



**EUROPEAN
SPALLATION
SOURCE**



DMSC STAP

Updates from DRAM
(Data Reduction, Analysis and Modelling)

TORBEN NIELSEN

April 2026

DRAM

Data Reduction, Analysis and Modelling



Data reduction

- **scipp** will be used for all instruments
- Possibly in combination with other software for NMX & Imaging
- Are looking for partners

Data analysis

- **easyScience** for powder, sxtal & reflectometry
 - EasyDynamics for QENS
 - EasyImaging for TOF imaging
 - (Always) In combination with other libraries (cal. engines)
- **SasView** for SANS
- **(Py)MuhRec** for Tomography.
- **PACE** for spectroscopy

Data modelling

- **McStas** for instrument simulations
- Now also with Python API **McStasScript**
- **NCrystal**

DRAM

Data Reduction, Analysis and Modelling - Staff



Simon Heybrock



Neil Vaytet



Jan-Lukas Wynen



Sunyoung Yoo



Johannes Kasimir



Mridul Seth



Piotr Rozyczko



Andrew Sazonov



Henrik Jacobsen



Christian Vedel



Ales Kutsepau*



Peter Willendrup



Mads Bertelsen



Thomas Kittlemann



Damian Rodriguez**

- 3 teams (15+ persons)
- 1. Data Reduction (scipp)
- 2. Data Analysis (SasView, SpinW, EasyScience, external collaborations)
- 3. Modelling (McStas++, pan-learning.org, Detector Group)

*/** Post. Doc. / Lund

Scope

The DRAM group is responsible for providing the data reduction, analysis and modelling soft-ware for all instruments at ESS.

1

Data Reduction / General updates



Scipp Data reduction - parallel tracks

Overall focus on two tracks during the last 12 months:

1. Focused on data reduction workflow (with input from CIS/IDS)

2. Focused on deployment & operational readiness



1. Data reduction workflow

Main focus for data reduction group - Business as usual

Scipp – Updates 2 (from STAP April 2024)

New instrument specific packages (breaking ess pacakge apart)

- Sciline
- ESSreduce
- ESSsans
- ESSnmx
- ESSdiffraction
- ESSreflectometry
- ESSpolarization
- ESSspectroscopy*
- ESSimaging*

essdiffraction

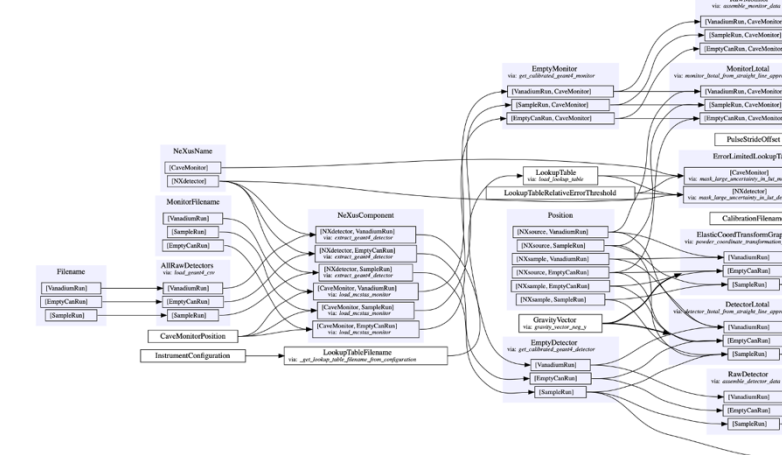
Section Navigation

- POWGEN data reduction
- On this page
- Introduction
- Define reduction parameters
- Create pipeline using Sciline
- Use the pipeline
- Compute final result
- Save reduced data to file
- Compute intermediate results
- Group by scattering angle

essreflectometry

Section Navigation

- Divergent data reduction for Amor
- On this page
- Make a (λ, θ) map
- [Save data](#)



essnmx

Section Navigation

- Workflow
- On this page
- Build Pipeline (Collect Parameters and Providers)
- Build Workflow
- Compute Desired Types
- Export Results

esssans

Section Navigation


- Installation
- LoKI
- Direct beam iterations for LoKI
- ISIS instruments
- Common tools
- On this page

2. Deployment & operational readiness - VISA



Reproducible environments and performance benchmarking



- A) VISA is the user interface for data and compute.
- A) We build and distribute reproducible conda environments which installs DMSC developed and third-party software that will be used for reduction, analysis and visualization.
- A) VISA instances are **prepopulated** with Code  Shelf notebooks and scripts.
- B) VISA and the external scientific user workflow was **tested successfully** multiple times at various schools like DMSC Summer School, Winter School and HERCULES.
- C) We also found **performance issues** with VISA, which boils down to old hardware being used by the current OpenStack backend. This is in process of being upgraded.
- C) We still need to make sure that the **UX is improved and remove all friction** in terms of latency and performance on VISA.

CodeShelf

Repository for production notebooks and scripts

- Group on DMSC GitLab (git.esss.dk/dram/code-shelf)
- Git repos for **production notebooks and scripts**
- Deployed to VISA
- Automatically tested in CI
 - Partial solution, still need to figure this out
- Scientists can contribute











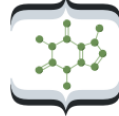







Code Shelf


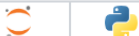
Users	Maintainers	Reference
Click here if you want to get code from Code Shelf or add new code.	Guide for maintaining Code Shelf and its support package Bibliotekar.	API reference for Bibliotekar.

Code Shelves

Instruments

BEER  	BIFROST  	DREAM  	ESTIA  
LoKI  	NMX  	ODIN  	Test Beamline  

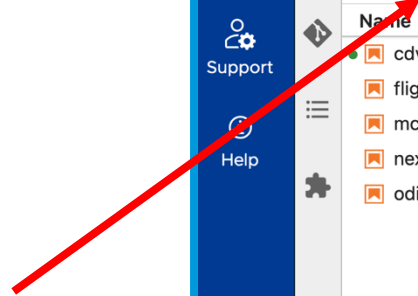
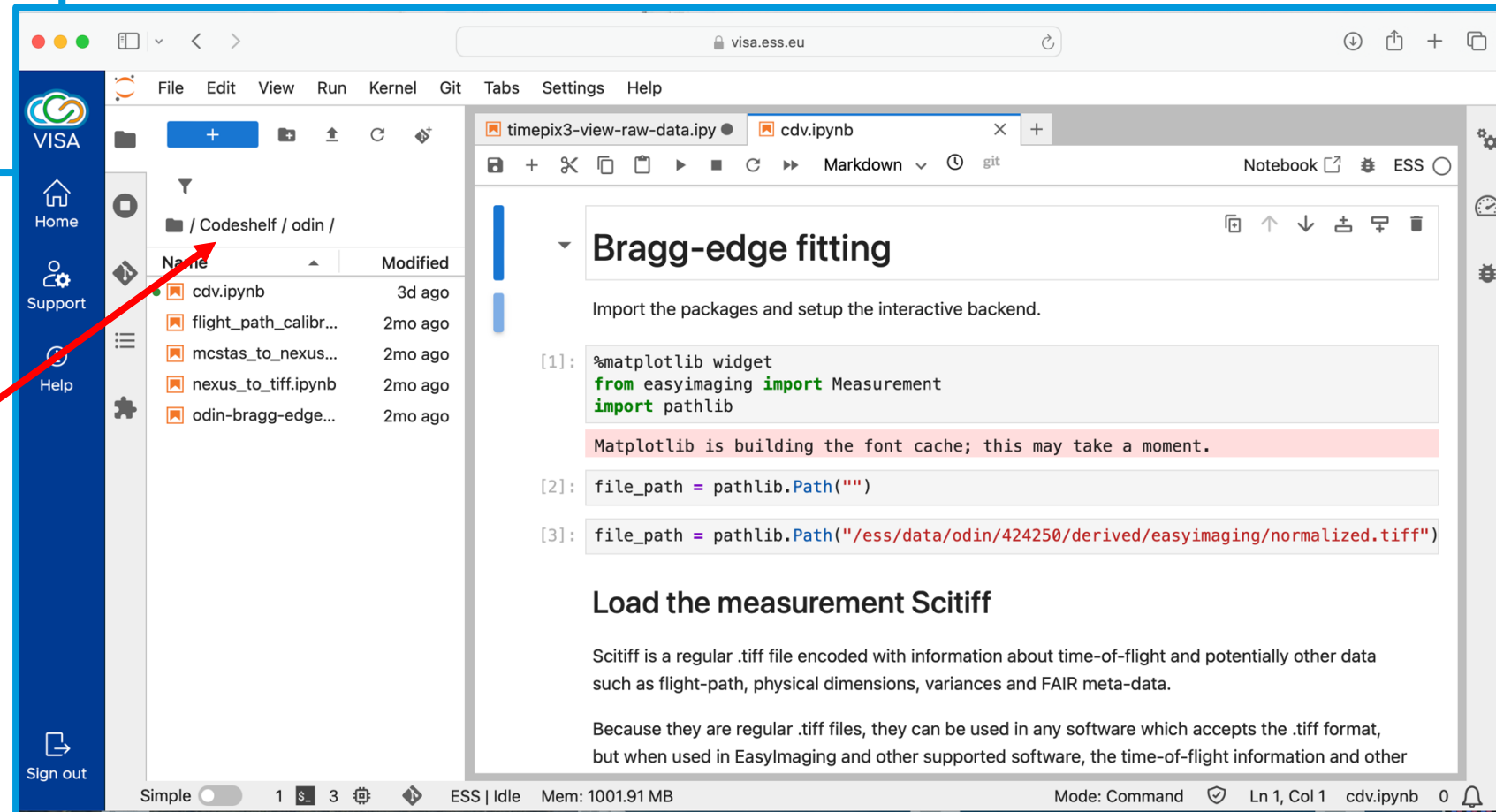
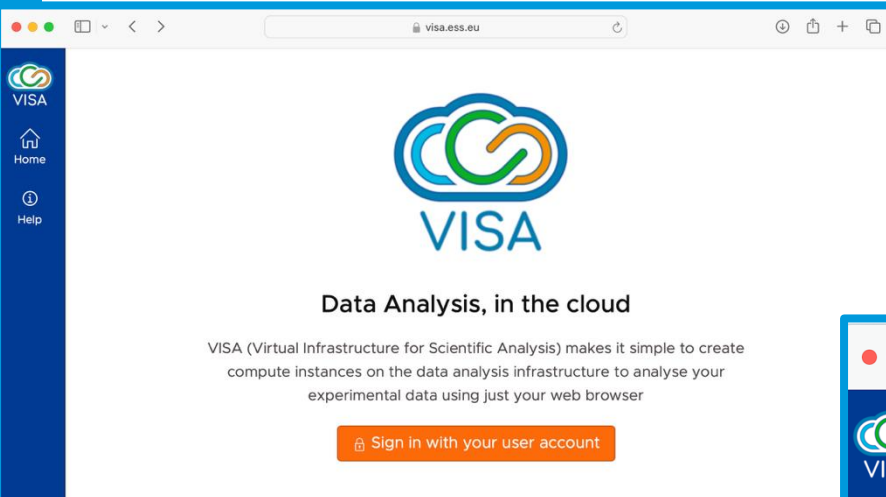
Miscellaneous

SciCat  

VISA

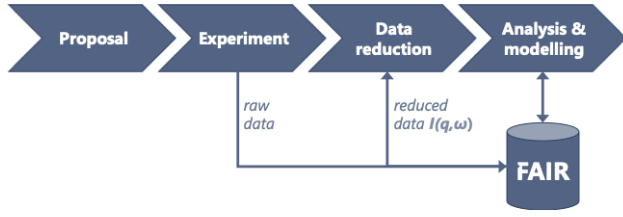


- Scipp docs on <https://scipp.github.io> are numpy / scipy like
- We also need something more "plug & play" like
- And something that can be linked to data
- `file_path = pathlib.Path("/ess/data/odin/424250/derived/easyimaging/normalized.tiff")`



DMSC

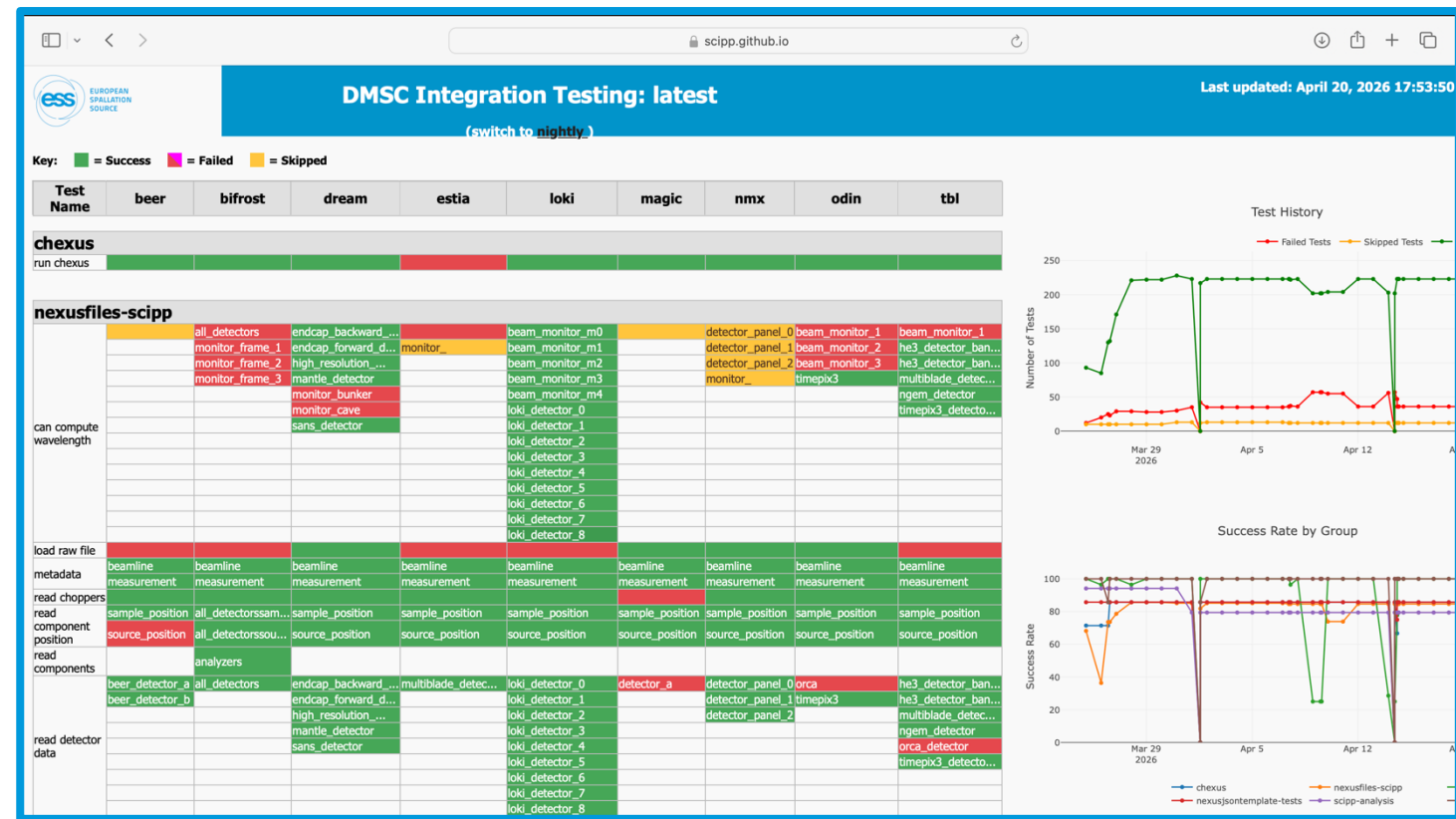
Integration tests – Spring 2026






Collaborative spirit across SIMS; ECDC & DMSC teams

Mitigates major risk related to changes upstream


- Dashboard for developers
- Check status of new changes and features
- Weekly meetings
- Daily check in on Slack
- N.B. jobs are ordered alphabetically
 - "compute tof" before "load raw data"

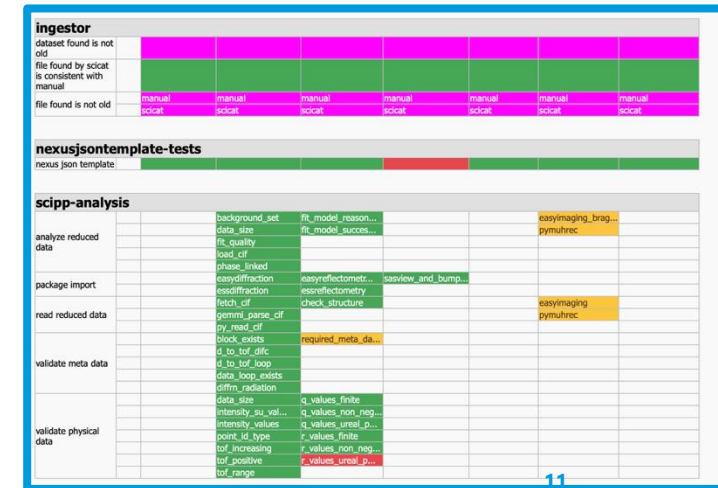


 **Neil Vaytet** 5:41 PM
And now we also have Magic 
<https://scipp.github.io/nightly-dashboard/latest/index.html>

 **Neil Vaytet** 3:43 PM
We now have beer tests running: <https://scipp.github.io/nightly-dashboard/latest/index.html>

1 reply

 **Jan-Lukas Wynen** 2 minutes ago
This failure is on us. Fixing...

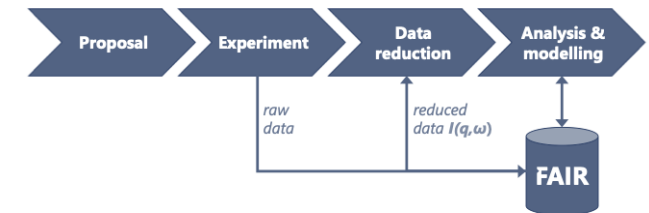


DMSC-wide integration nightly testing



Expanding test coverage to more instruments and packages

- Files are continuously written by the ECDC group for 7(9) instruments in the Continuous Data (CODA) proposal.

