Lessons learned from the assessment of successive problematic XDS versions over the past year

Global Phasing Consortium Meeting
Cambridge, UK

June 30 - July 2 2025

Lessons learned #1

- A fully collaborative effort between
  - Gérard Bricogne, Claus Flensburg, Clemens Vonrhein
  - Gleb Bourenkov, Ashwin Chari
  - Wolfgang Kabsch, Kay Diederichs

- Lots of information and encouragement from consortium members, beamline scientists and academic users.

- Being able to contact all/most stake holders quickly was crucial.

Timeline - 1/2

XDS saga (22nd July 2024 - May 2025)

See also (in order, with public material shown in bold and external contributions in italic):

- [20240722] First contact with new XDS version - to replace 20230630 version
- [20240729] Voiced serious concern about data quality
- [20240801] [internal confluence] XDS Version June 30, 2024 - testing and comparison
- [20240802] Asked/suggested for provision of extended 20230630 binary to users
- ***[20240820]*** *[**ccp4bb**] "A few little things have changed so you may have to adapt your scripts."*
- [20240831] Expiry of 20230630 binary
- [20240906] [internal confluence] XDS Version June 30, 2024 - testing and comparison of released version
- [20240910] [internal confluence] XDS Version June 30, 2024 - testing on simulated data
- [20240911] [internal confluence] XDS Version June 30, 2024 - changes in parameter naming, usage and meaning
- [**20240917**] [**Consortium-discuss**] Urgent request for feedback regarding performance of latest XDS version within autoPROC+STARANISO
- [**20240917**] [**synchrotron-announce**] Request for comments regarding performance of autoPROC+STARANISO using latest XDS version
- [**20240925**] **autoPROC Wiki** - ComparisonProcessing202409
- [**20240925**] [**Consortium-discuss**] Update on the investigation of the performance of the latest XDS version within autoPROC+STARANISO
- [20240926] [internal confluence] XDS Version June 30, 2024 - more details about public report(s)
- [**20240926**] [**synchrotron-announce**] Update on the investigation of the performance of autoPROC+STARANISO using the latest XDS version
- [20240927] [internal confluence] Phenix.refine handling of input reflection data (judge data quality judged by refinement)
- [**20241002**] [**ccp4bb**] Re: Problem of data reprocessing with XDS
- ***[20241003]*** *[**phenixbb**] "... at least as good as version 20230630, and often better."*
- [20241004] [internal confluence] XDS output ... why CORRECT, XSCALE and XDSCONV are problematic
- [20241008] [internal confluence] XDS and dynamic shadowing
- [**20241015**] [**Consortium-discuss**] Followup to XDS 2024 version analysis
- [**20241015**] [**synchrotron-announce**] Followup to XDS 2024 version analysis
- [20241028] [internal confluence] xds-zcbf plugin - modifications (202410) to dynamically combine simulated diffraction, background and shadowing images
- [20241121] Extended 20230630 binary available (expiry 20250331)

Timeline - 2/2

- [20250127] [internal confluence] XDS/DELPHI and thread conundrum
- [20250206] [internal confluence] XDS (upcoming) version
- **[20250213] autoPROC Wiki** - ComparisonProcessing202502
- **[20250213]** [**Consortium-discuss**] Another followup to XDS version analysis
- [20250215] [internal confluence] 5SP6
- [20250306] Extended 20230630 binary removed
- **[20250307]** [**Consortium-discuss**] Another followup to XDS version analysis
- **[20250307]** [**synchrotron-announce**] Followup to XDS 2024 version analysis - now looking at the 2025 versions
- [20250313] [internal confluence] XDS version
- **[20250313]** [**Consortium-discuss**] XDS version analysis: a request for factual information -- this is URGENT
- **[20250313]** [**synchrotron-announce**] XDS version analysis: a request for factual information -- this is URGENT
- [20250317] [internal confluence] XDS versions 2024-2025 - reflection comparison
- **[20250318]** [**Consortium-discuss**] XDS version analysis: a request for factual information -- results of questionaire and follow-up
- [20250324] [internal confluence] XDS version(s) - background generation (INIT) issues
- [20250326] [internal confluence] XDS version(s) - misfits, Wilson plot, I/sigI analysis
- **[20250327]** [**Consortium-discuss**] Latest update from Global Phasing about XDS matters
- **[20250327]** [**synchrotron-announce**] Latest update from Global Phasing about XDS matters
- ***[20250328]*** *[**ccp4bb**] new academic version XDS20250327 posted*
- **[20250331] autoPROC Wiki** - ComparisonProcessing202503
- **[20250423] autoPROC Wiki** - ComparisonProcessing202504
- **[20250424]** [**Consortium-discuss**] Assessment of recently posted XDS binary (20250409)
- **[20250424]** [**synchrotron-announce**] Assessment of recently posted XDS binary (20250409)
- [20250428] [internal confluence] XDS version 20250409 - CB224_xtl004_G1B1_138
- **[20250429]** [**Consortium-discuss**] Assessment of recently posted XDS binary (20250409) - followup
- **[20250429]** [**synchrotron-announce**] Assessment of recently posted XDS binary (20250409) - followup
- [20250505] [internal confluence] autoPROC at DLS
- **[20250506]** [**Consortium-discuss**] Assessment of latest XDS binary (20250430)
- **[20250506]** [**synchrotron-announce**] Assessment of latest XDS binary (20250430)
- **[20250520] ISPyB/MXCuBE presentation**
- **[20250528]** [**Consortium-discuss**] The XDS saga: epilogue and perspective
- **[20250530]** [**ccp4bb**] Vagaries of XDS releases [was: Requesting help with Twin Refinement]
- [20250623] [internal confluence] autoPROC at ESRF

GΦL
Global Phasing Limited

Why does it matter? Back of envelope calculation ...

- Let's assume we have 60 MX beamlines worldwide and **30** are active/productive in any meaningful way.
- Let's further assume that they run **20 days a month** and use **8 hours a day** for full data collections.
- We can take 12 datasets (**5 min per dataset**) collected per hour - resulting in:
  - 30 x 20 x 8 x 12 = **57600 datasets each month**
  - potentially processed multiple times with XDS (**3** different XDS-based pipelines on average)
  - 9 months of uncertainty and problematic XDS behaviour (minus 2 for holidays/shutdowns)
  - 57600 x 3 x 7 = **1.2 Mio processing results** that might be affected (to varying degrees)
  - each job (**10 min**, 12 cores, 64GB memory, Xeon E5-2683 v4) uses 1.73 kWh
  - would take about 23 years on such a machine - or ca 3 months if we had 100 of them
  - ca 2 Gwh (@0.245£ = ca **500k GBP**)
  - equivalent to ca **750 t CO2e**

Environmental impact

Details about your algorithm

To understand how each parameter impacts your carbon footprint, check out the formula below and the methods article

Runtime (HH:MM): 10 : 0

Type of cores: CPU

Number of cores: 12

Model: Xeon E5-2683 v4

Memory available (in GB): 64

Share your results as a csv file!

