

CrystalPlan: an Experiment Planning Tool for Crystallography

Janik Zikovsky,* Peter Peterson, Xiaoping Wang, Matthew Frost and Christina Hoffmann

Spallation Neutron Source, Oak Ridge National Laboratory, P.O. Box 2008 MS-6477, Oak Ridge, TN 37831-6477 USA. Correspondence e-mail: zikovskyj@ornl.gov

Beam time at large x-ray and neutron scattering facilities is always at a premium. The CrystalPlan program can calculate the data coverage of a crystal in reciprocal space in a single-crystal diffraction time-of-flight experiment. CrystalPlan can help a user build an experiment plan that will acquire the most data possible, with sufficient coverage but limited redundancy, therefore increasing scientific productivity. A user friendly GUI including a 3D viewer, an automated coverage optimizer, and an option to reorient the crystal for the measurement of selected hkl's on specific detector positions are among its useful features. A sample use case of the program with the TOPAZ beamline at SNS will be presented.

© 2010 International Union of Crystallography
Printed in Singapore – all rights reserved

1. Introduction

The TOPAZ beamline at the Spallation Neutron Source (SNS) at Oak Ridge National Lab (ORNL), currently finishing commissioning, is a neutron time-of-flight Laue instrument with high wavelength resolution. It is designed to acquire single-crystal diffraction data of small-to-moderate sized unit cells with high throughput. The final design calls for an array of 48 detectors covering a large fraction of 2π steradians. However as of this writing, only 14 of the 48 detectors are installed, giving only a fraction of 2π steradian coverage. In order to have sufficient data to perform a fit, an experimenter will typically want to measure $> 85\%$ of the peaks within a certain range of d-spacings. However, wide wavelength band combined with TOPAZ's complex geometry makes it very difficult to come up with an effective set of sample orientations that will cover all the required peaks without unnecessary redundancy.

Existing diffraction experiment planning software programs (for example, CrysAlis Pro from Oxford Diffraction; COSMO from Bruker AXS) were designed for X-ray experiments where a single incident wavelength and a large number of sample orientations or sweeps are used to collect data. These programs are not suitable for TOPAZ, since as a time-of-flight instrument, it is able to collect data from a wide bandwidth of incident neutrons, using a stationary crystal. Additionally, this means that relatively few sample orientations are needed in order to complete a measurement: whereas an X-ray measurement might have hundreds or thousands of individual frames, a TOPAZ run may have as few as 10.

The goal of CrystalPlan was to make an easy-to-use software tool to aid users in planning their experiments. By quickly making an experimental plan that will be sufficient for the needs of the scientist (enough peaks measured), but without spending time needlessly measuring the same region in reciprocal space, CrystalPlan will greatly increase the scientific productivity and throughput of TOPAZ and other beamlines. Users will be able to measure more samples or more fully explore parameters such as temperature and pressure.

2. Architecture

CrystalPlan is designed with these capabilities:

- Load an instrument geometry.
- Calculate the orientation of a crystal at different goniometer settings.
- Calculate the coverage of these orientations in reciprocal space, in volume and for single peaks.
- Take crystal Bravais lattice symmetries into account.
- Optimize the experiment plan automatically.
- Compare predicted and actual measurement of peaks.

2.1. Instrument Configuration

CrystalPlan supports area detectors with flat faces and arbitrary rectangular dimensions; this handles the detector types in use at SNS (Anger cameras and 8-pack tube detectors), but the software could easily be extended for more complicated detector shapes as needed. The detector configuration is loaded from a text file containing the location, size, orientation and number of pixels of each detector. Other settings to be set for an instrument include the detectable range of neutron/x-ray wavelength.

2.2. Sample Orientation

CrystalPlan is designed to be able to use a variety of sample orientation goniometers. A base Goniometer class is subclassed for specific types of goniometers, allowing a unified interface. This base class acts as a *universal goniometer*, which uses the conventional goniometer angles of ϕ , χ , ω as defined in (Busing & Levy, 1967). Each class inheriting from Goniometer can convert the desired ϕ , χ , ω angles to the internal motor positions it requires. The API allows for limited goniometers: for example, TOPAZ is currently equipped with a goniometer where χ is fixed at 45 degrees. CrystalPlan handles this by limiting the available sample orientation angles accessible to the user.

