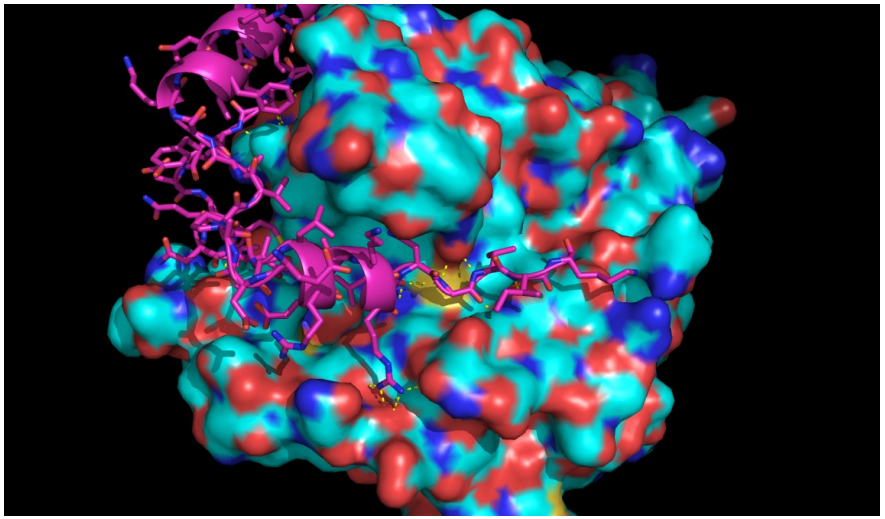
Introduction to CrystFEL

Data processing for serial crystallography



Thomas White

BioXFEL Data Analysis Workshop
Berkeley, 21 Aug 2014

HELMHOLTZ ASSOCIATION

CFEL SCIENCE

DESY

Contributors to CrystFEL

- Led by me, Thomas White (Center for Free-Electron Laser Science, DESY Hamburg).
- Code contributions from:
  - Rick Kirian (ASU, now CFEL), Nadia Zatsepin (ASU).
  - Karol Nass (CFEL, now MPI-MF Heidelberg), Andrew Aquila, Andrew Martin, Lorenzo Galli, Kenneth Beyerlein, Chun Hong Yoon, Cornelius Gati, Anton Barty, Fedor Chervinskii, Valerio Mariani (CFEL).
  - Takanori Nakane (MRC-LMB).
  - Wolfgang Brehm (U Konstanz).
- Testing/feedback from the above and many additional people, especially:
  - Francesco Stellato, Markus Metz, Dominik Oberthur, Oleksandr Yefanov (CFEL).
  - Shibom Basu, Haiguang Liu (ASU).
- Words of wisdom from many people, in particular:
  - Henry Chapman (CFEL), Petra Fromme (CFEL), James Holton (ALS/UC Berkeley), John Spence (ASU/LBL), Kay Diederichs (U Konstanz).

What is CrystFEL?

- CrystFEL: data processing for “snapshot serial” crystallography (usually using an FEL, but not necessarily).
- Development since late 2009. Currently at version 0.5.3a.
- Licence: GPLv3 (*free* and *open source* software)
- User-oriented development.
- **http://www.desy.de/~twhite/crystfel**

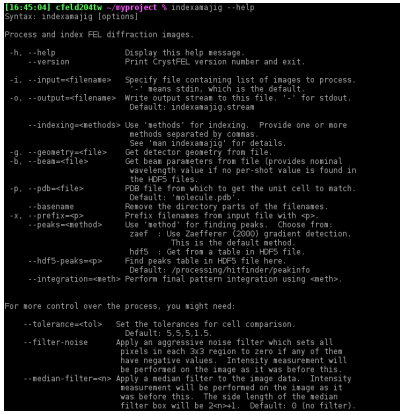
“Official” citation: White et al., J. Appl. Cryst. 45 (2012) p335

What do you get? Core programs

indexamajig	Bulk indexing and integration of patterns
ambigator	Resolve indexing ambiguities
process_hkl	Merge using “Monte Carlo” technique
hdfsee	View images in HDF5 format (with geometry)
cell_explorer	Examine unit cell parameter distributions
compare_hkl	Compare merged reflection data
check_hkl	Evaluate merged reflection data
pattern_sim	Simulate diffraction patterns
partial_sim	Simulate partial reflection intensities
get_hkl	Reflection data “Swiss army knife”
render_hkl	Render plane sections of reciprocal space
partialator	Merge using partialities and post-refinement

What do you get? Documentation

- **Manual pages:** `$ man indexamajig`
- **Help mesages:** `$ indexamajig --help`
- **Website, particularly: tutorial, best practice, FAQ**
- **Mailing list for announcements**
- **User support**



What do you get? Scripts

> Auxiliary scripts for plotting graphs, collating results etc.

> Difference from core programs: you're meant to copy them to your working directory and modify them

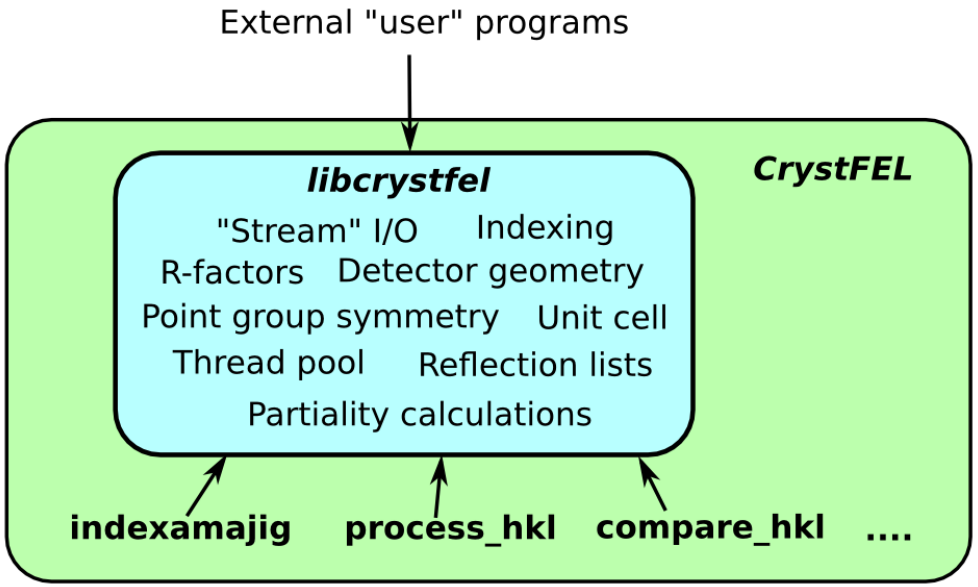
```
1 #!/usr/bin/perl -w
2
3 use strict;
4 use File::Basename;
5
6 my $args = join(" ", splice(@ARGV, 1, scalar(@ARGV)-1));
7 if ( !($args eq "") ) {
8         printf("Extra arguments for hdfsee: %s\n", $args);
9 } else {
10        # Default arguments - feel free to override!
11        $args = "--binning=2 --int-boost=10";
12        printf("Using default arguments for hdfsee: %s\n", $args);
13 }
14
15 open(FH, $ARGV[0]);
16 open(TMP, "> list.tmp");
17
18 my $in_image = 0;
```

What do you get? libcrystfel

> libcrystfel: shared library exposing high-level functions, e.g. “index this pattern” or “calculate these partialities”.

> Possible application: use CrystFEL for online data processing within a facility's monitoring software.

> API docs on website



Where can you get it?

- From the website: http://www.desy.de/~twhite/crystfel
  (or just type “crystfel” into your favourite search engine)

- “Release” versions: tested with a series of regression tests

- Git repository: very latest code, maybe broken, maybe awesome, slightly harder to install (need autotools installed).

- Packaged snapshots: latest code,

- Ubuntu, OpenSUSE and Fedora packages. MacPort. Soon: Debian.

Where can you get it?

> GNU General Public License (GPL):
"*free*" as in "*freedom*" (also as in "*free beer*")

> Freedom to use the software *for any purpose* (even commercially).

> Freedom to study how the software works.

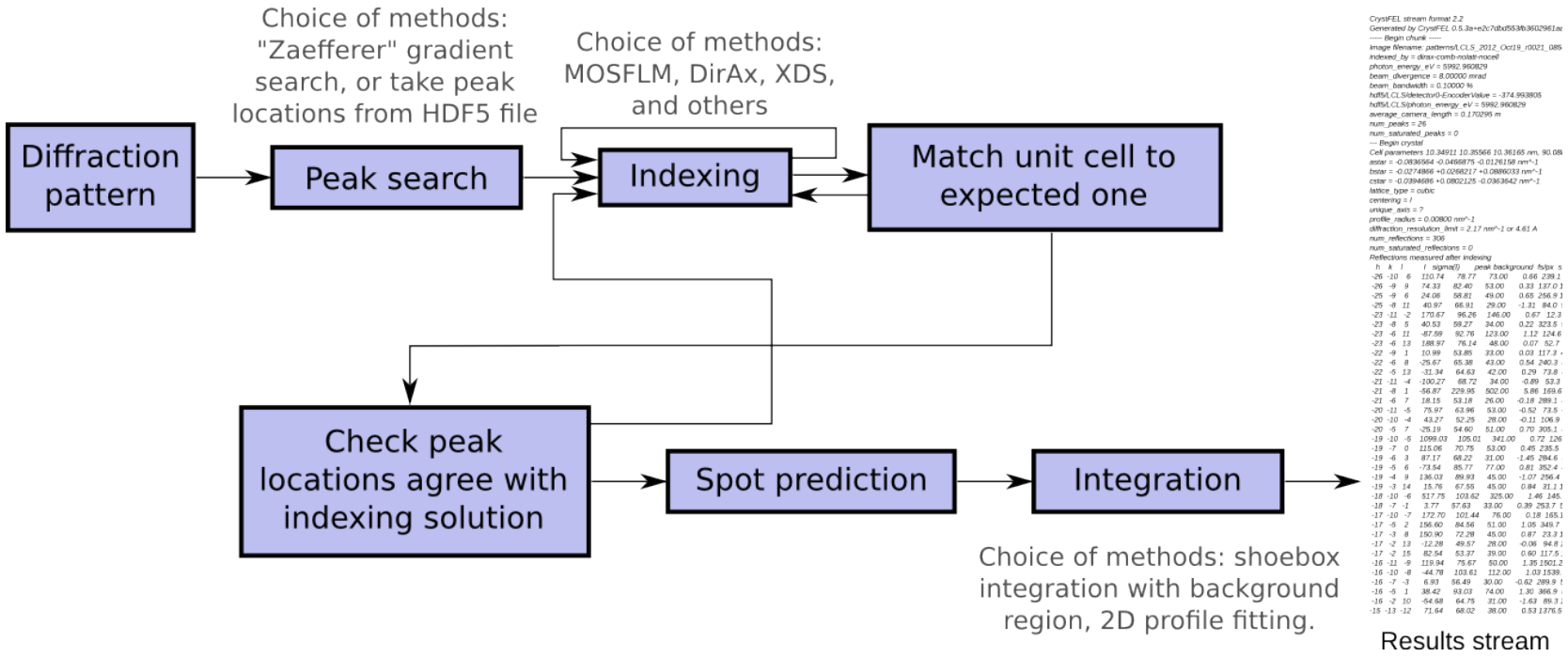
> Freedom to change the software, e.g. to fix bugs or add features.

