Visualizing the raw diffraction pattern with *LABELIT*

Nicholas K. Sauter[a]

[a]*Physical Biosciences Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720*

Correspondence email: NKSauter@LBL.Gov

Introduction

This article focuses on creating publication-quality pictures that illustrate diffraction data. While numerous tools are available for the routine conversion of raw data files into common image formats, other plots require specialized markup and can only be produced by custom-written software. A familiar example is the need to label each Bragg spot in the diffraction pattern with its proper Miller index. Also, synchrotron beamlines with very fast detectors such as the Pilatus-6M have emphasized the advantage of collecting data with thin rotation slices, which improve the signal-to-noise ratio but also leave the image sparsely populated with Bragg spots. This raises the need for a visualization tool where several consecutive images are summed together to give a more recognizable lattice. Another mechanism to conveniently examine the lattice is to plot the signal that corresponds to plane sections through reciprocal space. Old-style precession cameras would generate this type of photograph experimentally, but for modern rotation geometry it is necessary to synthesize such images with pixels taken from different shots over the whole data set. Code for all of these applications is available within the *LABELIT* package, which is distributed with *PHENIX* and available for download at www.phenix-online.org.

All examples discussed here are executed through the command line, as *LABELIT* has not yet been incorporated into the *PHENIX* graphical interface. Figures are documented in a special subdirectory within the source code: `labelit/publications/ccn_visualization` and can be reproduced by following the instructions contained therein. General documentation for *LABELIT* is at cci.lbl.gov/labelit.

Prerequisite indexing with *labelit.index*

Since the desired illustrations depend on knowledge of the Miller indices, we first perform an autoindexing step to identify the principle axes and unit cell dimensions of the crystal. It is assumed that the *PHENIX* package (any release subsequent to 1 Jan 2011) is installed and added to the path by sourcing the appropriate setup file (`phenix_env` or `phenix_env.sh`). As currently implemented, the program *MOSFLM* must also be placed on path under the alias `ipmosflm`; this is used during the autoindexing step to obtain a firm estimate of the resolution limits.

Indexing is done in a new current working directory (`cwd`), with the raw data frames placed either in `cwd` or any other directory:

```
cwd> labelit.index <data_path>/file_template_1_###.img 1 90
```

In this example the diffraction pattern is indexed from two 1° rotation images (#1 and #90) spaced widely apart to achieve the highest accuracy. Details are found in the *LABELIT* documentation. Of interest below, any lattice with higher than triclinic symmetry can be described in multiple Bravais settings:

```
LABELIT Indexing results:
Solution  Metric fit  rmsd  #spots  crystal_system  unit_cell
:)    5     0.197 dg  0.100    494  orthorhombic oP  84.6 123.3 174.2 90.0 90.0 90.0
:)    4     0.197 dg  0.105    496    monoclinic mP  84.6 123.4 174.3 90.0 90.0 90.0
:)    3     0.197 dg  0.099    493    monoclinic mP  84.6 174.2 123.3 90.0 90.0 90.0
:)    2     0.052 dg  0.081    496    monoclinic mP 123.3  84.6 174.0 90.0 90.2 90.0
:)    1     0.000 dg  0.082    498     triclinic aP  84.6 123.3 174.0 90.2 90.0 90.0
MOSFLM Integration results:
Solution  SpaceGroup Beam x   y  distance  Resolution Mosaicity RMS
:)    5         P222  94.08  94.11  179.99       2.20    0.050       0.040
      1           P1  94.08  94.09  180.02       2.14    0.050       0.027
```