From an initial data set, the data analysis program ISAW (Mikkelsen *et al.*, 2005) is used to calculate the sample mounting orientation UB matrix, which is saved to a text file along with the lattice parameters. CrystalPlan can load this UB

matrix file and will use that UB matrix in coverage calculations.

2.3. Reciprocal Space Representation.

The reciprocal space to be measured is divided into a cubic 3D matrix, the size of which is determined by the user, who specifies the minimum d-spacing (d_{min} , in Å) that he or she is interested in. The resolution of the 3D matrix is also specified (in Å⁻¹), which determines the number of points (voxels or volumetric pixels) to be modeled. Available machine memory limits the resolution that can be achieved in practice.

Each point of the reciprocal space 3D matrix holds an integer representing the number of times that voxel of **Q**-space (with **Q** defined as $\frac{2\pi}{d}$) has been measured – this is the total coverage matrix. To speed up the calculation of the total, a coverage matrix for each sample orientation is calculated and saved in memory. This means that the total coverage matrix can be computed by simply adding multiple 3D matrices corresponding to each sample orientation.

2.4. Calculating Coverage of One Sample Orientation

For each sample orientation, a rotation matrix combining the sample's mounting orientation (the **U** matrix loaded from ISAW) and the goniometer rotation matrices (**Φ**, **χ**, **Ω**) is computed. The coverage in reciprocal space is computed in this way: for each pixel in the face of a detector, the direction of the scattered beam is known. Then, two incident beam vectors are considered, corresponding to the λ_{min} and λ_{max} wavelength limits of the detectors and/or source. The scattered beam direction and the two possible incident beam vectors are used to find two \vec{q} vectors; these point in the same direction. At the high wavelength we find \vec{q}_{min} , and the low wavelength limit scattering off of \vec{q}_{max} . Since the instrument has a wide bandwidth, we fill in the 3D **Q**-space matrix by filling all points from \vec{q}_{min} to the \vec{q}_{max} , for all pixels in the face of the detector. This quickly defines a volume of coverage, as shown schematically in two dimensions in Figure 1.

To reduce memory usage yet allow for quickly trying different detector configurations, the 3D matrix of each sample orientation is saved a 32- or 64-bit integer, with each bit representing a particular detector. A simple binary mask can then be applied to quickly compute the coverage if some detectors are disabled, for example.

2.5. Crystal Parameters and Single Reflection Coverage

Crystal lattice parameters (lengths a , b , c and angles α , β , γ) are typed in or loaded from the ISAW UB matrix file. CrystalPlan then generates a list of hkl peaks with d-spacings larger than the d_{min} specified. For each sample orientation being simulated and for each peak, a function calculates at what wavelength λ and in which direction it scatters. If λ is within range of the limits of the instrument, the beam direction is projected onto each detector face to determine where, on each detector surface, it will be measured.

Each hkl peak is a separate object that holds a list of predicted measurements, including which detector measured it, at what position on the detector face, at what wavelength, and for

which sample orientation. These data will be used to graphically display coverage and calculate statistics such as redundancy.

2.6. Crystal Symmetry

When entering the lattice parameters, the crystal's point group symmetry can also be entered. This information is then used to calculate the increased measurement redundancy (Giacovazzo *et al.*, 1992). A list of n 3×3 multiplication matrices is generated based on the point group; any hkl vector can be multiplied by these matrices to find the n equivalent hkl reflections; for example, for orthorhombic (mmm) symmetry, $n = 8$; therefore 8 multiplication matrices are generated that multiply hkl to give $\pm h$, $\pm k$, $\pm l$.

For the single reflections, each hkl is tracked separately, but a primary hkl is found that points to a list of equivalent hkl 's. The GUI can then display either all hkl , ignoring crystal symmetry; or combine them to display only part of the reflections. Primary hkl 's are picked so that they make a contiguous block.