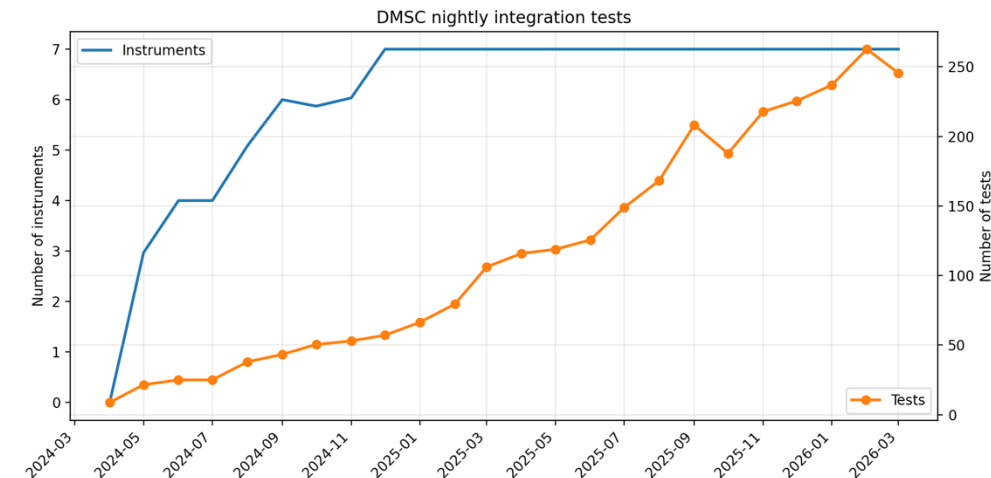


- Every night, we verify that:

- File was correctly ingested into Scicat
- Reduction software can read the metadata, geometry, event data and compute wavelength
- Analysis software can read reduced data and perform some routine analysis

- Number of tests has doubled over the past 12 months

- These nightly tests have proved invaluable for detector tests at instruments**



DMSC-wide integration nightly testing



Expanding test coverage to more instruments and packages

	Bifrost	Dream	Estia	Loki	NMX	Odin	TBL
Read geometry	✓	✓	✓	✓	✓	✓	✓
Read events	✓	✓	✓	✓	✓	✓	✓
Compute wavelength	✓	✓	✓	✓	✗	✓	✓
Scicat ingested	✓	✓	✓	✓	✓	✓	✓
Run analysis	✗	✓	✓	✗	✗	✗	✗

: Covered by tests

: Added in the last 12 months

: Not yet covered

DMSC Schools

Summer and Winter schools



- **2025** DMSC Summer School (last year)
 - New track (powder diffraction) added
- **2026** DMSC Winter School (first time)
 - **Internal Employees only**
 - Data Reduction courses more in details
 - Most materials were reusable from summer school.
 - 2 x 2 days
- **2026** DMSC Summer School (in preparation)
 - Expected little cost this year (Not planning any new tracks)

Reduction with scipp

Introduction to Scipp
 Coordinate transformations
 Introduction to Scilime
 Powder diffraction data reduction
 QENS data reduction
 SANS data reduction

It is similar to our own graph but involves additional intermediate results and can also be used to compute the scattering angle `two_theta`.

We can also add functions to compute the many more coordinates such as *d*-spacing, *Q*, or *hkl* indices: (If you don't know the syntax, simple read `(**a, **b)` as merging the two dicts `a` and `b` into a single dict.)

```
graph = {
    **scn.conversion.graph.beamLine.beamLine(scatter=True),
    **scn.conversion.graph.tof.elastic(start="tof"),
}
sc.show_graph(graph)
```

Histogram the data in d-spacing

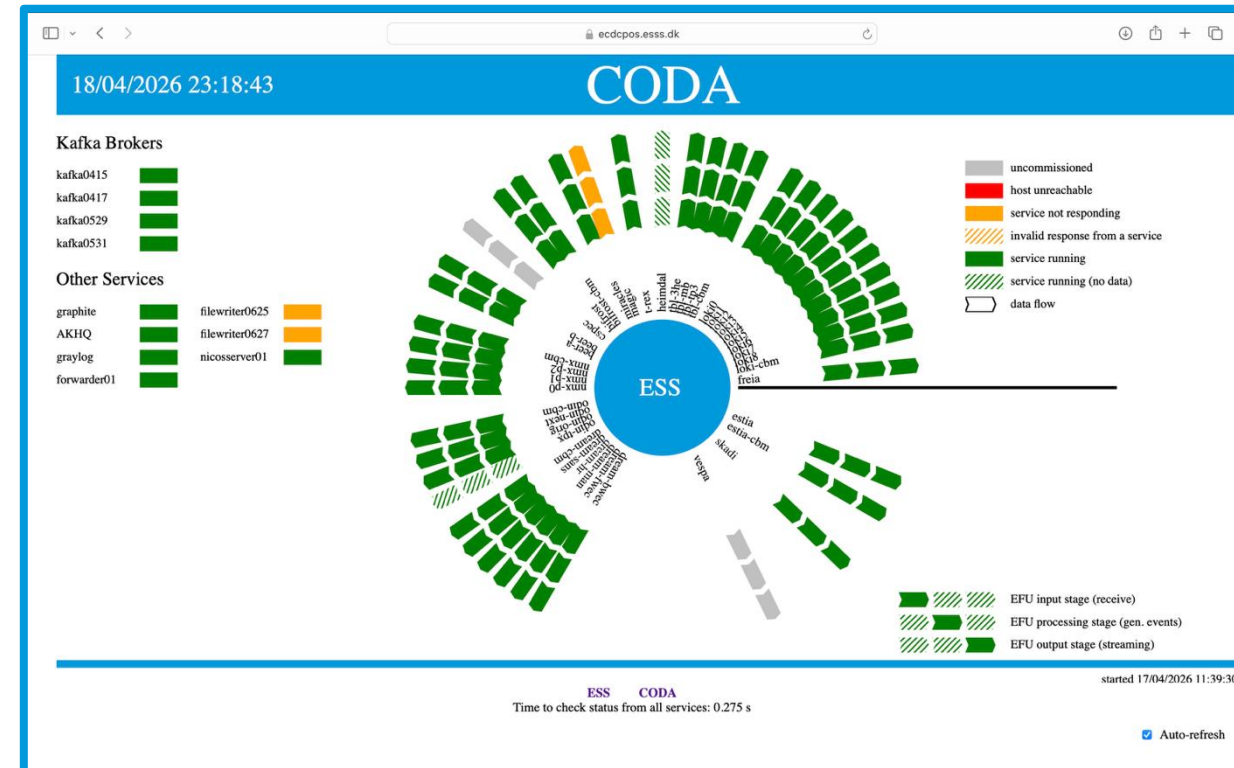
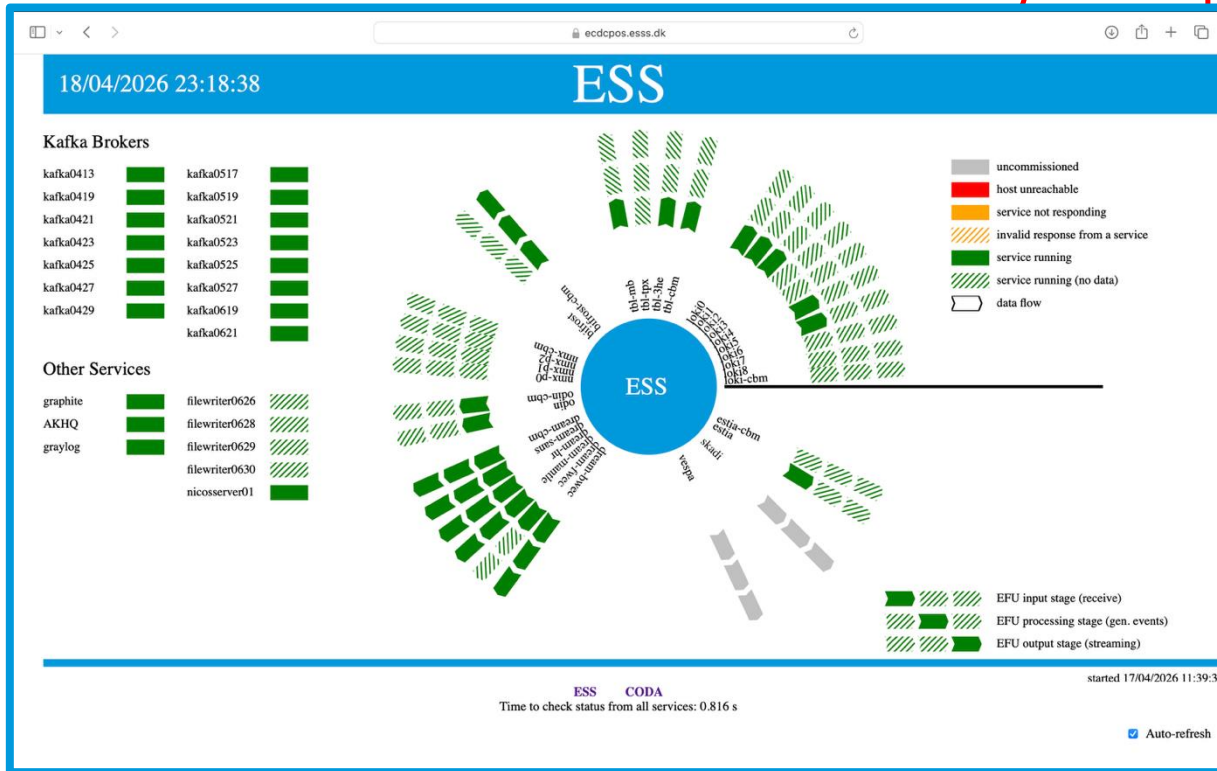
The final step in processing the sample run is to histogram the data into *d* bins.

```
dbins = sc.linspace("dspacing", 0.7, 2.5, 901, unit="angstrom")
ntheta = 128
si_dspacing_hist = si_dspacing_hist(two_theta=ntheta, dspacing=dbins)
si_dspacing_hist.plot(norm="log", vmin=1.0e-3)
```

```
si_dspacing_hist.sum("two_theta").plot()
```



ESS/ECDC infrastructure - Production & CODA --> Linking Scipp to infrastructure



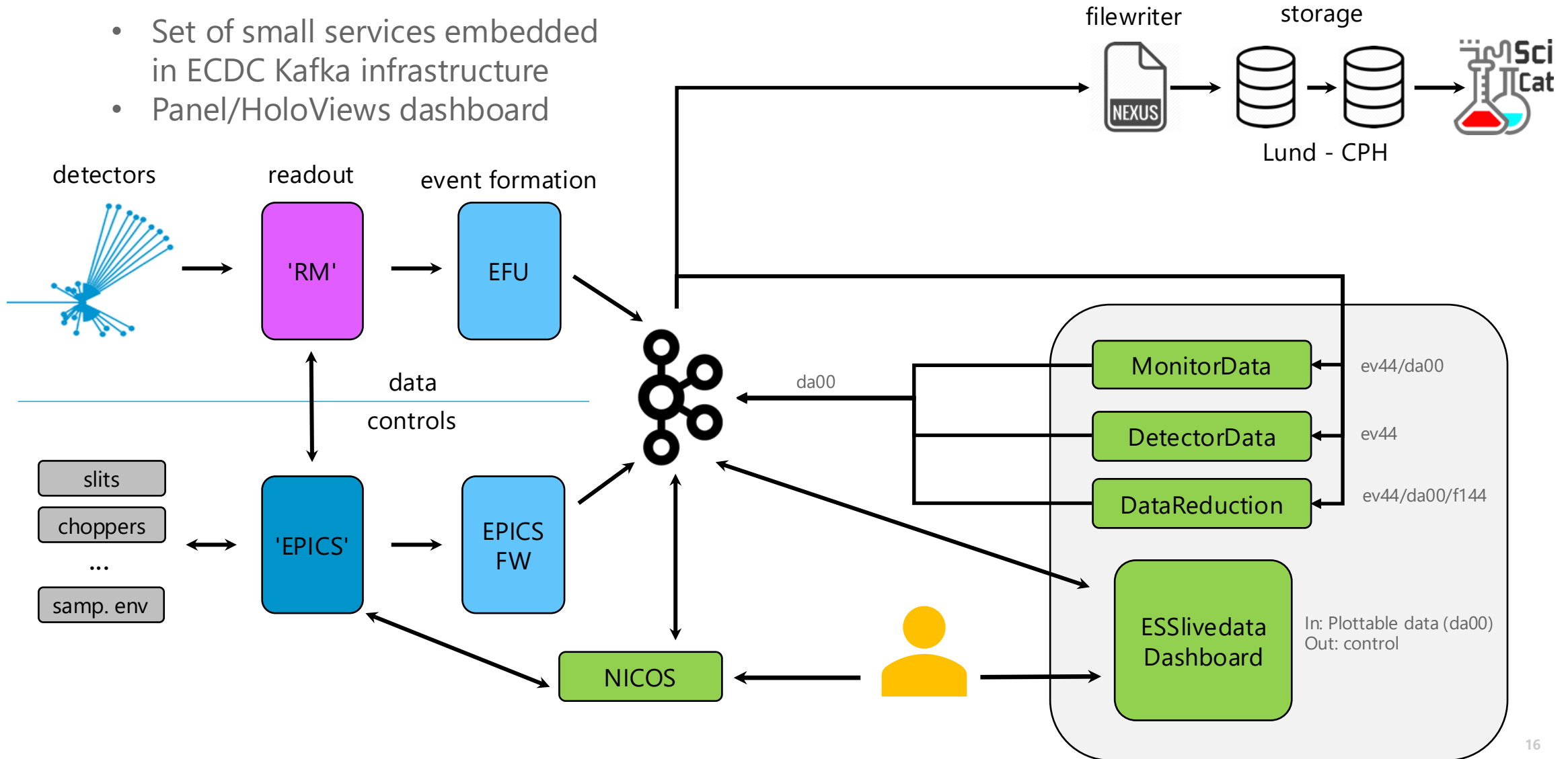
ECDC status pages: <https://ecdcpos.esss.dk/ess>