Import resuts
Drag and drop or click to select your .csv file.

601.43 gCO2e
Carbon footprint

1.78 kWh
Energy needed

0.66 tree-months
Carbon sequestration

3.44 km
in a passenger car

0.55%
of a flight Paris-Dublin

x 1.2 Mio

4 Mio km

6600 flights

2400 flights London to New York

Lessons learned #2

- Anything related to mission-critical workhorses like XDS matters - in terms of time, money and environmental impact

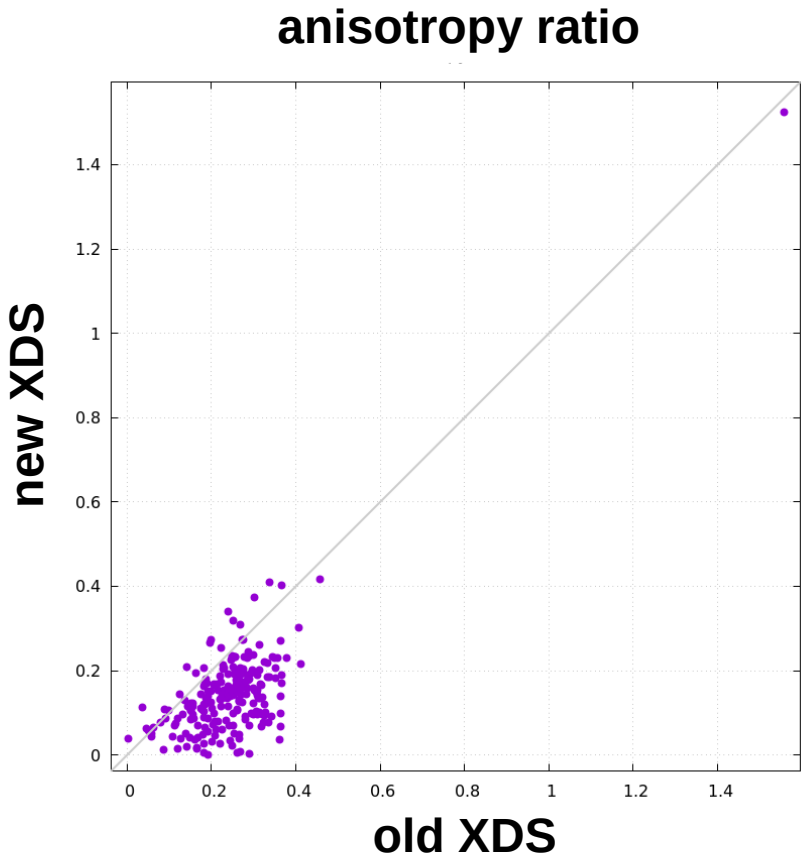
What did we do when, why and how?

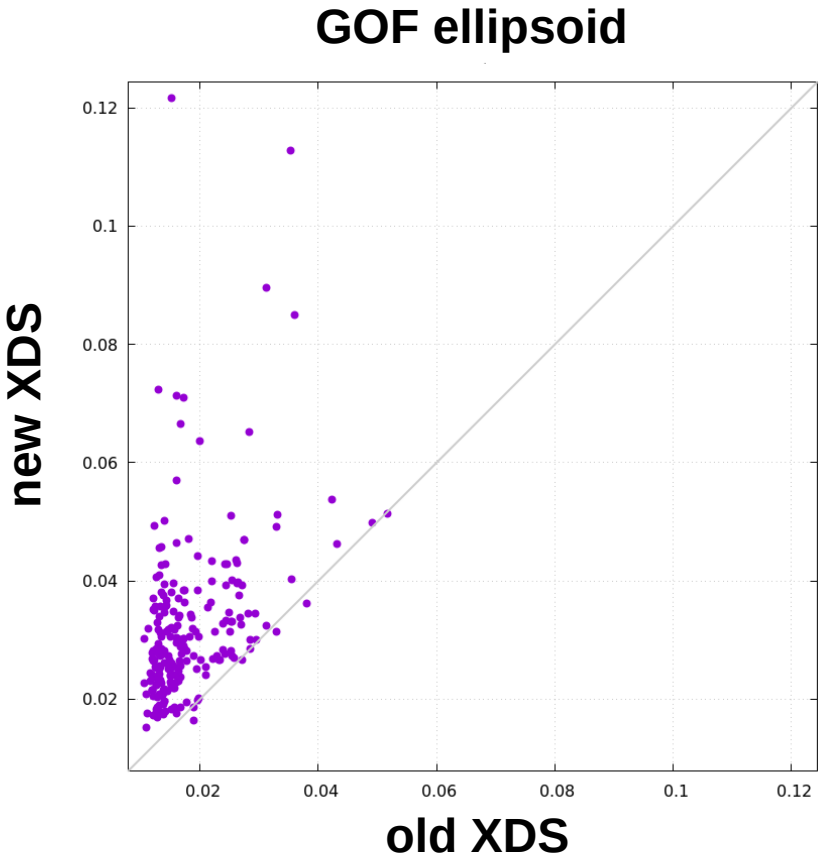
- When first contacted by Kay (**22nd July 2024**) to have a look at the upcoming XDS version (while getting ready for and taking part in the Diffraction Methods meeting in Berlin, 22nd - 27th July), we managed to run nearly 300 public datasets using the previous (**20230630**) and the upcoming XDS binary (**20240712**) through autoPROC. Concentrating on those PDB identifiers where both runs used the same set of images (i.e. no difference due to automatic exclusion of poor image ranges) gave us **222 PDB identifiers**.