In volume coverage mode, a 4D array is generated where the first 3 dimensions correspond to **Q**-space, and the 4th dimension is a list of n entries. Each entry in the 4th dimension points to a voxel that is an equivalent due to crystal symmetry. This allows CrystalPlan to quickly compute a volume coverage map that takes into account crystal symmetry: at each voxel, all n voxels are checked to see if any of them were measured.

2.7. Automatic Coverage Optimizer

CrystalPlan allows the user to manually enter lists of sample orientations to simulate; however, finding an optimal plan by trial and error would be very time consuming. To make the process easier, CrystalPlan includes an Automatic Coverage Optimizer, which takes as its inputs the desired number of sample orientations and uses a genetic algorithm to find a solution that satisfies a user-specified coverage criterion (for example, the user may ask for 85% of peaks to be measured, or for 90% of **Q**-volume).

Genetic algorithms have been covered in the literature before, see e.g. (Goldberg, 1989). In this application, the genes consist of goniometer angles. For instance, if a goniometer has freedom in 3 angles and the user requests a 10 different sample orientations, then each individual has 30 genes. The initial population is created randomly and is made to evolve using mutations and crossover. Individuals whose goniometer angles turn out to be impossible to reach are eliminated. For each individual set of sample orientations, the coverage statistics are computed (taking crystal symmetry into account if desired), and this is used as the fitness of the given individual in the normal genetic algorithm process.

The Automatic Coverage Optimizer can find an acceptable solution (if any is possible) in times ranging from a few seconds to several minutes, depending on the difficulty of the problem and the computing hardware. Given that at a neutron facility, a measurement of a single orientation may take a few hours, this feature can result in significant beam-time savings.

3. Implementation

3.1. Language and Libraries

The program needs to be cross-platform, and run equally well on Linux, Mac OS and Windows, although Linux is our primary target operating system. Another design goal was to use of open-source software and toolkits as much as possible.

To this end, CrystalPlan was written in Python (van Rossum, 1995), a powerful, open, cross-platform scripting language. Python was chosen for its ease of cross-platform deployment, and for its attractive language features. The numpy and scipy libraries (two open-source, commonly installed Python libraries) were used for many of the calculations (Oliphant, 2007; Jones *et al.*, 2001–). Some of the critical calculations were also written with inline C code, which speed up some calculations by up to 400× as compared with the pure Python version (which can still be used, in case of incompatibilities).

3.2. Code Layout

The calculation code of CrystalPlan is placed in a model layer and kept completely separate from eventual GUI elements, using a Model-View-Controller architecture (Gamma *et al.*, 1995). This makes it possible to run calculations using Python scripts, if desired. Unit tests for each part of the calculations are used to ensure consistent results on different platforms and systems.

3.3. Graphical User Interface

3.3.1. Design. An attractive and easy-to-use GUI was considered an important part of CrystalPlan, since the program is meant to be used interactively as a guide to the experimenter. The user interface was written using wxPython (Talbot, 2000), a mature cross-platform GUI toolkit, as well as the Enthought Traits GUI. Some of the 2D plots use matplotlib (Hunter, 2007), and the 3D visualization elements use the Mayavi2 toolkit for Python (Ramachandran, 2003), itself based on VTK (Schroeder, 2003). All of these libraries are open-source.

3.3.2. Main Program Workflow. A screenshot of the main window GUI is shown in Figure 2. The main window consists of a series of tabs, with workflow proceeding intuitively from left to right:

- **Q-Space tab:** define the volume of interest (minimum d-spacing) and resolution of reciprocal space.
- **Detectors tab:** load instrument detector geometry files, and enable or disable individual detectors.
- **Goniometer tab:** to choose the goniometer and degrees of orientation freedom of the instrument.
- **Sample tab:** enter or load the crystal's lattice parameters and UB matrix.
- **Try an Orientation tab:** interactively rotate a sample and observe the changes in coverage in real-time.
- **Add Orientations tab:** type in a list of sample orientation angles to add to the experiment plan.
- **Experiment Plan tab:** a list of the previously calculated sample orientations. Here you can manage the orientations you will use in the experiment: add, delete, or temporarily disable them to see the effect on coverage. Once

complete, the plan can be sent to the SNS data acquisition computers in order to begin a run (Zolnierczuk & Riedel, 2010).