> Freedom to share the software, even a modified version.

> No license forms to sign, send etc.

> The GPL ensures that these rights are never restricted.

Pipeline in “indexamajig”



> Input: HDF5 (very common scientific format, many language bindings)

> Note: choice of methods for **peak search**, **indexing** and **integration**.

> Output: plain text “stream” - don't be afraid to look inside!
(tip: use “less”, not “gedit” - file can be very large)

Peak search methods

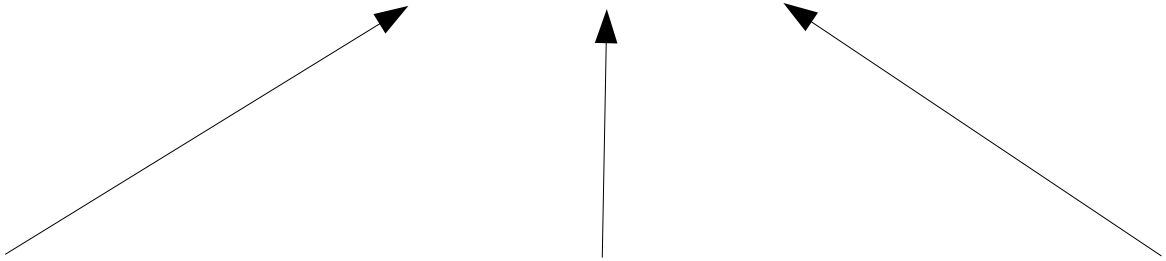
> `indexamajig [....] --peaks=hdf5 [....]`
>
> `....` means “get peak list from HDF5 file”.

> `indexamajig [....] --peaks=zaef [....]`
>
> `....` means “use internal peak search algorithm”.

Indexing methods

> **`indexamajig [...] --indexing=method1,method2,.. [....]`**
List of indexing methods will be tried in order.

> **Example method:** **`mosflm-raw-latt`**



Invoke mosflm to index the pattern using the peaks from the peak search

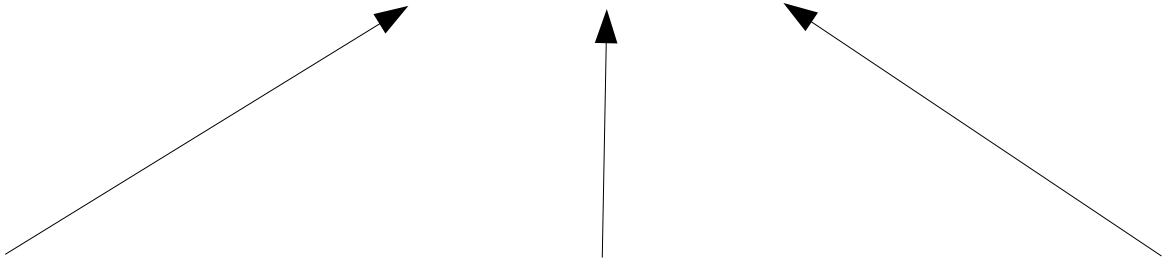
Skip the step where the unit cell parameters would be checked against the reference

Take the Bravais lattice information from the reference into account, e.g. “look for tetragonal P cells only”

Indexing methods

> **`indexamajig [...] --indexing=method1,method2,.. [....]`**
> **List of indexing methods will be tried in order.**

> **Example method:** **`mosflm-axes-latt`**



Invoke mosflm to index the pattern using the peaks from the peak search

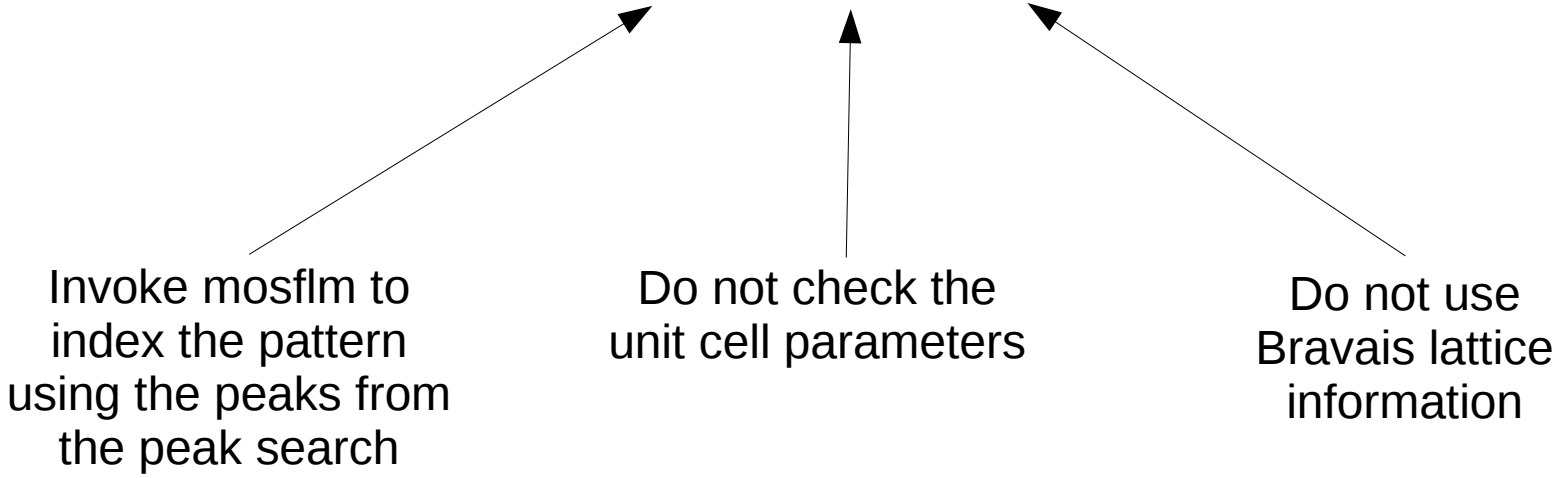
Check the unit cell parameters against the reference

Take the Bravais lattice information from the reference into account, e.g. “look for tetragonal P cells only”

Indexing methods

> **`indexamajig [...] --indexing=method1,method2,.. [....]`**
> **List of indexing methods will be tried in order.**

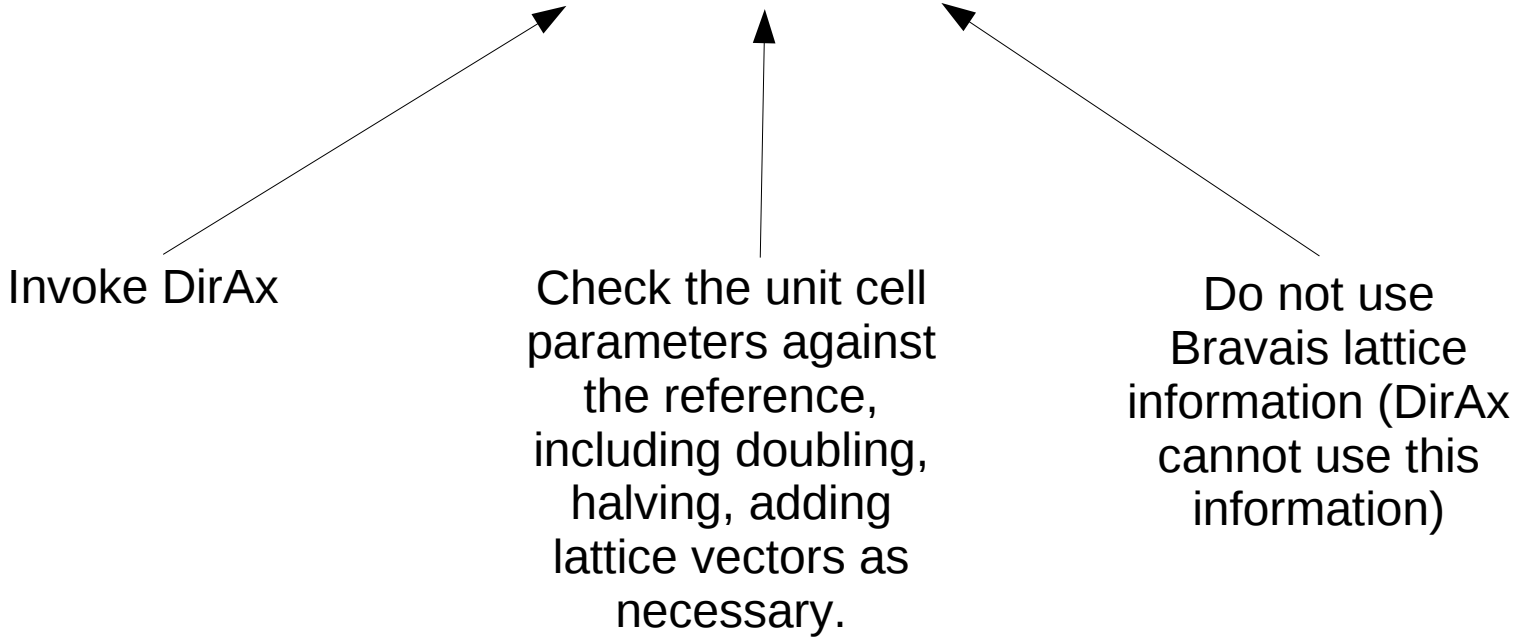
> **Example method:** **`mosflm-raw-nolatt`**