Lessons learned #3

- Testing under time pressure is a very bad idea.
- Enforced expiry dates for binaries creates unnecessary pressure
  - on developers (need to provide a new binary)
  - on users (need to switch to new binary)
  - on everyone (unable to run side-by-side comparisons “the official way”)

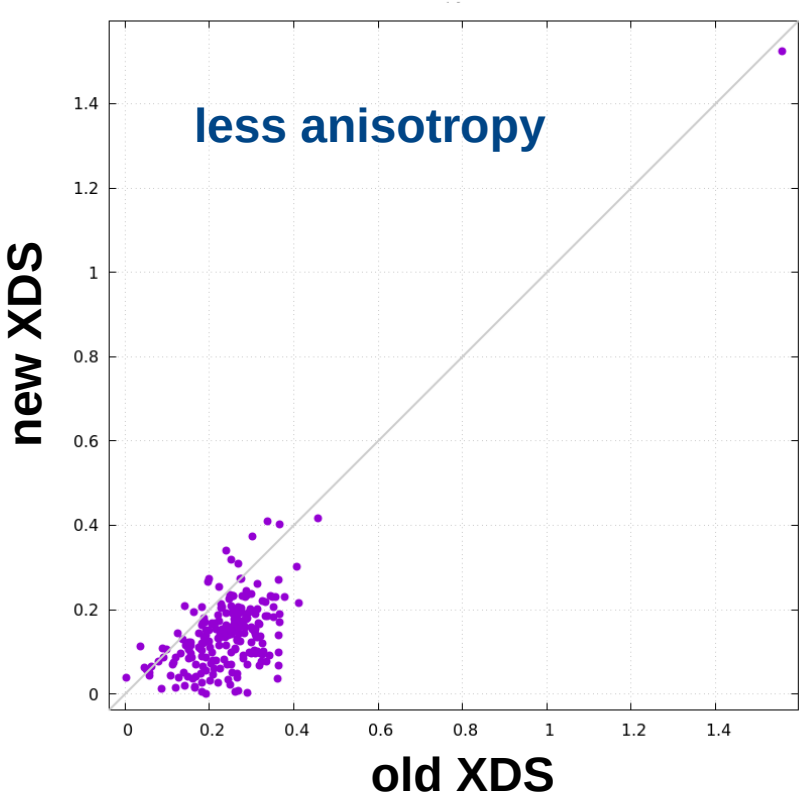
Looking at STARANISO analysis



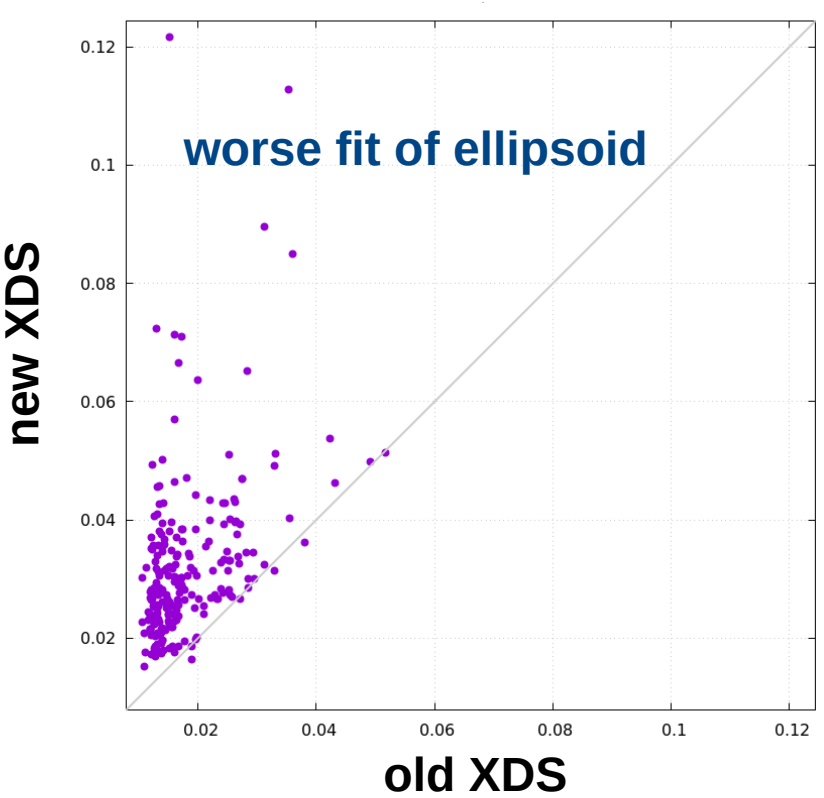


Looking at STARANISO analysis

anisotropy ratio

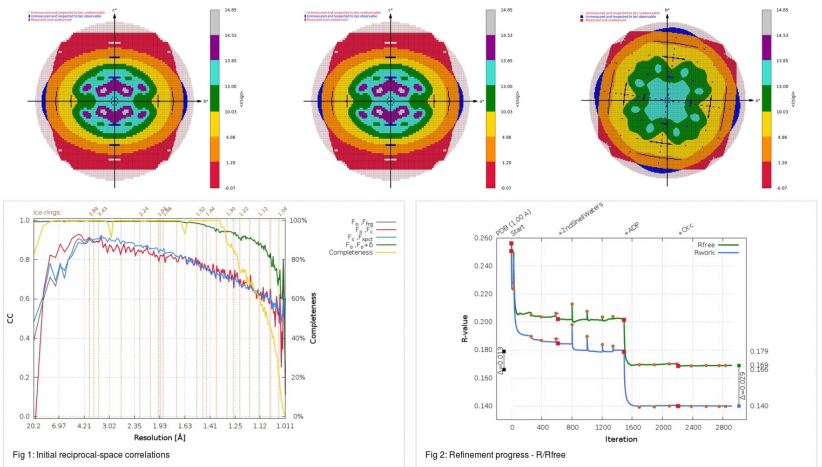


GOF ellipsoid

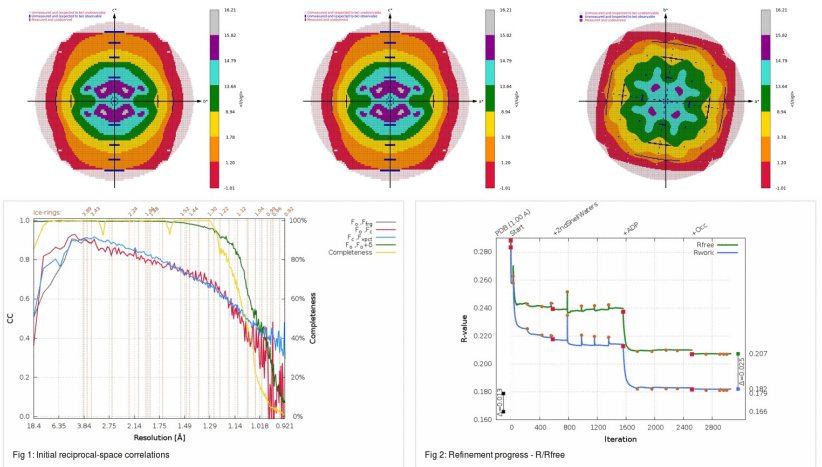


First and immediate alarm bells

- Looking at some aB_autorefine (BUSTER) jobs using the two processing results showed some concern:



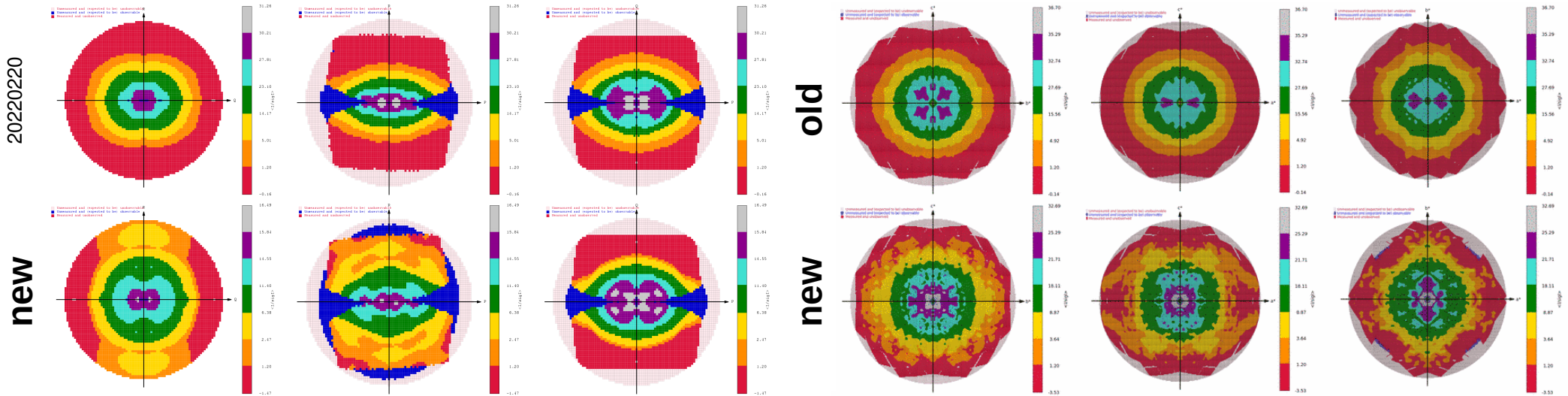
5RVG



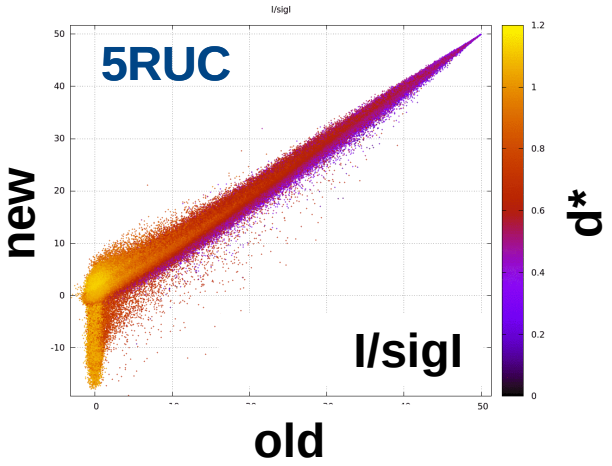
old **new**

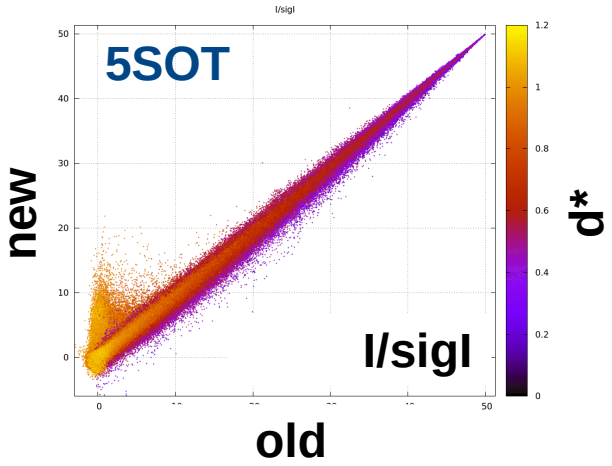
First and immediate alarm bells

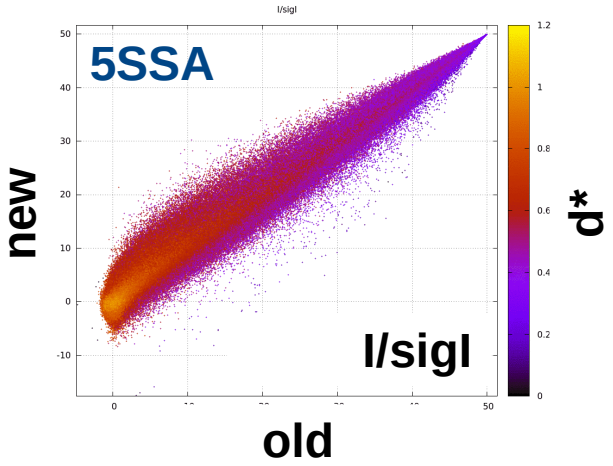
- confirmed also by external users:

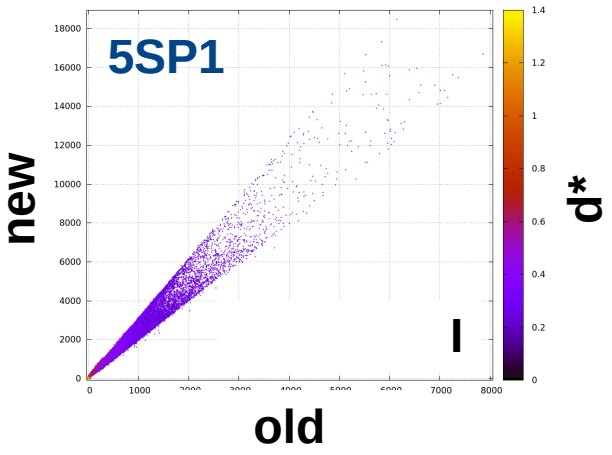


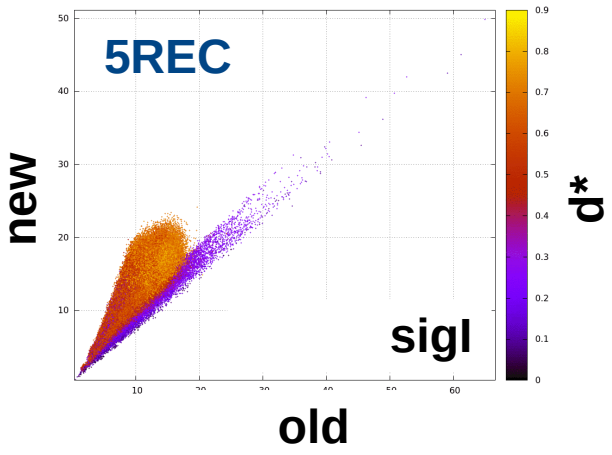
Comparison of I, sig(I) and I/sig(I)











