Data reduction tools

Egor Sobolev

Data Analysis, SPB/SFX, European XFEL

On behalf of the many colleagues contributing to the data reduction:

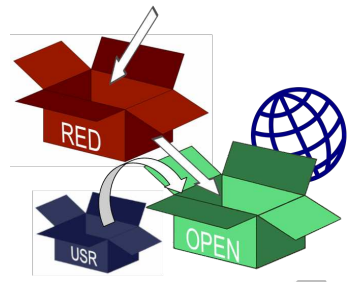
European XFEL data workshop @ European XFEL Users' Meeting 2025

20 January 2025, Hamburg

Data reduction: red box concept

We produce more data than we can store for long time

- New Scientific Data Policy introduces the **mandatory data reduction**
- Within **six month**, select the data for long-term storage and future processing:
  - RAW $< 50\mathrm{TiB}$, no reduction required
  - RAW $< 500\mathrm{TiB}$, reduce to 50 TiB
  - RAW $> 500\mathrm{TiB}$, reduce to 10% of RAW



- RED box may consist of any raw or processed data in approved formats

EXDF-tools: extendable reduction of stored data

How to fill RED box?

- Tools to semantically reduce and compare already recorded data offline. Seamlessly maintain EXDF data structure and compatibility
- Reduction expressed by extendable series of operations on data or its structure
  select-trains, select-entries, subslice-keys, compress-keys, ...
- Reproducible and serializable representation
- Used to reduce multiple prior proposals, extendable by users
- Command line interface, Web-GUI in future

```
% exdf-reduce p5555:r555:raw
    --remove-sources=SPB_EXP_SYS/CAM/*
    --remove-keys=*.XGM.*,current.*
    --remove-trains=1227549381:1227549391
    --output-layout voview
    --output-folder red
```

Reduction of stored data: examples

SPB: SPI retrospective reduction (pilots)

RED box content:

- hit selection (raw + proc)
- a few reference runs without reduction

Freed up 5.5 PB relative to RAW data size

propno	red, TiB	raw, TiB	ratio
2734	102	2355	23.1
2995	258	874	3.4
2746	116	670	5.8
5476	54	2253	41.7
Total	530	6152	11.6

- CSV format

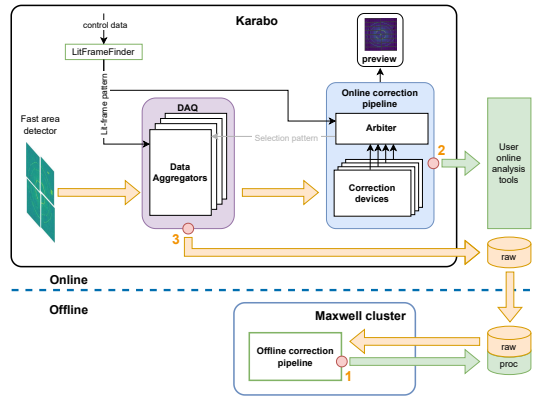
```
trainId;pulseId;flag
1227549381;1400;1
1227549381;1404;0
1227549382;0;0
1227549382;4;0
1227549382;8;1
...
```

- HDF5 format

```
hitlist.h5
+-entry_1
  +- trainId        [uint64]
  +- pulseId        [uint64]
  +- flag           [uint8]
```

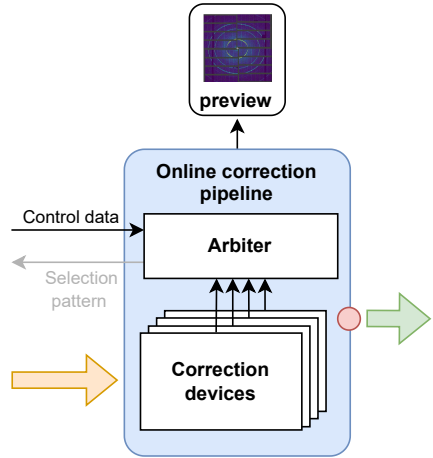
Data reduction integration points

1. Offline processing
   Most reproducible and safe, still large impact on user analysis
2. Online processing
   Mitigate bandwidth and computing power limitations
   Simplify real-time analysis
3. Acquisition
   Maximal impact downstream, no turning back