Indexing methods

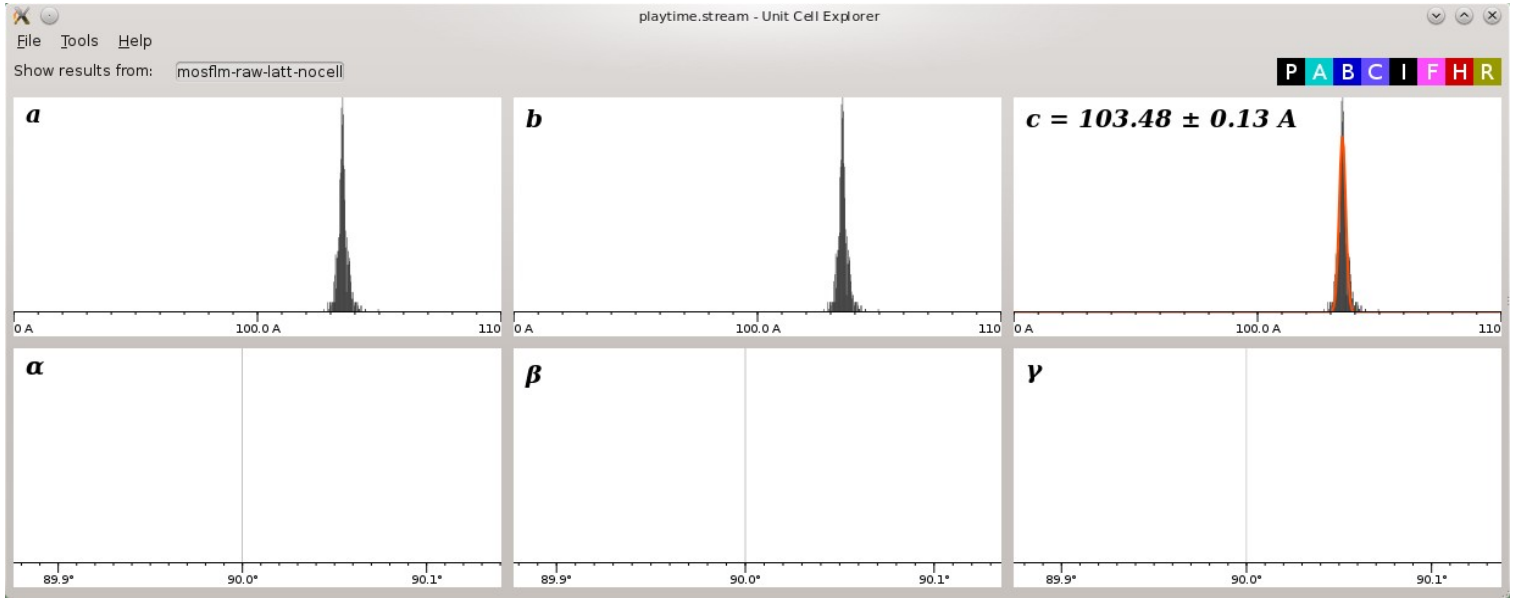
> **`indexamajig [...] --indexing=method1,method2,.. [....]`**
> **List of indexing methods will be tried in order.**

> **Example method:** **`dirax-comb-nolatt`**



The Unit Cell Explorer

```
> cell_explorer output.stream
```



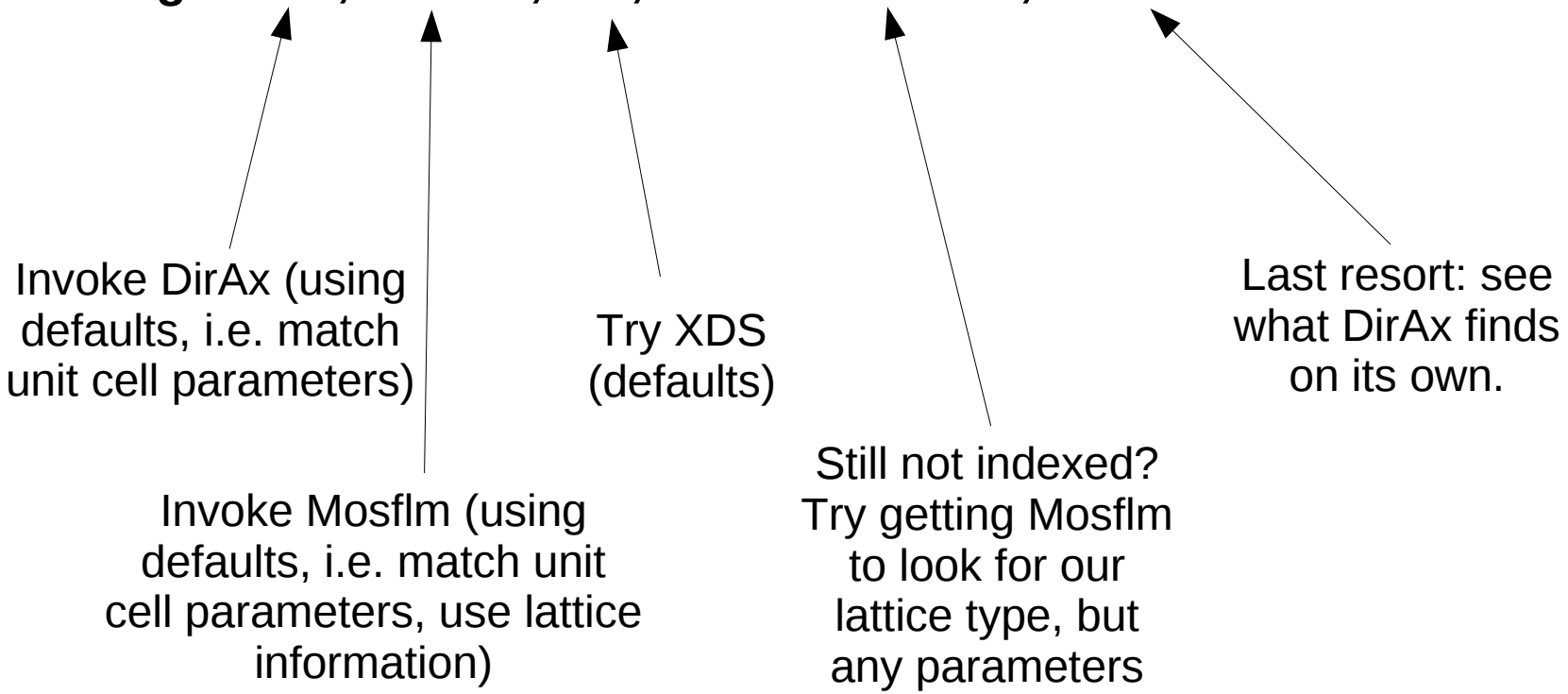
a = b = c = 103.5 Å
α = β = ɣ = 90.0°
Body centered (I).

Possible space groups: I23, $I2_13$, I432, $I4_132$, I422 $I4_122$, $I4_1$, I4, I222

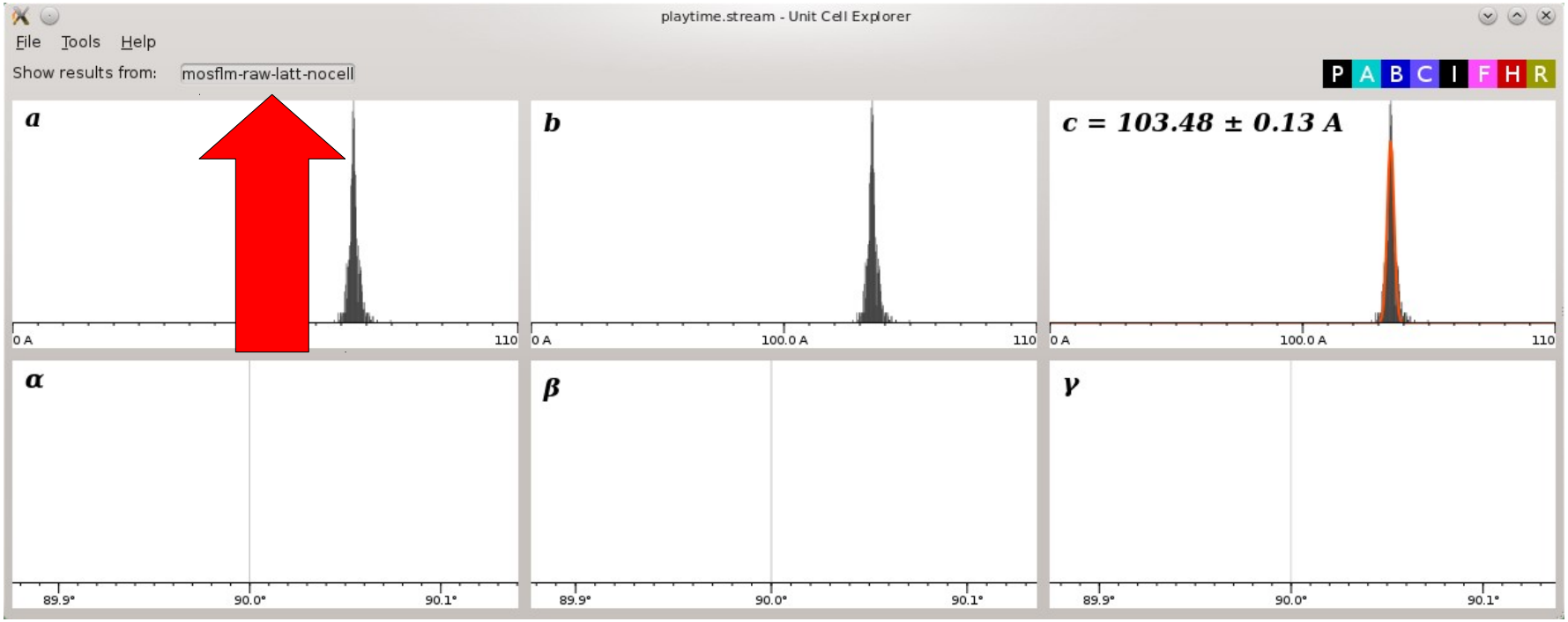
Indexing methods

> **`indexamajig [...] --indexing=method1,method2,.. [....]`**
> **List of indexing methods will be tried in order.**

> **--indexing=dirax,mosflm,xds,mosflm-raw-latt,dirax-raw**



The Unit Cell Explorer



Geometry and beam files

```
> indexamajig [...] --beam=lcls.beam --geometry=cspad.geom [....]
```

```
beam/bandwidth = 0.001
beam/divergence = 0.008
profile_radius = 0.008e9
beam/photon_energy = /LCLS/photon_energy_eV
beam/photon_energy_scale = 1.0

beam/fluence = 1.0e12
beam/radius = 10.0e-6
```

Important for spot prediction

Used only by pattern_sim - ignored for indexing

Beam files will probably go away in CrystFEL 0.6.0.

A menace ☹

"man crystfel_geometry", templates distributed with CrystFEL, or ask a friendly PCS collaborator, beamline scientist or us.

Analysis in CrystFEL can be surprisingly robust to inaccurate geometry.