- some very odd/funny "features"

Lesson #4

- A picture says more than a thousand words.
- Running through a large number of data and creating easy to visualise comparisons make the task simpler: scatter plots are great :-)

More test data needed

- From that point onwards we were ruinning a large set of PDB structures (with deposited diffraction data) every time a new XDS binary became available and reported those findings to the XDS developers, our users (mainly through **consortium-discuss mailing list**, but also through **proc-announce mailing list** for all our academic users) and to synchrotron/beamline staff (through our **synchrotron-announce mailing list**). This quickly settled from an initial 222 PDB identifiers into a set of 60:

```
4FQN 4QKI 5AUI 5E9I 5FBO 5OD9 5ONZ 5RG0 5SP6 5VZR 6BLI 6CK7 6CW0 6DEX 6NQY 6P8P 6P8U 6R16 6TPI 6UCA
6VZQ 6VZW 7AO5 7DK1 7KDS 7MJB 7S87 7SY9 7TM9 7UV5 7WDA 7Z1V 7Z1Y 7Z41 8AGQ 8B1N 8BXT 8DA3 8E5V 8E60
8EGN 8EPS 8EW7 8FG7 8FT8 8GCA 8K4Q 8PQC 8R5Q 8RCC 8SDW 8SHR 8SIO 8SLU 8SO5 8TCA 8UFN 8VEV 8WT3 9CPL
```

- Two of those always failed (8E5V, 8E60) while 8SLU sometimes worked. They were left in to also test error handling in autoPROC itself. But realistically we had 57 test datasets:
  - 0.97 - 2.8A diffraction limits
  - Eiger 1M, Eiger 9M, Eiger 16M, Pilatus 2M, Pilatus 6M
  - 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.40, 0.50 and 1.0 image width
  - 0.774880 - 1.603110 Å wavelength
  - 180 - 3600 images (03/2012 - 09/2023)
  - ALS (5.0.1, 5.0.2, 8.3.1), APS (19-ID, 22-ID, 23-ID-D, 24-ID-C, 24-ID-E), Diamond (I03, I04-1), ELETTRA (11.2C, 5.2R), ESRF (ID23-2, ID29, ID30A-1, ID30B, MASSIF-3, NSLS X25), SSRL (12-2, BL9-2)

Using common software stack(s)

- Throughout the tests and analysis, various tools and jiffies have been developed (some not yet in the released autoPROC version) that have been very useful in keeping an eye on changes and performance.
- Although we don't recommend the (non-default) use of data that went through either XDS/CORRECT or XSCALE, a lot of comparison and analysis did involve data and analysis from XDS/CORRECT:
  - This way, the XDS developers and we were looking at the same software stack throughout.
  - The effects on the default and recommended autoPROC (XDS/INTEGRATE - AIMLESS - STARANISO) path might be different.
  - We were mainly interested in analysing any changes in the actual integration (sine as far as we know nothing changed on the scaling, rejection, merging part of CORRECT/XSCALE).

Lessons learned #5

- For the full history of the type of analysis we made public: see autoPROC wiki at https://www.globalphasing.com/autoproc/wiki/