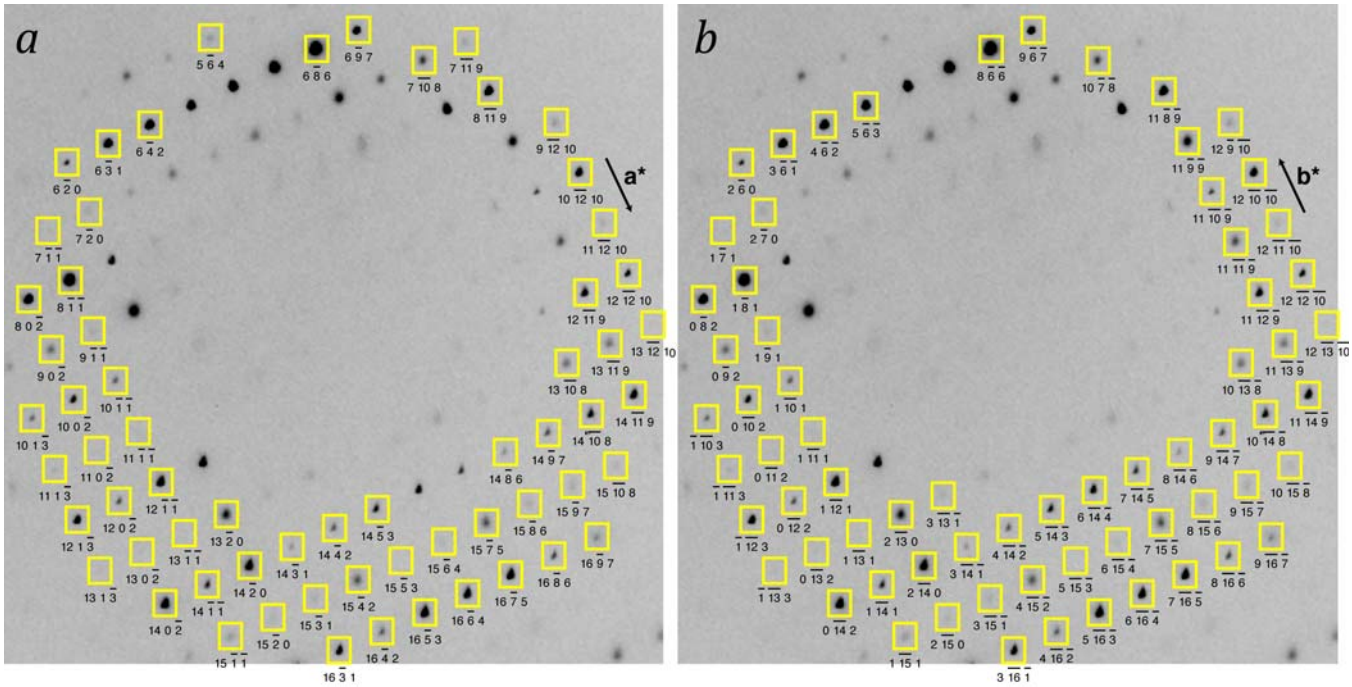

Figure 1. Detail of a rotation image from the 2qyv dataset, illustrated with `labelit.image`. The Bravais lattice is orthorhombic in model (*a*) and monoclinic in model (*b*), corresponding to the above-listed "*LABELIT* solution" numbers 5 and 2, respectively. While describing the same physical data, the two models differ in the orientations of the reciprocal cell axes (**a***, **b***, **c***) and the resultant Miller index labels attached to each Bragg spot. As the two lattice solutions are refined separately, slightly different subsets of Bragg spots are predicted by the two models (yellow boxes), but this distinction is normally erased in the subsequent steps of postrefinement and data integration.

As the printout shows, this orthorhombic lattice may be viewed either in the orthorhombic setting, or alternately in the triclinic or three different monoclinic settings. These are the indexing results from a dataset used by the Joint Center for Structural Genomics (JCSG) to solve the structure of Protein Data Bank entry 2qyv. The JCSG data repository, available for download from www.jcsg.org (Elsliger *et al.*, 2010), is used for the examples in figures 1, 3 & 4.

To create the subsequent illustrations, the indexing results must be kept in their cached location in the files `cwd/DISTL_pickle`, `cwd/LABELIT_pickle` and `cwd/LABELIT_possible`. Therefore, if multiple datasets are to be indexed, separate working directories should be created. Cached information in `cwd` can be deleted with:

```
cwd> labelit.reset
```

PDF-format rendering of diffraction images: *labelit.image*

The first product of interest is a simple picture of a rotation photograph, with Bragg spots labeled as shown in figure 1.

As with other programs in the *PHENIX* family, keyword input for `labelit.image` may be provided either at the command line or from an "effective parameter file" listing the desired keywords in a structured format. Relevant keywords can be listed out with:

```
cwd> labelit.image
cwd> labelit.image help # detailed descriptions for each keyword
```

Output from the undecorated `labelit.image` command may be used as a template to create the

effective parameter file `param.eff`:

```
bravais_choice = 5
image_number = 1
window_fraction = 0.4
window_center_x = 0.5
window_center_y = 0.5
image_brightness = 1.0
pdf_output{
  file = output.pdf
  box_size = 500
}
markup{
  bragg_spot{
    box = True
    linewidth = 0.04
    profile_shrink = 0
    color = yellow
  }
  miller_index{
    legend = False
    font_size = 10
    color = black
    vertical_offset = 10
  }
  inliers = False
}
```

The final image is computed with the command

```
cwd> labelit.image param.eff
```

or by conveniently supplying scoped command-line keywords for all non-default values:

```
cwd> labelit.image bravais_choice=5 \
                   image_number=1 \
                   pdf_output.file=output.pdf \
                   markup.miller_index.legend=True
```

