NTNU

Innovation and Creativity

An easy to use XRD data analysis framework for users and scientists, online and offline

Amund Hov, trainee ID15

NTNU, Institute of Physics

July 31st 2013

Motivation

- When doing Xray diffraction experiments it is often necessary to perform several steps of data reduction and analysis in order to evaluate a result.
- In some cases data might even be found to be completely useless upon inspection after the experiment has concluded.
- Online analysis at the beamline can be useful in these cases.

NTNU
Innovation and Creativity

Motivation contd.

- Users are not necessarily specialists in xray diffraction or scientific computing
- It would be nice to be able to give external users not just their data, but also the tools to process them, wouldn't it?

NTNU
Innovation and Creativity

Implementation

Started xraylib as intern three months last summer. Continued this year from march as part of master degree “technical physics“ at NTNU, Norway.

Python

- High level $\Rightarrow$ Quick and easy to develop and prototype.
- Yet high performance when using compiled libraries
- Evergrowing library of tools for scientific computing.
- No license necessary (unlike matlab)

NTNU
Innovation and Creativity

Dependencies

- No dependency on ESRF computing infrastructure
- Python libraries
  - pyFAI Fast azimuthal integration on GPU / CPU
  - scipy / numpy Almost all you need for arrays/images/matrices
  - scikits-image Branch of scipy. More filters, including tomographic reconstruction (FBP, SART)

NTNU
Innovation and Creativity

Installation

- Scripts for setting up xraylib for use *and* development are provided.
- In theory this is all you should need if you have system-wide access

```
git clone https://github.com/amundhov/xraylib.git
cd xraylib
./install.sh or ./develop.shp
```

- Python library dependencies maintained in **requirements.txt** and can be parsed by standard python package tools such as **pip** and **distutils**.

NTNU
Innovation and Creativity

Installation cntd.

- In practice it is best to have your system administrator install the required python libraries. On debian based systems these are all available through APT.
- For development or local installation make use of python's virtual environments:

```
virtualenv –system-site-packages venv
source venv/bin/activate
[ perform steps from previous slide ]
```

- Whenever venv/bin/activate is sourced xraylib will be available on your path along with the scripts in scripts/

NTNU
Innovation and Creativity

What's in it (for me)?

Some examples

- Calculation of sample properties such as xray absorption at given energy.
- Simplified file access (EDF, multiframe EDF, hdf5, anything fabio supports)
- Utility functions and examples for making small scripts and workflows.

We'll have a look at applications shortly.

NTNU
Innovation and Creativity

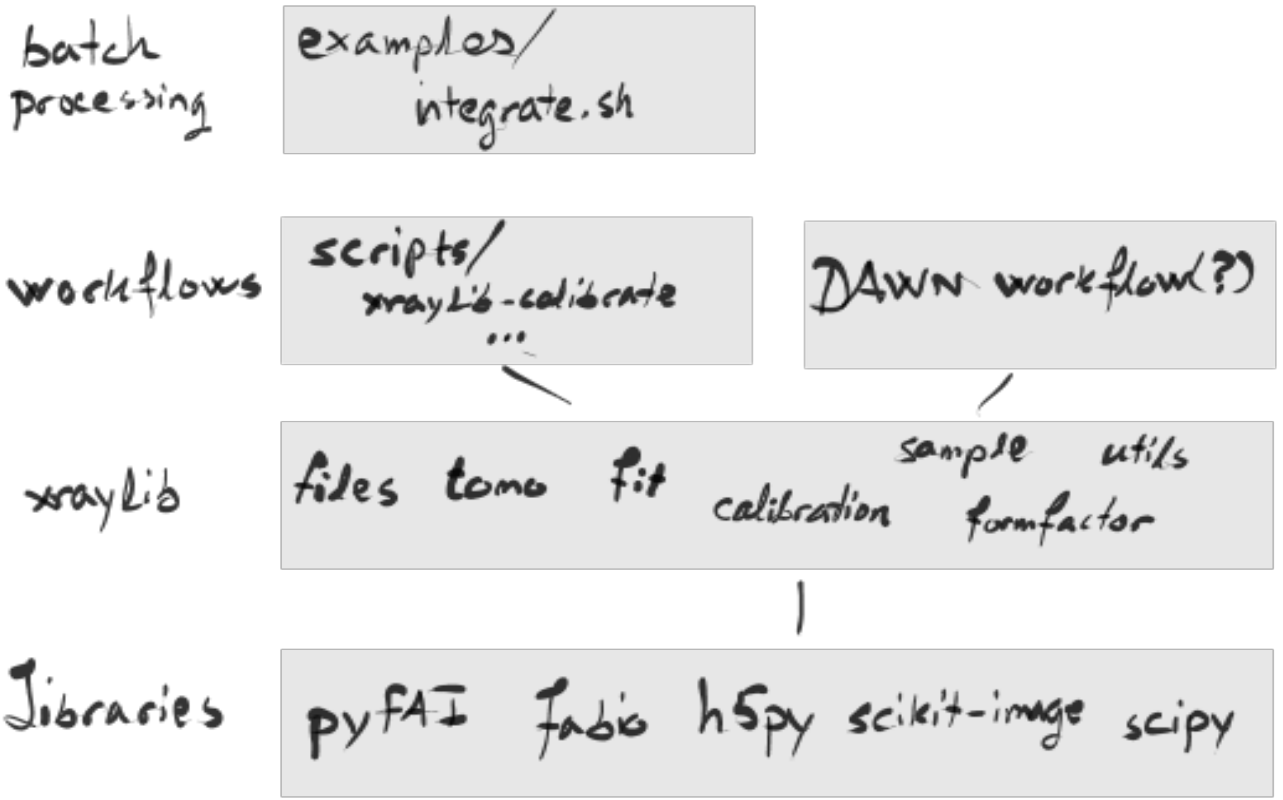
xraylib + DAWN

- Data exploration. Rich support for slicing and visualising data in HDF5 format.
- Peak fitting and export.
- Integration with python scripts possible in their workflow tool. Lots of functionality coming, but currently not used.

In summary: process data with xraylib, visualise with DAWN.