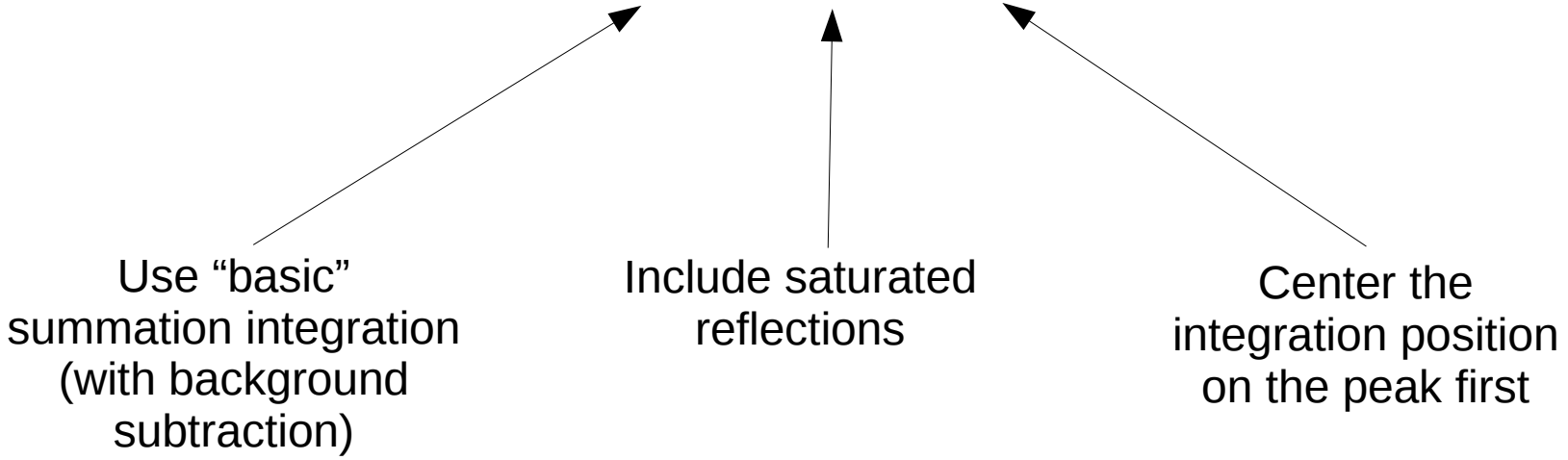
"Calibration mode" in hdfsee.

Geometry refinement tool to be added in CrystFEL 0.6.0.

Integration methods

> **`indexamajig [...] --integration=method [....]`**
> **(only one integration method allowed)**

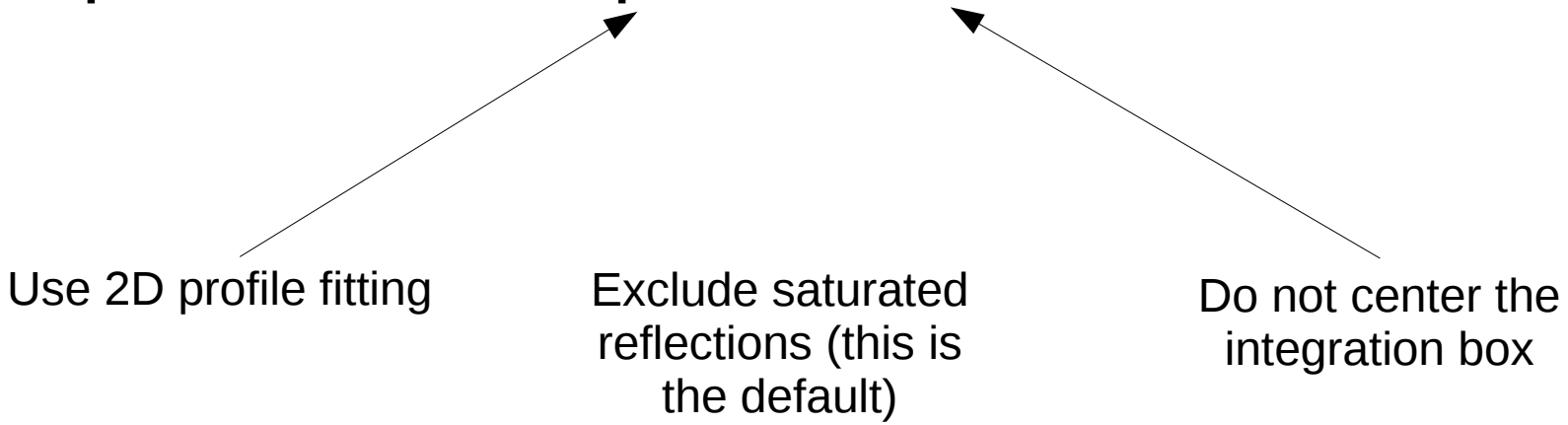
> **Example method:** **`rings-sat-cen`**



Integration methods

> **`indexamajig [...] --integration=method [....]`**
> **(only one integration method allowed)**

> **Example method:** **`prof2d-nocen`**

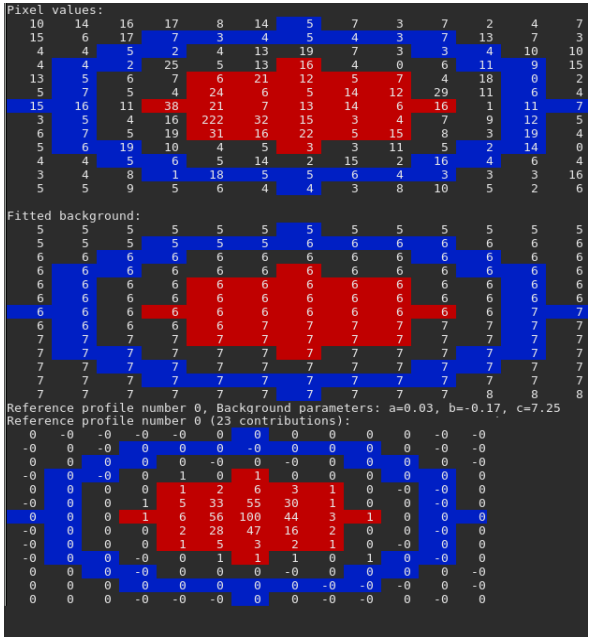


Integration diagnostics

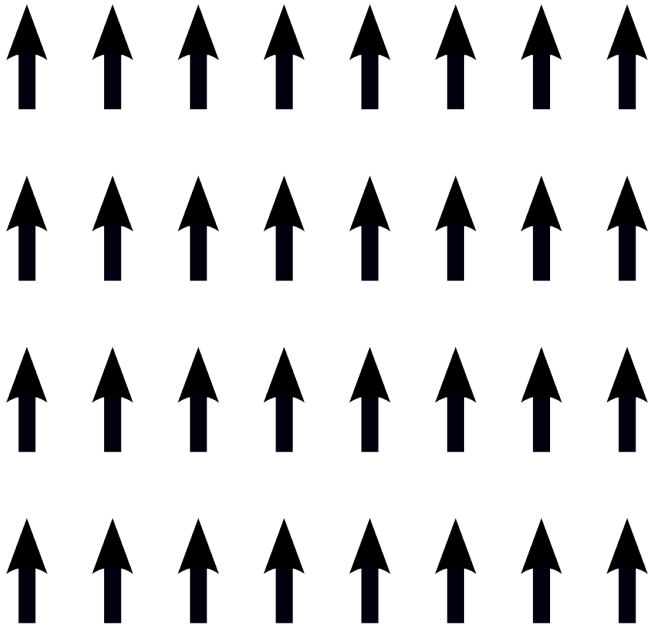
> **`indexamajig [...] --int-diag=<something>`**

> <something> can be:

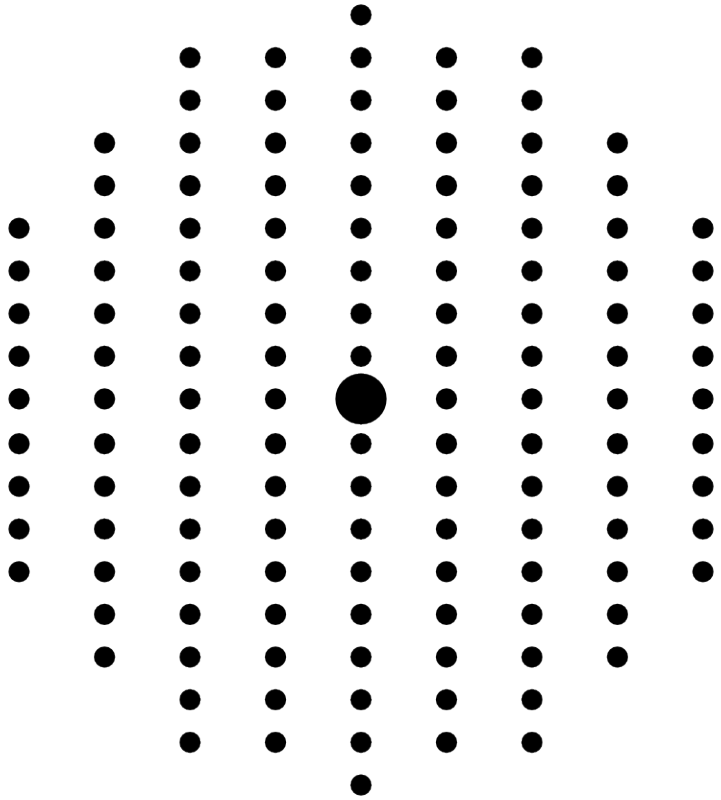
- `none` (this is the default)
- A set of indices, e.g. `4,3,5`
- `random` – show a random 1% of reflections
- `implausible` – show reflections with I < -5sigma
- `negative` – show reflections with I < -3sigma