Keywords and default values are as follows:

`bravais_choice=None`

This required keyword identifies the integer Bravais setting number to use for labeling the Bragg spots, as enumerated in the "*LABELIT* solution" column of the *LABELIT* output. The list of possible settings can be viewed again with the command `labelit.stats_index`. Fig. 1 illustrates how the output varies with different Bravais choices; the same physical data are displayed but differently numbered Miller indices are attached to each spot. The correct Bravais choice is not necessarily known at the time of indexing. In the case shown (PBD entry 2qyv) the published symmetry happens to be $P2_12_12$, corresponding to `bravais_choice=5` (figure 1a).

`image_number=None`

The integer image sequence number to use in the illustration (required). It is not necessary to supply the full file name, as the directory and file template are already cached by `labelit.index`.

```
window_fraction=1.0
```

Fractional length of the full image x and y dimensions to be used for illustration. A window_fraction of 0.5 would render a 1500 × 1500 square section of a 3000 × 3000 pixel raw image. To zoom in on an image detail, pick progressively smaller values.

```
window_center_x=0.5
window_center_y=0.5
```

Fractional offset on the full image to be used as the center of the section to be illustrated. The center of the raw image is at the coordinates (`window_center_x = 0.5`, `window_center_y = 0.5`), with x and y being the slow and fast directions on the image, respectively. On the printed page, slow is vertical and fast is horizontal, with the origin in the upper left corner.

```
image_brightness=1.0
```

Factor used to multiply the pixel values to produce a customized brightness. By default, a brightness scale is automatically calculated for each image, such that the 90th-percentile pixel is shown as saturated (black). An image_brightness > 1.0 makes pixels more saturated (darker).

```
pdf_output.file=None
```

Required file name for the output illustration. The top of the printed page will show the image file name and relevant information taken from the file header, while the `labelit.index` results and `labelit.image` command will be summarized at the bottom.

```
pdf_output.box_size=500
```

Number of points (unit of length, 1/72 inch) for the square edge of the illustration on the printed page.

```
markup.bragg_spot.box=True
```

Boolean value to toggle the boxes that locate the predicted position of each Bragg spot.

```
markup.bragg_spot.linewidth=0.04
```

Width of the printed lines used to outline each Bragg spot (in mm). Adjust this value to improve the clarity of the illustration if the Bragg spots are too congested.

```
markup.bragg_spot.profile_shrink=0
```

Number of pixels to shrink the box edge for outlining Bragg spots. By default, the rectangular box is sized to contain the average profile of the bright spots used by `labelit.index` for indexing, plus a two-pixel margin on each side. Use this keyword to improve clarity if necessary.

```
markup.bragg_spot.color=yellow
```

Color used for the Bragg spot boxes, as defined in the PDF-generating package *Reportlab*.

```
markup.miller_index.legend=True
```

Boolean value to toggle the inlining of Miller index HKL values underneath each Bragg spot.