NTNU
Innovation and Creativity

Structure



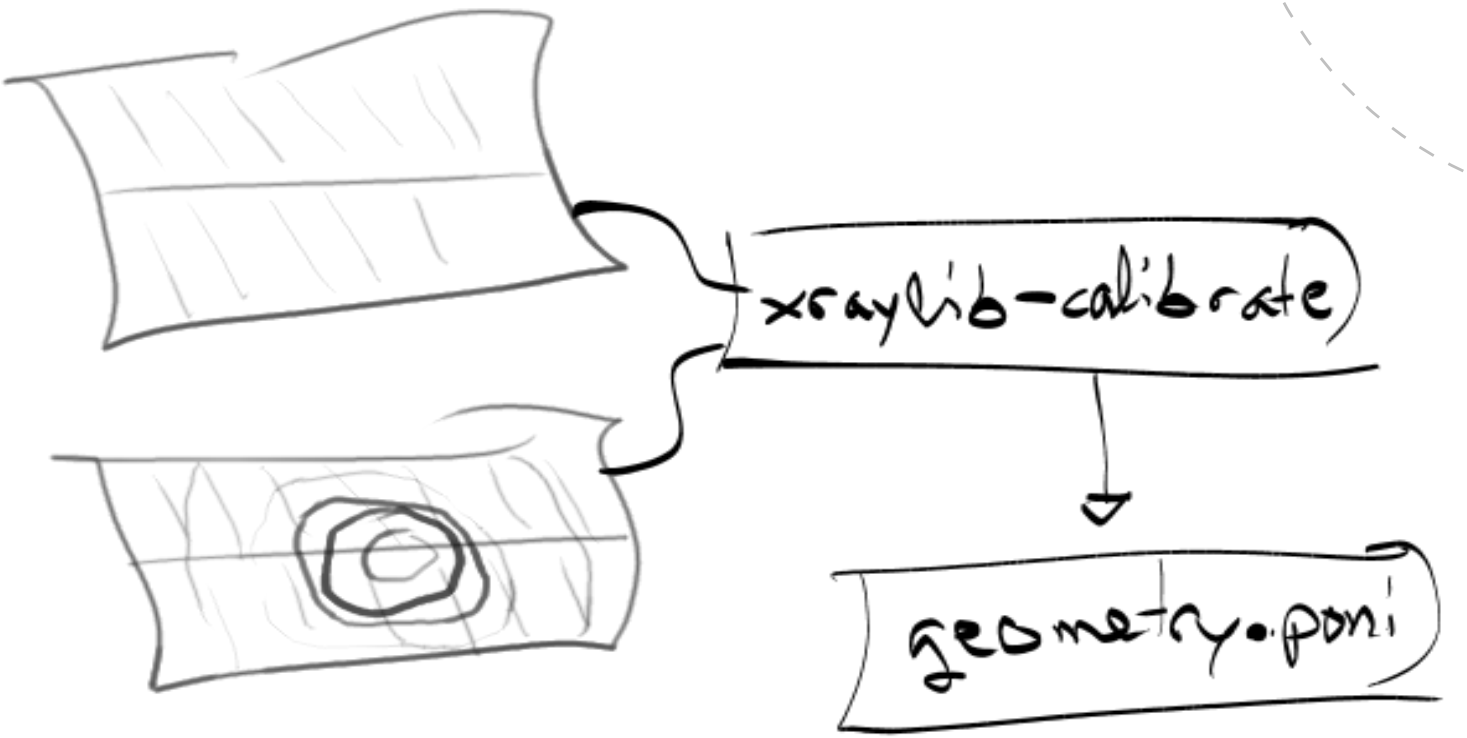
NTNU
Innovation and Creativity

Current applications of the framework

1. Detector calibration. Detector tilt and beam origin.
2. Azimuthal integration.
3. Sinogram assembly of specific phases in diffraction tomography experiments [DSCT]
4. Sinogram correction and reconstruction

NTNU
Innovation and Creativity

scripts/xraylib-calibrate



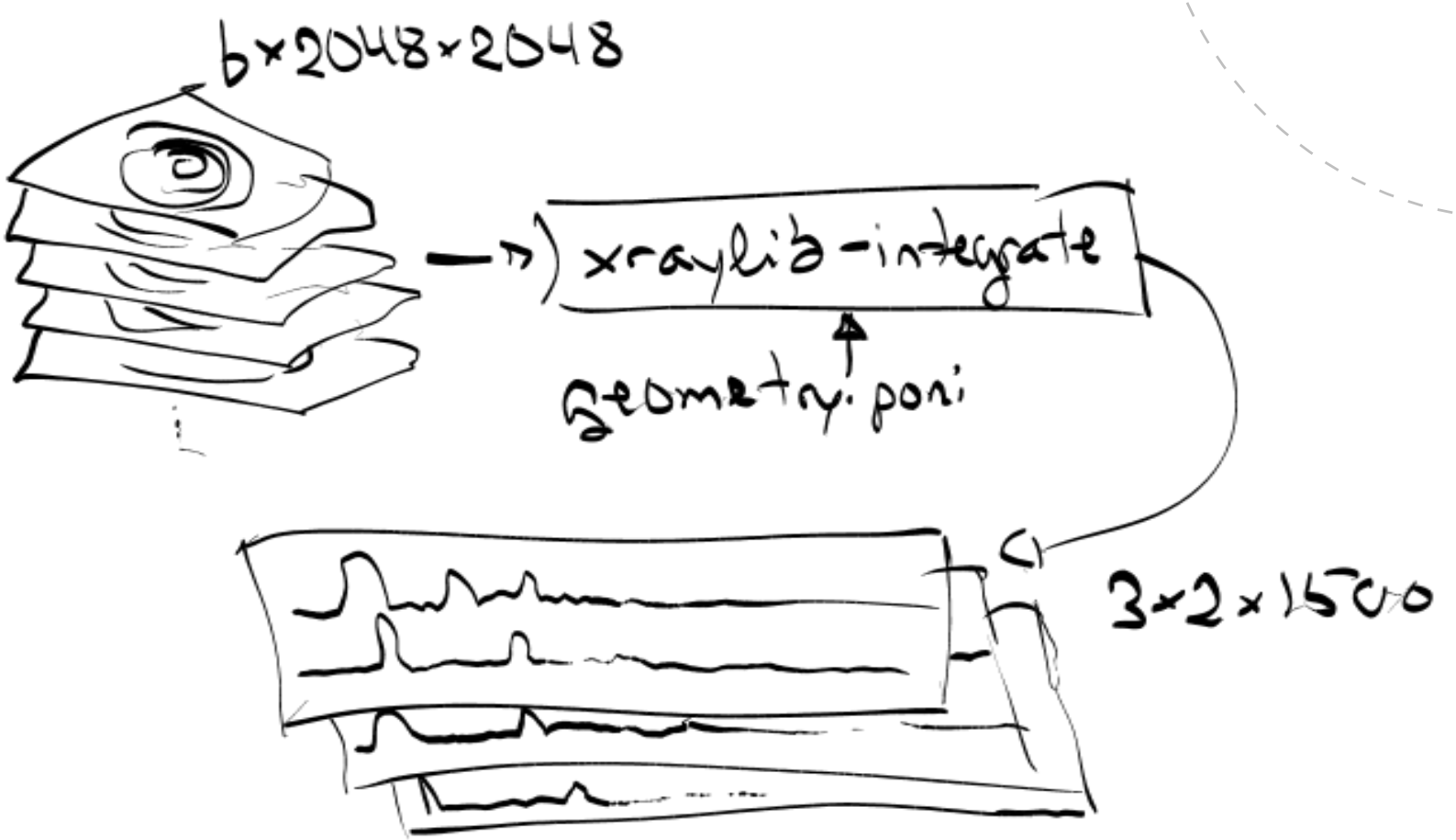
NTNU
Innovation and Creativity

scripts/xraylib-calibrate

```
Usage: xraylib-calibrate <options> CALIBRATION_IMAGE [DARK_CURRENT1,DARK_CURRENT2,...]

         XRD calibration routine based on ring shape.

Options:
  -s, --silent          supress output to terminal.
  -t, --timings         Report execution times.
  --pixels              Origin and calibration limits in pixels instead of mm.

  File options:
    --out=FILE          Save calibration image after subtracting darkcurrent.
    --data-set=STRING   Location to save data set.
    -p=FILE, --poni=FILE
                        File to save detector geometry.

  Detector options:
    -D DETECTOR_NAME, --detector=DETECTOR_NAME
                        Detector name
    --distance=distance [mm]
                        Detector distance from sample
    --binning=x y       Number of pixels that detector is set to group.
    --tilt=a b [degrees]
    --origin=x y [mm]   Initial detector origin wrt beam

  Calibration options:
    --limits=lower upper
                        Radial distance [mm] to use for calibration.
```

