BEAM-ish 2.0: a graphical user interface for the physical characterization of macromolecular crystals

Jeff Lovelace and Gloria E. O. Borgstahl
**BEAM-ish 2.0: a graphical user interface for the physical characterization of macromolecular crystals**

Jeff Lovelace and Gloria E. O. Borgstahl*

Eppley Institute for Cancer Research, 987696 Nebraska Medical Center, Omaha, NE, 68198-7696, USA. Correspondence e-mail: gborgstahl@unmc.edu

Received 15 April 2003
Accepted 22 May 2003

**Keywords:** protein crystals; mosaicity; crystal quality; fine phi slicing

1. The crystallographic problem

Crystal mosaicity is determined from the measurement of the reflection angular width and can be used as an indicator of crystal perfection (Helliwell, 1988). A new method has been developed that combines the use of synchrotron radiation, super-fine \( \phi \)-slicing and a CCD area detector to measure simultaneously the mosaicity of hundreds of reflections (Bellamy et al., 2000). Highly monochromatic unfocused synchrotron radiation of known horizontal and vertical divergence was used to minimize reflection broadening by the beam. The X-ray beam characteristics and Lorentz correction are deconvoluted from the resulting reflection widths to calculate the true crystal mosaicity (Greenhough & Helliwell, 1982). Information on crystalline mosaic block structure can be extracted from the reflection topographs and by curve fitting of reflection profiles.

2. Method of solution

**BEAM-ish 2.0** is a graphical user interface (Fig. 1) that manages the processing of multiple super-fine \( \phi \)-sliced diffraction images for crystal quality analysis (Bellamy et al., 2000), seamlessly links together several programs, provides a central storage directory for files generated at all stages of the analysis, and allows the user to visualize and check the results. The core operation of **BEAM-ish** has remained unchanged since version 1.0 (Lovelace et al., 2000). There are several major improvements in version 2.0. The first is the addition of an animator. The animator allows the user to play the fine slices around a reflection. It is tied to the ‘Zoom Window’ to determine the...
magnification and to the ‘Profile Animator’ to display the location of the slice along the reflection profile. Next a Differential Evolution Algorithm (DEA) was written to perform Gaussian peak fitting (Wormington et al., 1999). The DEA is a subset of the genetic algorithm family and is primarily driven by random numbers and parameter constraint ranges. The DEA breeds a family of possible solutions and then screens for the strongest offspring to breed successive generations. The DEA implementation is faster, provides better fits in terms of minimizing the least squares, is not sensitive to initial conditions, and does not require the Matlab libraries to function, in comparison with the Levenburg–Marquardt algorithm (Marquardt, 1963) used in BEAM-ish 1.0 (Lovelace et al., 2000).

An HTML-based online help system has been incorporated into BEAM-ish 2.0. The help files can also be read with any HTML reader, such as Netscape or Internet Explorer. An inter-process communication core has been added to allow multiple instances of BEAM-ish running on the same computer to share information between one another. Primarily this is used when searching for symmetry-related reflections across multiple data sets. A collection of menu tools have been added to make it easier to extract information from the BEAM-ish window and other BEAM-ish files for the purpose of including the data in publications or performing additional analysis with other programs. These tools have been helpful in extracting subsets of data to perform three-dimensional reflection reconstructions. DEA fitting, accumulating symmetry-related reflections and three-dimensional reconstructions have been described by Vahedi-Faridi et al. (2003). BEAM-ish allows the image oscillation information to be masked so that in cases where the collection software fails to write the oscillation information properly in the image header the data can still be processed. Several internal searching and sorting algorithms have been improved to enhance performance. Finally, BEAM-ish has been updated to handle ADSC, SBC and MarCCD image formats automatically with the ability to add future detector formats easily as needed.

3. Software and hardware environment

BEAM-ish 2.0 is designed to run under the IRIX or Linux operating systems with XWindows release 11 or better. The source code for BEAM-ish was developed entirely in C++ and relies heavily on Troll Tech’s QT library of graphical widgets. Additionally, Matlab’s runtime library is included in the SGI version and provides the LM algorithm for Gaussian peak fitting. Under Linux, the only peak-fitting algorithm provided is the DEA. SGI O₂ systems or better, or a PC running Linux and XWindows, should be able to run the program.

4. Documentation and availability

Currently, BEAM-ish, HALFSLICE, PEAKSTATS and associated documentation are available ‘as is’ on a collaborative basis with the authors. BEAM-ish comes with HTML-based online help. The manual is also available at http://www.unmc.edu/Eppley/sbl/beamish/home.htm.

We would like to thank Dr Edward Snell for helpful discussions. This work was funded by NASA grants NAG8-1380 and NAG8-1825.

References