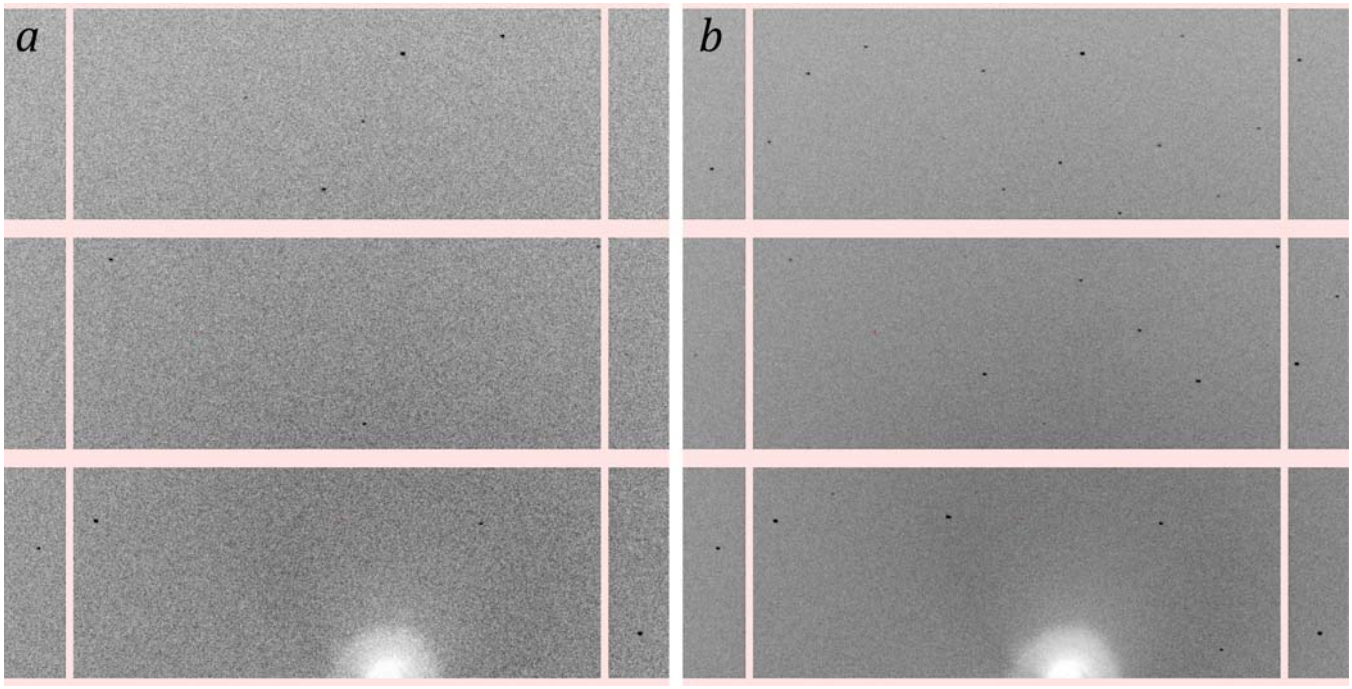

Figure 2. Detail from a cubic-lattice diffraction pattern taken from a single image (*a*, Δφ=0.2°) and a stack of five images (*b*, Δφ=1.0°). The rectangular areas are 195 × 487-pixel modules on a Pilatus-6M detector.

```
markup.miller_index.font_size=10
markup.miller_index.color=black
markup.miller_vertical_offset=10
```

For the inlined HKL values, font size in points, ink color as defined in *Reportlab*, and vertical offset in the downward direction in points, so that the HKL value does not overlap the spot.

```
markup.inliers=False
```

Boolean value to toggle dots locating the bright spots used by `labelit.index` for indexing. Red dots indicate the spot center of mass, pink dots show the maximum pixel.

Summation of consecutive thin-sliced images

Diffraction photographs of still crystals, or those with an extremely small rotation angle, will record a correspondingly thin slice through reciprocal space. Such photographs may exhibit few Bragg spots, especially if the unit cell is small and those that are captured will represent partial slices through the rocking curve, not full intensities. A sparse diffraction pattern from a 0.2° rotation photograph is shown in figure 2a.

For the purpose of illustrating the diffraction, it can be advantageous to stack consecutive rotation shots on top of each other, thus summing the partial intensities and filling out the layer slices so that the lattice pattern is more readily apparent. Such a construction is shown in figure 2b. The picture was created by supplying an `image_range` keyword:

```
cwd> labelit.image image_range=1,5
```

Keywords `image_range` and `image_number` are mutually exclusive and cannot be supplied together.

It is hoped that the availability of this method for stacking images will encourage crystallographers to

make it a common practice to acquire very finely sliced rotation images, which is now practical with the introduction of fast pixel array detectors such as the Pilatus-6M. Thin-sliced data (Pflugrath, 1999) have several advantages including improved signal-to-noise and the ability to model the Bragg spots with three-dimensional profiles. If it is easy to stack images for routine viewing, this removes the objection that thinly-sliced images are difficult to examine visually.

Note: it is possible to go immediately from raw images to PDF-format pictures *without* indexing first, if the object is to simply render the image without any markup. A separate command is provided for this purpose:

```
cwd> labelit.pdf <image template (/home/data/lysozyme_###.img)>
```

Keyword options are:

```
image_number = 1
image_range = 1,5
window_fraction = 0.4
window_center_x = 0.5
window_center_y = 0.5
image_brightness = 1.0
pdf_output{
  file = output.pdf
  box_size = 500
}
```

Synthesis of pseudo-precession photographs: *labelit.precession_photo*

While molecular structure is ideally explored with perfect crystals that give sharp Bragg peaks, it has been the imperfections that have posed large challenges over the years. The visual examination of images certainly plays an important role in diagnosing specific types of disorder (Nave, 1999). Aside from mosaicity, or isotropic disorder that gives rise to wide rocking curves (the diffraction of a Bragg spot over a large rotation angle), recent papers have highlighted specific types of long-range disorder that produce recognizable signatures at the "non-Bragg" positions of the diffraction pattern. For example, incommensurate modulation (Borgstahl *et al.*, 2009), a periodic distortion of the crystal lattice, generates discrete satellite Bragg spots; while lattice translocation disorder (Tsai *et al.*, 2009), the slight displacement of successive crystal layers, creates a pattern of streaks on specific spots.