NTNU
Innovation and Creativity

scripts/xraylib-integrate



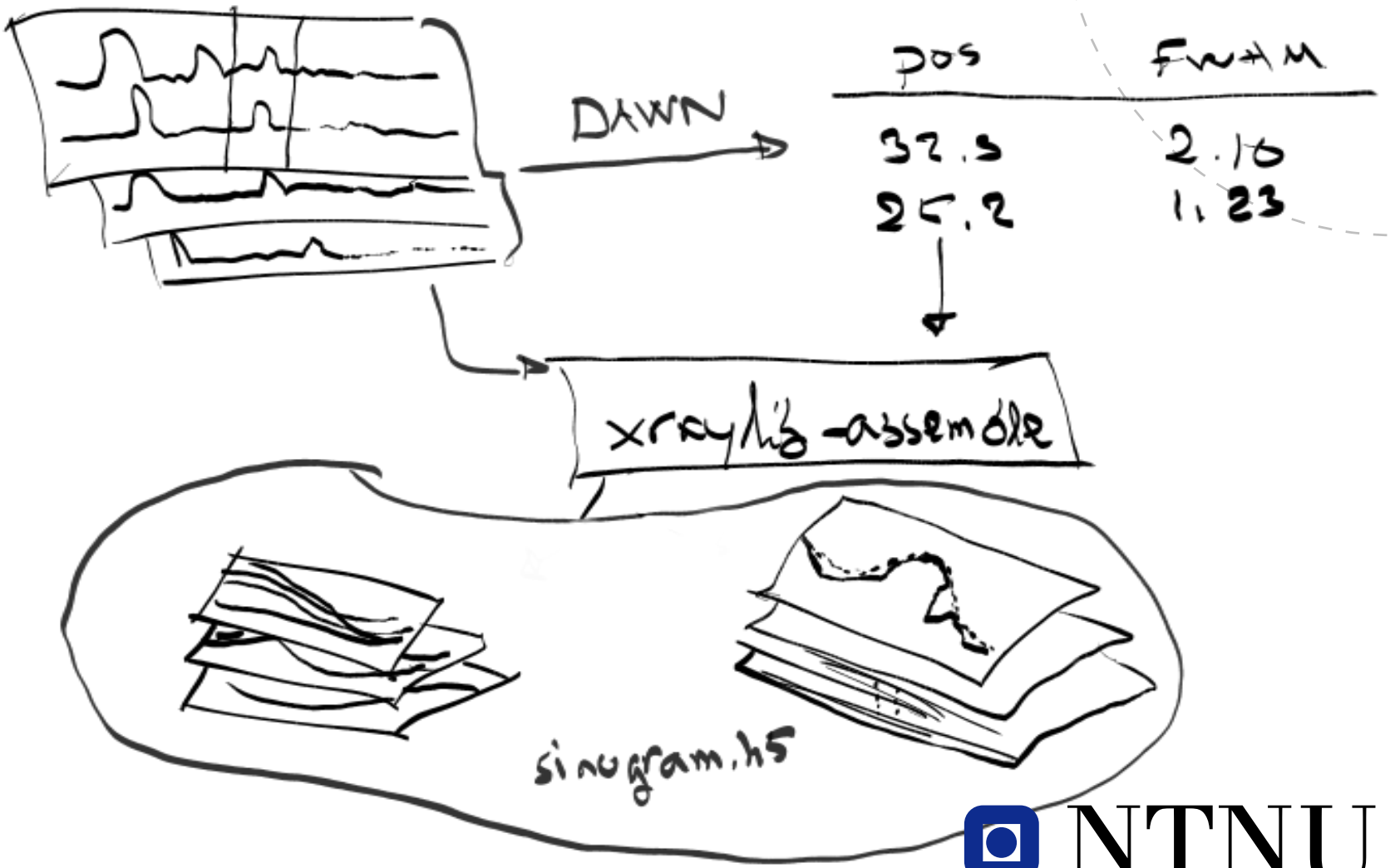
NTNU
Innovation and Creativity

scripts/xraylib-integrate

```
Usage: <options> --data-prefix=/mnt/data/../EXPERIMENT_...

         Integrate diffraction images and assemble into dataset.
Experiment parameters are assumed to be separated by underscore
NAME_xxx_yyy_zzz.edf will produce a data set with powder profiles in
dimensions x,y,z.

Options:
  -s, --silent          supress output to terminal.
  -t, --timings         Report execution times.
  --nbuffer=NBUFFERS    Number of files to buffer when loading files
  --points=POINTS       Number of points to keep radially.

  Input options:
    --data-path=/path/IMAGE_xyz_
                        Integrate files starting with this path.
    --dark-path=/path/DARK_xyz_
                        Average darkcurrent files starting with this path.
    --dark=DARK         Use single darkcurrent image.
    -p PONI_FILE, --ponit=PONI_FILE
                        Name of poni file with detector geometry.

  Output options:
    -o FILE, --out=FILE
                        File to save integrated dataset.
    --data-set=STRING   Location to save data set.
```