3.3.3. Reciprocal Space Coverage Visualization. Figure 3a) shows a screenshot of the reciprocal space coverage 3D interface. The 3D view can represent **Q**-space in two ways: volume coverage, or single-peak coverage. The view shown can be rotated, translated or zoomed using the mouse or keyboard.

The volume coverage shows a solid isosurface representing the volume measured at least once. Redundant regions can also be shown using semi-transparent isosurfaces; or the coverage can be inverted, showing non-measured volumes only; or sliced between user-selectable q_{min} and q_{max} .

The reflections view (Figure 4a) displays each hkl as a small sphere; the color of which indicates the number of times the hkl is predicted to be measured in the current experiment plan.

Both views can take into crystal symmetry in their display using the techniques described previously. The volume coverage (Figures 3b) fills in the sphere of coverage; whereas the single reflection view (Figure 4b) leaves only the primary reflections (one eighth of the total, in this example).

Finally, both views display quick coverage statistics in the lower right corner, showing the percentage of volume (or of the number of peaks) measured, as well as the proportion of redundancy (voxels or peaks measured 2 or more times).

3.3.4. Single Reflection View. Figure 5 shows a screenshot of the GUI allowing one to observe in detail the predicted measurements of a particular peak, by clicking in the 3D view or by typing in the hkl values. The position on the detector and wavelength of measurement are displayed. An additional window (not shown) allows the user to place the peak on a particular position on a detector by automatically finding the necessary sample orientation angles. This is used, for example, to center important peaks in the most sensitive part of a detector.

4. Sample Use Case

As part of the TOPAZ beamline commissioning process, a Natrolite ($\text{Na}_2\text{Al}_2\text{Si}_3\text{O}_{10} \cdot 2\text{H}_2\text{O}$) single crystal was measured on a fixed 45° χ goniometer at room temperature and ambient atmosphere. An initial short run (wavelength range from ~ 0.5 Å to 3.5 Å) was processed using ISAW to determine the UB matrix and cell parameters of the crystal.

The resulting UB matrix was loaded into CrystalPlan, where the Automatic Coverage Optimizer was used to generate an experiment plan with 8 sample orientations (Figure 6a) that covered 97% of the peaks, taking into account the crystal symmetry (orthorhombic or mmm) (Figure 6b). The plan was left to run unattended overnight.

Peaks were integrated using ISAW, and the resulting .integrate file was loaded into CrystalPlan for comparison. The predicted and actual measurement positions on each detector face were compared for the 554 integrated peaks found. Very good agreement was found; the RMS difference between the predicted and actual measurement wavelength was only 0.016 Å.

research papers

The offsets in the peak position were plotted, which helped show that the main contributor to this error was a slight misalignment of the goniometer.

5. Conclusions and Future Development

We have described CrystalPlan, a software program designed to quickly produce efficient experiment plans for wide bandwidth time-of-flight Laue diffractometers. This program will help increase scientific productivity at SNS beamlines such as TOPAZ.

One planned future development for CrystalPlan include extending the software to help predict coverage in inelastic neutron scattering experiments. Another desired feature is the introduction of feedback loops to help determine the right time to end data acquisition at a particular sample orientation. CrystalPlan would interface with data analysis software such as ISAW and use recently developed capabilities for real-time data analysis to monitor the signal to noise ratio on particular peaks. When the signal becomes sufficient, the sample will automatically be moved to the next sample orientation, further minimizing data acquisition time. Combined with advanced sample changers coming on-line at TOPAZ, this will allow for extremely efficient, unattended data acquisition.

Acknowledgements

This research is supported by UT Battelle, LLC under Contract No. DE-AC05-00OR22725 for the U.S. Department of Energy, Office of Science.