Historically, the availability of Buerger precession cameras (*e.g.*, Blundell & Johnson, 1976) made it easy to examine specific planar layers of the reciprocal lattice, after painstaking alignment of the principal crystallographic axes (**a***, **b***, **c***) relative to the camera reference frame. In certain cases (Bragg & Howells, 1954), differences in the Bragg spot shape could be described as a function of Miller index. Achieving this type of convenient plot with modern rotation data requires a software calculation for two reasons: first, each image represents a curved surface of reciprocal space, not a plane; and secondly, the crystal axes are now rarely prealigned with the camera.

Assuming that images are available from a wide enough rotational range, the requisite planar section can be synthesized. However, it is best to keep in mind that there are implicit limitations. We assume, for example, that the crystal is rigidly fixed to the goniometer rotor, so its orientation is exactly known for each source image from the dataset. Deviations from this ideal will degrade the synthesized image, particularly at higher scattering angles. Also, our implementation does not apply scaling corrections. Thus, factors such as accumulated radiation dose that change the sample over time will cause symmetry-related reflections to appear unequal in intensity. Geometric approximations are unavoidable: in order to create a mapping between the raw image and reciprocal space coordinates, it is assumed that each raw image represents the center of its rotation range (for example, a 1° rotation image covering φ=[0°,1°] is uniformly assigned the value φ=0.5°). Moreover, image pixels far from the

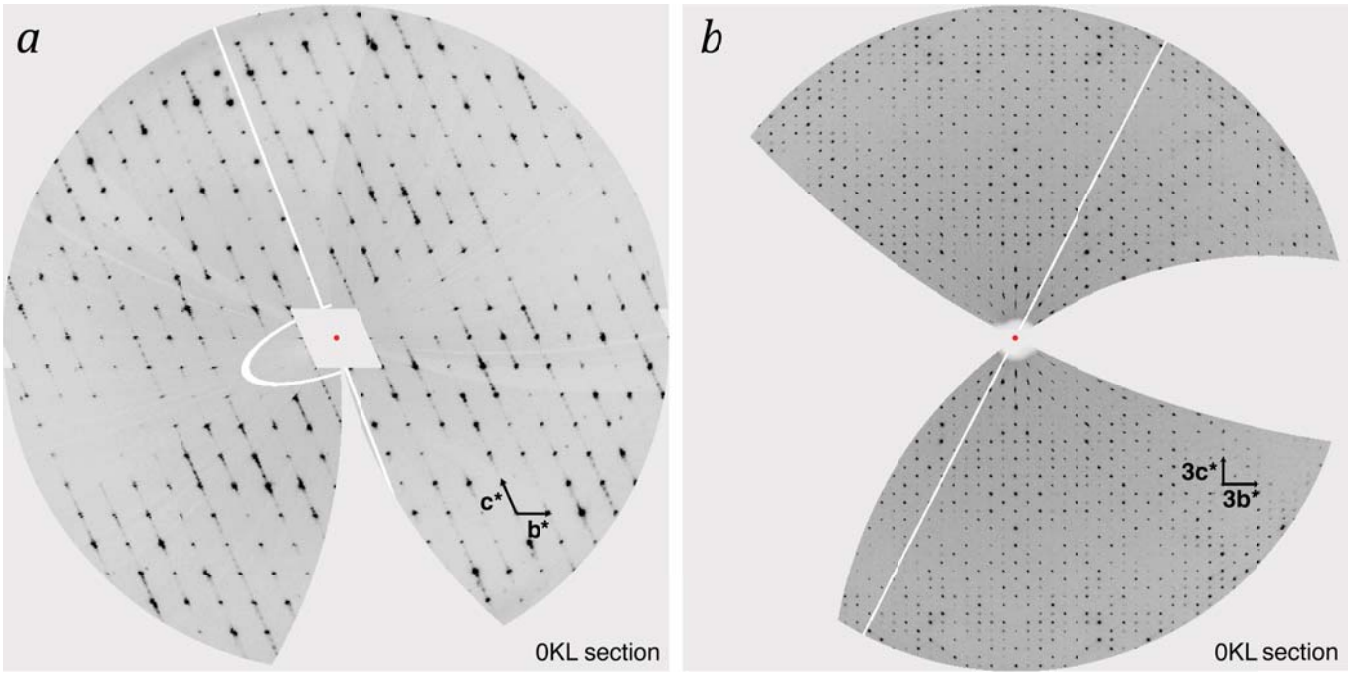

Figure 3. Reciprocal space sections illustrated with `labelit.precession_photo` for structures 1vk8 (*a*) and 2qyv (*b*). The origin of the streaks extending along the **c*** axis in (*a*) was not examined in the original publication (Dermoun *et al.*, 2010); but is presumably associated with lattice disorder.

rotation axis map to a larger rotational path through reciprocal space and appear as large quadrilaterals on the synthesized image. Clearly, the best sampling is obtained with fine rotational slicing (Pflugrath, 1999) and small image pixels.