NTNU
Innovation and Creativity

scripts/xraylib-assemble



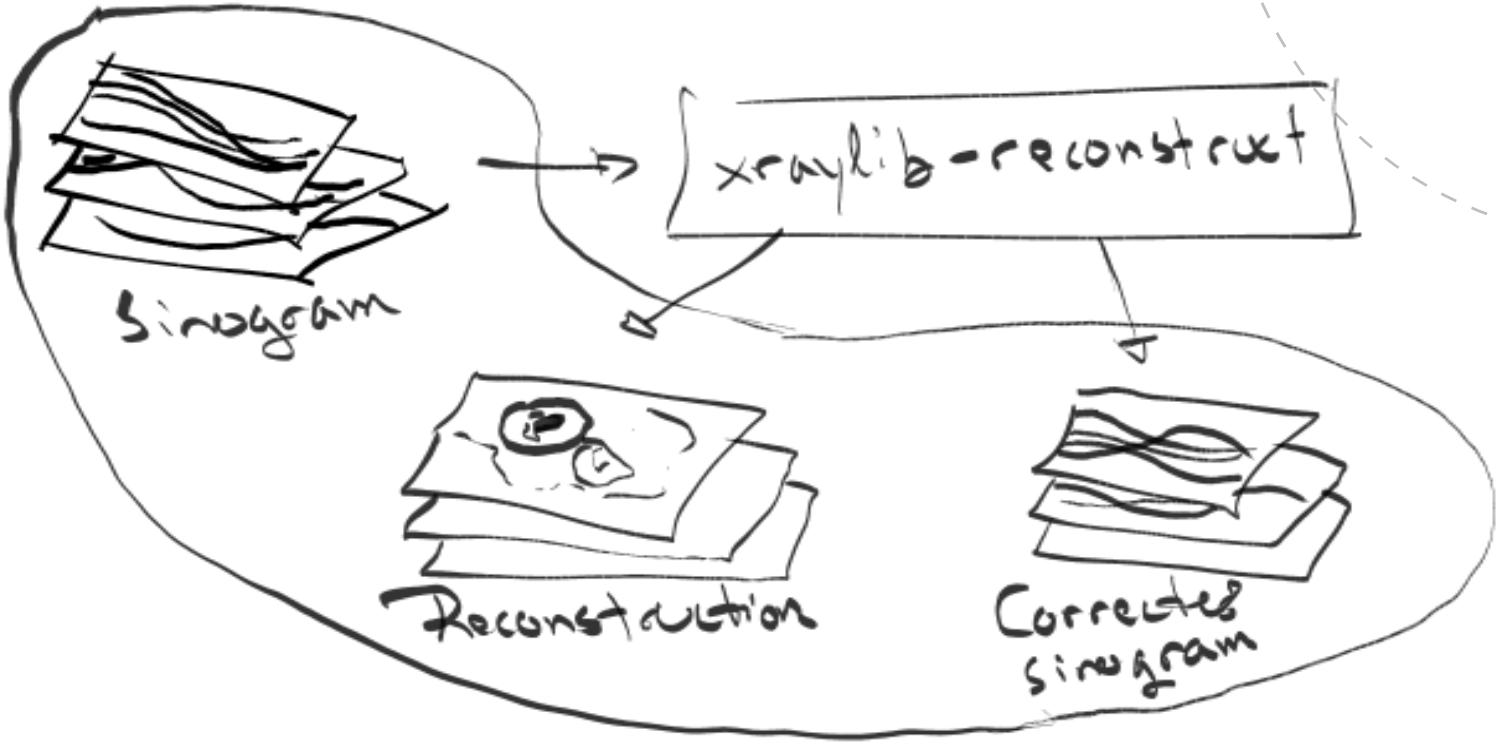
NTNU
Innovation and Creativity

scripts/xraylib-assemble

```
Usage: xraylib-assemble <options> diffractogram.h5

Options:
  -s, --silent          supress output to terminal.
  -t, --timings         Report execution times.
  --flip                Flip every other scan line.

  Input options:
    --position=PEAK_POSITION
                        Peak channel position
    --fwhm=PEAK_FWHM    Peak FWHM full width at half maximum
    --shape=PEAK_SHAPE  Peak shape. [gaussian|delta]
    --peak-file=FILE    Exported peaks from DAWN.
    --input-set=STRING  Override data set [default /xraylib/diffractogram].

  Output options:
    -o FILE, --out=FILE
                        Output file.
```

NTNU
Innovation and Creativity

scripts/xraylib-reconstruct



NTNU
Innovation and Creativity

scripts/xraylib-reconstruct

```
Usage: xraylib-reconstruct <options>

         Correct sinograms and perform reconstruction.
By default the sinogram is split into odd and even rows
which are correlated to correct for offset in interleaved
scans, followed by a search for the center of rotation.

Options:
  -s, --silent          supress output to terminal.
  -t, --timings         Report execution times.

  Input options:
    --sinogram-group=SINOGRAM_GROUP
    --iterations=SART_ITERATIONS
                        Number of iterations to perform SART reconstruction.
                        [default=1]
  Output options:
    --output-sinogram
    --output-reconstruction
    -o FILE, --out=FILE
                        Write reconstruction slices to file.
```

NTNU
Innovation and Creativity

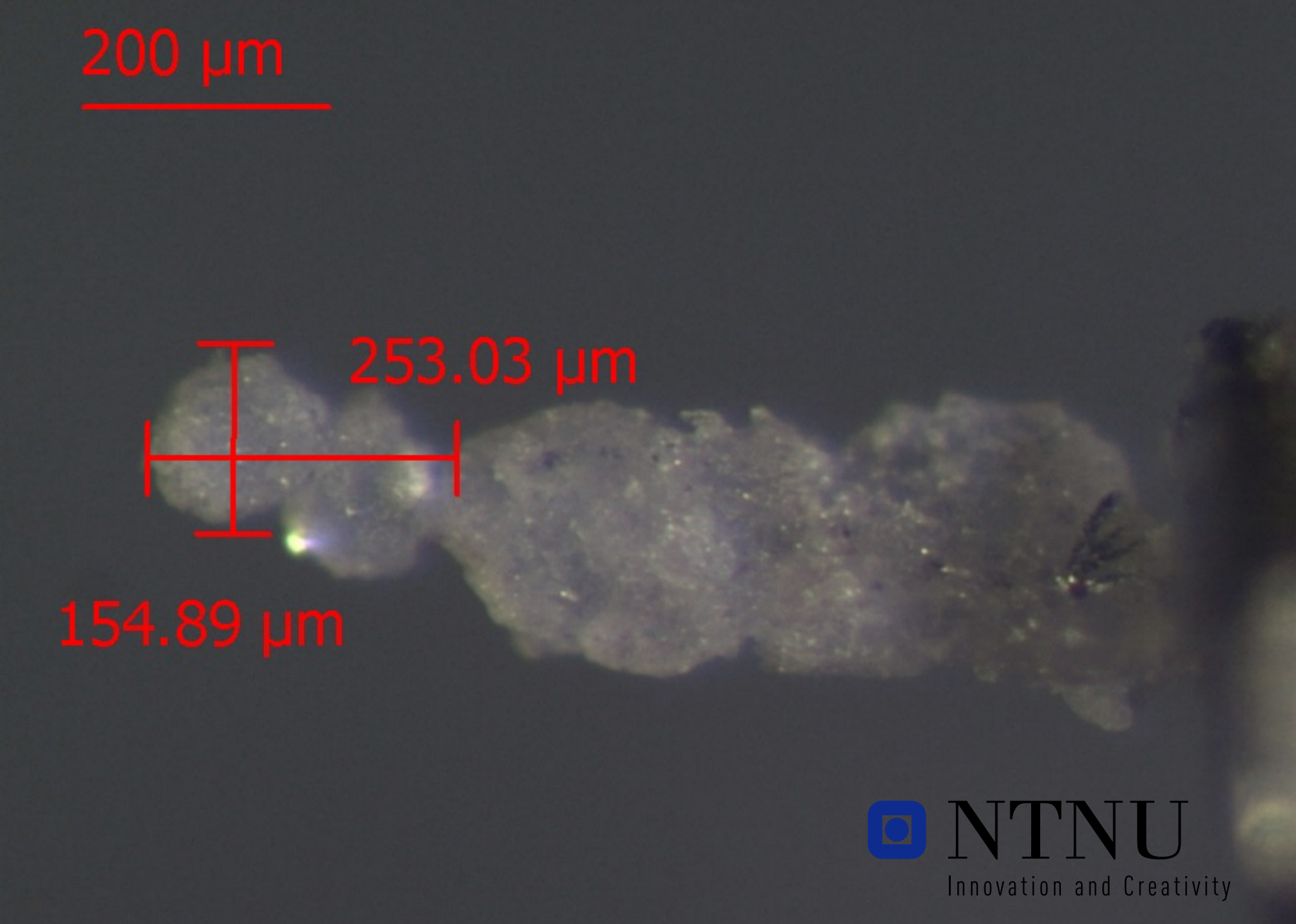
Experimental application

Just two weeks ago diffraction tomography data was collected to test the diffraction tomography workflow.
Dummy sample consisting of small ($\approx 150 \mu m$) glass spheres suspended from a needle using wax.

NTNU
Innovation and Creativity

[unintentionally blank]