References

- Busing, W. & Levy, H. (1967). *Acta Cryst.* **22**(4), 457.
- Gamma, E., Helm, R., Johnson, R. & Vlissides, J. (1995). *Design Patterns*. Boston, MA: Addison-Wesley.
- Giacovazzo, C., Monaco, H., Artioli, G., Viterbo, D., Ferraris, G., Gilli, G., Zanotti, G. & Catti, M. (1992). *Fundamentals of Crystallography*. Oxford University Press.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Kluwer Academic Publishers.
- Hunter, J. D. (2007). *Computing in Science and Eng.* **9**, 90–95.
URL: <http://matplotlib.sourceforge.net/>
- Jones, E., Oliphant, T., Peterson, P. *et al.*, (2001–). SciPy: Open source scientific tools for Python.
URL: <http://www.scipy.org/>
- Mikkelsen, D. J., Schultz, A. J., Mikkelsen, R. & Worlton, T. G. (2005). *IUCR Compcomm Newsletter*, **5**, 32–39.
URL: <http://www.iucr.org/resources/commissions/crystallographic-computing/newsletters/5>
- Oliphant, T. E. (2007). *Computing in Science and Engineering*, **9**(3), 10–20.
URL: <http://numpy.scipy.org/>
- Ramachandran, P., (2003). Scientific data visualization with mayavi.
URL: <http://code.enthought.com/projects/mayavi/>
- van Rossum, G. (1995). *Python tutorial, Technical Report CS-R9526*. Tech. rep. Centrum voor Wiskunde en Informatica (CWI).
URL: <http://www.python.org/>
- Schroeder, W. (2003). *The Visualization Toolkit*. Kitware, Inc., 3rd ed.
URL: <http://www.vtk.org/>
- Talbot, H. (2000). *Linux J.* **2000**.
URL: <http://www.wxpython.org/>
- Zolnierczuk, P. A. & Riedel, R. A. (2010). In *17th Real-Time Conference*. Lisbon, Portugal.

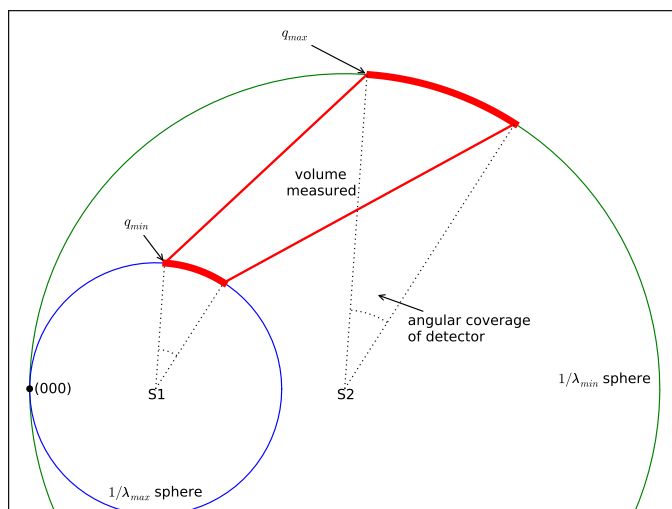


Figure 1

2D Ewald sphere construction showing reciprocal-space detector coverage for a wide-bandwidth Laue diffractometer. The smaller blue circle represents the $1/\lambda_{\max}$ limit of measurement, with the larger green circle representing $1/\lambda_{\min}$. The origin of reciprocal space (000) is common to both spheres, but the “virtual” sample position in the construction (S1 or S2) depends on λ . The angular coverage of the detector is represented by the dashed wedge, and projects an arc onto each sphere of radius $1/\lambda$. The red wedge between the two λ is built by projecting straight lines between the q_{\min} and q_{\max} limits for each pixel on the detector face, giving the volume in reciprocal space that is measured at a single sample orientation.