ESSlivedata (previously Beamtime) – 1/4



Live data reduction and visualization during beam time

- Set of small services embedded in ECDC Kafka infrastructure
- Panel/HoloViews dashboard

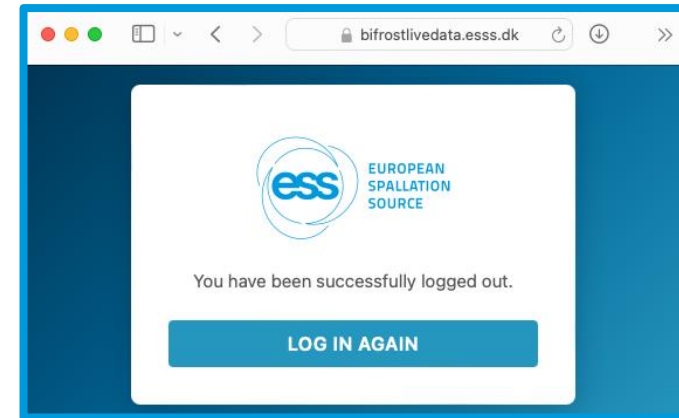
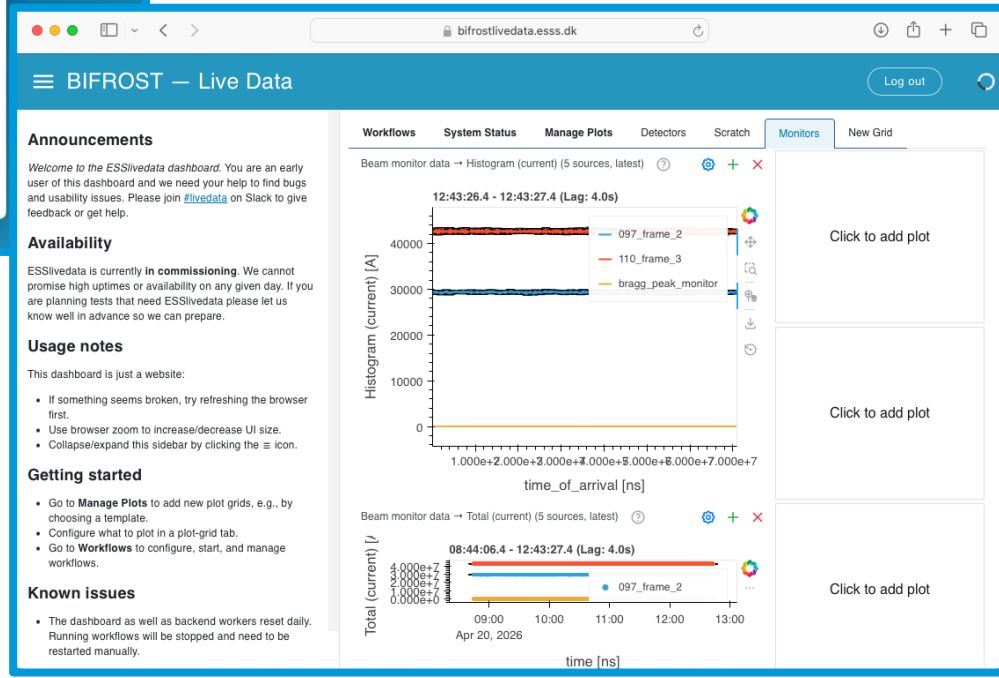
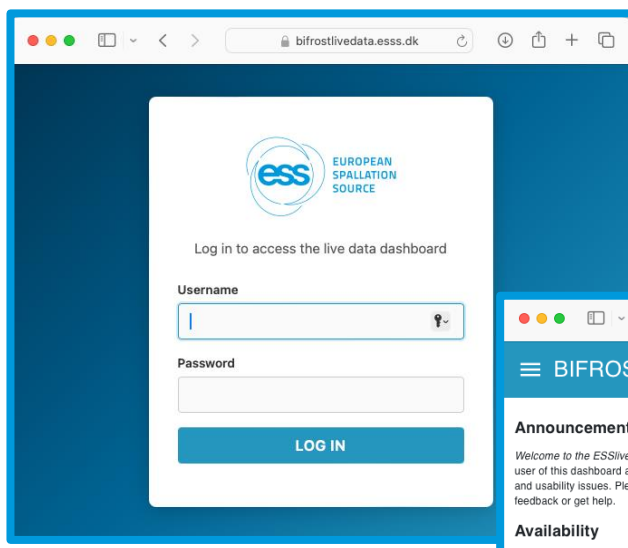




ESSlivedata – 2/4

Where we are now

- Working backend services + interactive dashboard
- Experimental deployments at 1+6 beamlines (BIFROST, LOKI, ESTIA, DREAM, NMX, ODIN, TBL) during detector commissioning
- **Each deployment uncovered issues, none were showstoppers** -> system is maturing through real world testing
- Generally positive feedback from Instrument Scientists



ESSlivedata – 3/4



What we learned / what's hard

- Performance and multi-session app with Panel/HoloViews/Bokeh
 - Significant learning curve
 - Contributing upstream performance improvements
- Deployment and integration with DST+ECDC systems
- **Deployment expectations:** need clearer communication about readiness levels ("commissioning" vs "production")
- Each instrument has **different needs:** balancing generality vs. instrument-specific customization

The screenshot shows the 'LOKI – Live Data' interface. At the top, there is a navigation bar with a hamburger menu, the title 'LOKI – Live Data', and a 'Log out' button. Below the navigation bar, there are tabs for 'Workflows', 'System Status', 'Manage Plots' (which is active), 'Overview', 'Tubes', and 'Detectors'. The 'Manage Plots' section contains a form for configuring a grid. The form has fields for 'Grid Title' (set to 'Overview'), 'Rows' (set to 2), and 'Columns' (set to 3). There are 'Save Changes' and 'Save as Copy' buttons. To the right of the form, a preview shows a 2x3 grid of plots. The top row contains 'Tube View Total (current)' and 'Timeseries data Delta'. The bottom row contains 'Beam monitor data Histogram (cumulative)', 'Beam monitor data Total (current)', and 'Timeseries data Delta'. Below the grid, there is a list of active plots: 'Overview', 'Tubes', and 'Detectors'. Each plot has a dropdown arrow, a pencil icon for editing, an eye icon for visibility, a download icon, and a close icon.

Customizable grids of plots

ESSlivedata – 4/4

NICOS and hutch integration

- “Done”
 - Adopted shared Kafka message schemas and documented interfaces
 - Experimental display of results and controls in NICOS
- **But actually...**
 - Need lots more work to make it work smoothly with good UX
 - Decide what we and users actually need
 - Where is the scope boundary between ESSlivedata and NICOS?
 - How can we avoid confusion for users who need to interact with both?



The screenshot shows the LOKI Live Data dashboard. At the top, there is a navigation bar with a hamburger menu, the text 'LOKI – Live Data', a 'Log out' button, and a refresh icon. Below the navigation bar, there are tabs for 'Workflows', 'System Status' (which is active), 'Manage Plots', 'Overview', 'Tubes', and 'Detectors'. The main content area is divided into three sections: 'Dashboard Sessions', 'Backend Workers', and 'Notifications'. The 'Dashboard Sessions' section shows one active session for 'simon (xW73WcXhMoQf)' with a status of 'You' and a last heartbeat of '2 seconds ago'. The 'Backend Workers' section shows four workers, all in a 'RUNNING' state. The 'Notifications' section shows three success messages: 'Started 5/5 jobs for 'Beam monitor data'', 'Started 9/9 jobs for 'Tube View'', and 'Started 9/9 jobs for 'Detector XY Projection''. Two handwritten annotations are present: 'Multi-session support (hutch computer + laptop)' is written diagonally over the session table, and 'Compute runs in backends with (limited) scalability' is written diagonally over the notifications section.

Session	Status	Last Heartbeat
simon (xW73WcXhMoQf)	You	2 seconds ago

Namespace	Worker ID	Status	Uptime	Stats
loki:detector_data	c3610f80	RUNNING	Up: 1.6 hours	Jobs: 18 Msgs: 736.1k
loki:monitor_data	e71e9f44	RUNNING	Up: 1.6 hours	Jobs: 5 Msgs: 77.6k
loki:data_reduction	c070c19b	RUNNING	Up: 1.6 hours	Jobs: 0 Msgs: 814.0k
loki:timeseries	47d1341b	RUNNING	Up: 1.6 hours	Jobs: 0 Msgs: 596

Time	Status	Message
08:20:09	SUCCESS	Started 5/5 jobs for 'Beam monitor data'
08:20:07	SUCCESS	Started 9/9 jobs for 'Tube View'
08:19:58	SUCCESS	Started 9/9 jobs for 'Detector XY Projection'

1

Data Reduction / Instrument updates

Readiness of data reduction for H.C.

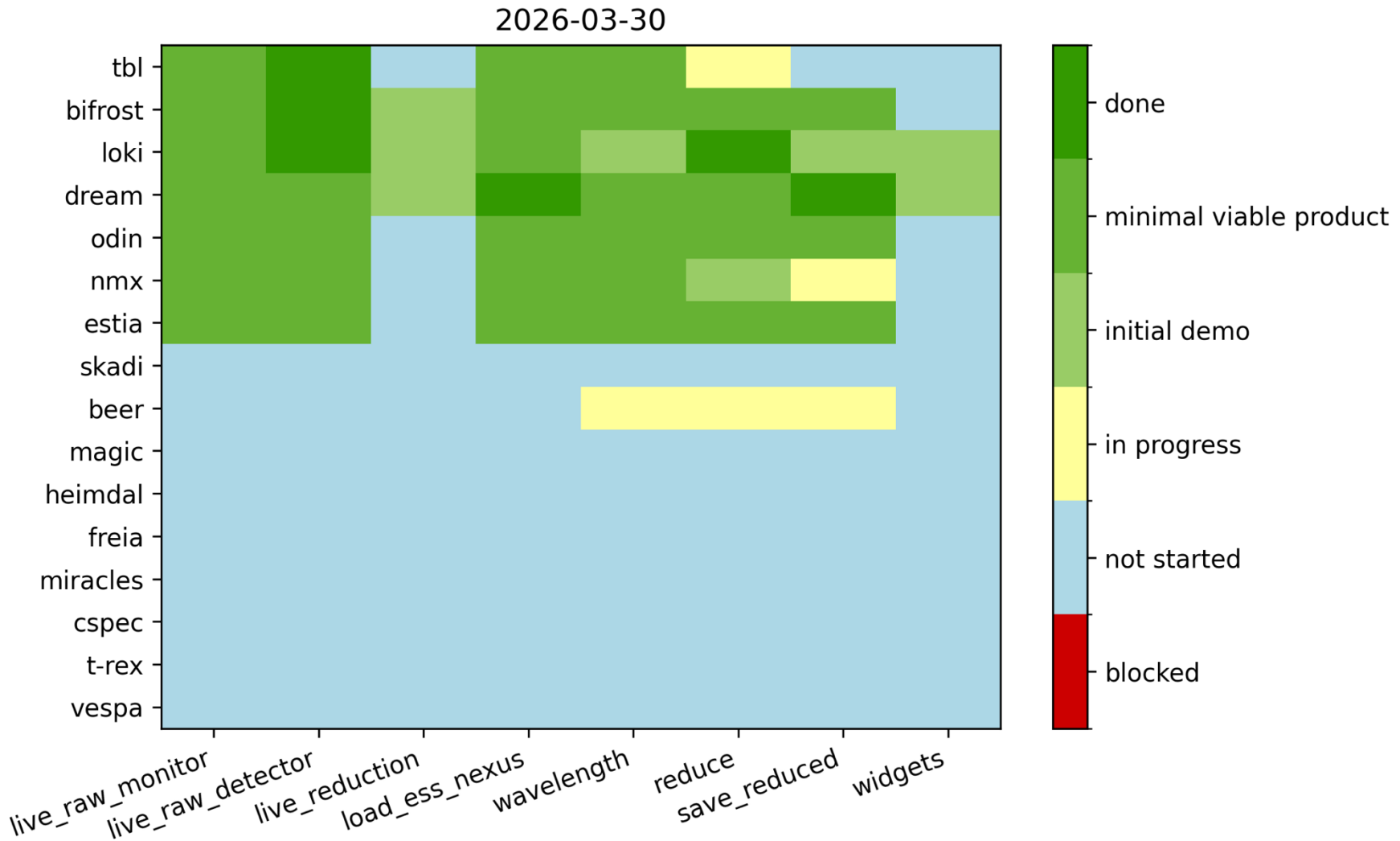


BIFROST		
ODIN		TOF fitting on ROI (analysis) still work in progress
LOKI		
DREAM		
NMX		
ESTIA		Needs entries in NeXUS for polarisation. CIS resigned
SKADI		Can reuse LOKI workflow but also needs polarisation
FREIA		CIS resigned
MAGIC		CIS started
BEER		Multiplexing demonstrated
MIRACLES		CIS started
HEIMDAL		Can reuse DREAM (powder) and MAGIC (Sxtal). CIS approved.
CSPEC		CIS started
TREX		CIS started
VESPA		



Instruments

Status matrix (approximate)