Despite these caveats, `labelit.precession_photo` can be used (figure 3) to clearly illustrate phenomena that we recently cited: streaky Bragg spots of unknown origin associated with PDB code 1vk8 (Sauter & Poon, 2010) and a pattern of alternating weak and strong Bragg spots due to pseudotranslational symmetry in PDB structure 2qyv (Sauter & Zwart, 2009). The command line keywords for `labelit.precession_photo` are handled exactly as described above for `labelit.image`:

```
bravais_choice=None
image_range=None
pdf_output.file=None
pdf_output.box_size=500
```

Identical to the parameters described for `labelit.image`. A "`None`" value indicates required input. Here it is advantageous to specify an `image_range` covering the entire dataset, so that the coordinate grid of the synthesized image is filled in to the largest extent.

```
pixel_width=600
```

The width of the synthesized coordinate grid in pixels, which is then fit into the `pdf_output.box_size` dimension expressed in points.

```
resolution_outer=3.0
```

The high resolution limit of the requested plot, expressed in Ångstroms.

`intensity_full_scale=256`

Intensity value on the raw image that is treated as fully saturated (black).

`plot_section="H,K,0"`

Determine which principle axes are in the plane of the printed page; either **a***, **b*** (`H,K,0`); **b***, **c*** (`0,K,L`); or **a***, **c*** (`H,0,L`). Also, upper- and lower-layers can be sectioned, *i.e.*, "`H,K,1`"; "`H,K,-1`"; etc.

`layer_width=0`

The width of the reciprocal space section to be illustrated, given in fractional Miller index units. For example, if `plot_section` is "`H,K,0`" and `layer_width` is 0.05, then all image pixels mapping to reciprocal space coordinates between `H,K,-0.025` and `H,K,0.025` are plotted, with overlapping pixels being averaged. By default `layer_width=0`, corresponding to a section thickness of one pixel.

`apply_symmetry=None`

Point-group symmetry used to average the data. This option is not recommended by default, but is provided to respond to user comments that `labelit.precession_photo` printouts do not always look like true precession photographs. The full reciprocal space layer is not always covered. This is because the experimental rotation range is often less than the full 180° required for full coverage. Can point-group symmetry be applied to get an illustration that *looks* like a full precession photograph? Caution must be exercised, as such an operation could erase distinctions that might be important! As noted above the synthesized image normally reflects differences due to radiation damage, as well as variations in other factors such as the incident beam flux and the length of the absorption path. Moreover, the true symmetry of the diffraction may not be as high as implied by the Bravais choice, as with a monoclinic crystal with β angle close to 90°, which can be plotted within an orthorhombic cell. With these warnings in mind, the user can choose to impose point-group symmetry on the diffraction pattern if desired, with the `apply_symmetry` keyword.

Two choices must be made when selecting the point-group symmetry. First, should Friedel symmetry, $HKL = \overline{HKL}$, be imposed or not? Second, for certain crystal systems (such as tetragonal) there are alternate Laue groups to select. The Laue group cannot be selected automatically based on indexing alone, as it is necessary to compare symmetry-equivalent intensities after integration and scaling. To make the full matrix of choices clear, the user should type the undecorated command:

```
cwd> labelit.precession_photo
```

which outputs a table enumerating the point group choices for each possible Bravais setting, for example:

Bravais_choice	Lattice	Laue-group	Reflection-symmetry	Friedel-only
9	tP	4/mmm	422	-1
9	tP	4/m	4	-1
8	oC	mmm	222	-1
7	mC	2/m	2	-1
6	mC	2/m	2	-1
5	oP	mmm	222	-1
4	mP	2/m	2	-1
3	mP	2/m	2	-1
2	mP	2/m	2	-1
1	aP	-1	1	-1