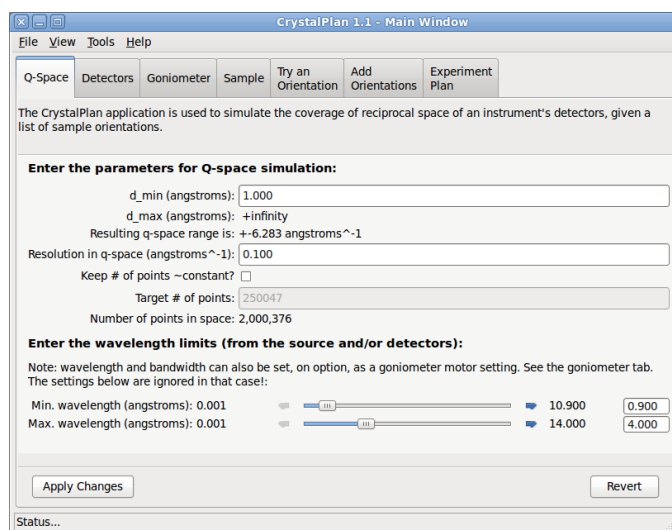
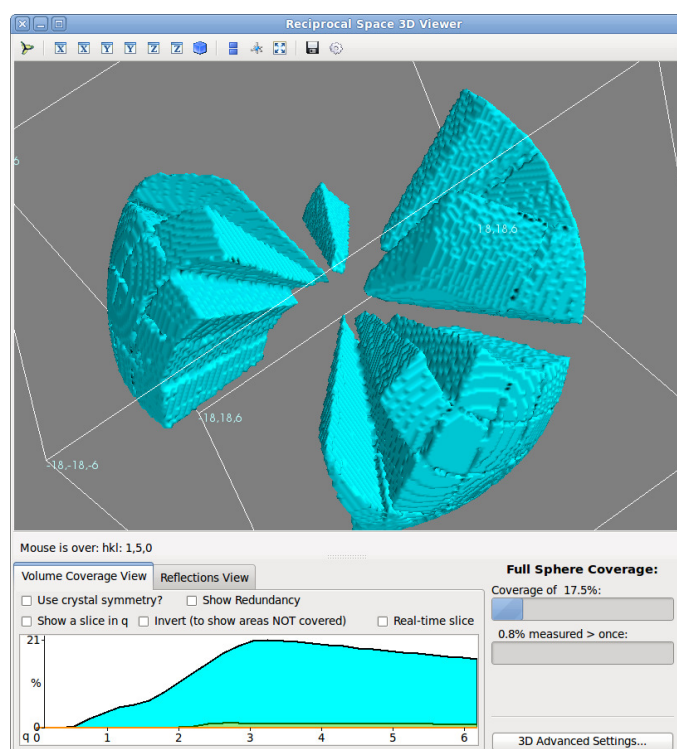
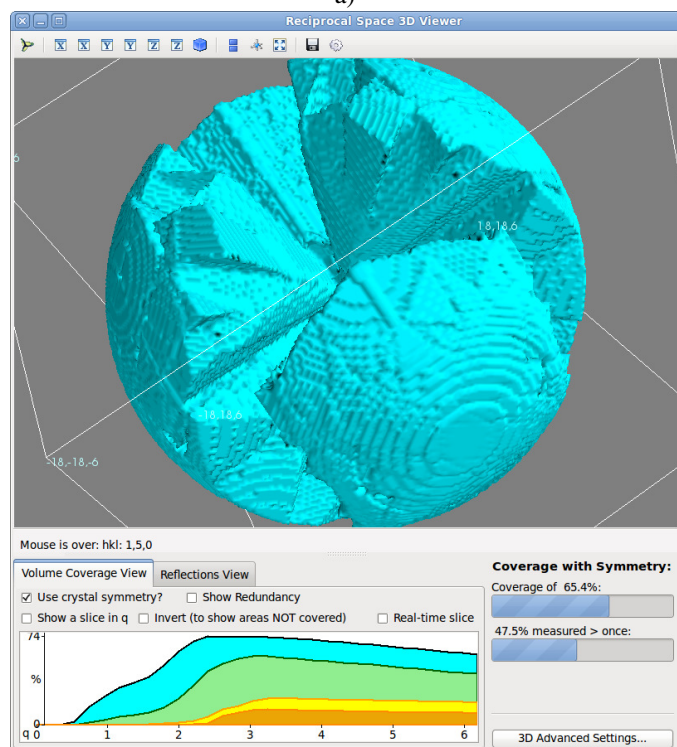


Figure 2

Screenshot of the main GUI window. Six more interface windows (not shown) are presented to the user when he or she clicks on the tabs at the top of the main window.