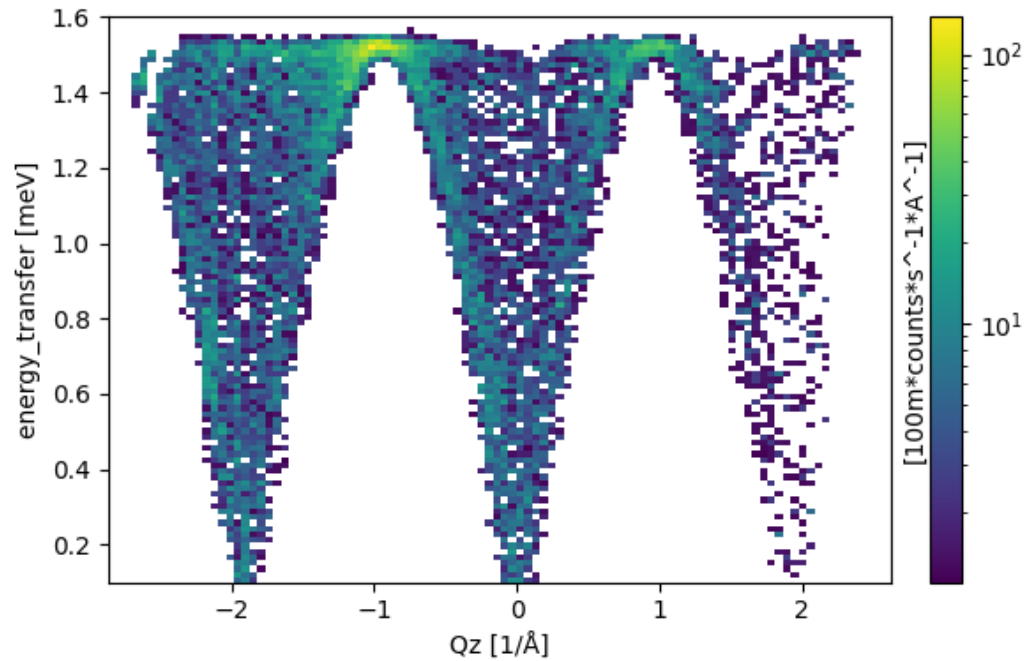
Instruments

BIFROST

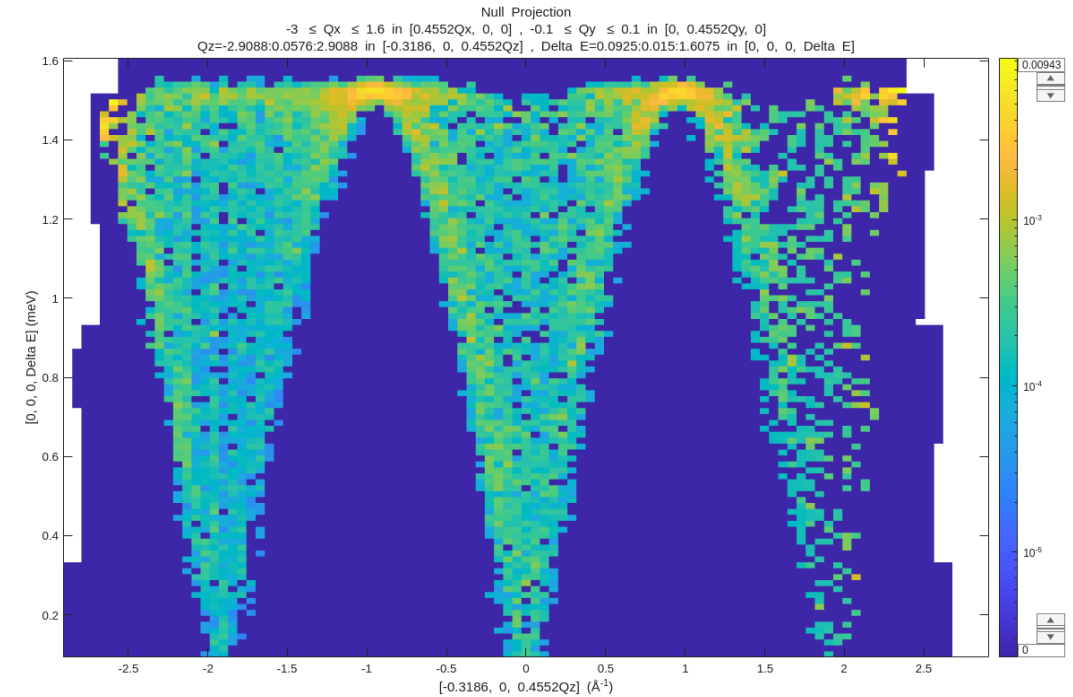


- Can save reduced data to SQW and load into HORACE

Scipp



HORACE



- Implemented a SXD workflow for Bragg peak monitor
- Working on support for moving detector

2

Modelling / General updates

Modelling Group / "McCode++"



Main projects & activities

■ Projects

- McStas/McXtrace
- McStas Union components [Simulation of complex systems]
- McStasScript [Python API]
- McStasToX [Python converter package]
- MCPL [Binary format]
- Ncrystal
- Simplebuild-dg code [Geant4-based simulation framework]

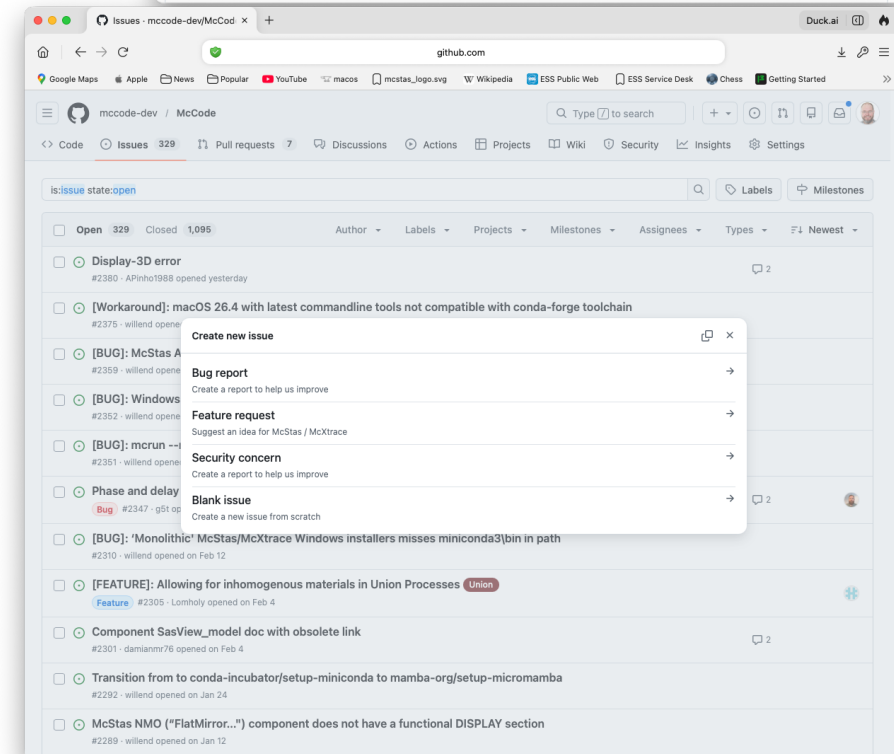
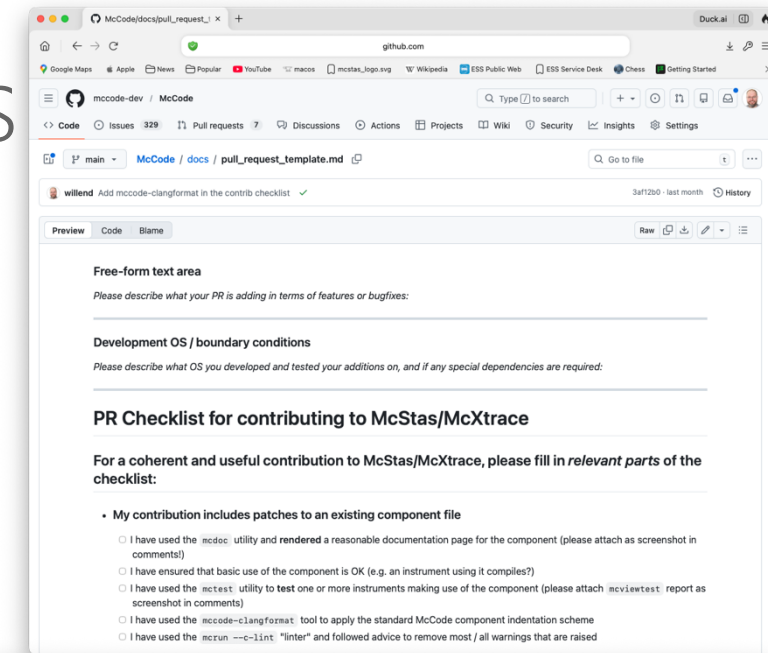
■ McCode++ support activities

- **Maintenance and develop new features**
- Support to CIS/IDS, ESS IS teams & McStas community
- McCode++ workshops
- DMSC Schools
- Papers
- Student projects
- New ESS instruments

McStas/McXtrace 3.5.x-3.6.x series

Infrastructure and Project

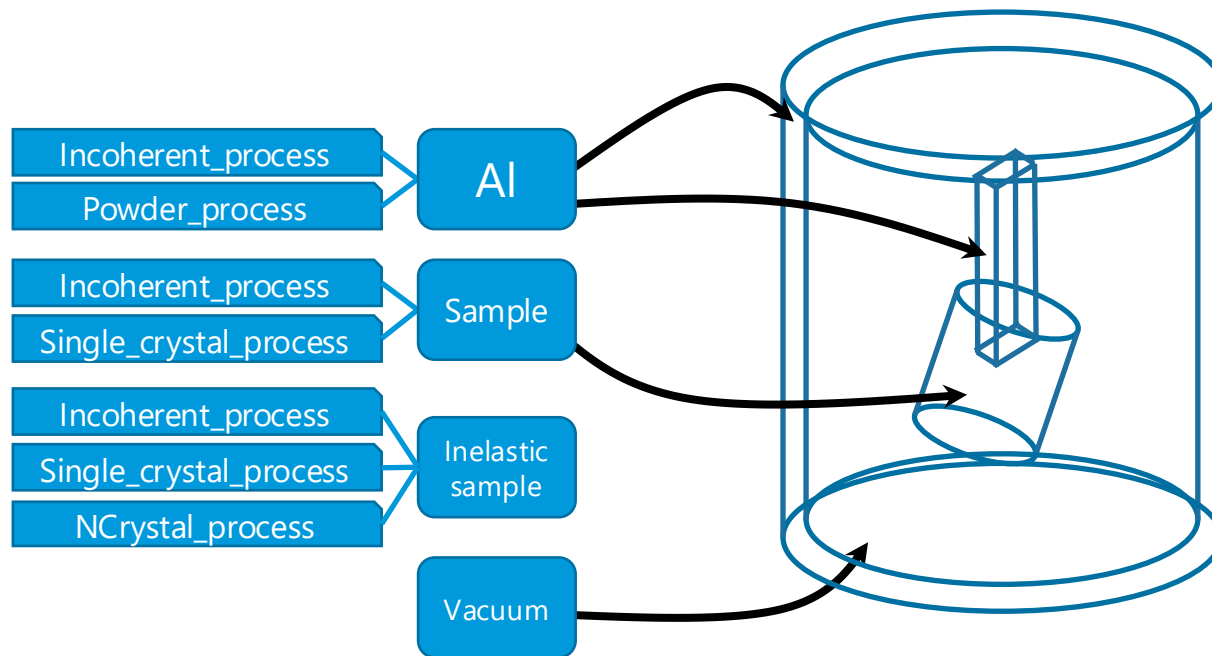
- Common release and versioning for McStas & McXtrace - **support for 2.x and 1.x now officially deprecated**
- We apply conda-forge for platform-independence
- We release often/rolling
 - At 3.5.18 in January 2025, 3.6.11 in March 2026
- Common McStas/McXtrace release cycle
 - Automated builds for McStas/McXtrace releases
- Focus on testing and reproducibility
 - CI (GitHub + conda) in place
 - PR-only, “production grade” main branch
 - **PR-template in place for improved quality external contributions**
 - Issue-templates in place more coherent error reporting



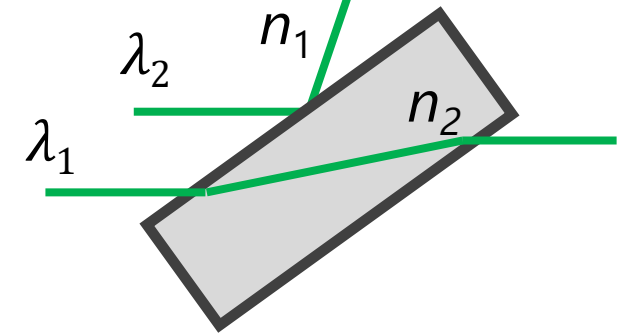
Union components

Overview

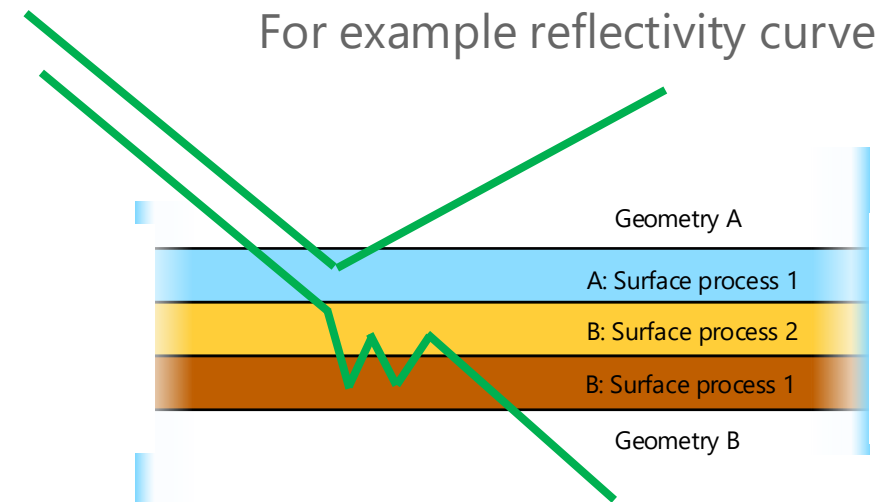
- Assemble materials from scattering processes
- Assemble geometry from basic shapes



This year refraction was added



As well as ability to attach surfaces to faces
For example reflectivity curve





McStasToX

Export McStas simulation data

- Reads McStas NeXus file
 - Pixel ID / position and time
 - Transformed to one coordinate system
- Can export Scipp object
- Plan is to support other output formats
- Eventually match instrument NeXus files

- Detectors can be simulated with Union
 - Ongoing: MAGIC, TREX, DREAM, SKADI, NMX
 - Include multiple scattering in detector
 - Events from these can be read with McStasToX

```
import mcstastox
with mcstastox.Read(file_path) as loaded_data:
    scipp_object = loaded_data.export_scipp_simple(source_name="source",
                                                sample_name="sample_position")

scipp_object
```

scipp.DataArray (1.40 MB)

► Dimensions: (events: 36699)

▼ Coordinates:

position	(events)	vector3	m	[-0.25051048 -0.00666667 1.7533107],	☰
sample_position	()	vector3	m	[0. 0. 2.]	☰
source_position	()	vector3	m	[0. 0. 0.]	☰
t	(events)	float64	s	0.001, 0.001, ..., 0.003, 0.001	☰

▼ Data:

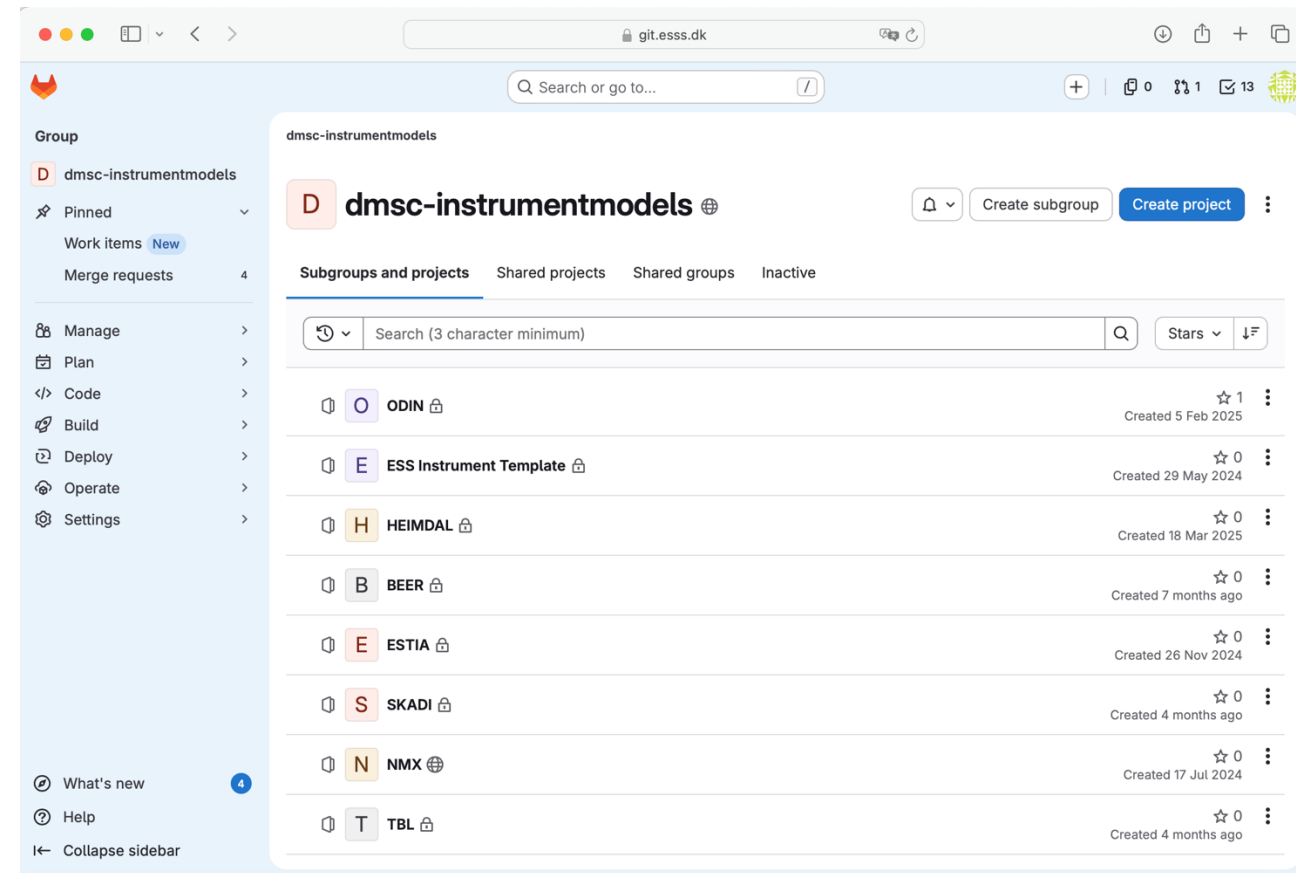
	(events)	float64	counts	7.479e-12, 9.837e-12, ..., 6.517e-12, 9.07	☰
--	----------	---------	--------	--	---