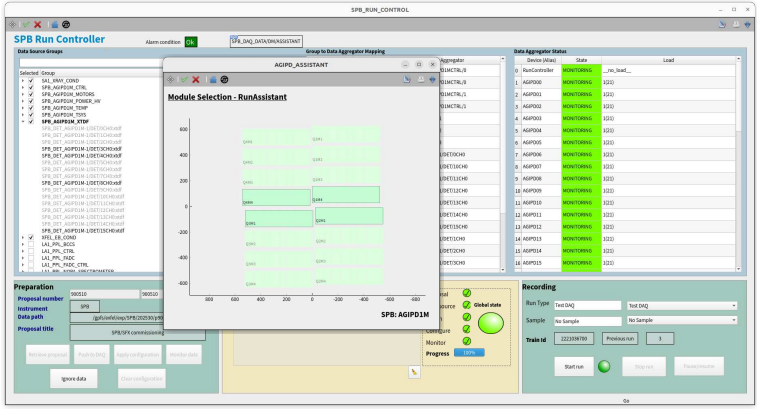
Online processing: correction addons and arbiter kernel

- Correction addon running within the online correction pipeline
  - Distributed across online cluster resources
    Data is local on datacenter-grade GPUs
  - May morph correction result as needed
    Compute statistics and attach metadata
- Arbiter kernel running in central arbiter
  - Combines metadata across all detectors
    May take data reduction decision and feed back to DAQ or to filter output streams
- Configure and combine provided implementations or integrate your own fully custom code



TOOLS

DAQ: module selection



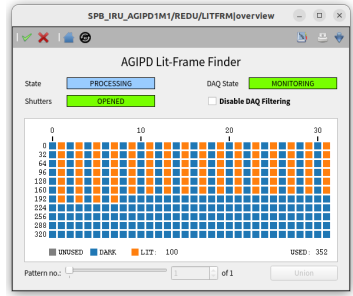
DAQ: data reduction stage

- Gain suppression
  - automatic in fixed gain mode
  - forced in adaptive gain mode
- Frame selection based on decision vector
  - decision vector is boolean vector from any device

Lit-frame selection

EXDF tools

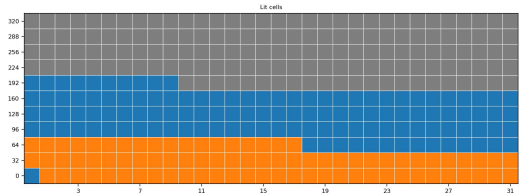
- Annotates the detector frames using
  - Pulse pattern
  - Detector configuration
  - Trigger timing
  - Veto pattern
  - Shutter states
- Online:
  - Karabo device (`LitFrameFinder`)
  - Filtering in DAQ
- Offline:
  - python package (`extra-redu`)
  - annotation and filtering at offline correction



Lit-frames: offline analysis

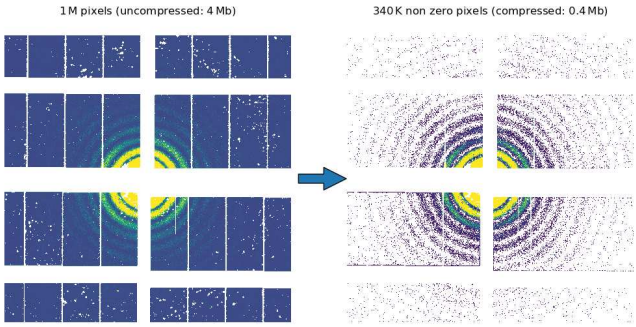
```
from extra_data import open_run
from extra_redu.litfrm import (
    make_litframe_finder, FrameSelection, SelType)
from extra_redu.litfrm.draw.agipd import draw_cells

run = open_run(propno, runno, data="raw")

dev = make_litframe_finder("SPB", run)
r = dev.read_or_process()
sel = FrameSelection(r, guess_missed=True)

# draw lit-frames
cflag = r.output.dataFramePattern[frmno]
ncell = r.output.nFrame[frmno]
draw_cells(cflag[:ncell])

# filter images
flags = sel.litframes_on_trains(
    train_ids, count, cell_ids, [SelType.CELL])
images[flags[0][0]]
```

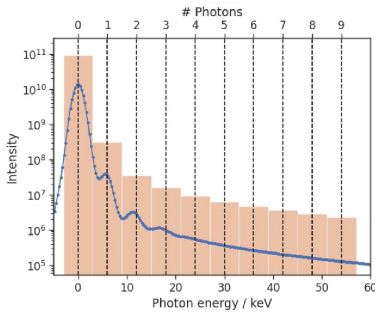
- Integrated in the offline correction pipeline
- You are also welcome to use it



Conversion to photons and compression

- If most of the area is illuminated by a low signal
- Convert to photons, round and compress with `deflate` method