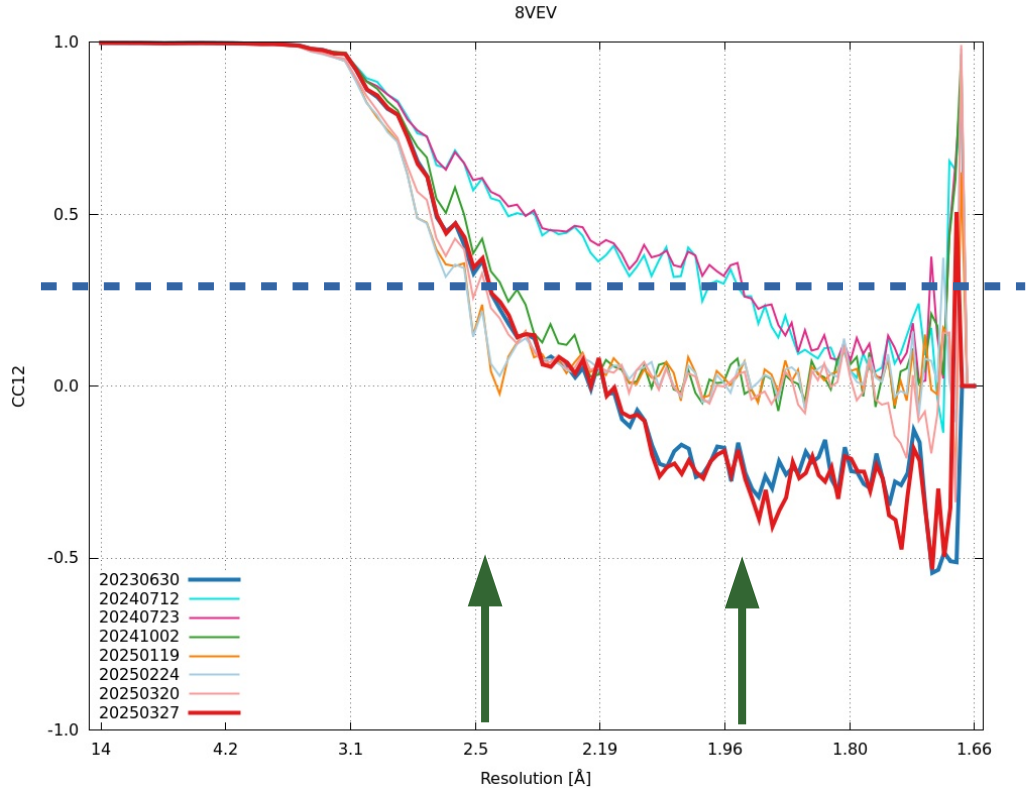
2024/09 -

2025/02 -

2025/03 -

2025/04 -

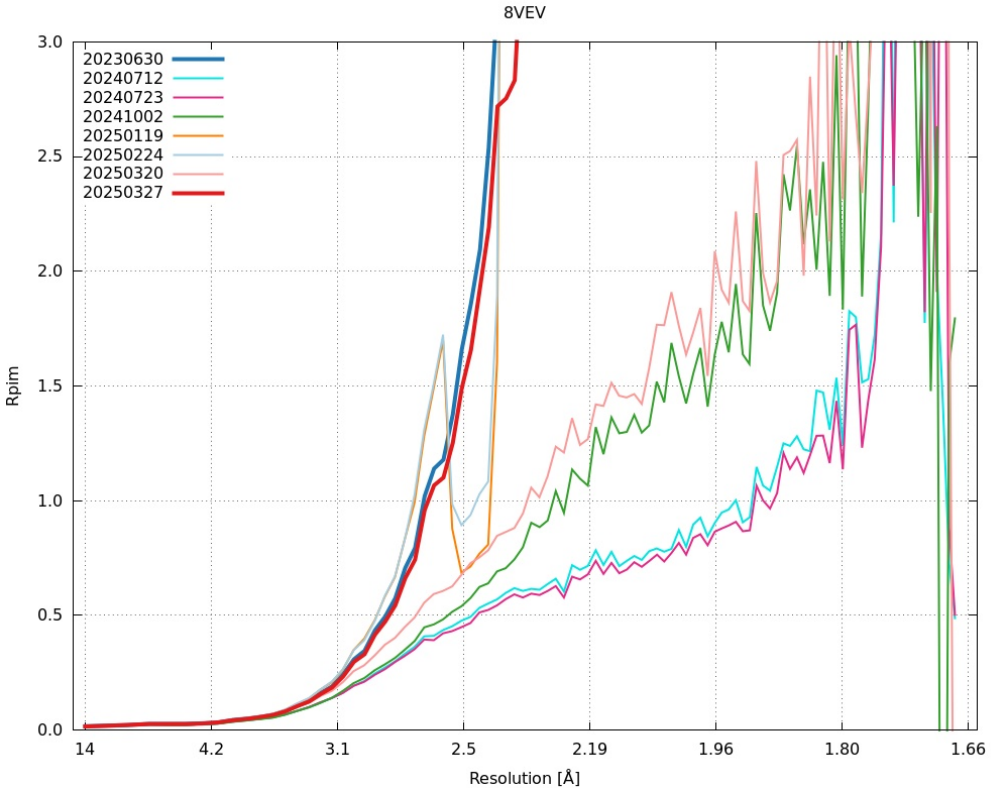
Misleading statistics - 8VEV



- CC(1/2) versus resolution
- XDS/CORRECT (XDS_ASCII.HKL)
- If using a spherical cut-off based on CC(1/2)>0.3: 2.5 or 1.9 Å

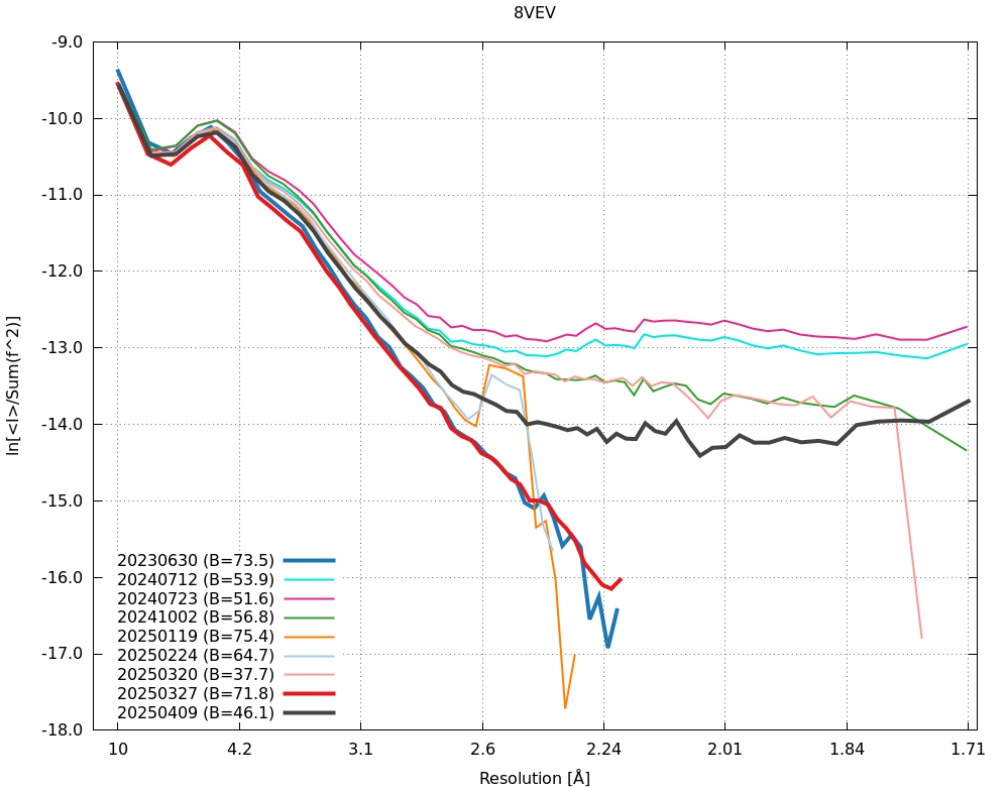
- CC(1/2) provides correlation of symmetry-related measurements of unique reflections
- If background handling involves some radial averaging/smoothing these measurements are no longer independent and artificial correlations can be introduced.

Misleading stats - 8VEV



- Rpim versus resolution
- XDS/CORRECT (XDS_ASCII.HKL)
- For certain XDS versions those merging R-values no longer “explode” (as expected).

Misleading stats - 8VEV



- Wilson plot
- XDS/CORRECT (XDS_ASCII.HKL)
- XDSCONV - CTRUNCATE
- For most XDS versions we see a very odd and unexpected shape
- Fitting a straight line to it (with poor correlation) leads to an artificially low B-factor:
  - If a new algorithm could really change the Wilson B from 74 to 54 (or even below 40), it would be a massive improvement
  - Goal: determine **correct** value (not to get one that is deemed “better”).