McStas support efforts 1/2

Internal ESS Instruments



- Simulation support for ESS instruments. On dmsc-instrumentmodels:
 - **NMX**
 - Single_crystal large-unit cell optimisations
 - [McStasToX](#), reduction workflow in Scipp
 - **ESTIA**
 - Ported McStas 3.x
 - Reduction workflow in Scipp
 - **ODIN**
 - [McStasToX](#), reduction workflow in Scipp
 - **HEIMDAL**
 - Samples added
 - Initial reduction workflow in Mantid
 - **BEER**
 - Initial reduction workflow in Mantid
 - [McStasToX](#), reduction workflow in Scipp
 - **TBL**
 - **Ported McStas 3.x**
 - [McStasToX](#), reduction workflow in Scipp
 - **SKADI**
 - Initial reduction workflow in Mantid



McStas support efforts 2/2

Internal ESS Instruments/Source



In general:

- Work with the IDS/CIS and Instrument Teams
- Focus on developments that let them progress

Important, incoming:

- ESS_butterfly source term
 - Updated docs, do's and dont's
 - Work with target/moderator/neutronics group
 - MCPL-based workflow for commissioning

Instrument Data Scientists (IDS/CIS)



McStas instrument model @ ESS beamlines

Name	Instrument(s)	Associated Instrument Division
Aaron Finke	NMX	Large-scale structures
Celine Durniak	DREAM, BEER	Diffraction & Imaging
Gregory Tucker	BIFROST	Spectroscopy
Christian Beck	MIRACLES, CSPEC	Spectroscopy
Bing Li	TREX	Spectroscopy
Iurii Kibalin	MAGIC	Diffraction & Imaging
Nicolo Paracini	ESTIA, FREIA	Large-scale structures
Oliver Hammond	LOKI, SKADI	Large-scale structures
Søren Schmidt	ODIN	Diffraction & Imaging
???	HEIMDAL	Diffraction & Imaging
???	VESPA	Spectroscopy

Instrument	McStas - TRL	CIS/IDS
TBL	●	No CIS/IDS
BIFROST	●	GT
LOKI	●	OH
ODIN	●	SS
DREAM	●	CD
NMX	●	AF
ESTIA	●	new CIS Ref. ?
SKADI	●	OH
BEER	●	CD
FREIA	●	new CIS Ref. ?
MAGIC	●	IK – just started
MIRACLES	●	CB – just started
HEIMDAL	●	new CIS Diff. ?
CSPEC	●	CB – just started
T-REX	●	BL – just started
VESPA	●	



ESS instrument models - Migration



Project: ess-instrumentmodels

Master branch: ess-instrumentmodels

Name	Last commit	Last update
BEER	Merge branch 'patch-2' into 'master'	8 months ago
BIFROST	MCPL-split front- and backends (single e...	4 years ago
C-SPEC/McStas	Import of C-SPEC from Pascale	7 years ago
DREAM	Add mcpl output and different option for ...	1 year ago
ESTIA	Import another Estia model from the dg_l...	7 years ago
FREIA/McStas	Old FREIA model(s) from Hanna	7 years ago
HEIMDAL/McStas	Pick Dan's version as sent by email	2 years ago
LOKI/McStas	Rectify Vertical_Bender for GPU use	1 year ago
MAGIC/McStas	Update MAGIC.instr	4 years ago
MIRACLES/McStas	Update with new model from Bilbao and ...	2 years ago
NMX/McStas	The edits provided here seems to rectify ...	2 years ago
NON-ESS	Work on J-PARC instrument BL18-SENJU...	2 years ago
ODIN/McStas	Add 2.x model with NeXus backend	4 years ago
SKADI/McStas	McStas script file to use in Jupyter noteb...	10 months ago
V20/McStas	Beautify header before release with McSt...	7 years ago
VESPA/McStas	Import VESPA from NOSG	7 years ago
VOR/McStas	Import of old VOR model from Pascale	7 years ago
.gitignore	Add .gitignore from NOSG repo (with ma...	7 years ago
README.md	Shorten urls	4 years ago

Group: dmsc-instrumentmodels

Subgroups and projects

- ODIN
- MAGIC
- T-REX
- ESS Instrument Template
- HEIMDAL
- BEER
- ESTIA
- SKADI
- NMX
- TBL

- ❑ Old
- ❑ One mono repo
- ❑ All in on one

- ❑ New organization
- ❑ One repo pr instrument
- ❑ All old design parameters stripped away

3

**Data Analysis /
General updates**

EasyScience: Scope and Overview



- The *EasyScience* framework is the common base used by all *Easy** projects.
- It provides **technique-agnostic** reusable components, and it also defines a set of Architecture Decision Records (ADRs) and shared organizational patterns.
- The main goal is to **reduce repeated work** and make projects easier to start and maintain.
- In the last 12 months, the focus was to turn existing practice into clear standards and shared tools.

The screenshot shows the GitHub repository for EasyScience. The repository is verified and has 20 followers. It is located in Copenhagen, DK, and has a website at https://easyscience.software. The repository contains a README.md file with the following content:

Domain-Specific Projects

These projects focus on different scientific techniques for data analysis and reduction.

Project	Tag	Description	Home	Library	Application
easy diffraction	ED	Diffraction data analysis	diffraction	diffraction-lib	diffraction-app
easy reflectometry	ER	Reflectometry data analysis	reflectometry	reflectometry-lib	reflectometry-app
easy imaging	EI	Bragg-Edge Imaging data analysis	imaging	imaging-lib	imaging-app
easy dynamics	EQ	Quasielastic Neutron Scattering data analysis	dynamics	dynamics-lib	dynamics-app
easy texture	ET	Texture data reduction			texture-app
easy shapes	ES	Molecular Structure Generation multiscale simulation workflows	shapes		shapes-app

Core Framework Modules

These are essential building blocks that support the domain-specific projects.

Project	Tag	Description	Library
easy science	ESc	EasyScience Core Module backend	core
easy utilities	EUt	Shared Utilities for EasyScience backend	utils

EasyScience: Completed



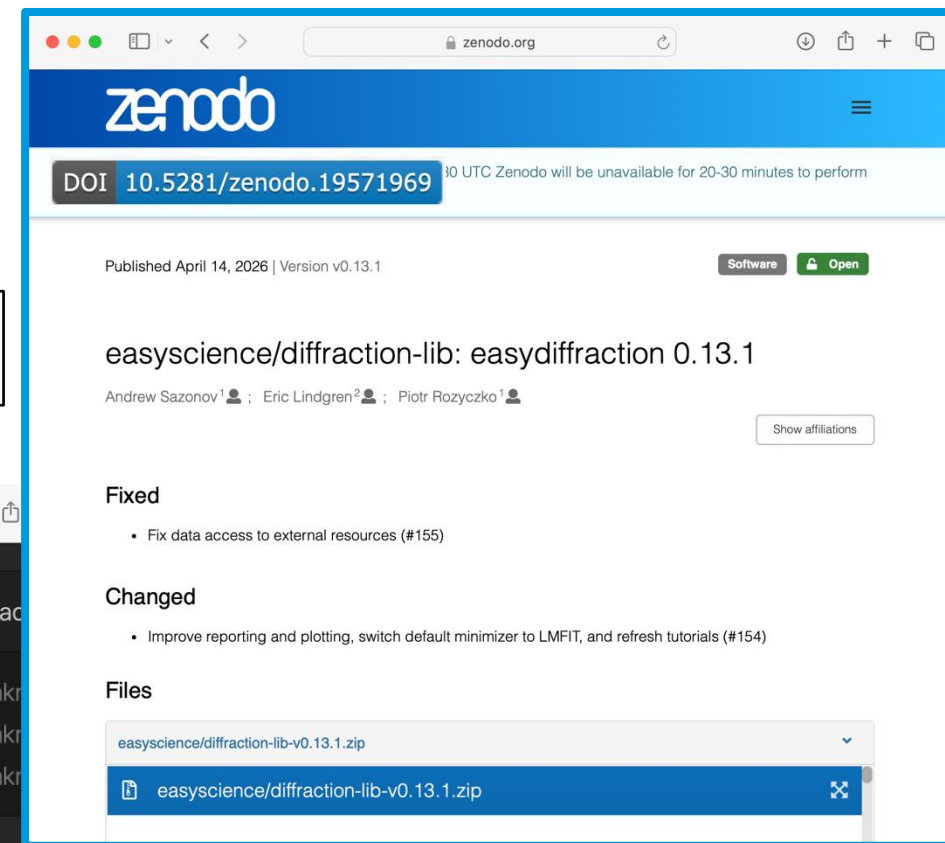
- A solid set of ADRs was proposed, approved, or is now under discussion.
- Approved ADRs now cover release flow, branching, environments, GitHub secrets, automation, PyPI publishing, **Zenodo DOI archiving**, labels, docstrings, license headers, and more.
- A shared template system was introduced for repository layout, CI/CD workflows, documentation, code quality checks, release setup, and more.
- **Templates** were already applied to EasyDiffraction, **EasyDynamics**, EasyShapes, EasyScience core, and EasyUtilities.
- **Automatic HTML dashboards were created to compare code quality** and CI status across repositories.
- EasyUtilities was created as a shared place for helper code used in more than one project.

Zenodo & Code quality

Docs --> Citation

Citation

If you use **EasyDiffraction** library in your work, please cite the specific version you used.



zenodo

DOI 10.5281/zenodo.19571969 30 UTC Zenodo will be unavailable for 20-30 minutes to perform

Published April 14, 2026 | Version v0.13.1 Software Open

easyscience/diffraction-lib: easydiffraction 0.13.1

Andrew Sazonov¹; Eric Lindgren²; Piotr Rozyczko¹ Show affiliations

Fixed

- Fix data access to external resources (#155)

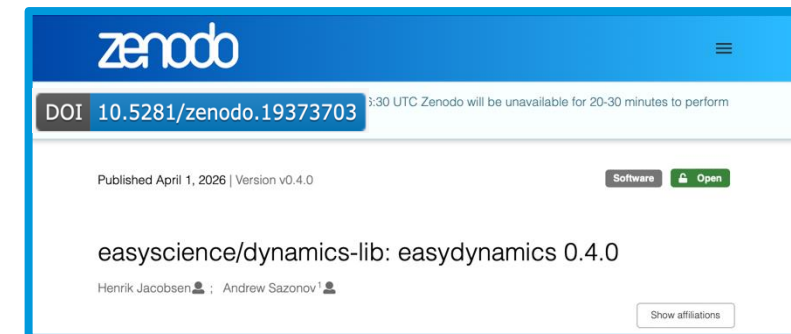
Changed

- Improve reporting and plotting, switch default minimizer to LMFIT, and refresh tutorials (#154)

Files

easyscience/diffraction-lib-v0.13.1.zip

easyscience/diffraction-lib-v0.13.1.zip



zenodo

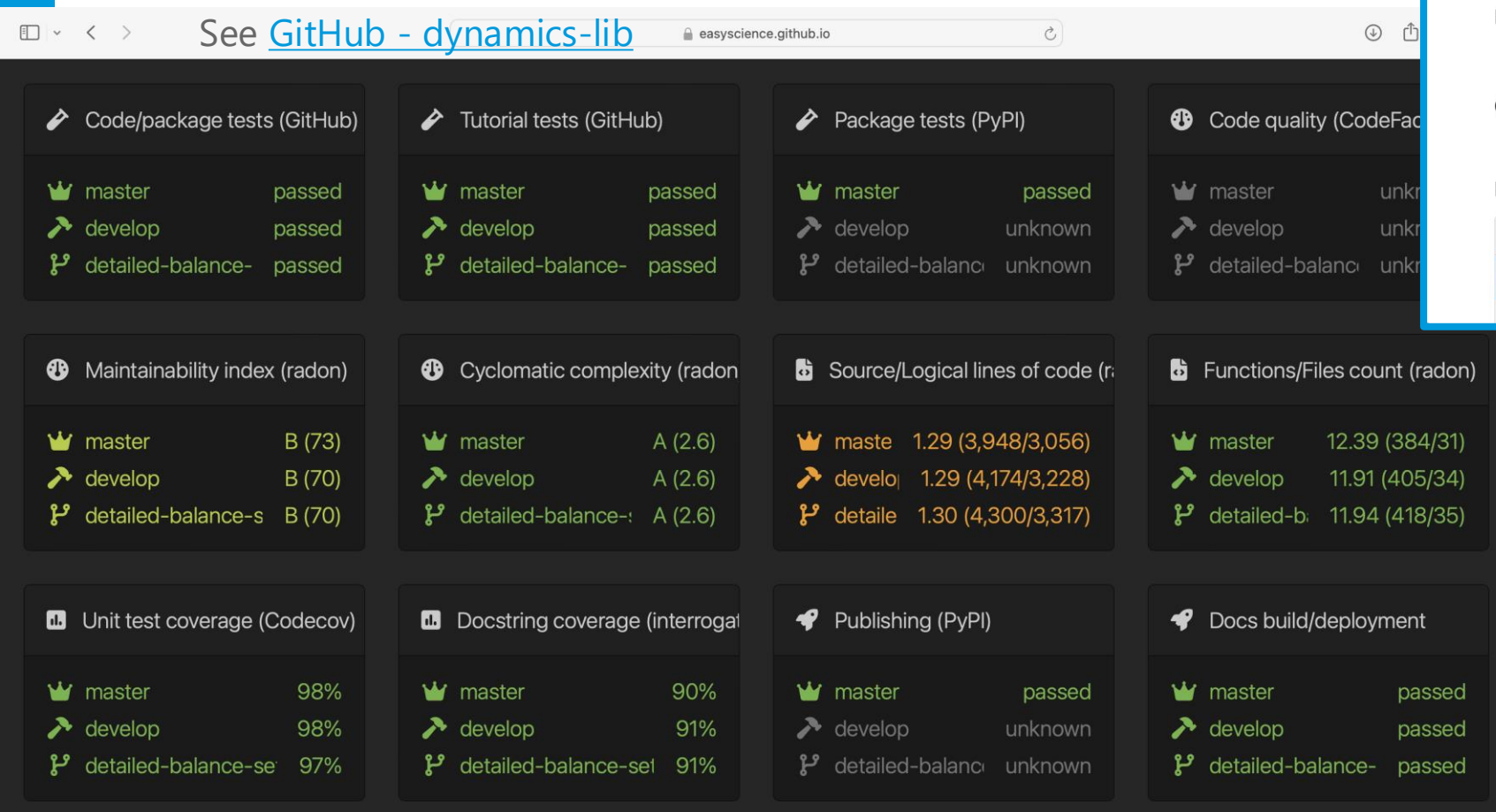
DOI 10.5281/zenodo.19373703 30 UTC Zenodo will be unavailable for 20-30 minutes to perform