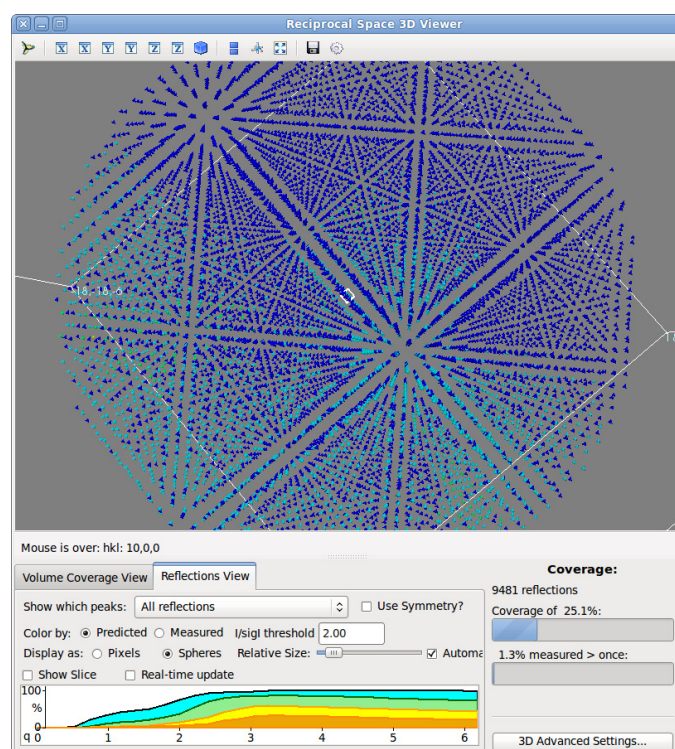
a)



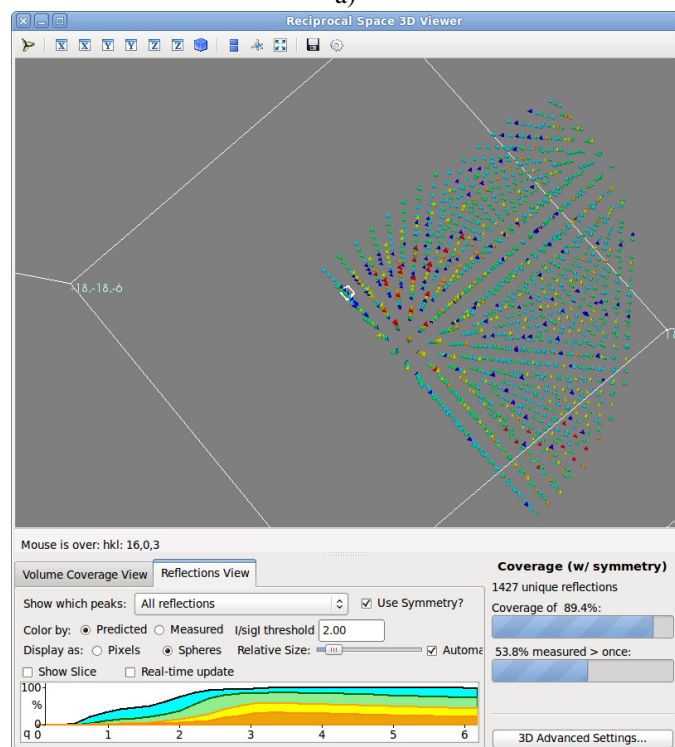
b)

Figure 3

Sample screenshot showing the 3D view of the predicted coverage of reciprocal space volume for 3 sample orientations and 14 detectors in TOPAZ, from 0.9 Å to 4.0 Å: a) without crystal symmetry, 17% coverage; b) with crystal symmetry (mmm 8-fold symmetry), coverage increases to 65%.



a)



b)

Figure 4

Sample screenshot showing the 3D view of single-crystal reflection measurements. Each sphere represents an *hkl* peak, color-coded to indicate the number of measurements (dark blue: 0, cyan: 1, green: 2, etc.). a) without crystal symmetry, 2375 of 9481 reflections are measured, or 25%. b) with crystal symmetry, 1276 of 1427 primary reflections are measured, or 89%, and redundancy is also increased.