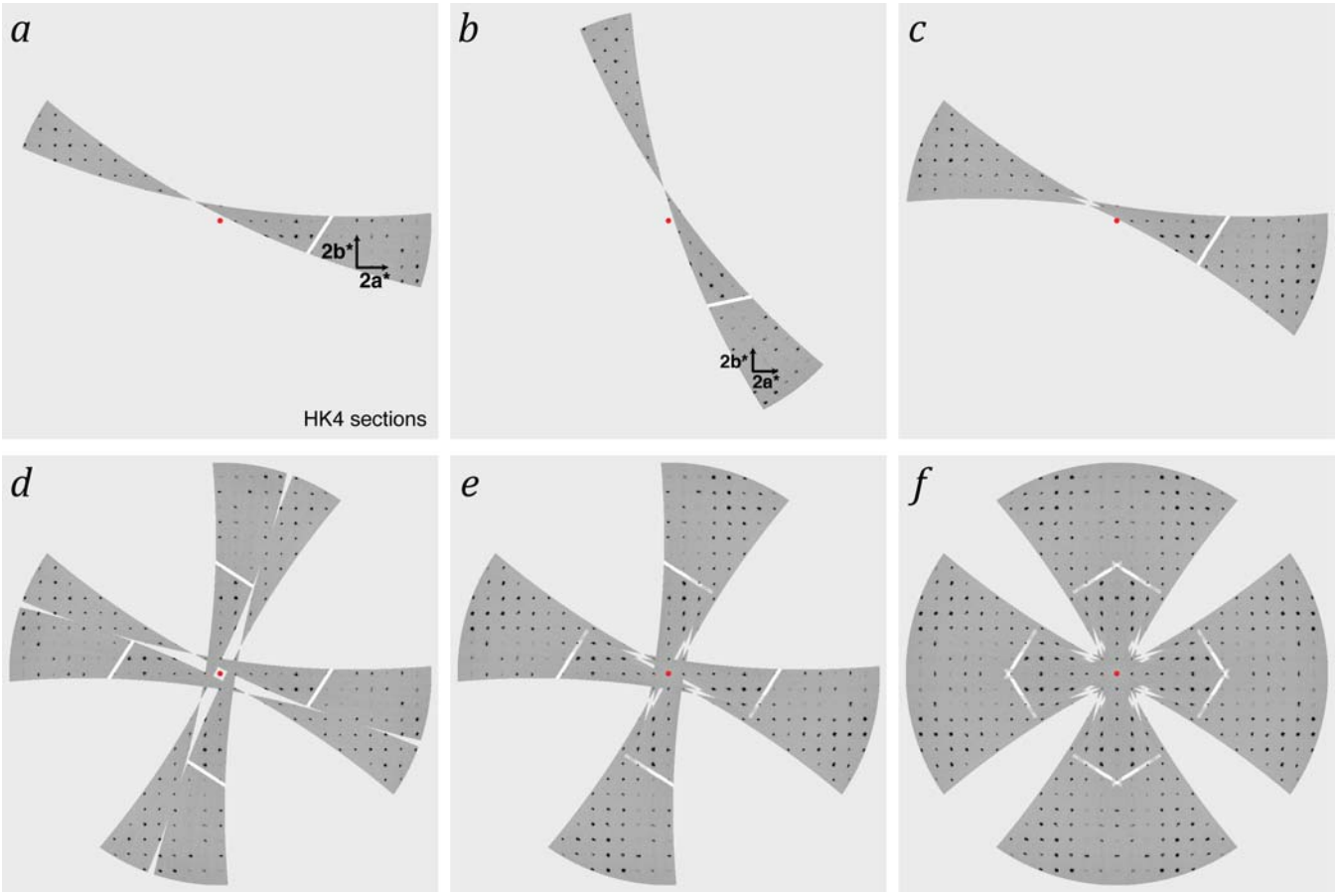

Figure 4. L=4 sections from the 1o3u diffraction pattern (Eriandsen *et al.*, 2004). The space group of the structure is $P4_12_12$, and the data are plotted either in the primitive tetragonal setting (*a*, *c*-*f*) or the *C*-centered orthorhombic setting (*b*). Point group symmetries imposed on the data are $\bar{1}$ (*c*), 4 (*d*), 4/m (*e*), and 4/mmm (*f*).

Various combinations of `bravais_choice` and `apply_symmetry` produce drastically different pictures, as illustrated in figure 4. Each panel plots the same 20° of rotation data from a tetragonal diffraction pattern. The `bravais_choice` keyword changes the axes on which the data are plotted without altering the intensities that are displayed, as seen by comparing panels (*a*, tP) and (*b*, oC). In contrast, the `apply_symmetry` option has the effect of increasing the reciprocal space coverage and/or averaging symmetry-redundant measurements of the displayed reciprocal space coordinates. For example, the application of Friedel symmetry (*c*) brings data from the L=–4 section on to the L=4 layer. Application of 4-fold symmetry (*d*) produces a clover-leaf pattern around the L-axis, while the combination of both Friedel and 4-fold symmetry (*e*) combines both effects. Finally, 4/mmm symmetry (*f*) yields two mirror planes at H=0 and K=0.