Published April 1, 2026 | Version v0.4.0 Software Open

easyscience/dynamics-lib: easydynamics 0.4.0

Henrik Jacobsen; Andrew Sazonov

Show affiliations



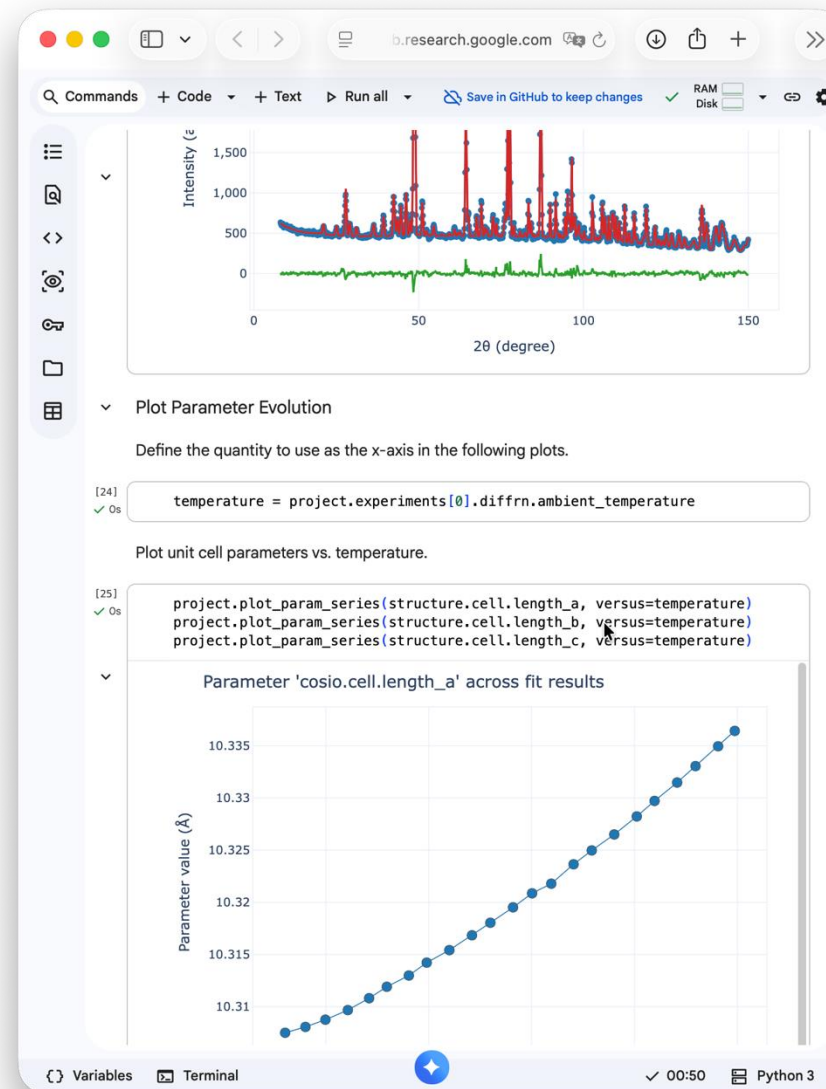
See [GitHub - dynamics-lib](#) easyscience.github.io

Category	Branch	Status
Code/package tests (GitHub)	master	passed
	develop	passed
	detailed-balance-	passed
Tutorial tests (GitHub)	master	passed
	develop	passed
	detailed-balance-	passed
Package tests (PyPI)	master	passed
	develop	unknown
	detailed-balance-	unknown
Code quality (CodeFactor)	master	unknown
	develop	unknown
	detailed-balance-	unknown
Maintainability index (radon)	master	B (73)
	develop	B (70)
	detailed-balance-s	B (70)
Cyclomatic complexity (radon)	master	A (2.6)
	develop	A (2.6)
	detailed-balance-	A (2.6)
Source/Logical lines of code (radon)	master	1.29 (3,948/3,056)
	develop	1.29 (4,174/3,228)
	detailed-balance-	1.30 (4,300/3,317)
Functions/Files count (radon)	master	12.39 (384/31)
	develop	11.91 (405/34)
	detailed-b-	11.94 (418/35)
Unit test coverage (Codecov)	master	98%
	develop	98%
	detailed-balance-se	97%
Docstring coverage (interrogate)	master	90%
	develop	91%
	detailed-balance-se1	91%
Publishing (PyPI)	master	passed
	develop	unknown
	detailed-balance-	unknown
Docs build/deployment	master	passed
	develop	passed
	detailed-balance-	passed

EasyDiffraction: Scope and Overview



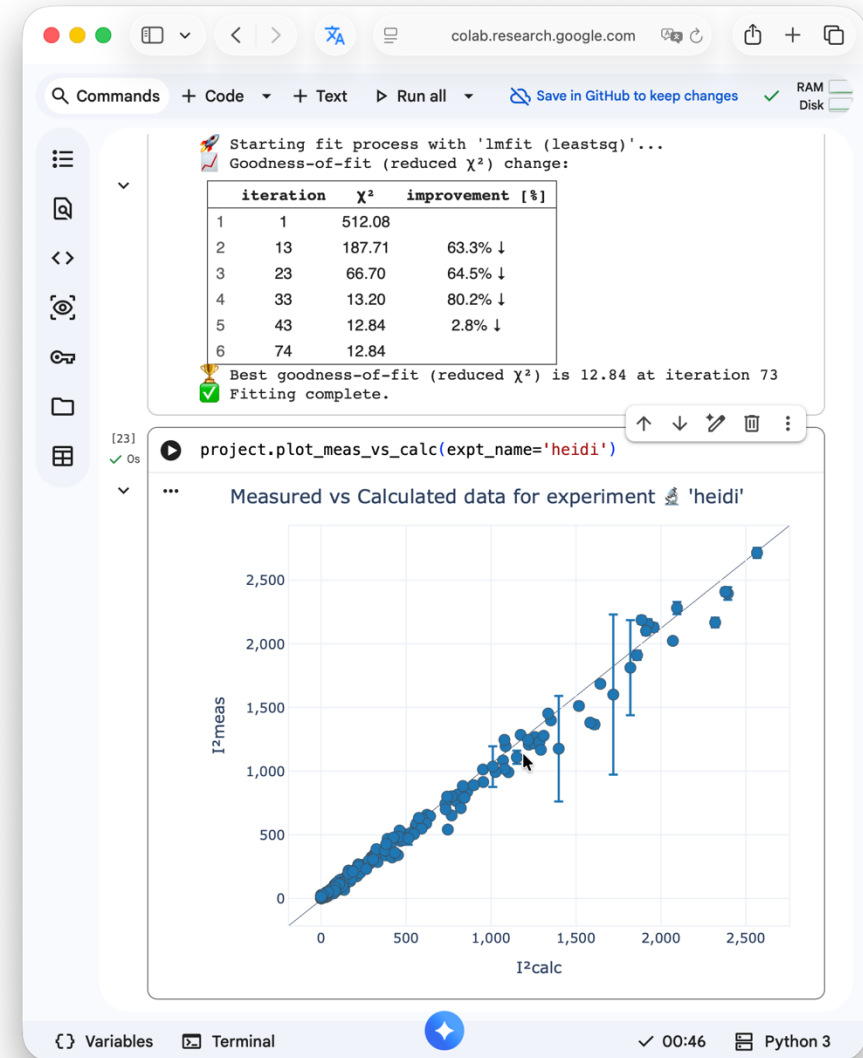
- EasyDiffraction is the diffraction-specific layer.
- It provides both a **GUI and a Python library** for the structural refinement of diffraction data.
- The current scope includes **powder and single-crystal diffraction; total scattering/PDF**.
- It relies on external calculation engines such as **CrysPy, CrysFML, and PdfFit2**.
- **It uses common EasyScience infrastructure** for testing, packaging, documentation, and releases.



EasyDiffraction: Completed (list of some)



- There was a steady release flow over the last 12 months for EasyDiffraction Library, reaching version 0.11.1 on March 31, 2026.
- The user-facing API became more consistent and more stable.
- **Fitting workflows improved**, including excluded regions, constraints, joint fitting, multi-dataset fitting, and clearer handling of results.
- **The scope was extended** beyond standard powder diffraction, with initial PDF support and single-crystal support.



EasyDiffraction: Ongoing



- Support for anisotropic atomic displacement parameters.
- Better support for joint fitting with different experiment types and different calculators.
- Improving sequential fitting of multiple datasets.
- Continuous improvement of documentation and tutorials.
- **Continued collaboration with ILL on PyCrysFML / CrysFML for features needed by EasyDiffraction.**
- **Better packaging and CI for PyCrysFML across platforms and Python versions.**
- A clearer written roadmap and clearer diffraction-related design decisions.
- Steps toward better preparation for DREAM hot commissioning.

EasyDiffraction: To-Do (improvements)



- Some **important cases are still missing**, especially magnetic structures and polarised diffraction.
- Support is not yet balanced between CrysPy and CrysFML.
- Development is also **not balanced** between the library and the GUI: the library has more functionality, while the GUI has received less attention for some time.
- **Packaging of external engines**, especially CrysFML, is still a work in progress.
- End-to-end readiness also depends on other teams and systems, not only on EasyDiffraction.

EasyReflectometry Library & Application

- Used successfully during the 2026 HERCULES School at ESS
- New releases made
- 1.2.0 (App. - Mar 2026)
- 1.4.1 (Lib. - Oct 2025)
- 1.5.0 (Lib. - Mar 2026)

```

# --- Reflectivity ---
for i, rid in enumerate(refl_nums):
    qz = data['coords'][f'Qz_{rid}'].values
    r = data['data'][f'R_{rid}'].values
    sr = np.sqrt(data['data'][f'R_{rid}'].variances)
    color = color_cycle[i % len(color_cycle)]

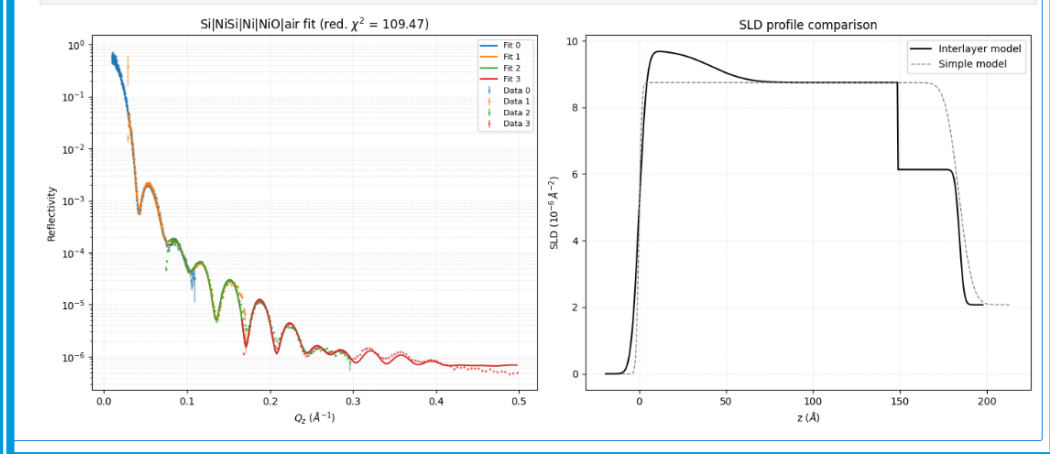
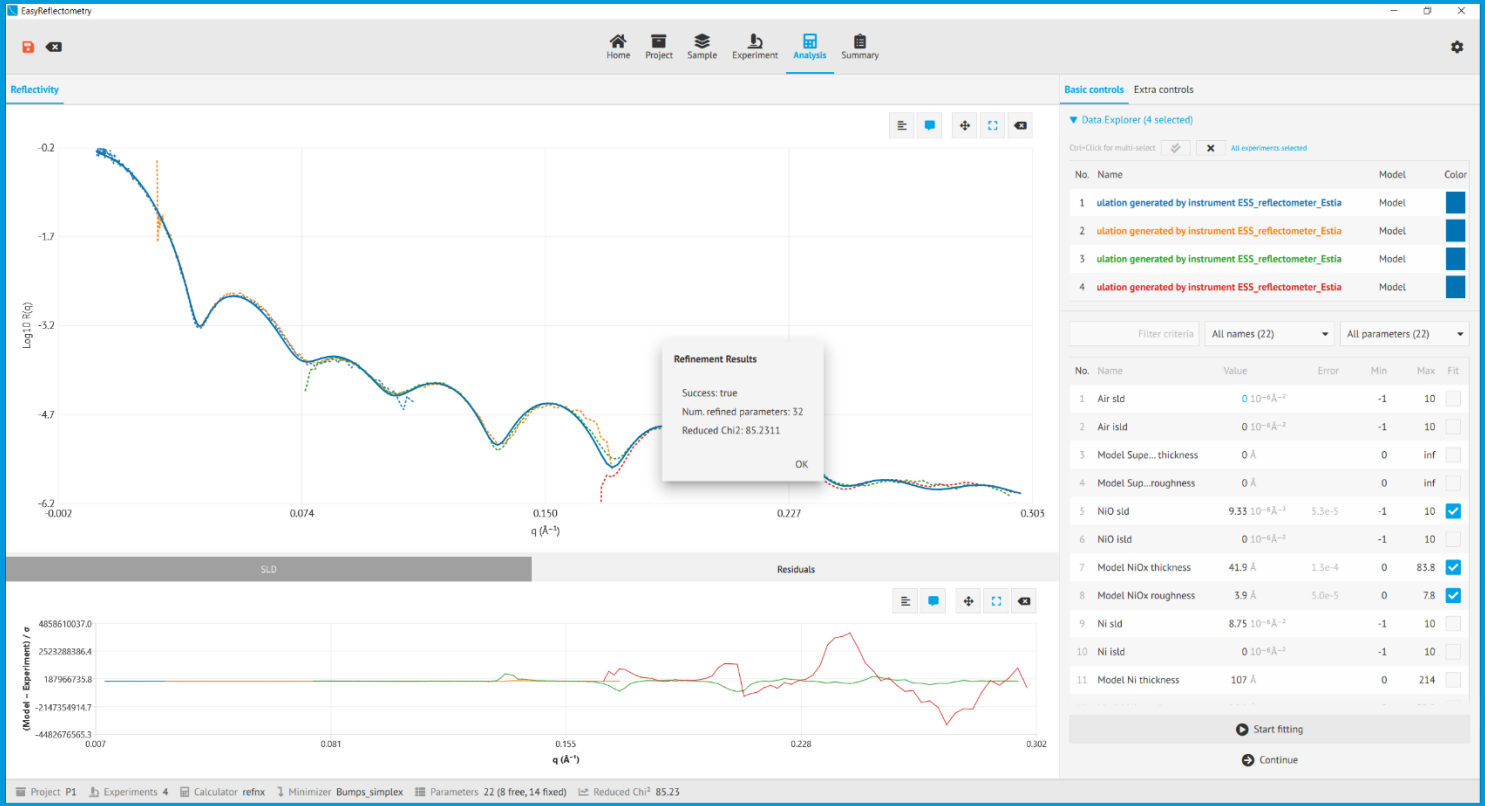
    ax3.errorbar(
        qz, r, yerr=sr,
        fmt='.', markersize=3, capsize=0, color=color,
        alpha=0.5, label=f'Data {rid}',
    )
    r_model2 = model2.interface.fit_func(qz, model2.unique_name)
    ax3.plot(qz, r_model2, '-', color=color, linewidth=1.5, label=f'Fit {rid}')

ax3.set_yscale('log')
ax3.set_xlabel(r'$Q_z$ ($\text{\AA}^{-1}$)')
ax3.set_ylabel('Reflectivity')
ax3.set_title(f'Si|Ni|Ni|NiO|air fit (red. $\chi^2$ = {reduced_chi2:.2f})')
ax3.legend(fontsize=8, loc='upper right')
ax3.grid(True, which='both', linestyle=':', alpha=0.4)

# --- SLD profile comparison ---
ax4.plot(z_sld2, sld_vals2, '-', color='black', linewidth=1.5, label='Interlayer model')
ax4.plot(z_sld, sld_vals, '-', color='gray', linewidth=1.0, label='Simple model')
ax4.set_xlabel(r'$z$ ($\text{\AA}$)')
ax4.set_ylabel(r'$SLD$ ($10^{-6} \text{\AA}^{-2}$)')
ax4.set_title('SLD profile comparison')
ax4.legend()
ax4.grid(True, linestyle=':', alpha=0.4)

fig2.tight_layout()
plt.show()