NTNU
Innovation and Creativity

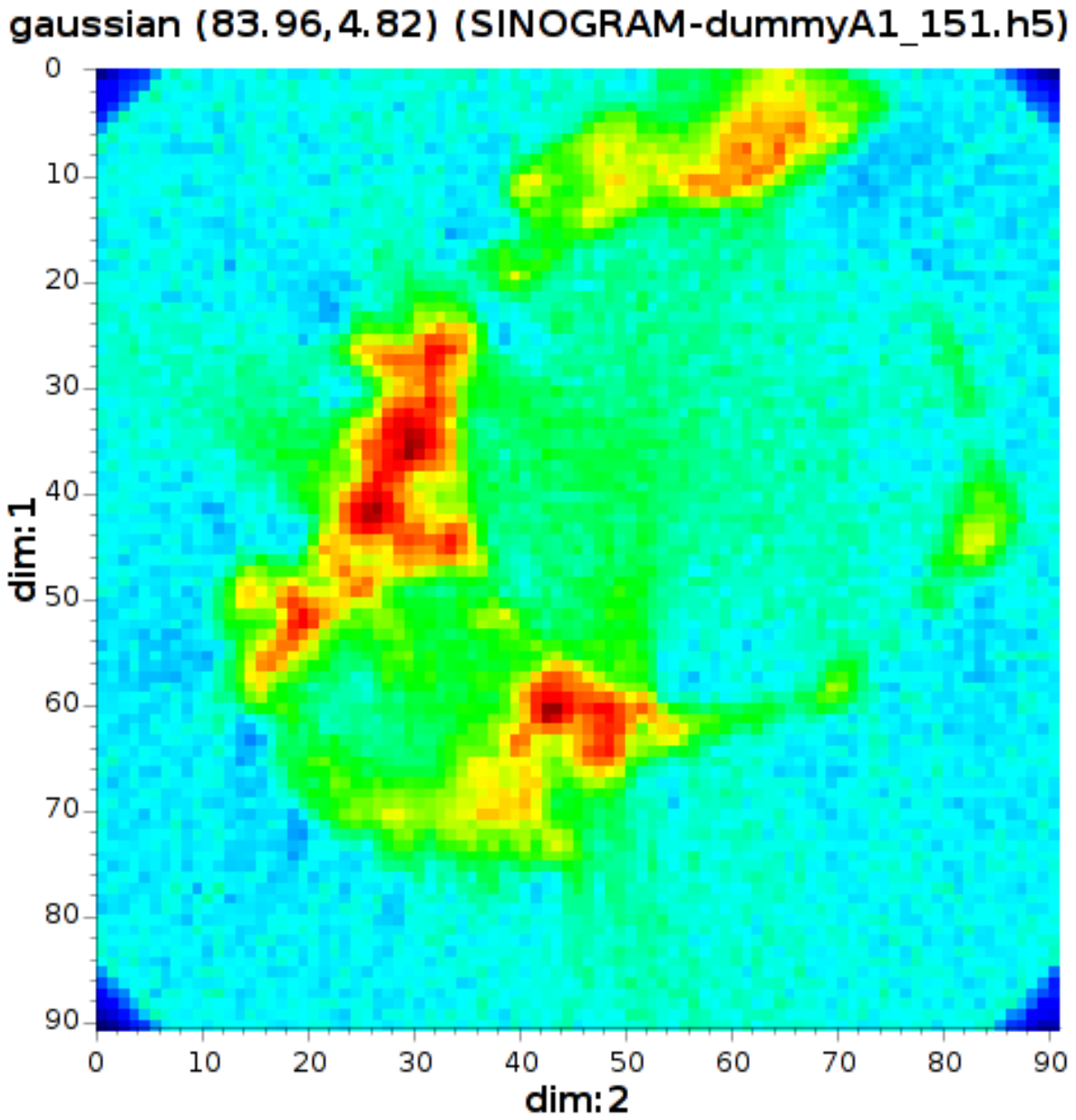


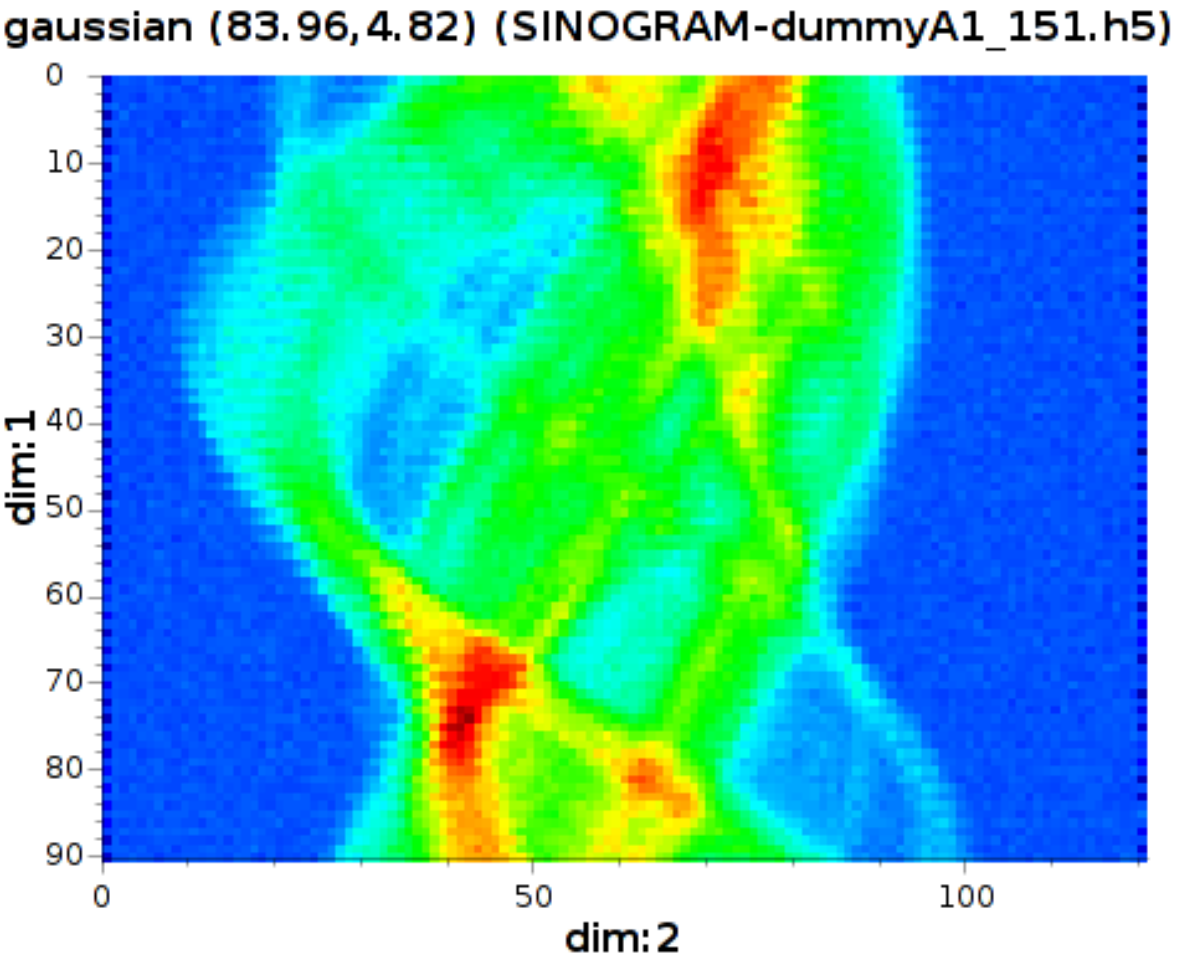
NTNU
Innovation and Creativity

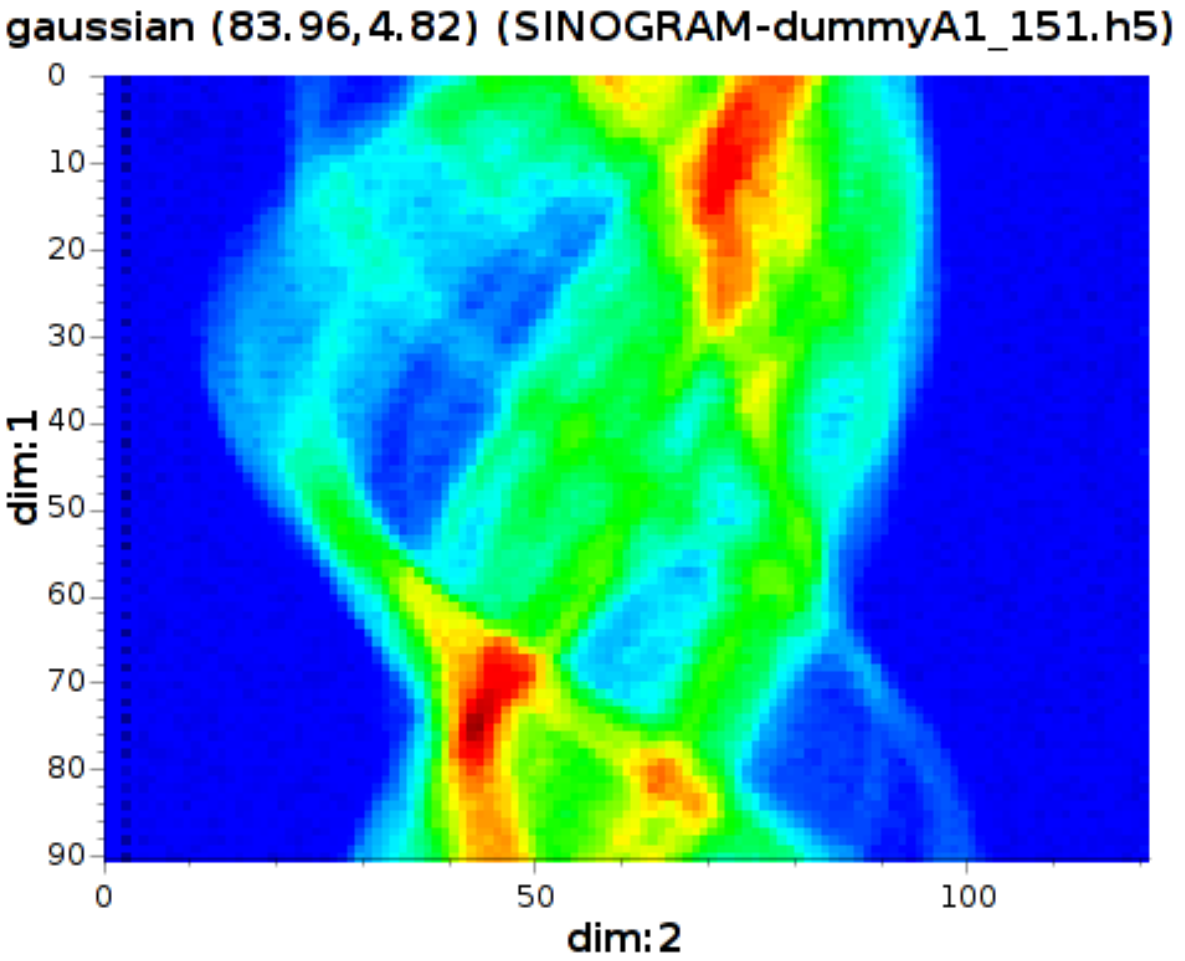