Indexing ambiguities

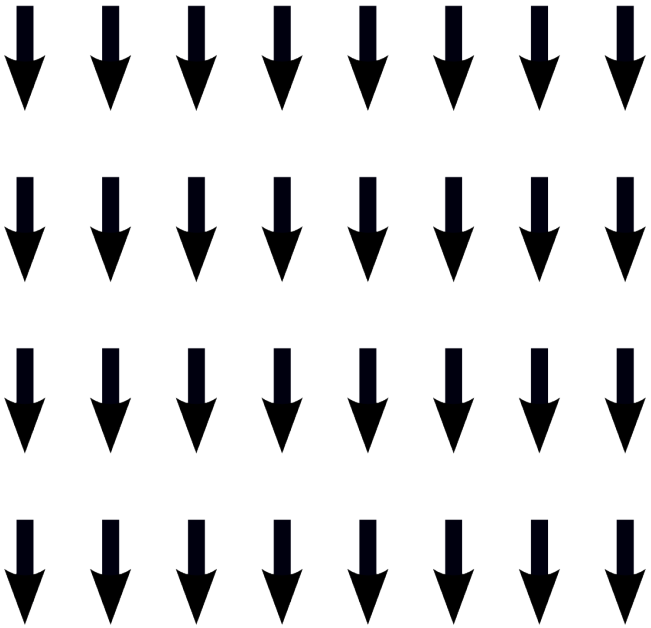


A crystal structure

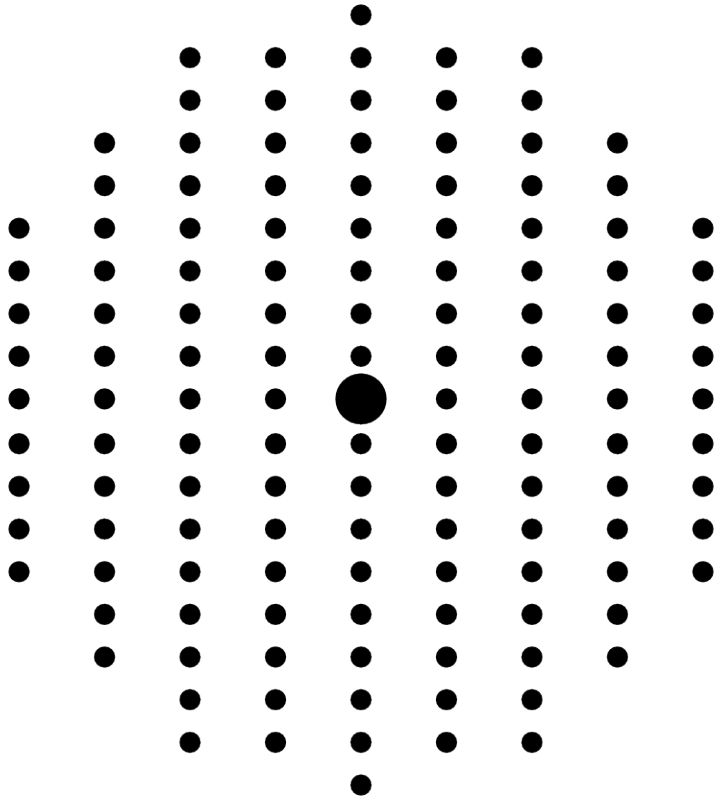


Schematic diffraction pattern

Indexing ambiguities

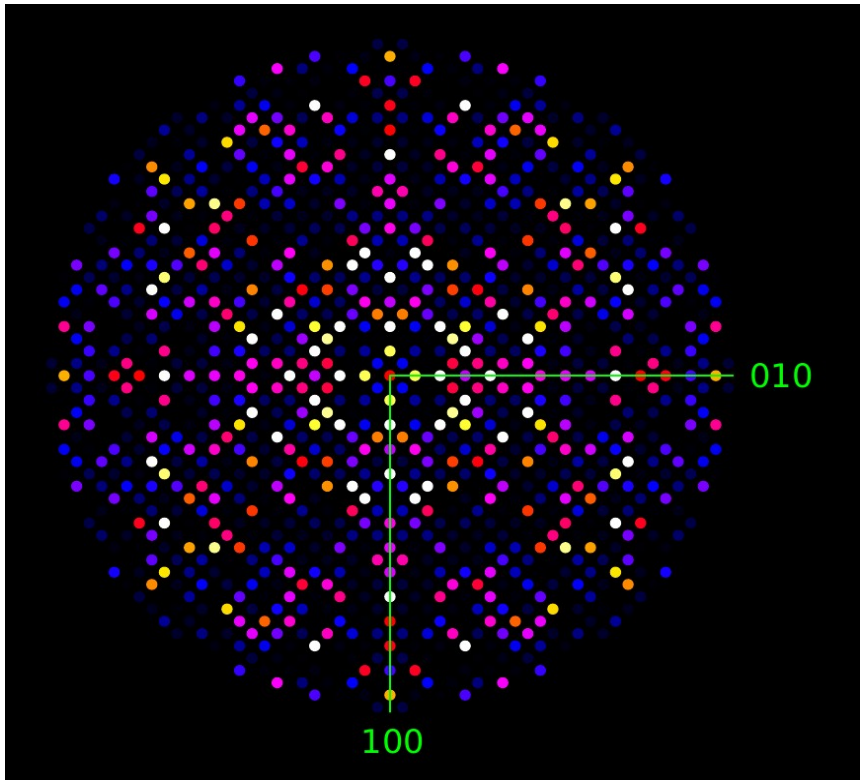


A crystal structure

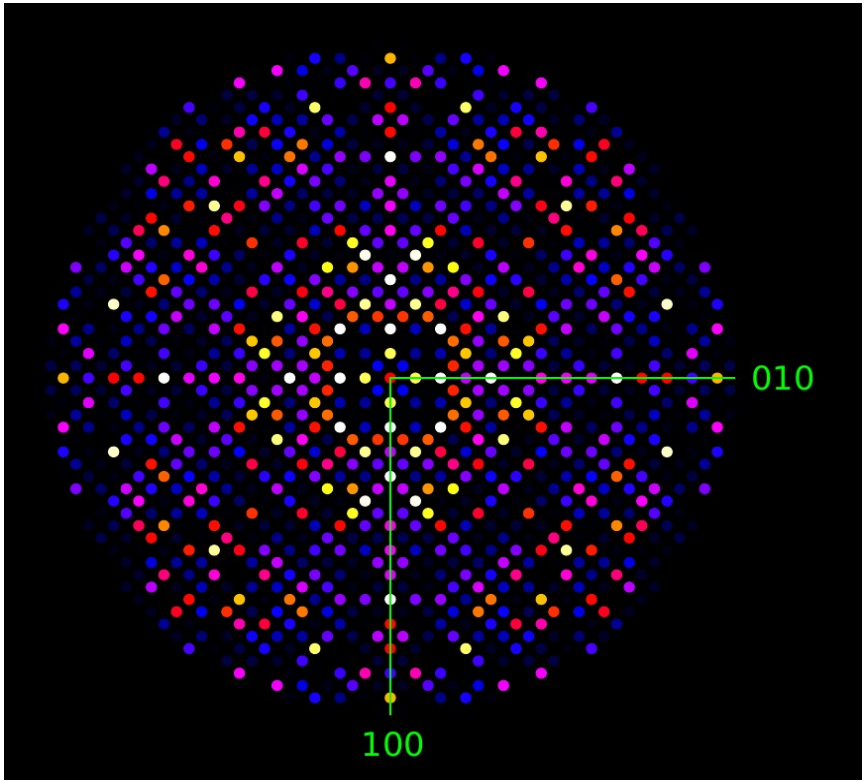


Schematic diffraction pattern

Indexing ambiguities



Actual zone axis intensities
Twofold symmetry



Zone axis intensities if indexing ambiguity is not resolved.
Fourfold symmetry

Indexing ambiguities

Triclinic lattice

$\bar{1}$	**1**	$P\bar{1}$	**P1**

Monoclinic lattice

	m		Pm, Pc, Cm, Cc
2	2/m	**P2, $P2_1$, C2**	P2/m, $P2_1$/m, C2/m, P2/c, $P2_1$/c, C2/c

Orthorhombic lattice

	mm2		Pmm2, $Pmc2_1$, Pcc2, Pma2, $Pca2_1$, Pnc2, $Pmn2_1$, Pba2, $Pna2_1$, Pnn2, Cmm2, $Cmc2_1$, Ccc2, Amm2, Aem2, Ama2, Aea2, Fmm2, Fdd2, Imm2, Iba2, Ima2
222	mmm	**P222, $P222_1$, $P2_12_12$, $P2_12_12_1$, $C222_1$, C222, F222, I222, $I2_12_12_1$**	Pmmm, Pnnn, Pccm, Pban, Pmma, Pnna, Pmna, Pcca, Pbam, Pccn, Pbcm, Pnnm, Pmmn, Pbcn, Pbca, Pnma, Cmcm, Cmce, Cmmm, Cccm, Cmme, Ccce, Fmmm, Fddd, Immm, Ibam, Ibca, Imma

Tetragonal lattice

	$\bar{4}$							
4	$\bar{4}2m$	$\bar{4}m2$	4/m	4mm	**P4, $P4_1$, $P4_2$, $P4_3$, I4, $I4_1$**	$P\bar{4}2m$, $P\bar{4}2c$, $P\bar{4}2_1m$, $P\bar{4}2_1c$, $I\bar{4}2m$, $I\bar{4}2d$	$P\bar{4}m2$, $P\bar{4}c2$, $P\bar{4}b2$, $P\bar{4}n2$, $I\bar{4}m2$, $I\bar{4}c2$	P4/m, $P4_2$/m, P4/n, $P4_2$/n, I4/m, $I4_1$/a	P4mm, P4bm, $P4_2$cm, $P4_2$nm, P4cc, P4nc, $P4_2$mc, $P4_2$bc, I4mm, I4cm, $I4_1$md, $I4_1$cd
422	4/mmm				**P422, $P42_12$, $P4_122$, $P4_12_12$, $P4_222$, $P4_22_12$, $P4_322$, $P4_32_12$, I422, $I4_122$**	P4/mmm, P4/mcc, P4/nbm, P4/nnc, P4/mbm, P4/mnc, P4/nmm, P4/ncc, $P4_2$/mmc, $P4_2$/mcm, $P4_2$/nbc, $P4_2$/nnm, $P4_2$/mbc, $P4_2$/mnm, $P4_2$/nmc, $P4_2$/ncm, I4/mmm, I4/mcm, $I4_1$/amd, $I4_1$/acd		

Rhombohedral lattice

3	$\bar{3}$	3m	**R3 (H3)**	R$\bar{3}$ (H$\bar{3}$)	R3m (H3m), R3c (H3c)
32	$\bar{3}$m		**R32 (H32)**	R$\bar{3}$m (H$\bar{3}$m), R$\bar{3}$c (H$\bar{3}$c)	