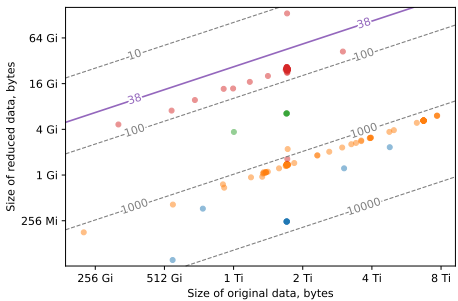


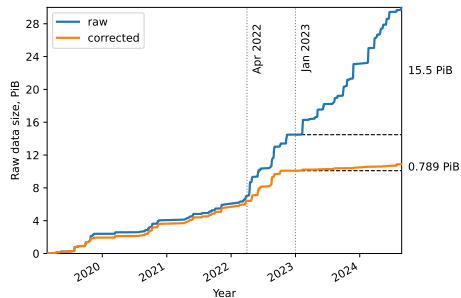
Outcome of corrected images compression

MID: XPCS, Bragg CDI, SAXS/WAXS, etc

Reduction of corrected data, ratio 20 times since 2023

- lit-frame selection
- rounding to photon counts and compression





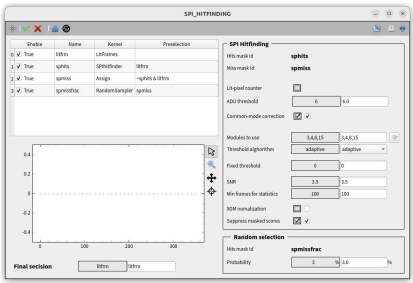
SPI hit-finding

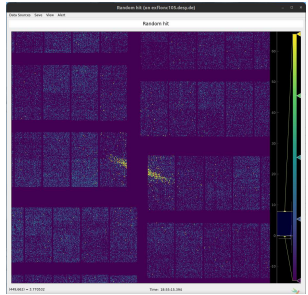
- Threshold binary classifier based on the number of lit-pixels
- Automatic threshold adjustment
- Routinely applied online, filtering data streamed for user analysis
- Applied in offline correction to annotate frames

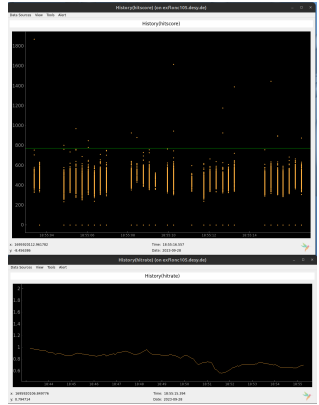
- Online:
  - Lit-pixel counter correction addon
  - Hit-finder arbiter kernel
- Offline:
  - python package (`extra-redu`)
  - hit annotation at offline correction

Hit-finding for SPI: online pipeline

- Hit-finding and filtering with online correction pipeline
- Visualization in Hummingbird







Hit-finding for SPI: offline analysis

- Lit-pixel counter and hit annotation are integrated in offline correction pipeline
- You are also welcome to use it
- Results are stored in EXDF-format

```
from extra_data import open_run
from extra_redu.spi import LitPixelCounter, SpiHitfinder
from extra_redu.fileutils import StackedPulseSource, exdf_save

dc = open_run(propno, runno, data="proc")
src = StackedPulseSource.from_datacollection(
    dc, r"SPB_DET_AGIPD1M-1/DET/(?P<key>\d+)CH0:xtdf", "image")

litpx_counter = LitPixelCounter(src, threshold=adu_threshold)
with mp.Pool(num_proc) as pool:
    chunks = src.split_trains(trains_per_part=1)
    results = pool.imap_unordered(litpx_counter, chunks)
    for _ in results: pass

hitfinder = SpiHitfinder(modules=[3, 4, 8, 15])
hitfinder.find_hits(litpx_counter)

sources = {
    "SPB_DET_AGIPD1M-1/REDU/LITPX_COUNTER": litpx_counter,
    "SPB_DET_AGIPD1M-1/REDU/SPI_HITFINDER": hitfinder,
}
exdf_save(".", "REDU00", runno, sources, sequence_size=3500)
```

Hit-finding for SPI: access hit-finding results

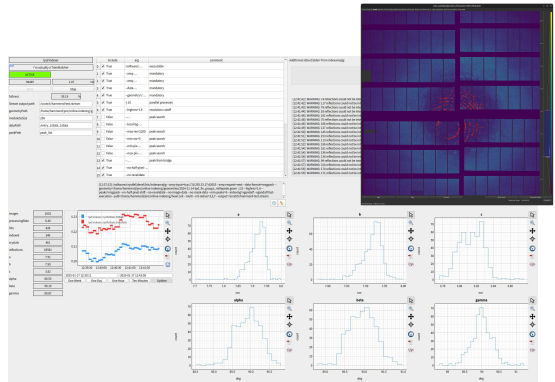
Read results (hit-scores and flags) in Python

```
from extra_data import open_run
from extra_redu.spi import SpiHitfinder

run = open_run(propno, runno, data="proc")
hitfinder = SpiHitfinder.from_datacollection(
    run, "SPB_DET_AGIPD1M-1/REDU/SPI_HITFINDER")

hitfinder.plot_hitrate()
plt.show()

hitfinder.plot_hitscore()
plt.show()
```