```

Refinement Results

- Success: true
- Num. refined parameters: 32
- Reduced Chi2: 85.2311

No.	Name	Value	Error	Min	Max	Fit
1	Air std	$0 \cdot 10^{-4} \text{\AA}^{-2}$		-1	10	<input type="checkbox"/>
2	Air isld	$0 \cdot 10^{-4} \text{\AA}^{-2}$		-1	10	<input type="checkbox"/>
3	Model Supe...thickness	0 Å		0	inf	<input type="checkbox"/>
4	Model Sup...roughness	0 Å		0	inf	<input type="checkbox"/>
5	NiO std	$9.33 \cdot 10^{-4} \text{\AA}^{-2}$	$5.3e-5$	-1	10	<input checked="" type="checkbox"/>
6	NiO isld	$0 \cdot 10^{-4} \text{\AA}^{-2}$		-1	10	<input type="checkbox"/>
7	Model NiOx thickness	41.9 Å	$1.3e-4$	0	83.8	<input checked="" type="checkbox"/>
8	Model NiOx roughness	3.9 Å	$5.0e-5$	0	7.8	<input checked="" type="checkbox"/>
9	Ni std	$8.75 \cdot 10^{-4} \text{\AA}^{-2}$		-1	10	<input type="checkbox"/>
10	Ni isld	$0 \cdot 10^{-4} \text{\AA}^{-2}$		-1	10	<input type="checkbox"/>
11	Model Ni thickness	107 Å		0	214	<input type="checkbox"/>

Start fitting
Continue

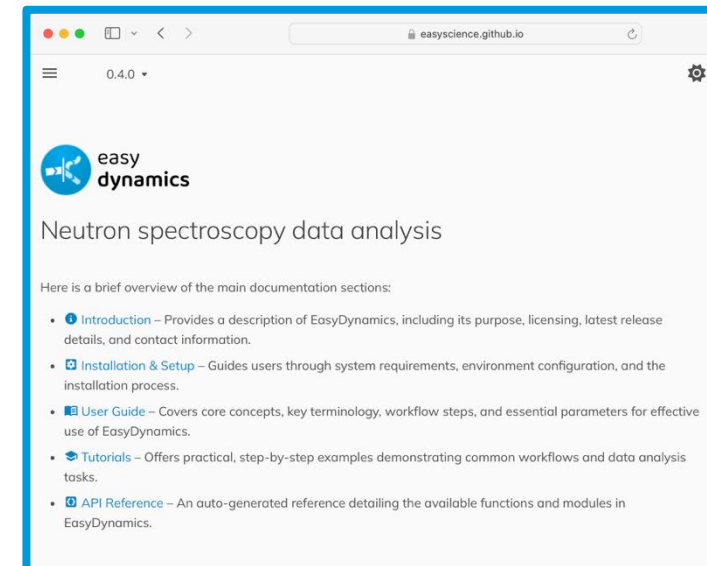
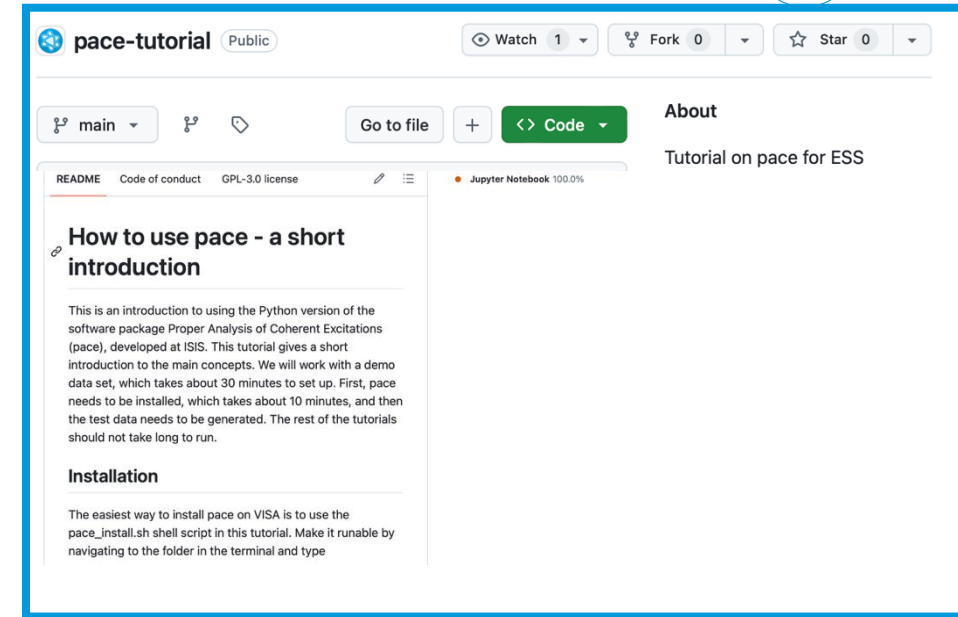
Spectroscopy: single crystal and powder



□ Spectroscopy analysis can be divided into two categories:

□ Single crystal: 4-dimensional data sets: $I(Q_x, Q_y, Q_z, E)$, mainly from **BIFROST** and **CSPEC**. Data will be analysed using **pace_neutrons**

□ Powder and amorphous materials: 2-dimensional data sets: $I(|Q|, E)$, mainly from **CSPEC** and **MIRACLES**. Data will be analysed using **EasyDynamics**

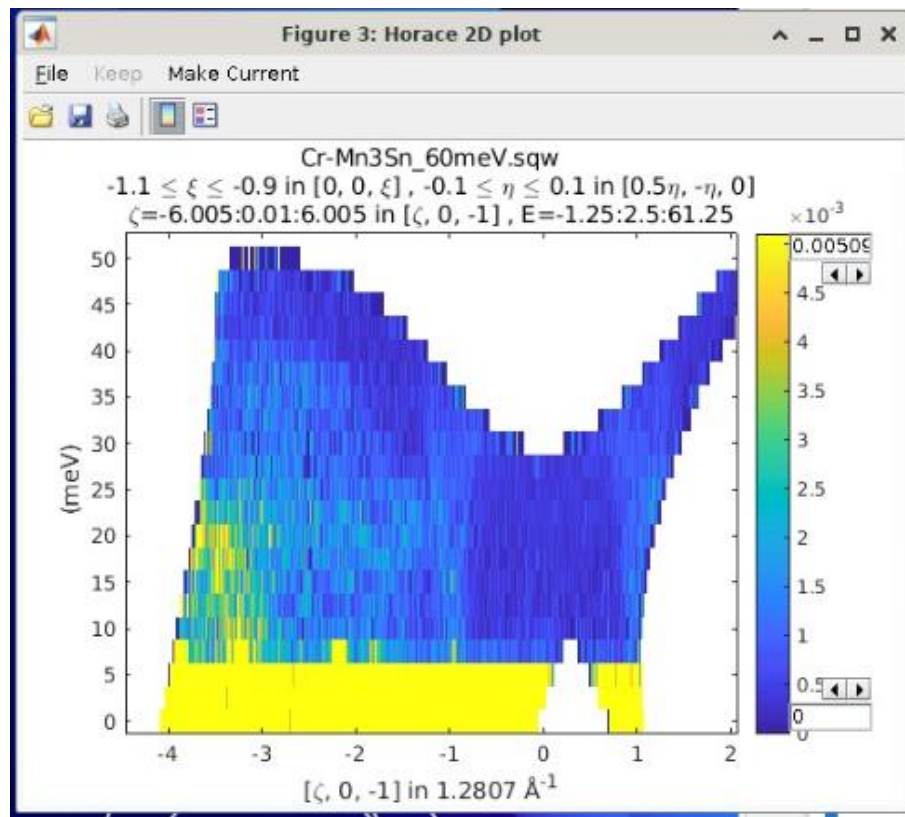


pace-neutrons on VISA



Visualise, manipulate and analyse single crystal spectroscopy data

- pace-neutrons is a Python wrapper of Horace with almost all of the functionality



```
%matlab_plot_mode windowed
my_slice = m.cut_sqw(sqw_file, proj, [-6, 0.01, 6], [-1-0.1, -1+0.1], [-0.1, 0.1], [0,2.5,60])
m.plot(my_slice)
m.keep_figure()

*** Step 1 of 12; Read data for 10408311 pixels -- processing data... -----> retained 2358880 pixels
*** Step 2 of 12; Read data for 9904008 pixels -- processing data... -----> retained 2472596 pixels
*** Step 3 of 12; Read data for 9652739 pixels -- processing data... -----> retained 3043601 pixels
*** Step 4 of 12; Read data for 9059583 pixels -- processing data... -----> retained 1796985 pixels
*** Step 5 of 12; Read data for 9422482 pixels -- processing data... -----> retained 3275038 pixels
*** Step 6 of 12; Read data for 10091763 pixels -- processing data... -----> retained 2842257 pixels
*** Step 7 of 12; Read data for 11050159 pixels -- processing data... -----> retained 1955642 pixels
*** Step 8 of 12; Read data for 10045834 pixels -- processing data... -----> retained 2947709 pixels
*** Step 9 of 12; Read data for 9340458 pixels -- processing data... -----> retained 2351767 pixels
*** Step 10 of 12; Read data for 10143940 pixels -- processing data... -----> retained 2571891 pixels
*** Step 11 of 12; Read data for 9300646 pixels -- processing data... -----> retained 2301227 pixels
*** Step 12 of 12; Read data for 10247453 pixels -- processing data... -----> retained 2320584 pixels
*** Completed cat_pixels using 1 pages in 8.7437 sec.
Warning: colordef will be removed in a future release.
```

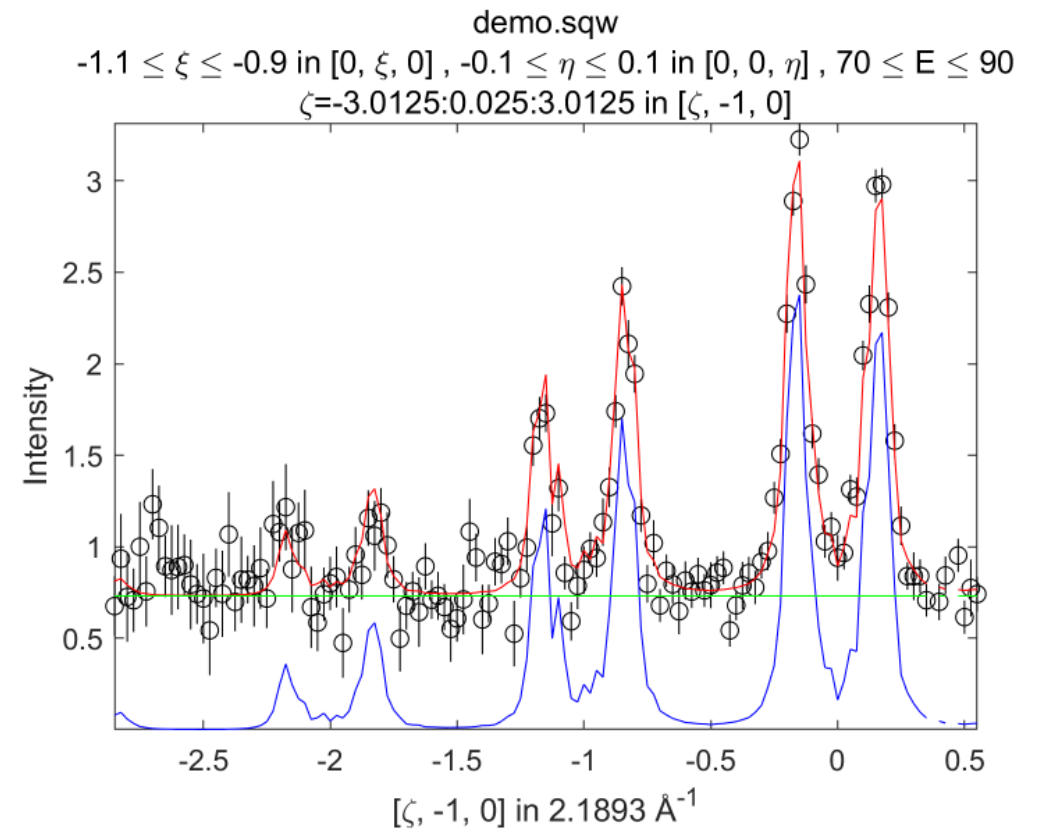
Figure: Spin waves in Cr-doped Mn₃Sn

pace-neutrons



Visualise, manipulate and analyse single crystal spectroscopy data

- **Completed:** make pace-neutrons available on VISA, make tutorials for users. Can fit data, e.g. using SpinW
- **Ongoing:** Test with simulated BIFROST and CSPEC single crystal data
- **Still needs to be done:** Develop roadmap for Python library and/or PACE/Horace
- **Where are the weak points:** Including resolution functions in models. pace-neutrons is a Python wrapper of the underlying Matlab library; a pure Python library would be better



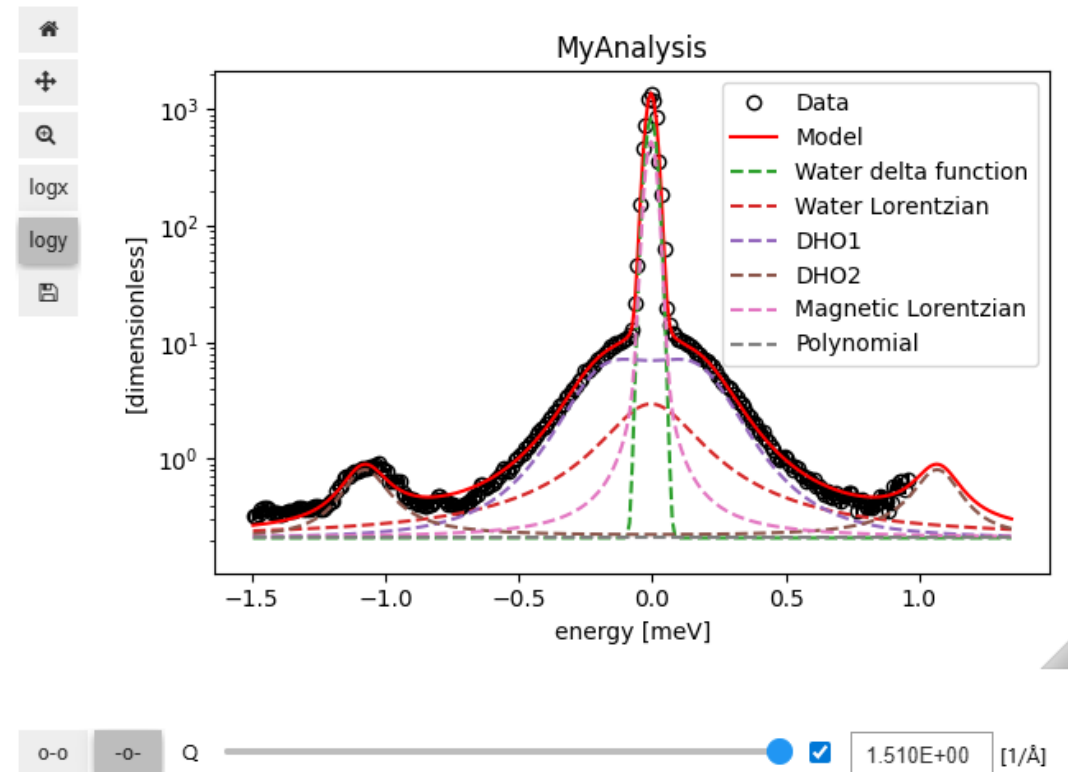
EasyDynamics



For QENS and powder INS data e.g. from CSPEC and MIRACLES

- Python library that supports
 - Fit of arbitrarily complex functions
 - Fit multiple Q simultaneously
 - Easy handling of resolution functions
- **Completed:** First release with core functionality and extensive documentation. Easy to install on all platforms. Interest in collaboration/use from ILL, HZB, MLZ,...
- **Ongoing:** Improve data loader, extend to larger data sets, more tutorials, ...
- **Still needs to be done:** Graphical interface
- Where are the weak points: Lack of graphical interface