Hexagonal lattice

3	$\bar{3}$			**P3, $P3_1$, $P3_2$**	$P\bar{3}$			
6	3m1	$\bar{6}$	31m	**P6, $P6_1$, $P6_5$, $P6_2$, $P6_4$, $P6_3$**	P3m1, P3c1	$P\bar{6}$	P31m, P31c	P6mm, P6cc, $P6_3$cm, $P6_3$mc
312	$\bar{3}$m1	$\bar{6}$m2	$\bar{3}$1m	**P312, $P3_112$, $P3_212$**	$P\bar{3}$m1, $P\bar{3}$c1	$P\bar{6}$m2, $P\bar{6}$c2	$P\bar{3}$1m, $P\bar{3}$1c	
321	6/m	62m		**P321, $P3_121$, $P3_221$**	P6/m, $P6_3$/m	$P\bar{6}2m$, $P\bar{6}2c$		
622	6/mmm			**P622, $P6_122$, $P6_522$, $P6_222$, $P6_422$, $P6_322$**	P6/mmm, P6/mcc, $P6_3$/mcm, $P6_3$/mmc			

Cubic lattice

23	$\bar{4}$3m	m$\bar{3}$	**P23, F23, I23, $P2_13$, $I2_13$**	P$\bar{4}$3m, F$\bar{4}$3m, I$\bar{4}$3m, P$\bar{4}$3n, F$\bar{4}$3c, I$\bar{4}$3d	Pm$\bar{3}$, Pn$\bar{3}$, Fm$\bar{3}$, Fd$\bar{3}$, Im$\bar{3}$, Pa$\bar{3}$, Ia$\bar{3}$
432	m$\bar{3}$m		**P432, $P4_232$, F432, $F4_132$, I432, $P4_332$, $P4_132$, $I4_132$**	Pm$\bar{3}$m, Pn$\bar{3}$n, Pm$\bar{3}$n, Pn$\bar{3}$m, Fm$\bar{3}$m, Fm$\bar{3}$c, Fd$\bar{3}$m, Fd$\bar{3}$c, Im$\bar{3}$m, Ia$\bar{3}$d	

Indexing ambiguities

```
> ambigator input.stream -o output.stream
            -y 6/m -w 6/mmm -j 8 --lowres=6 --highres=3
```

Actual symmetry (note: centre of symmetry added, i.e. consider Friedel pairs as equivalent)

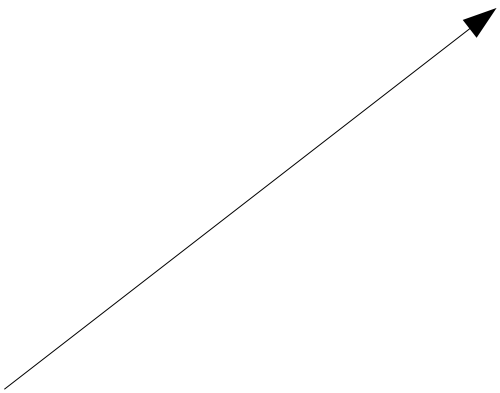
Apparent symmetry (can also give the ambiguity operator directly, using --operator)

Use 8 threads for calculations (go faster).

Resolution range (in A) to be considered. Needs to be big, but not too big.

Merging

```
> process_hkl -i input.stream -o output.hkl -y 6/m
```

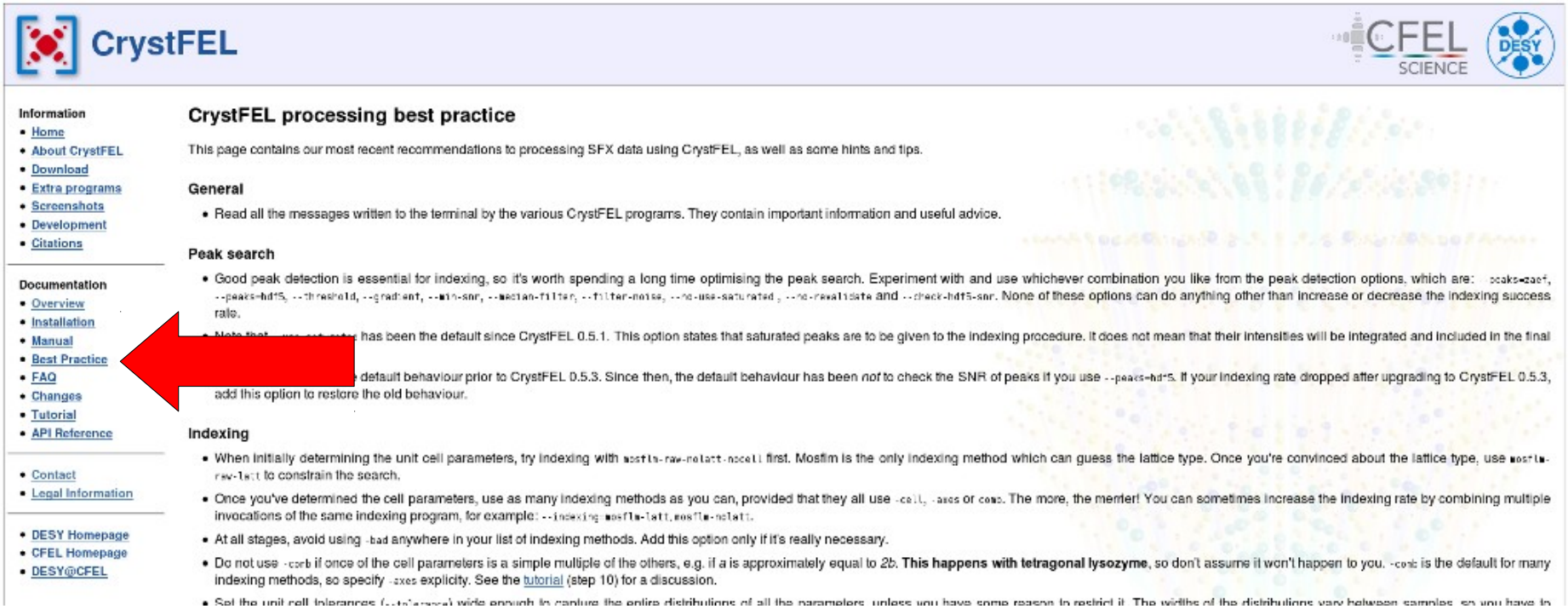


Output: text file, can be imported into most structure solution programs. Alternatively: “create-mtz” and “create-xscale” scripts.

Actual symmetry (note: centre of symmetry added, i.e. merge Friedel pairs)

Confused about point group notation? “man crystfel” for a list of all possibilities.

“Best practice” guidelines



CrystFEL

CFEL SCIENCE DESY

Information
- Home
- About CrystFEL
- Download
- Extra programs
- Screenshots
- Development
- Citations

Documentation
- Overview
- Installation
- Manual
- Best Practice
- FAQ
- Changes
- Tutorial
- API Reference

- Contact
- Legal Information

- DESY Homepage
- CFEL Homepage
- DESY@CFEL

CrystFEL processing best practice

This page contains our most recent recommendations to processing SFX data using CrystFEL, as well as some hints and tips.

General

- Read all the messages written to the terminal by the various CrystFEL programs. They contain important information and useful advice.

Peak search

- Good peak detection is essential for indexing, so it's worth spending a long time optimising the peak search. Experiment with and use whichever combination you like from the peak detection options, which are: `--peaks=zaef`, `--peaks=hdf5`, `--threshold`, `--gradient`, `--min-snr`, `--median-filter`, `--filter-noise`, `--no-use-saturated`, `--no-revalidate` and `--check-hdf5-snr`. None of these options can do anything other than increase or decrease the indexing success rate.
- Note that `--no-sat-corr` has been the default since CrystFEL 0.5.1. This option states that saturated peaks are to be given to the indexing procedure. It does not mean that their intensities will be integrated and included in the final
- default behaviour prior to CrystFEL 0.5.3. Since then, the default behaviour has been *not* to check the SNR of peaks if you use `--peaks=hdf5`. If your indexing rate dropped after upgrading to CrystFEL 0.5.3, add this option to restore the old behaviour.

Indexing

- When initially determining the unit cell parameters, try indexing with `mosflm-raw-nolatt-nocell` first. Mosflm is the only indexing method which can guess the lattice type. Once you're convinced about the lattice type, use `mosflm-raw-latt` to constrain the search.
- Once you've determined the cell parameters, use as many indexing methods as you can, provided that they all use `-cell`, `-axes` or `cons`. The more, the merrier! You can sometimes increase the indexing rate by combining multiple invocations of the same indexing program, for example: `--indexing mosflm-latt,mosflm-nolatt`.
- At all stages, avoid using `-bad` anywhere in your list of indexing methods. Add this option only if it's really necessary.
- Do not use `-corb` if once of the cell parameters is a simple multiple of the others, e.g. if *a* is approximately equal to *2b*. **This happens with tetragonal lysozyme**, so don't assume it won't happen to you. `-cons` is the default for many indexing methods, so specify `-axes` explicity. See the tutorial (step 10) for a discussion.
- Set the unit cell tolerances (`--tolerance`) wide enough to capture the entire distributions of all the parameters, unless you have some reason to restrict it. The widths of the distributions vary between samples, so you have to

“Definitely use this option”
“Don't use this option except for testing”
“Try this option and see what happens, might be good or bad”
“Watch out for this 'gotcha'”
... etc ...

Future developments: CrystFEL 0.5.4 and 0.6.0

> Version 0.5.4: after this workshop, unless we find any show-stoppers.

- Incremental update, minor new features etc.

> Version 0.6.0: in a few months - bigger changes.

- Multiple images in one HDF5 file (use SLAC HDF5s directly, reduce load on filesystem).
- No more “beam” files.
- No more “-y” options (this can be determined automatically).
- ... some other goodies ...

Future developments: multi-lattice indexing

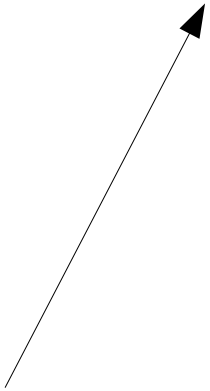
- Core of CrystFEL can already deal with any number of crystals per pattern.

- Limitation: underlying indexing algorithms.
  (indexing algorithms have not been a main theme of CrystFEL development - we take advantage of previous work by calling other programs).

- Coming soon (CrystFEL 0.6.0):
  - Multi-lattice MOSFLM
  - Felix (related to GrainSpotter / TotalCryst)
  - Multi-lattice indexing using internal “retry” mechanism (from Takanori Nakane).

Future developments: partialities and post-refinement

```
> partialator -i input.stream -o output.hkl -y mmm
              --model=sphere  --iterations=1
```



Partiality model (options: sphere, gaussian, scsphere, thin, unity).