Hit-finding for serial crystallography

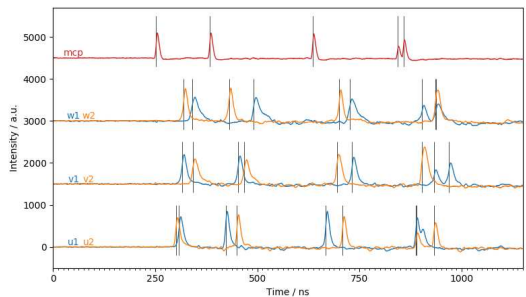
EXDF tools

- Online
  - Peak-finding on full data (`peakfinder9` add-on)
  - Hit-finding Arbiter kernel
  - Bridge to `indexamajig`
- Offline
  - `EXtra-Xwiz` pipeline (standalone)
    Visit poster “Automatic data processing and results overview during SFX experiments” by **Oleksii Turkot**



REMI event reconstruction

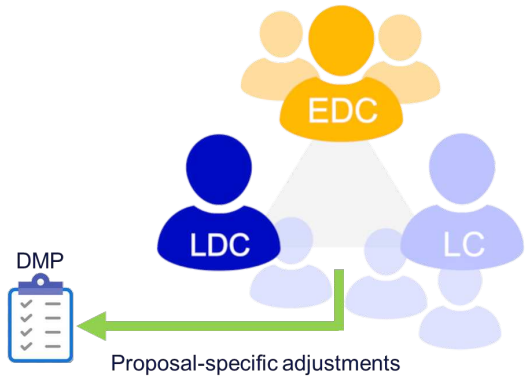
- Delay line detectors for REMI/COLTRIMS and RIXS sampled with GHz digitizers
- On-FPGA zero suppression of analog signal allows reduction by $50\times$
- Digitization and event reconstruction to $(\mathrm{x}, \mathrm{y}, \mathrm{t})$ tuples reduces data by another $> 40\times$, compared to raw $> 2000\times$




```
(-46.99, -30.75, 1277.01, 0)
(45.44, 7.41, 1678.84, 3)
(38.30, 11.44, 3679.09, 1)
(-27.46, -23.33, 6249.36, 15)
(-33.62, -19.56, 6249.64, 19)
```

Data reduction in your experiment

The choice of data reduction tools is specific for every experiment
Discuss the opportunities with Local Data Contact



More opportunities: visit our posters and contact us at da@xfel.eu

Operation-specific methods

are related to instrument operation itself, no analysis of detector images is usually required.

These methods are robust, low risk, and the feedback latency is compatible with online requirements.

- Lit-frame selection: 1–352
- Shutter-based train selection: $\gg$2
- Gain suppression: 2
- Region of Interests, e.g. modules: 1–16
- Compression: up to 40
  Low intensity, signal localized in small area

Technique-specific methods

require analysis of detector data, and typically involves tuning of certain parameters.

The associated risks are generally higher, as well as computational complexity, there are challenges for automation.

- Hit finding: 10–10000
  SFX, SPI
- Event reconstruction: ˜2000
  REMI/COLTRIMS, (tr-)RIXS
- Azimuthal integration: ˜1000
  SAXS, WAXS, Powder diffraction, XPCS
- Correlation functions: ˜1000
  XPCS, XCCA

Summary

- Data reduction is mandatory
- Reduction is automated with tools integrated in the infrastructure
- The portfolio of data reduction methods and tools is growing
- Your Local Data Contact helps to enable data reduction

* Special thanks to European XFEL users joining in pilot projects: Jonas Seilberg, Duane Loh, Filipe Maia, Xavier Paulraj, Kartik Ayyer and many others

* Sobolev, Schmidt et al: Data reduction activities at European XFEL: early results, Front. Phys. 12, 1331329 (2024), URL: http://dx.doi.org/10.3389/fphy.2024.1331329

Let's meet at our posters

Thank you for you attention

Data Analysis Group, da@xfel.eu, www.xfel.eu/data_analysis

Data reduction activities at the European XFEL have been partially funded through the European Union's Horizon 2020 research and innovation programme under grant agreement no. 101004728.