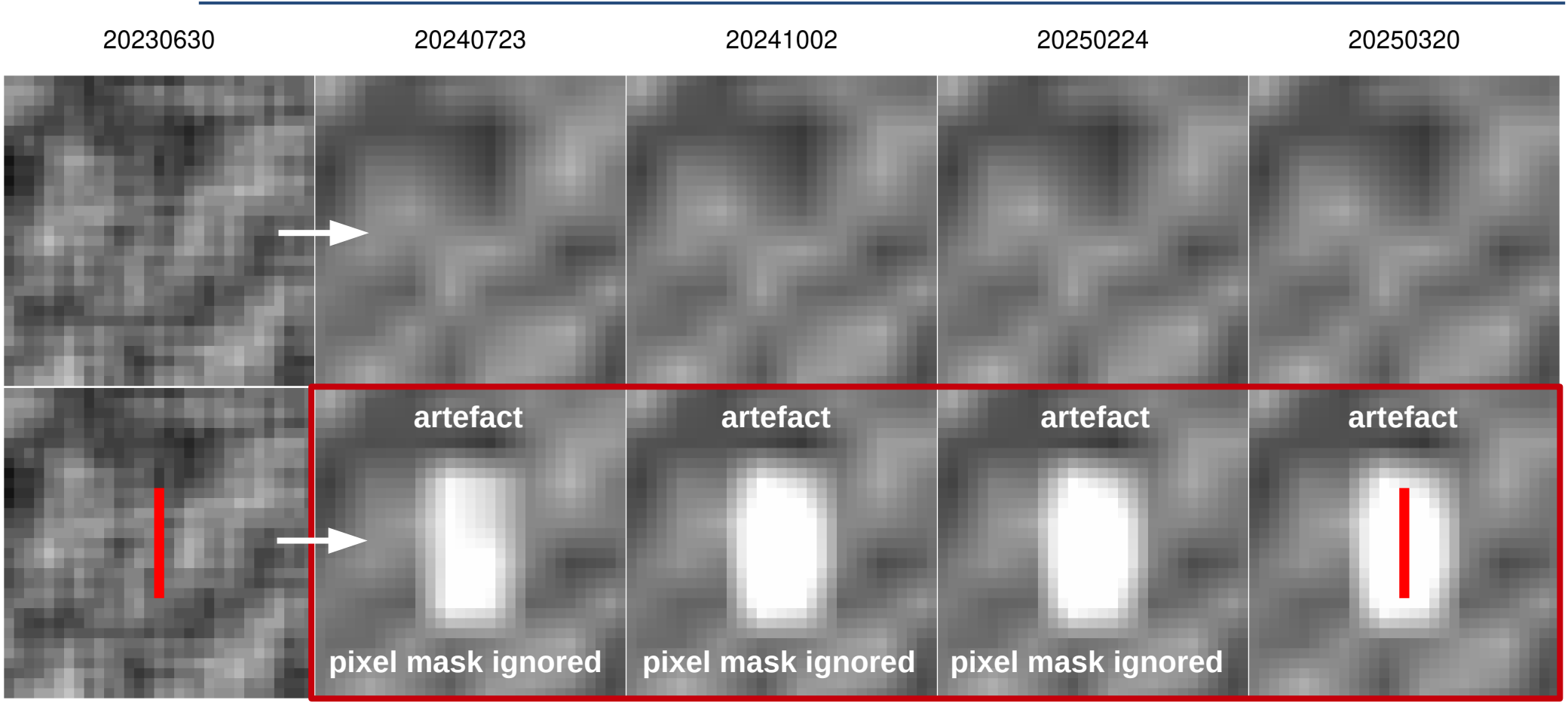
Lessons learned #6

- Careful when selecting metric to asses data quality!
- Ideally use something independent of the part to be assessed.
  - Changes in background handling during integration will impact both intensities but especially also sigmas.
  - Metrics like ISa, <I/sigI>, R{pim,meas}, CC(1/2) are highly influenced by error model (remember inverse-variance-weighted mean!).
  - Wilson plot or twinning statistics are much more independent and make good validators.
  - It is not just the Wilson B one has too look at, but consider the full story:
    - we assume that there is a linear region where one can fit a line and use that to estimate a B factor
    - if there is no fairly linear region, nothing coming out automatically afterward matters or should be considered

Background creation

- The initial background (INIT) is used for spot search (COLSPOT) and as input to DEFPIX (masking of e.g. beamstop shadow)
- The background (after DEFPIX) is then used in INTEGRATE for
  - scaling of each image (based on background)
  - **new**: to define background for integration itself
- Problems in creation and use of that background can have knock-on effects
- Test this by manually adding some masked (damaged) pixels to existing dataset and compare the resulting background images - also between versions

Background creation



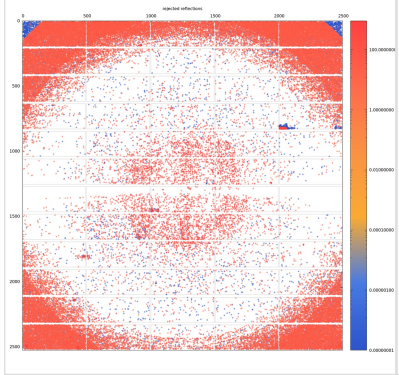
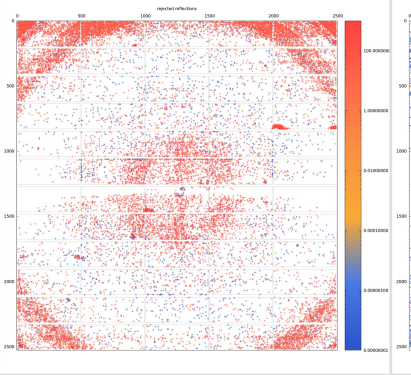
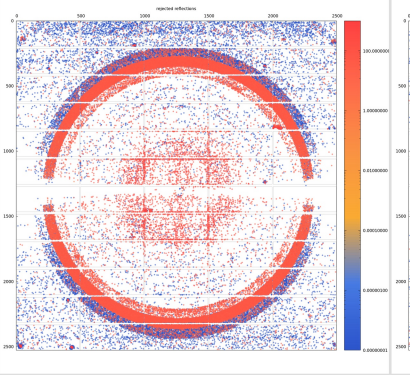
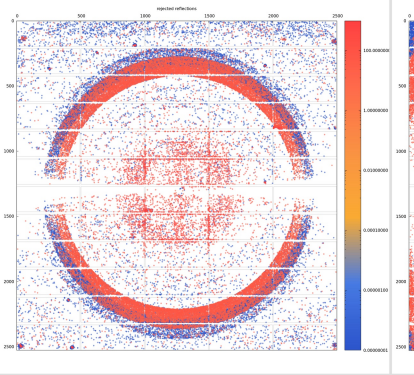
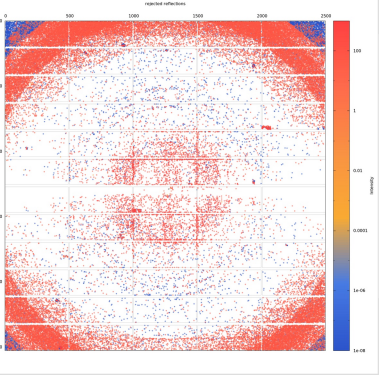
Lessons learned #7

- Working closely with raw diffraction images a must: reliably getting the exact counts as seen by the integration routine.
- Image simulation and editing can help testing at a very detailed and local level.