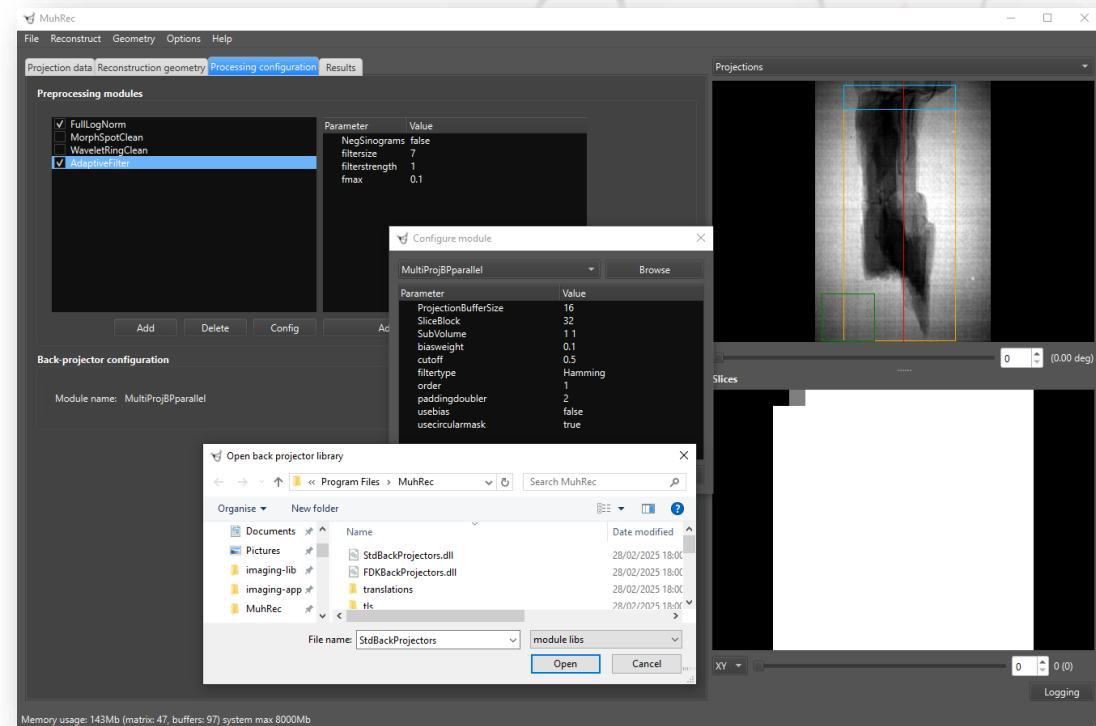
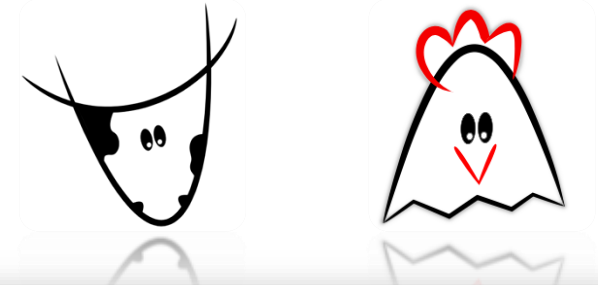
<https://easyscience.github.io/dynamics-lib/>



MuhRec

Tomography software

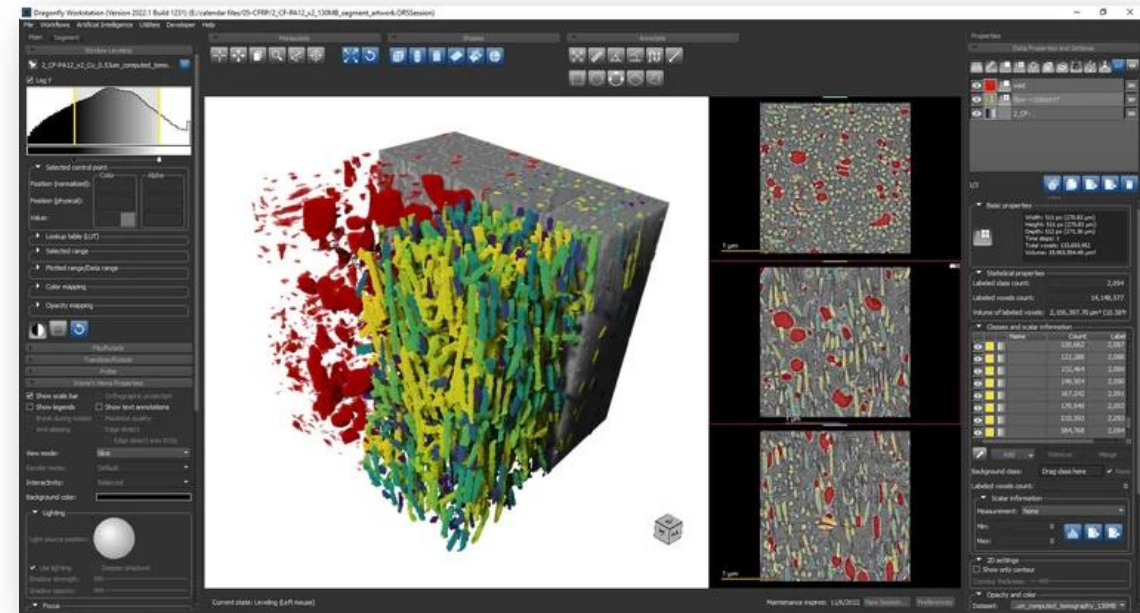
- **Yearly progress:** Maintenance
- **Completed:** App and Python library available and on VISA. Tomography user-notebook on code shelf.
- **Ongoing:** Release library on PyPi
- **To-do:** Improve performance with/without GPUs. Split up codebase into clear responsibilities for easier maintenance.
- **Concerns:** Primarily developed by Anders Kaestner at PSI – Slow improvements and potentially a single-point-of-failure.



Dragonfly

Image segmentation software

- **Yearly progress:** Bought Licenses. Tested on local infrastructure.
- **Completed:** Acquired licenses.
- **Ongoing:** Being added to VISA.
- **To-do:** Test on VISA. Test on neutron data created with MuhRec. Set-up user-script(s) on code-shelf.
- **Concerns:** Is our it-infrastructure able to handle the software? Is VISA?



EasyImaging Library

In-house Python library for fitting Bragg-edges



- **Yearly progress:** Can load data from tiff/scitiffs, visualize the data in various ways, re-normalize the image and create regions of interest by code or drawing

- **Completed:** Released v.0.0.1 on PyPi. Notebook including Bragg-edge fitting on code-shelf

- **Ongoing:** Creating the documentation.

- **To-do:** Add the physical model with Ncrystal, add instrument parameters, add fitting for ROI and pixel-by-pixel.



The Measurement class

The class responsible for holding and handing the measurement

```
[ ]: import scipp as sc
from easyimaging import Measurement
```

The **Measurement** class can be created in 3 different ways. The preferred way is by loadi

```
[ ]: measurement = Measurement.from_scitiff(filename='../small_scitiff.tiff')
```

Alternatively, the **Measurement** class can also be created using a regular Tiff stack using

```
[ ]: measurement = Measurement.from_tiff_stack(
    filename='../small_tiff.tiff',
    time_of_flights=sc.arange('t', 0, 240, 1, unit='s'), # mandatory
    # x_positions=sc.arange('x', 0, 50, 1, unit='mm'), # optional
    # y_positions=sc.arange('y', 0, 50, 1, unit='mm'), # optional
)
```

The *x_positions* and *y_positions* of the pixels can also be set or overwritten later with the r

```
[ ]: measurement.set_physical_coord_positions(
    x_positions=sc.arange('x', -25, 25, 1, unit='cm'),
    y_positions=sc.arange('y', -25, 25, 1, unit='cm'),
)
```

Finally the **Measurement** class can also be created by directly passing an appropriate sci

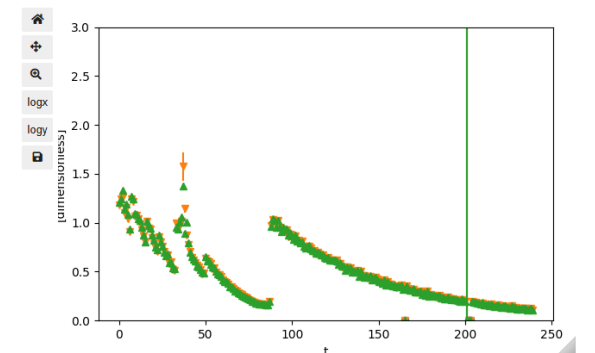
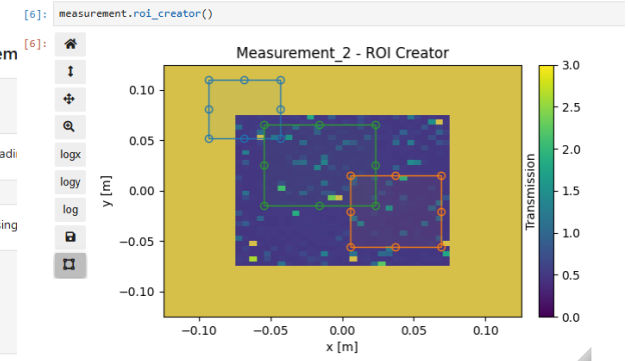
```
[ ]: tof = sc.arange('t', 0, 10, 1, unit='s')
x = sc.arange('x', 0, 7, 1, unit='m') # optional
y = sc.arange('y', 0, 7, 1, unit='m') # optional
data = sc.ones(dims=['x', 'y', 't'], shape=[6, 6, 10])
scipp_array = sc.DataArray(data=data, coords={'tof': tof, 'x': x, 'y': y})
```

Note that the *x* and *y* coordinates, corresponding to the previous *x_positions* and *y_positi*

Another point to note is that the time-of-flight coordinate is called 'tof' but it has to have

With this correct DataArray, we can then construct the **Measurement** class

```
[ ]: measurement = Measurement(data_array=scipp_array)
```

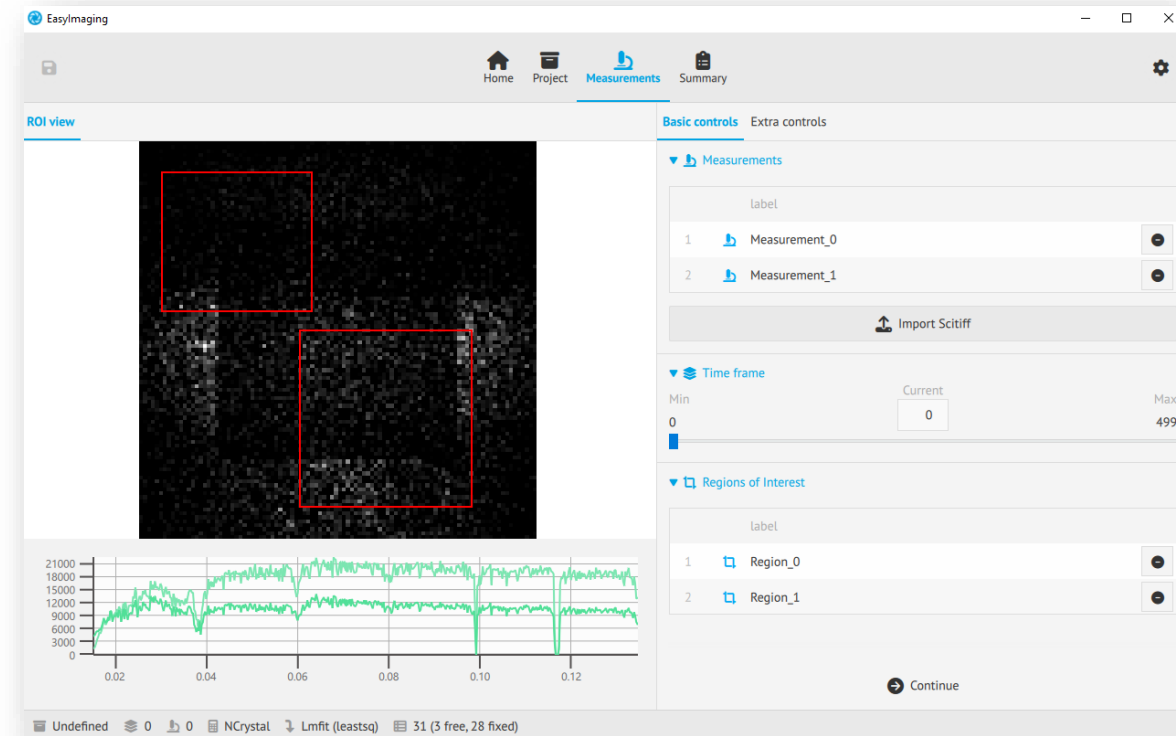


EasyImaging Application

In-house graphical application for fitting Bragg-edges



- **Yearly progress:** Can load in scitiffs, visualize the image-stack, define regions of interest and plot their spectrums.
- **Completed:** First alpha version was showcased at Neuwave-13.
- **Ongoing:** Update application to use the released version the easyimaging library
- **To-do:** The physical model page, The instrument page, The analysis page. Save and load project. Export python script/Jupyter notebook.

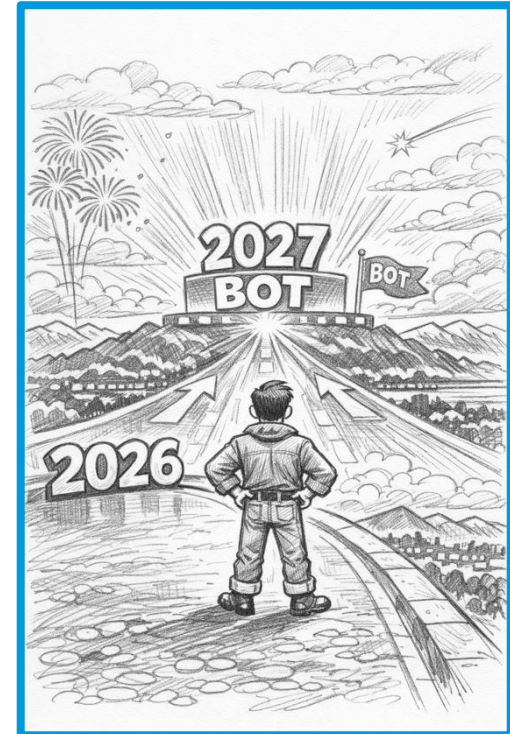




Plans for the rest of 2026

Major activities

- Scipp
 - Workflows for (new) CIS's
 - LiveData
 - Fine tune VISA UX
- Data Analysis
 - Continue to develop EasyScience projects
- McStas
 - Instrument models for (new) CIS's
 - ESS source description
 - Instrument models TRL / Digital Twins





Risks and weak points

What keeps us awake at night

- ❑ Commissioning of software – not always a turnkey solution
- ❑ Handover of software
- ❑ Lack of “acceptance criteria” and “definition of done”
- ❑ Orphan scope



Advise to the DRAM group

Are we on the right track?

- Advise in relation to new positions in materials modelling / ab-initio
 - Permanent vs Fixed?
 - University affiliated ?
 - Fast – Job out soon
- AI & ML?
- Any other advice?



Finish presentation