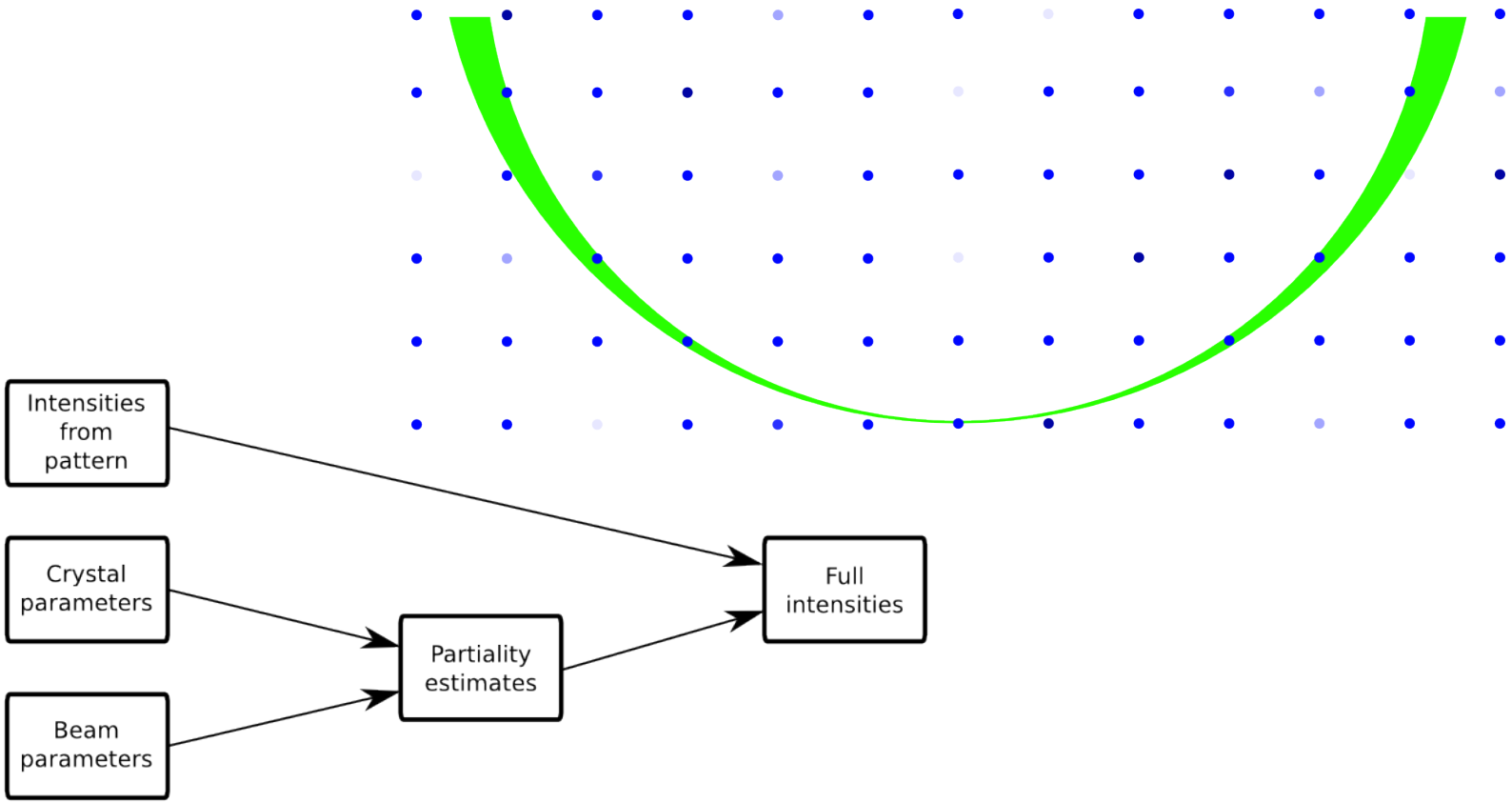
Number of cycles of post-refinement

Try --iterations=0 to simply apply the partialities according to the model.

Note: detector geometry, beam file and images not needed.

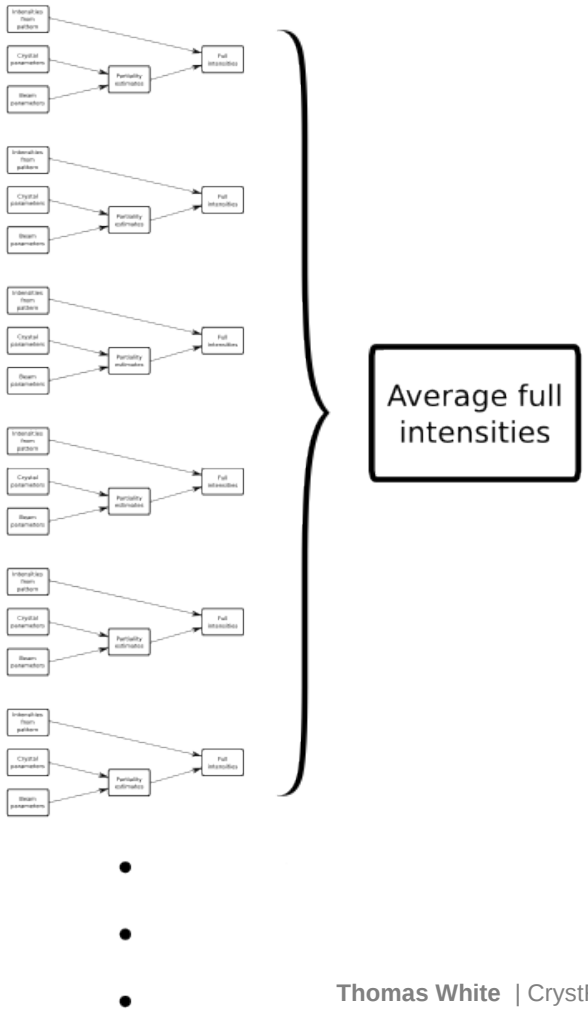
Future developments: partialities and post-refinement

> Basic scheme (in SFX context):
T. A. White, Phil. Trans. R. Soc. B (2014) 369 20130330.



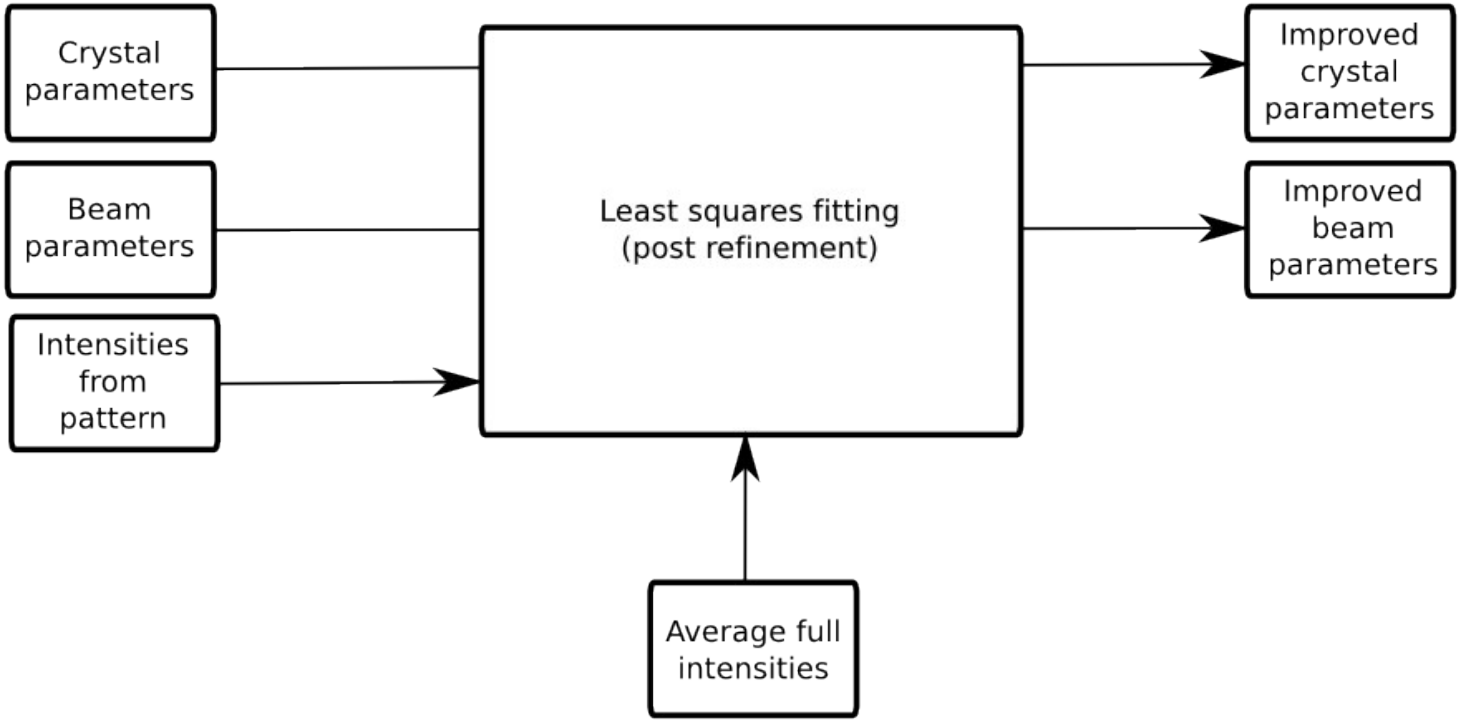
Future developments: partialities and post-refinement

> Basic scheme (in SFX context):
T. A. White, Phil. Trans. R. Soc. B (2014) 369 20130330.



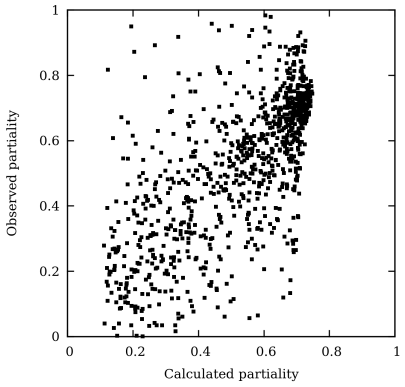
Future developments: partialities and post-refinement

> Basic scheme (in SFX context):
T. A. White, Phil. Trans. R. Soc. B (2014) 369 20130330.