Results

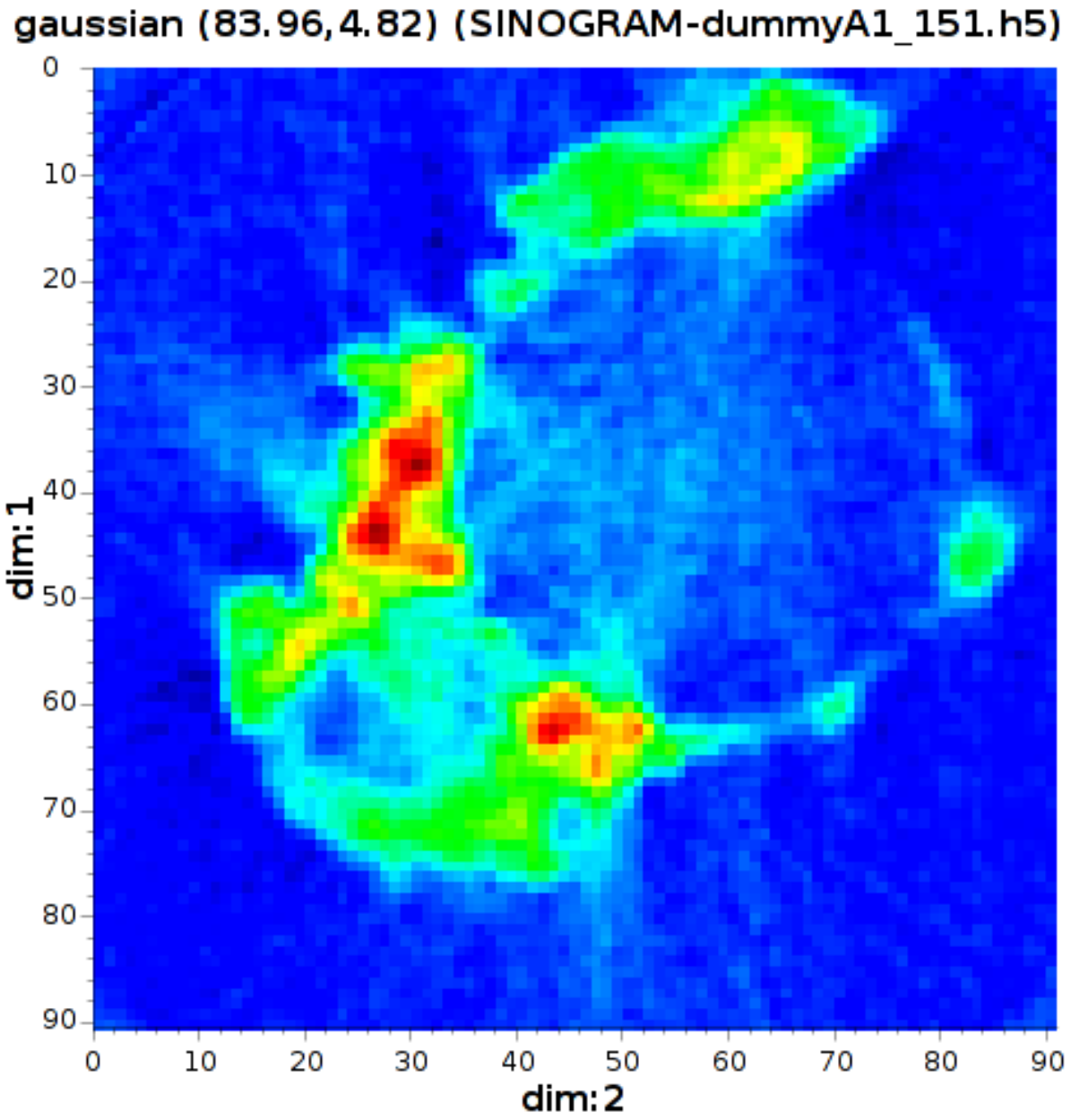
We will see the sinograms, corrected sinograms and their respective reconstructions using filtered back projection.