PNG- and GIF-format output

For completeness, we mention that *LABELIT* can also generate PNG-format images of the diffraction pattern:

```
cwd> labelit.png <image file> <output file.png> [-large]
cwd> labelit.overlay_distl <image file> <output file.png> [-large]
cwd> labelit.overlay_index <image file> <output file.png> [-large]
cwd> labelit.overlay_mosflm <image file> <output file.png> [-large]
```

The command `labelit.png` generates an undecorated image, while `labelit.overlay_distl` colors the subsets of bright spots either used (green) or not used (blue) for indexing. Commands

`labelit.overlay_index` and `labelit.overlay_mosflm` add markup of the predicted lattice for the highest Bravais choice, as refined by either *LABELIT* or *MOSFLM*, respectively. The only allowed keyword is `–large`, which imposes a one-to-one mapping between raw data pixels and pixels on the generated picture, otherwise the raw data pixels are binned in 2 × 2 squares.

An animated GIF-format movie can be generated of the entire dataset (this does not require indexing):

```
cwd> labelit.dataset_animation <template> <first image> <last image> <out>
cwd> labelit.dataset_animation /home/user/mydata/lyso_###.img 1 90 out.gif
```

Extensibility of Python code

Developers should be aware that the features discussed here could easily be extended by simple scripting in Python language. The applications discussed above are built on standard `cctbx` components for handling of detector formats (`iotbx.detectors`) and command-line keywords (`libtbx.phil`). Third party extensions are used for standard image formats (*Python Image Library*) and generation of PDF output (*Reportlab*).

Acknowledgments

Comments from software users were instrumental in developing the finished product. In particular, input from Tillman Heinisch (Universität Basel) and Jason Porta (University of Nebraska Medical Center) contributed significantly to `labelit.precession_photo`. The financial support of the National Institutes of Health / National Institute of General Medical Sciences under grant number R01-GM077071 is gratefully acknowledged. Operation of LBNL is partly supported by the US Department of Energy under Contract No. DE-AC02-05CH11231.

References

Blundell TL, Johnson, LN (1976). *Protein Crystallography*. London, Academic Press, Ltd.

Bragg WL, Howells ER (1954). X-ray diffraction by imidazole methaemoglobin. *Acta Crystallogr.* **7**, 409.

Dermoun Z, Foulon A, Miller MD, Harrington DJ, Deacon AM, Sebban-Kreuzer C, Roche P, Lafitte D, Bornet O, Wilson IA, Dolla A (2010). TM0486 from the hyperthermophilic anaerobe *Thermotoga maritima* is a thiamin-binding protein involved in response of the cell to oxidative conditions. *J. Mol. Biol.* **400**, 463-476.

Elsliger MA, Deacon AM, Godzik A, Lesley SA, Wooley J, Wüthrich K, Wilson IA (2010). The JCSG high-throughput structural biology pipeline. *Acta Crystallogr.* F**66**, 1137-1142.

Eriandsen H, Canaves JM, Elsliger MA von Delft F, Brinen LS, Dai X, Deacon AM, Floyd R, Godzik A, Grittini C, Grzechnik SK, Jaroszewski L, Klock HE, Koesema E, Kovarik JS, Kreusch A, Kuhn P, Lesley SA, McMullan D, McPhillips TM, Miller MD, Morse A, Moy K, Ouyang J, Page R, Robb A, Quijano K, Schwarzenbacher R, Spraggon G, Stevens RC, van den Bedem H, Velasquez J, Vincent J, Wang X, West B, Wolf G, Hodgson KO, Wooley J, Wilson IA (2004). Crystal structure of an HEPN domain (TM0613) from *Thermotoga maritima* at 1.75 Å resolution. *Proteins* **54**, 806-809.

Plugrath JW (1999). The finer things in X-ray diffraction data collection. *Acta Crystallogr.* D**55**, 1718-1725.

Sauter NK, Poon BK (2010). Autoindexing with outlier rejection and identification of superimposed lattices. *J. Appl. Crystallogr.* **43**, 611-616.

Sauter NK, Zwart PH (2009). Autoindexing the diffraction patterns from crystals with a pseudotranslation. *Acta Crystallogr.* D**65**, 553-559.