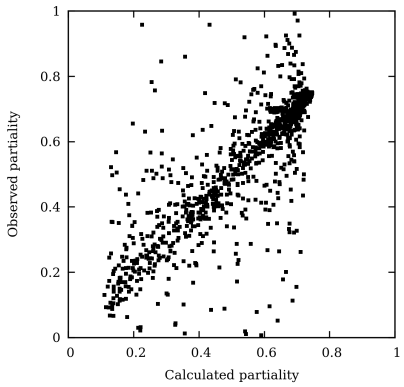


Future developments: partialities and post-refinement

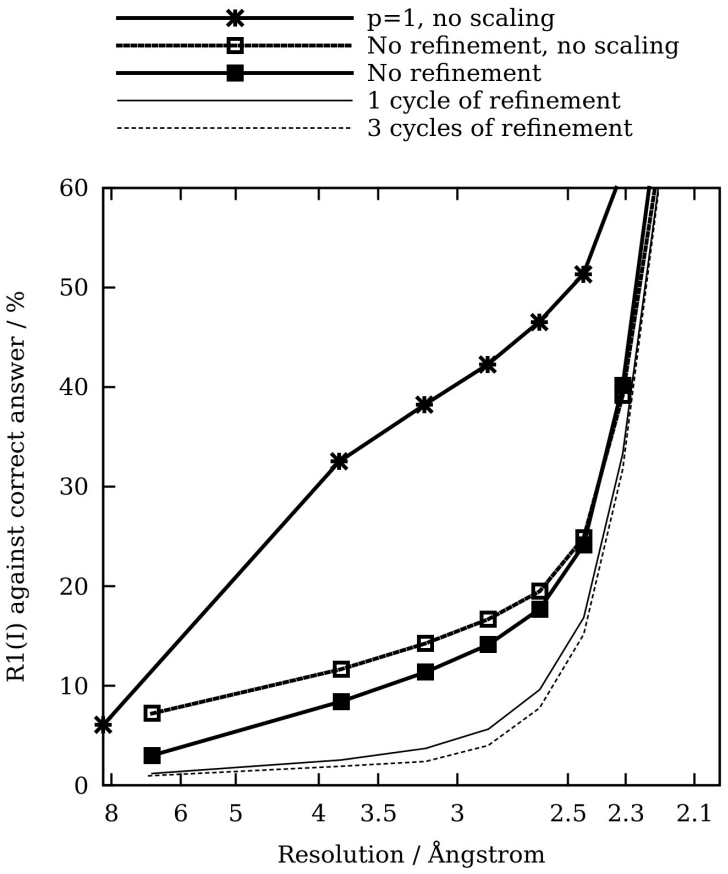
- Works really nicely for simulations of different kinds.
- Difficult to make it work on real data.



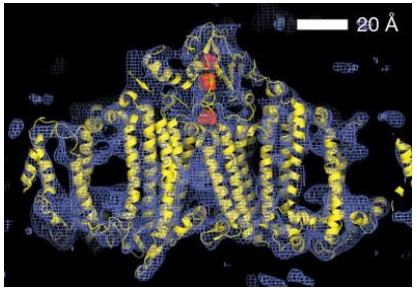
Before refinement



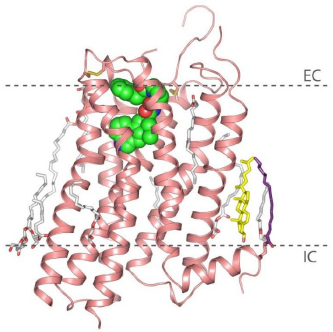
After 1 cycle of refinement



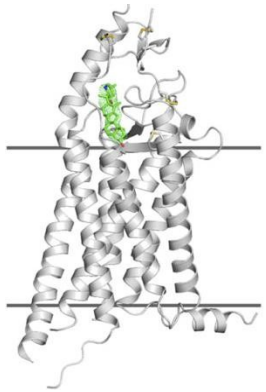
CrystFEL in the PDB (highlights)



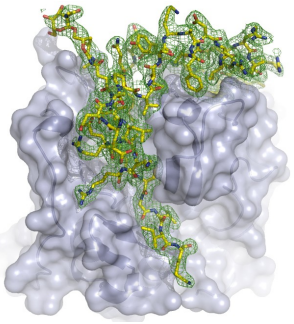
3PCQ
Photosystem I (the first SFX experiment)
Chapman et al., Nature 2011



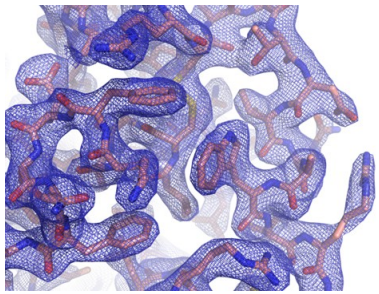
4NC3
Serotonin receptor bound to ergotamine
Lui et al., Science 2013



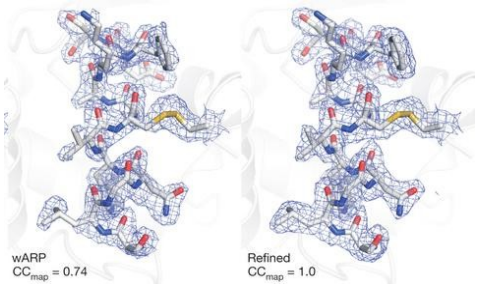
4O9R
Smoothened receptor using LCP injector
Weierstall et al., Nature Communications 2014



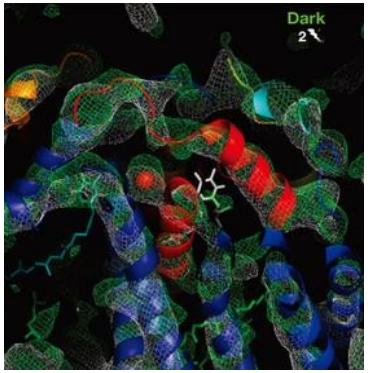
4HWY
Natively inhibited Cathepsin B
Redecke, Nass et al., Science 2013



4O34
Serial crystallography using a synchrotron beamline
Stellato et al., IUCrJ 2014



4N5R
Lysozyme (Gd derivative) ab initio phasing using SAD
Barends et al., Nature 2013



4Q54
Photosystem II in putative S_3 excited state
Kupitz, Basu et al., Nature 2014

CrystFEL in the PDB

2014

C. Kupitz, S. Basu, I. Grotjohann, R. Fromme et al. "Serial time-resolved crystallography of photosystem II using a femtosecond X-ray laser". Nature (2014). **4PBU** and **4Q54**.

M. Caffrey, D. Li, N. Howe and S. T. A. Shah. "'Hit and run' serial femtosecond crystallography of a membrane kinase in the lipid cubic phase". Phil Trans. Roy. Soc. B 369 (2014) 20130621.

O. Yefanov, C. Gati, G. Bourenkov, R. A. Kirian et al. "Mapping the continuous reciprocal space intensity distribution of X-ray serial crystallography". Phil. Trans. Roy. Soc. B 369 (2014) 20130333.

T. A. White. "Post-refinement method for snapshot serial crystallography". Phil. Trans. Roy. Soc. B 369 (2014) 20130330.

J. C. H. Spence, N. A. Zatsepin and C. Li. "Coherent convergent-beam time-resolved X-ray diffraction". Phil. Trans. Roy. Soc. B 369 (2014) 20130325.

F. Stellato, D. Oberthür, M. Liang, R. Bean et al. "Room-temperature macromolecular serial crystallography using synchrotron radiation". IUCrJ 1 (2014). **4O34**.

K. Qu, L. Zhou and Y.-H. Dong. "An improved integration method in serial femtosecond crystallography". Acta Cryst. D70 (2014) p1202.

M. Frank, D. B. Carlson, M. S. Hunter, G. J. Williams et al. "Femtosecond X-ray diffraction from two-dimensional protein crystals". IUCrJ 1 (2014) p95.

C. Gati, G. Bourenkov, M. Klinge, D. Rehders et al. "Serial crystallography on in vivo grown microcrystals using synchrotron radiation". IUCrJ 1 (2014) p87. **4N4Z.**

U. Weierstall, D. James, C. Wang, T. A. White et al. "Lipidic cubic phase injector facilitates membrane protein serial femtosecond crystallography". Nature Communications 5:3309 (2014). **4O9R.**

C. Song, K. Tono, J. Park, T. Ebisu et al. "Multiple application X-ray imaging chamber for single-shot diffraction experiments with femtosecond X-ray laser pulses". J. Appl. Cryst. 47 (2014) p18.

2013

W. Liu, D. Wacker, C. Gati, G. W. Han et al. "Serial Femtosecond Crystallography of G Protein-Coupled Receptors". Science 342 (2013) p1522. **4NC3** and **CXIDB ID 21.**

L. C. Johansson, D. Arnlund, G. Katona, T. A. White et al. "Structure of a photosynthetic reaction centre determined by serial femtosecond crystallography". Nature Communications 4:2911 (2013). **4CAS.**

T. R. M. Barends, L. Foucar, S. Botha, R. B. Doak et al. "De novo protein crystal structure determination from X-ray free-electron laser data". Nature (2013). **4N5R** and **CXIDB ID 20.**

H. Dermirci,et al. "Serial femtosecond X-ray diffraction of 30S ribosomal subunit microcrystals in liquid suspension at ambient temperature using an X-ray free-electron laser". Acta Crystallographica F69 (2013) p1066.

T. A. White, A. Barty, F. Stellato, J. M. Holton, R. A. Kirian, N. A. Zatsepin and H. N. Chapman. "Crystallographic data processing for free-electron laser sources". Acta Cryst. D69 (2013), p1231.

T. R. M. Barends, L. Foucar, R. L. Shoeman, S. Bari et al. "Anomalous signal from S atoms in protein crystallographic data from an X-ray free-electron laser". Acta Crystallographica D69 (2013) p838.

L. Redecke, K. Nass et al. "Natively Inhibited Trypanosoma brucei Cathepsin B Structure Determined by Using an X-ray Laser". Science 339 (2013) p227. **4HWY.**

2012

S. Boutet, L. Lomb, G. J. Williams, T. R. M. Barends et al. "High-Resolution Protein Structure Determination by Serial Femtosecond Crystallography". Science 337 (2012) p362. **4ET8**, **4ET9** and **CXIDB ID 17.**

L. C. Johansson, D. Arnlund, T. A. White, G. Katona et al. "Lipidic phase membrane protein serial femtosecond crystallography". Nature Methods 9 (2012) p263. **4AC5.**

R. Koopmann, K. Cupelli, L. Redecke, K. Nass et al. "In vivo protein crystallization opens new routes in structural biology". Nature Methods 9 (2012) p259.

A. Aquila, M. S. Hunter, R. B. Doak et al. "Time-resolved protein nanocrystallography using an X-ray free-electron laser". Optics Express 20 (2012) p2706.

A. Barty, C. Caleman, A. Aquila, N. Timneanu et al. "Self-terminating diffraction gates femtosecond X-ray nanocrystallography measurements". Nature Photonics 6 (2012) p35.

2011

H. N. Chapman, P. Fromme et al. "Femtosecond X-ray protein nanocrystallography". Nature 470 (2011) p73. **3PCQ.**

(see the website)