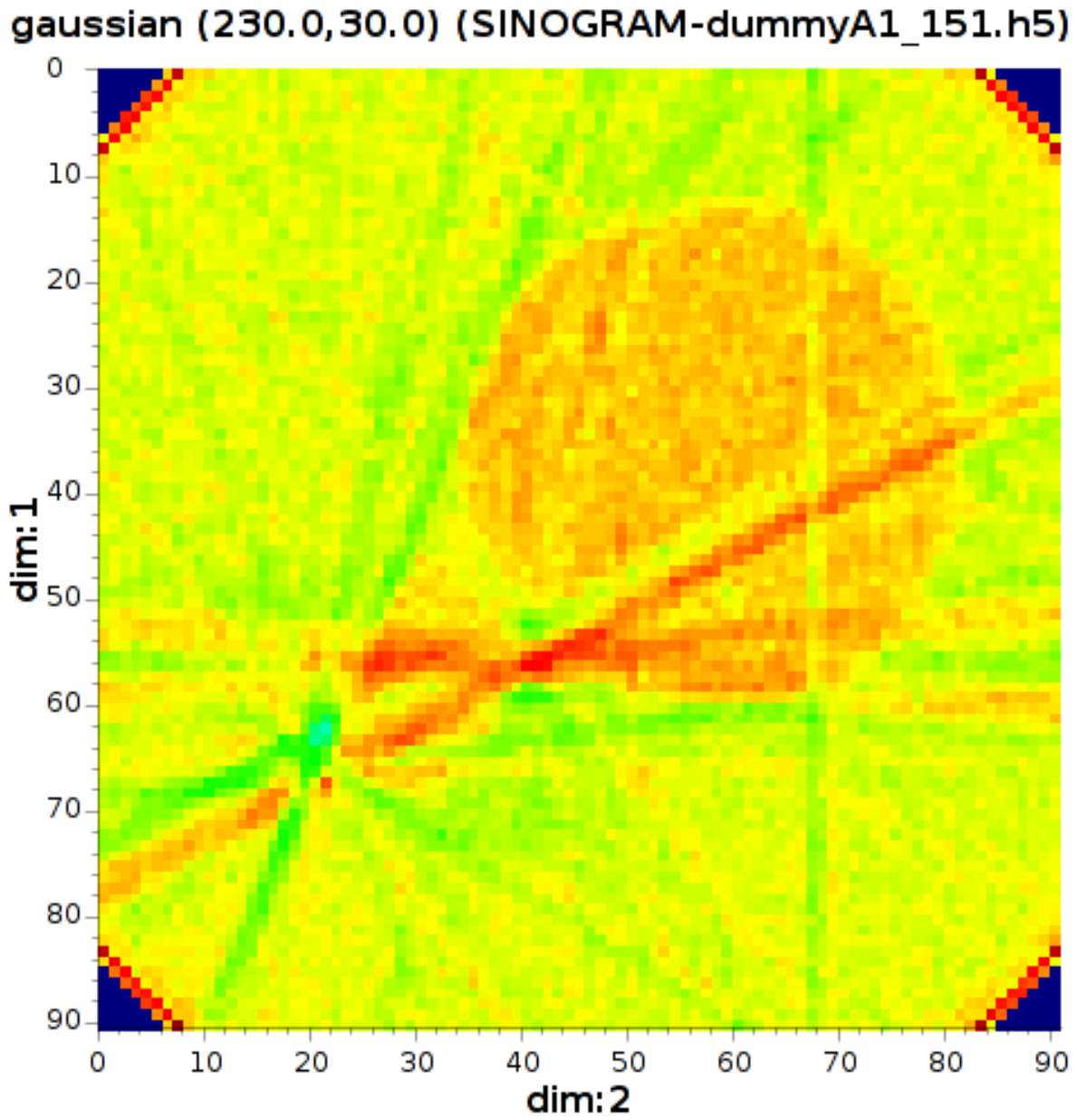
NTNU
Innovation and Creativity

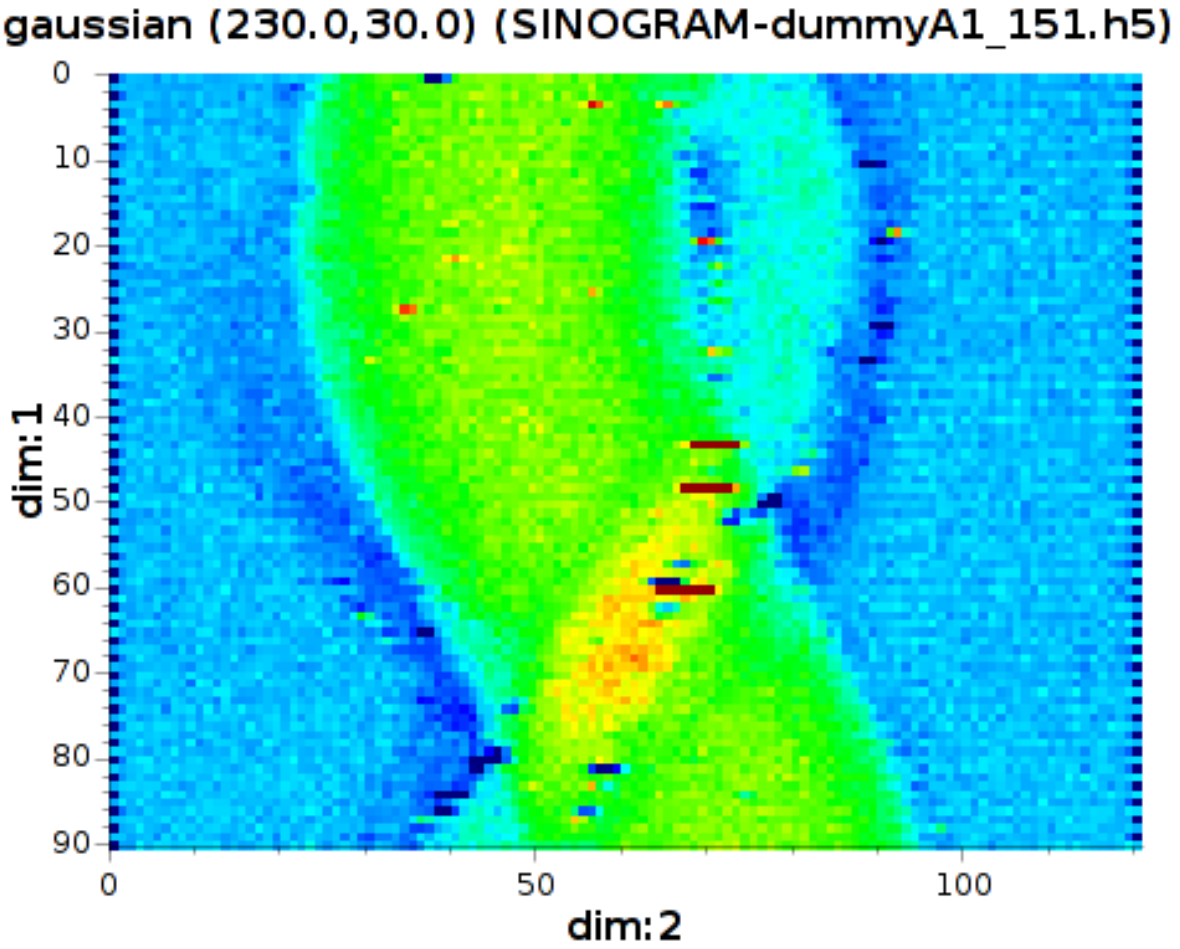

gaussian (83.96,4.82) (SINOGRAM-dummyA1_151.h5)
dim:1
dim:2
0
10
20
30
40
50
60
70
80
90

