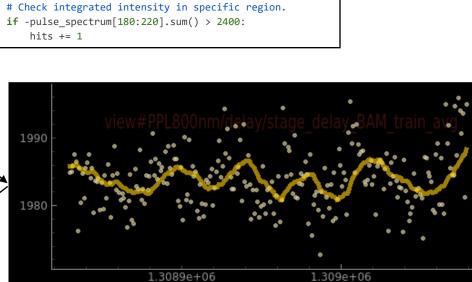


Introduction to Online Analysis

Bonus slides

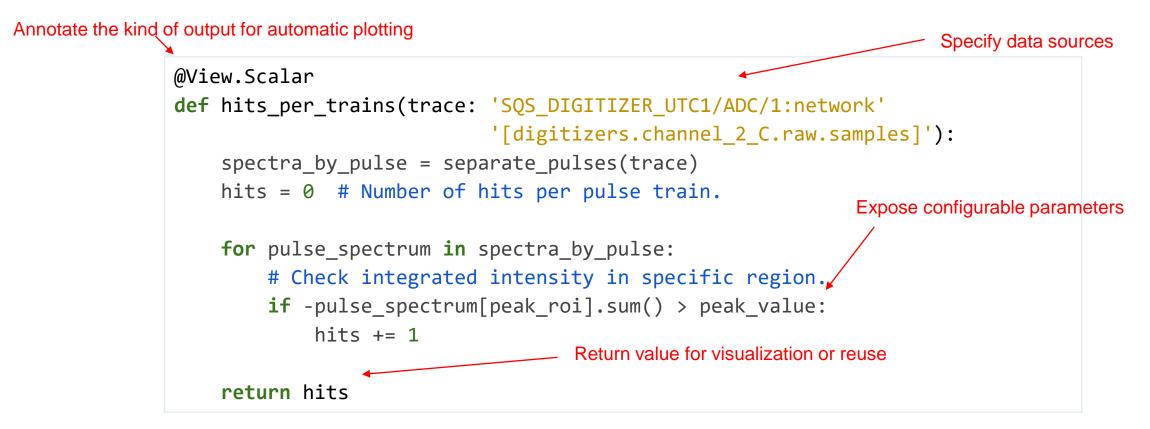
EXtra-metro: programmable custom online analysis framework





detector_frame # (n_pulses, samples) array with spectra. hits # Number of hits per pulse train. if hits > 0: return detector_frame

Conveniently run Python functions on data pipelines



Build analysis pipelines by combining small functions