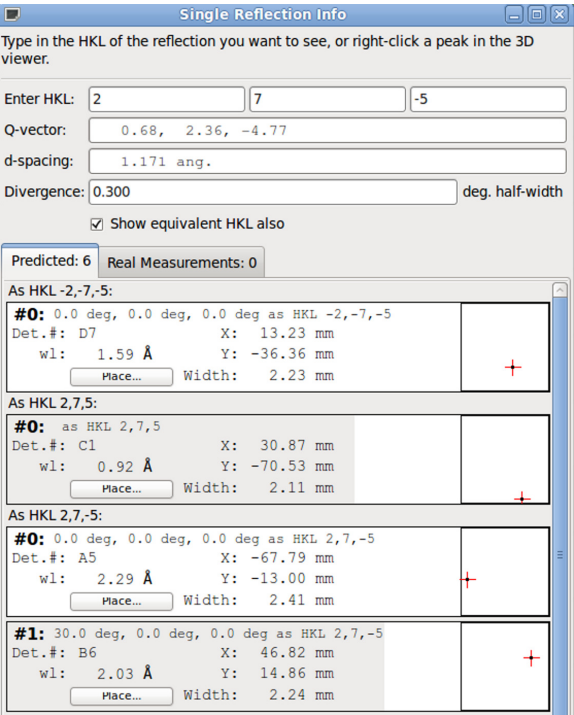


Figure 5
GUI showing details of the predicted measurement positions for a single *hkl*.

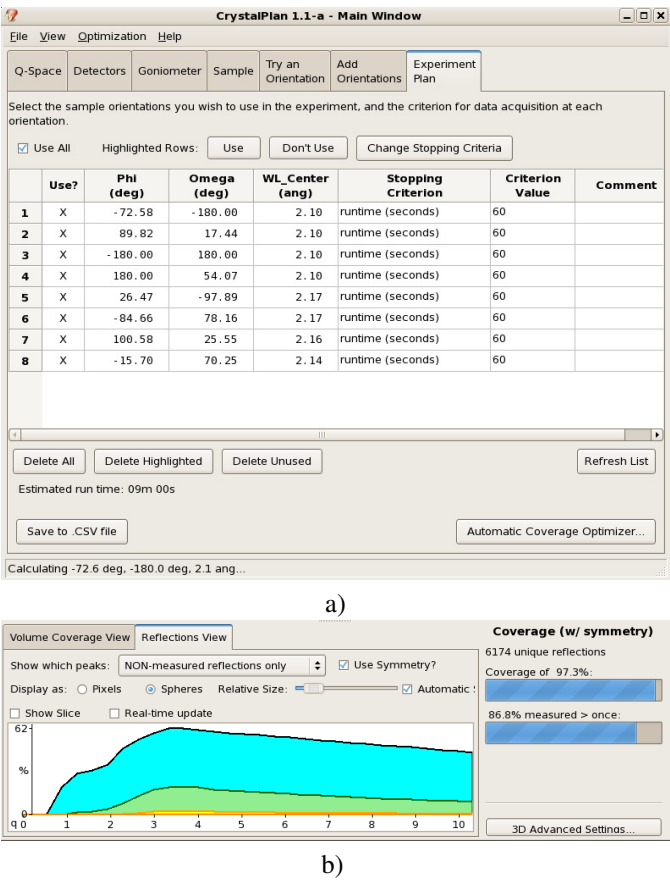


Figure 6
a) Screenshot of the Natrolite experiment plan generated using the Automatic coverage Optimizer. b) Screenshot of the resulting coverage statistics.