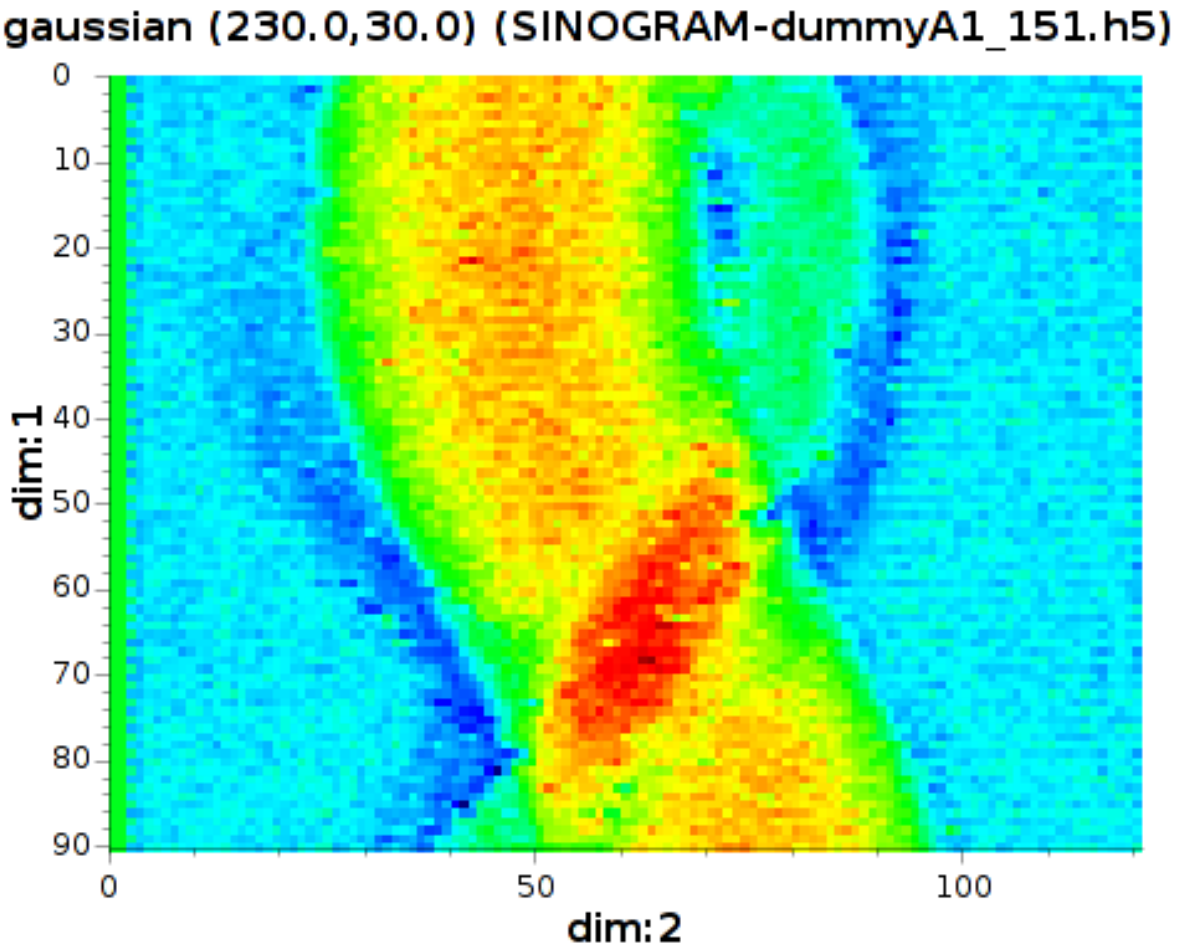

gaussian (83.96,4.82) (SINOGRAM-dummyA1_151.h5)
0
10
20
30
40
50
60
70
80
90
dim:1
0
50
100
dim:2

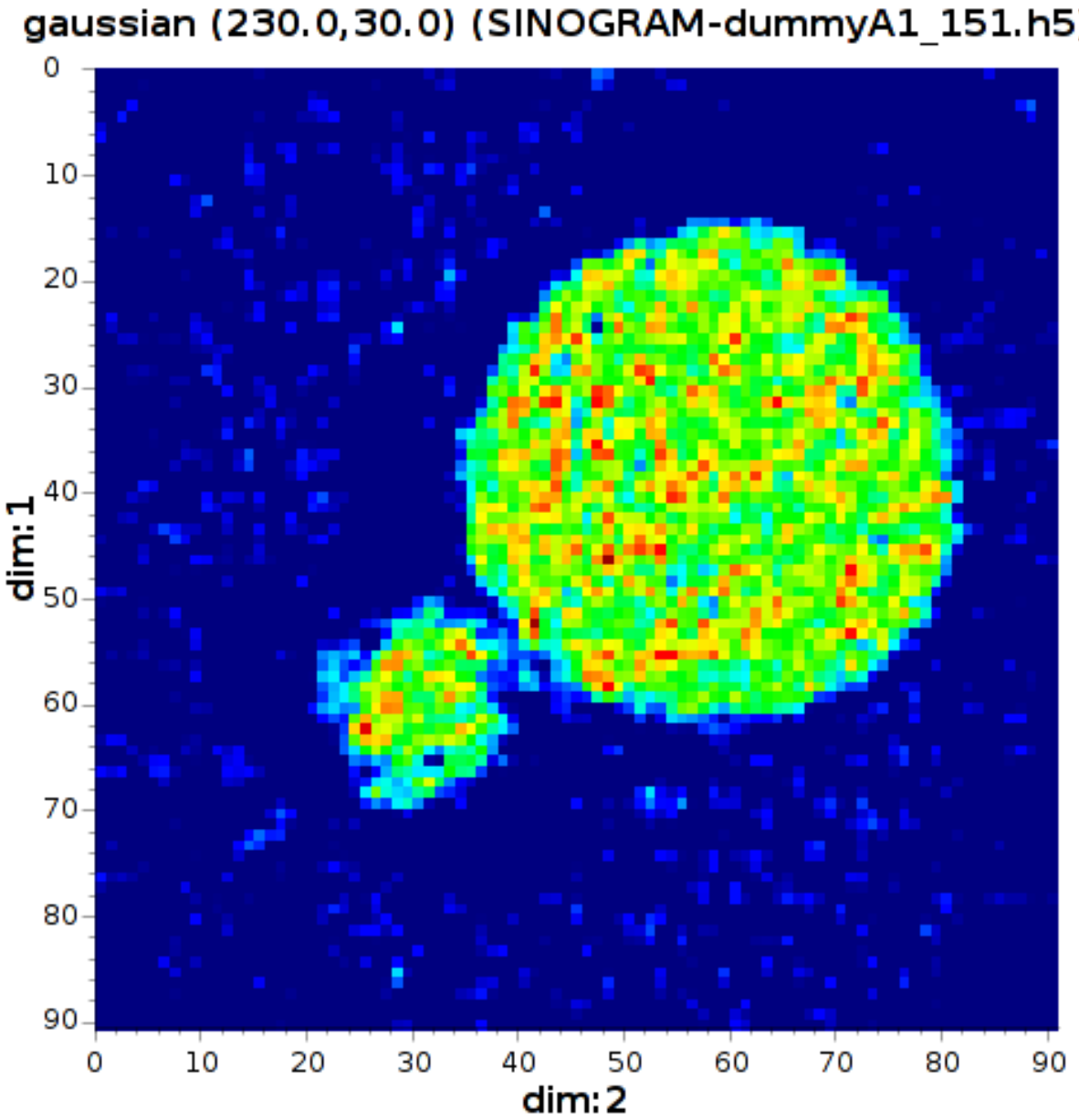

gaussian (83.96,4.82) (SINOGRAM-dummyA1_151.h5)
0
10
20
30
40
50
60
70
80
90
dim:1
0
50
100
dim:2


gaussian (83.96,4.82) (SINOGRAM-dummyA1_151.h5)
dim:1
dim:2
0
10
20
30
40
50
60
70
80
90


gaussian (230.0,30.0) (SINOGRAM-dummyA1_151.h5)
dim:1
dim:2
0
10
20
30
40
50
60
70
80
90


gaussian (230.0,30.0) (SINOGRAM-dummyA1_151.h5)
0
10
20
30
40
50
60
70
80
90
dim:1
0
50
100
dim:2


gaussian (230.0,30.0) (SINOGRAM-dummyA1_151.h5)
0
10
20
30
40
50
60
70
80
90
dim:1
0
50
100
dim:2


gaussian (230.0,30.0) (SINOGRAM-dummyA1_151.h5)
dim:1
dim:2
0
10
20
30
40
50
60
70
80
90

Issues / Further work

- Easy to use?
  - Not everyone comfortable working from the command line.
  - Keep the number of options down, or hidden for power-users
  - Investigate closer integration with DAWN.
  - Beamlines could provide presets for detector calibration and batch scripts.
- More applications. Use beyond basic data reduction and Diffraction Tomography.
- Extend test coverage beyond

NTNU
Innovation and Creativity

The biggest problem of them all!

We need to call the module something.
Xraylib is already taken.

Suggestions are welcome. . .

NTNU
Innovation and Creativity

Questions?

Live demonstration of results in DAWN.

NTNU
Innovation and Creativity