GΦL
Global Phasing Limited

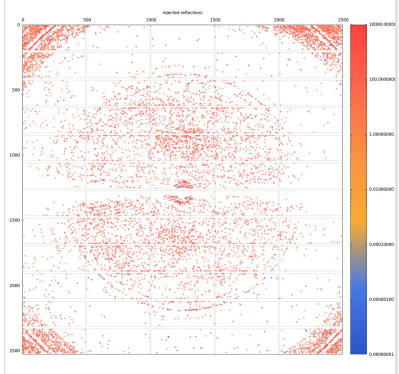
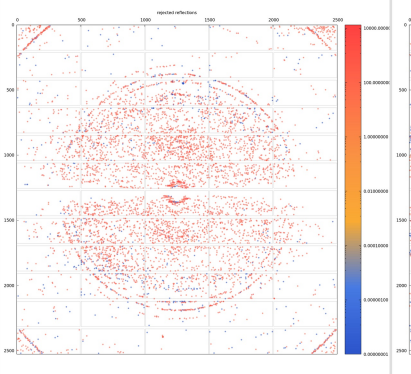
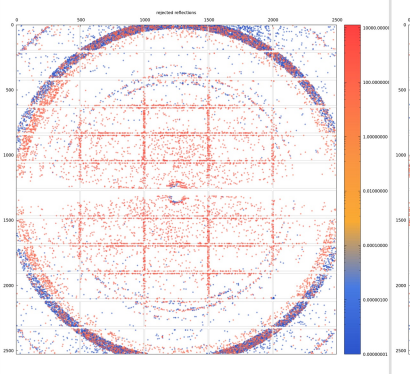
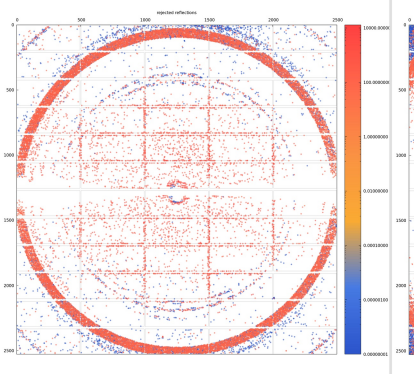
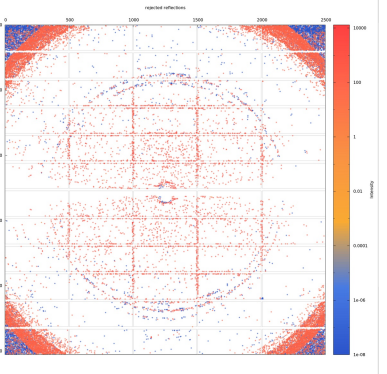
Artefact knock-on effects - rejections in XDS/CORRECT

8SO5

20230630	20241002	20250119	20250224	20250320
				

Intensity

6UCA

20230630	20241002	20250119	20250224	20250320
				

Intensity

Software stack at ESRF

- Not documented, but one can get a rough idea (if access/account is available)
- Version information by scraping logfiles on disk:

```
Jul 24 2024   20230630
Aug 27 2024   20240723
Sep 23 2024   20240723
Oct 17 2024   20240723
Oct 28 2024   20241002
Nov 26 2024   20241002
Dec 20 2024   20241002
Jan 23 2025   20241002
Feb  7 2025   20241002
May  5 2025   20250327
May 19 2025   20250327
May 27 2025   20250430
```

Lesson learned #9

- **Beamlines need to behave more like software**
  - Finally used data is not always processed directly at synchrotron (by auto-processing pipelines with specific beamline configurations and settings): a large number of power users take raw data and process at home (for consistency and control).
  - FAIR principles mean that someone else can make sense of data even N years later.
  - **Versioning** (to have a handle of beamline X at date Y)!
  - **Change log** (keeping track of changes).
  - **News** channel.
  - **Known issues** (no software or instrument is always perfect)!
  - The typical beamline-specific (externally visible) website is often depressingly inadequate:
    - often contains the CAD drawing of the original design
    - some highly technical information about mirrors or flux (of interest mainly to other beamline scientists?)
    - inadequate and out-of-date information about detectors or currently used software stacks
    - always mentions the need to reference/acknowledge the use of the beamline (but little about external software used and their need for citation)

Lesson learned #10

- Existing MX algorithms have been extremely well tested and debugged
  - XDS must've been run against 10s of Millions of datasets and is probably the most popular MX data processing package at the moment.
  - Any change or new development has to show that it is at least as good as the previous version:
    - which becomes more and more difficult with mature algorithms
    - requiring large number of tests and probably community efforts
    - this means that existing, known-to-work versions have to be maintained as a means to run comparisons (probably for years)
  - that is a definite maintenance/support burden

Lesson learned #11

- The **work going into method developments**, testing, maintaining and supporting those MX workhorses has to be **properly acknowledged**.
  - Not just in papers (or at least in supplementary material) and PDB depositions
  - Also in tools that use PDB depositions (wwPDB websites, overall statistics about software used etc)
  - And in synchrotron interfaces (ISPyB, EXI, ICAT, SynchWeb) and beamline web pages.
- Failing to do so creates the wrong impressions:
  - “autoPROC is a script that runs XDS” (no need to cite autoPROC ... XDS and AIMLESS is enough)
  - “If there isn't even a reference/citation for program X at the beamline or on a wwPDB website: why should I make the effort to hunt this down and provide it in my paper or during deposition?”
  - Licence conditions are optional (use of autoPROC requires citation) - since often invisible to synchrotron user.
  - "Was nichts kostet ist nichts wert.", i.e. if something doesn't cost anything it isn't worth anything.
- We should be leading by example - this is not a mere pecadillo (“Kavaliersdelikt”)